MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September - December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI611 Artificial Intelligence Fundamentals (8 ECTS)

Course purpose and objectives: The purpose of the course is to introduce students to the fundamental principles and techniques that underline software systems that exhibit “intelligent” behavior.

Learning outcomes: Upon completion of this course, students will have acquired a good understanding of modern Artificial Intelligence, the problems that it addresses and the fundamental solution methods that it uses. More specifically the students will know the main knowledge representation techniques and reasoning methods that underlie artificial intelligence problem solving and be able to develop simple solvers for artificial intelligence systems.

Required: Knowledge of a high-level programing language, object-based data concepts and structures.

Teaching methodology: Lectures, discussions of practical examples and (unsupervised) lab activities where the active learning element is encouraged and supported. Students would be strongly guided to view all topics presented and discussed with a critical eye, identifying the limits of AI both in its foundational years and the current situation characterized by an explosion of multimedia data of varying degrees of usability, quality and ethical considerations.

Assessment: Final exam (50%), midterm exam (30%) and homework (theoretical and/or programming assignments) (20%).

Main text:

S. Russel and P. Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson, 2021.

Other reading:

R. J. Brachman, H. J. Levesque, Knowledge Representation and Reasoning, Elsevier, 2004.

N. J. Nilsson: The Quest for Artificial Intelligence: A history of ideas and achievements, Cambridge University Press, 2010.

M. Ginsberg: Essentials of Artificial Intelligence, Morgan Kaufman,1993.

P. H. Winston: Artificial Intelligence, 3rd Edition, Addison-Wesley, 1992.

E. Keravnou, Artificial Intelligence and Expert Systems (in Greek), Greek Open University, 2000.

G.F. Luger and W.A. Stubblefield, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 5th edition, Addison-Wesley, 2005.

G. Weiss (editor), Multiagent Systems: a modern approach to distributed AI, The MIT Press, 2nd edition, 2013.

P. Jackson, Introduction to Expert Systems, 3rd edition, Addison-Wesley, 1999.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTRODUCTION

Historical Landmarks, AI Definitions and Symbolic versus Connectionist AI

CONTENTS

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this introductory unit, students will be able:

1. To present the important landmarks in the history of AI.
2. To appreciate the different meanings ascribed to the term “Artificial Intelligence” and hence the lack of a universally accepted definition.
3. To give and critically analyze several of these definitions.
4. To discuss Alan Turing’s seminal thesis on computing machinery and intelligence rooted on his famous test for machine intelligence, the imitation game.
5. To discuss Allen Newell and Herbert Simon’s foundational work on symbols, search and the physical symbol system hypothesis, that underline their proposition about computer science as an empirical enquiry.
6. To draw a distinction between symbolic and connectionist AI.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

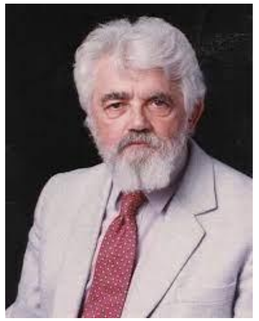
Tracing the history of Artificial Intelligence through its landmarks from Alan Turing’s imitation game to DeepMind

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Scientific achievements that paved the way

1956
Dartmouth College Summer Research Project on Artificial Intelligence
Organized by John McCarthy who coined the term



What followed

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Scientific achievements that paved the way

1763: Thomas Bayes invented a framework for reasoning about the probability of events – **Bayesian inference**

1854: George Boole specified the representation of logic in equations – **Logical reasoning**

1935: Alan Turing conceived the **principle of the modern computer** that could also learn from experience by altering its own instructions

1943: As a leading cryptanalyst, Turing led efforts to create the **first fully functioning digital computer**, named Colossus

1945: **ENIAC** the first electronic general-purpose computer was built

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Scientific achievements that paved the way

1949: Claude Shannon developed the first computer **chess playing** program

1950: Turing proposed a test for machine intelligence – the **imitation game**

1952: Arthur Samuel developed the first computer **checkers program** – early demonstration of machine learning

1955: Allen Newell developed the first program **to mimic human problem solving** – together with Herbert Simon they started working on the **Logic Theorist**

1956: Over the Xmas holiday Herbert Simon and Allen Newell invented a **thinking machine** – Simon earned in 1978 a Nobel prize for his pioneering work on **heuristic problem solving**

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

August 1955: A Proposal for the Dartmouth Summer Project on Artificial Intelligence

http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf

Submitted to the Rockefeller Foundation, that accepted to fund the project in the summer of 1956

Proposers: John McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Indicative aspects of the artificial intelligence problem in the McCarthy et. al. proposal

1. **Automatic computers**: if a machine can do a job, then an automatic calculator can be programmed to simulate the machine.
2. **How can a computer be programmed to use a language**: It may be speculated that a large part of human thought consists of manipulating words according to rules of reasoning and rules of conjecture.
3. **Neuron Nets**: How can a set of (hypothetical) neurons be arranged so as to form concepts.
4. **Theory of the size of a calculation**: If we are given a well-defined problem (one for which it is possible to test mechanically whether or not a proposed answer is a valid one) one way of solving it is to try all answers in order.
5. **Self-Improvement**: Probably a truly intelligent machine will carry out activities which may best be described as self-improvement.
6. **Abstractions**: A number of “types” of abstraction can be distinctly defined and several others less distinctly.
7. **Randomness and creativity**: A fairly attractive and yet clearly incomplete conjecture is that the difference between creative thinking and unimaginative competent thinking lies in the injection of some randomness.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Individual Research Proposals

Claude Shannon

- Application of information theory concepts to computing machines and brain models
- The matched environment – brain model approach to automata

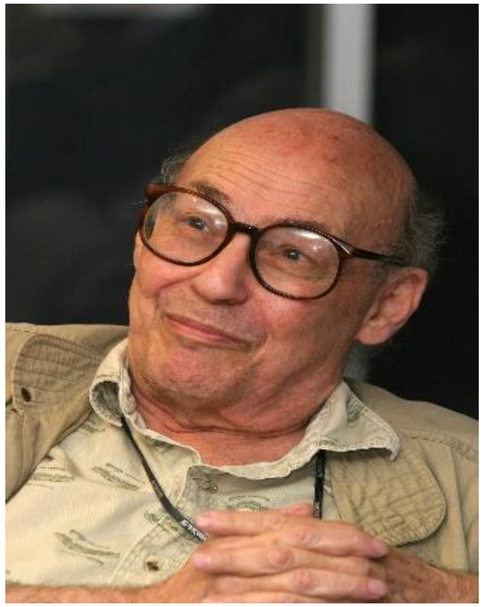
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Individual Research Proposals

Marvin Minsky

Design a machine that exhibits learning: given a criterion of “success” and “failure” a machine can be trained to exhibit goal-seeking behavior

Progress on a very tentative proposal given in his dissertation towards having a model of such a machine fairly close to the stage of programming in a computer



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Individual research proposals

Nathaniel Rochester

- Originality in machine performance
- The process of invention or discovery
- The machine with randomness

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

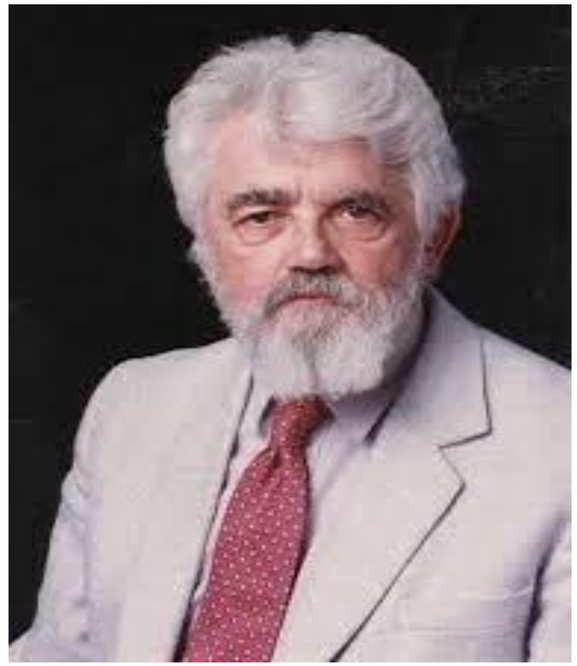
Individual research proposals

John McCarthy

Study the relation of language to intelligence

Construct an artificial language which a computer can use on problems requiring conjecture and self-reference

Using this language might be possible to program a machine to learn to play games well and to do other tasks



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The birth of Artificial Intelligence in the summer of 1956 at Dartmouth College

50th anniversary picture

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

What followed

1957: Frank Rosenblatt invented the perceptron algorithm and machine – precursor to **neural network** and **deep learning**

1958: The programming language **LISP** (list processing) devised by John McCarthy appeared – founded on the mathematical theory of recursive functions

1959: Herbert Simon, John Clifford Shaw and Allen Newell worked on the **General Problem Solver** (GPS) indented to work as a universal problem solver machine – in contrast to Logic Theorist, GPS used **means-ends analysis**

1959: Arthur Samuel, a gaming world pioneer coined the term "**machine learning**" – how a computer could be programmed to play a better game of checkers than its creator – mastering the process of learning

1961: First **industrial robot** called Unimate began working on an assembly line in the General Motors plant in New Jersey

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

What followed

1966: **ELIZA**, the world's **first chatbot** was created at the MIT AI Lab by Joseph Weizenbaum – demonstrates basic communications between humans and machines – example of early NLP program

https://web.njit.edu/~ronkowit/eliza.html

1970s: **Criticism** growing about AI researchers overpromising and underdelivering products or solutions with real-world impact

1976: Allen Newell and Herbert Simon, 10th Turing Lecture "Computer Science as Empirical Enquiry: **Symbols** and **Search**"

1980s: "**AI Winter**": During the late 1980s and early 1990s excitement surrounding AI fell.

1982: Japan's Ministry of International Trade and Industry began the **Fifth Generation Computer Systems** (FGCS) initiative – create computers using massively parallel computing and logic programming

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

What followed

1997: **Deep Blue** (IBM) – Chess-playing program defeating world champion Garry Kasparov

mid-2000s: **Resurgence of interest**

2007: a team from Princeton University assembled **ImageNet**, a large database of annotated images – in 2012 a CNN with 650,000 neurons reduced error rate to 15.3%

2009: Google started a secret project to develop a **driverless car** – in 2014 the first autonomous vehicle passed a U.S. state self-driving test

2011: **IBM Watson**, a cognitive computing platform, to compete in the tv show Jeopardy! – defeats two former champions. IBM launches Watson as a major move to bring AI to healthcare

2016: Microsoft made a major breakthrough in **speech recognition**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

What followed

2015: **AlphaGo** (DeepMind – Dennis Hassabits), using **reinforcement learning** defeating European Go champion – in 2016 AlphaGo beat the Korean Go champion and in 2019 the World Go champion

2017: **AlphaZero** (DeepMind) – superhuman play in multiple games and trained by self-play

2019: **AlphaStar** (DeepMind) – Deep reinforcement learning defeating StarCraft player

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Defining Artificial Intelligence: the birth of the name and its many guises

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Birth of the name “Artificial Intelligence”: 1956 Dartmouth College Summer Research Project

The present purpose of the Artificial Intelligence problem is taken to be that of making a machine behave in ways that would be called intelligent if a human were so behaving.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

John McCarthy: What is Artificial Intelligence?
http://jmc.stanford.edu/articles/whatisai/whatisai.pdf (2007)

It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

John McCarthy: What is Artificial Intelligence?

Intelligence involves mechanisms, and AI research has discovered how to make computers carry out some of them and not others. If doing a task requires only mechanisms that are well understood today, computer programs can give very impressive performance on these tasks. Such programs should be considered “somewhat intelligent”.

AI programs haven’t yet reached the level of being able to learn much of what a child learns from physical experience. Nor do present programs understand language well enough to learn much by reading.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

John McCarthy on chess

Chess is the Drosophila of AI: analogy with geneticists' use of the fruit fly to study inheritance

- Chess programs play at grandmaster level, but with limited intellectual mechanisms compared to those used by a human chess player
- Large amounts of computation are substituted for understanding
- Once human mechanisms are understood, human-level chess programs can do far less computation
- Deep Blue did millions of times as much computation

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

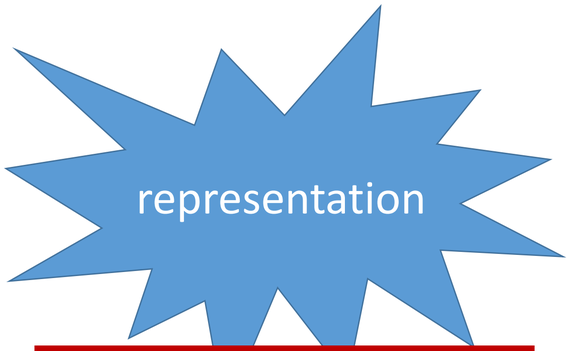
John McCarthy on computability, complexity and algorithms

Computability theory and computational complexity are relevant but don't address the fundamental problems of AI

What is important for AI is to have algorithms as capable as people at solving problems

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

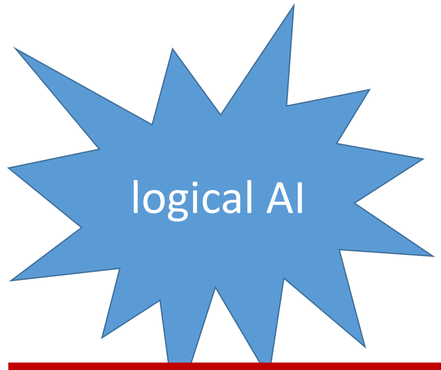
Branches of AI



Facts about the world must be represented in some way, e.g., languages of mathematical logic

Examine large numbers of possibilities

Compare observations with a pattern



Knowledge about the world, facts of a specific situation and goals represented as sentences of some mathematical logical language

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Branches of AI

inference

From some facts, others can be inferred (e.g., logical deduction, non-monotonic reasoning, default reasoning)

common sense knowledge and reasoning

Area in which AI is farthest from human-level

learning from experience

Connectionism and neural nets specialize in that – programs can only learn what facts or behaviors their formalisms can represent

planning

Generate a strategy for achieving a goal, often a sequence of actions

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Branches of AI

A study of the kinds of knowledge that are required for solving problems in the world

The study of the kinds of things that exist – kinds of objects and their properties

Rules of thumb for guiding the solution to a problem – not infallible

A technique for solving a task by mating and selecting fittest in millions of generations

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

AI applications

- Game playing
- Speech recognition
- Understanding natural language
- Computer vision
- Expert systems
- Heuristic classification

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Other definitions of Artificial Intelligence

Artificial Intelligence makes computers do things that would require intelligence if done by people

Artificial Intelligence is the development of computers whose observable performance has features which in humans we would attribute to mental processes

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Other definitions of Artificial Intelligence

Artificial Intelligence is the study of how to build and/or program computers to enable them to do the sorts of things that minds can do. Some of these things are commonly regarded as requiring intelligence:

- Medical diagnosis and/or prescription
- Legal or scientific advice
- Proving theorems in logic or mathematics

While others are not, as all adults can do them and typically do not involve conscious control:

- Seeing things in sunlight or shadows
- Finding a path through cluttered terrain
- Using one's common sense

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

A rather controversial definition

Artificial Intelligence is the science of intelligence in general or, more accurately, as the intellectual core of cognitive science

- Its goal is to provide a systematic theory that can explain (and perhaps enable us to replicate) both the general categories of intentionality and the diverse psychological capacities grounded in them

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Yet more definitions . . . No one is universally adopted

Artificial Intelligence is a branch of informatics that deals with the automation of intelligent behavior.

Artificial Intelligence is the study of mechanisms that underline intelligent behavior through the construction and evaluation of systems that represent these mechanisms

Artificial Intelligence is the development of computer systems to solve difficult problems that cannot be solved by exhaustive examination of all solutions since they can be too many

Artificial Intelligence is the study of how to make a computer do something that currently humans do better

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Critical periods in the lifetime of AI

Late 50s until early 60s: torrent of theories and experiments which moved AI forward and brought it in the public and government spotlight

1974 until 1980: the first AI Winter

1987 until 1993: the second AI Winter

From mid 2000s: Resurgence of interest in AI – it is not an exaggeration to say that AI generates value for the present and transforms the future

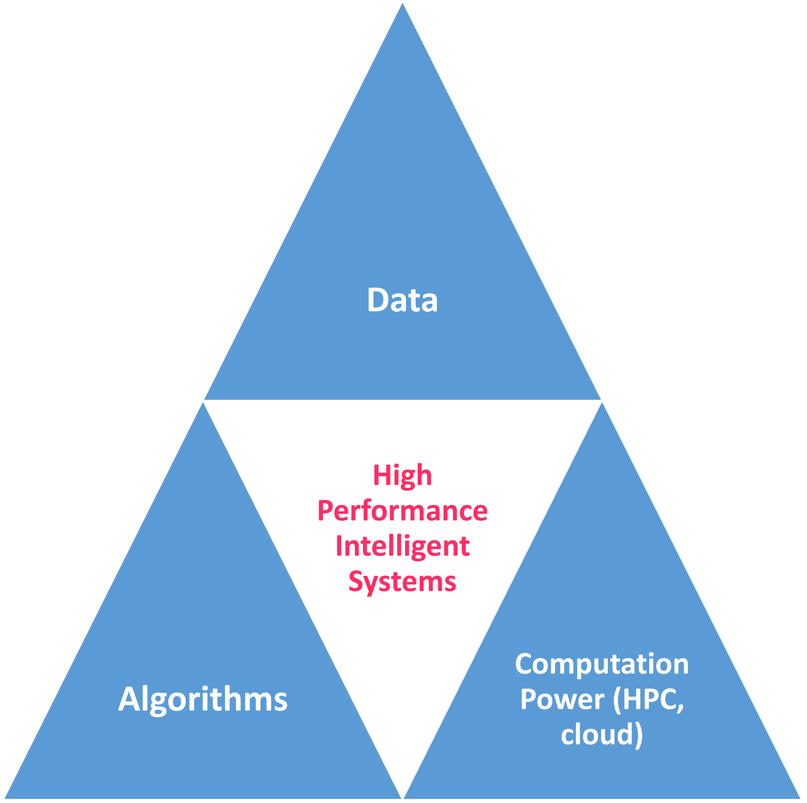
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Clear indications of the impact of AI

- Tech giants compete for supremacy
- Europe and most countries formulate strategies and pass legislation to regulate AI
- Companies train staff to adopt AI
- The recently announced “Informatics Reference Framework for School” of the European Coalition "Informatics for all" includes the suggestion for familiarization of children, from primary to high school, with elements of AI since the new generation of powerful computer systems is based on AI
- America is in "strategic competition" with China having the belief that the one who leads in AI will lead the world

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

High Performance Intelligent Systems



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Categorizing AI

- **Weak or Narrow AI:** is good at performing a single task
- **Strong or General AI:** the type that can understand and reason across its environment as a human would. Has always been elusive and not likely on the short-term horizon.
- **Artificial Super Intelligence:** a level of computer sophistication where machines become smarter than the humans that create them

- **Good Old-Fashioned AI (GOFAI):** The symbolic AI in the 1950s to the late 1980s. “Neat” AI mainly rooted on
  - Symbolic representations of problems
  - Top-down approaches of logic, problem solving and expert systems

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Alan Turing’s Computing Machinery and Intelligence (Oct 1950)

https://phil415.pbworks.com/f/TuringComputing.pdf

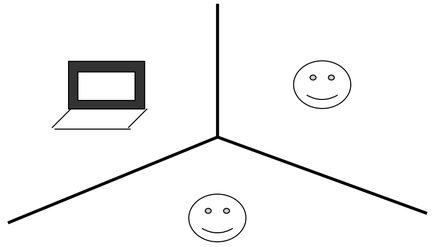
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The imitation game: Can machines think?

A machine is a digital, discrete-state, computer

- Draws a sharp line between the physical and the intellectual capacities of a human
- The interrogator cannot see, touch, or hear the competitors
- A machine to play the game satisfactorily, should try and provide answers that would naturally be given by a human



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Critique of Turing's Imitation Game as a Test for Machine Intelligence

- It molds machine intelligence to the measures of human intelligence
  - May not machines carry out something which ought to be described as thinking but which is very different from what a human does?
- Limits itself to problem-solving tasks in a symbolic way, completely ignoring other important elements of intelligence
- Is 'behavior' enough to prove intelligence?

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Objections addressed by Turing

- The Theological Objection:
  - Thinking is a function of a human being's immortal soul
- The "Heads in the Sand" Objection:
  - The consequences of machines thinking would be dreadful. Let us hope and believe that they cannot be so.
- The Mathematical Objection:
  - In any sufficiently powerful logical system, statements can be formulated which can neither be proved nor disproved within the system
    - Similar limitations apply to the human intellect

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Objections addressed by Turing

- The Consciousness Objection:
  - The only way by which one could be sure that a machine thinks is to **be** the machine and to feel oneself thinking
- The Various Disabilities Objection:
  - Machines cannot be kind, resourceful, beautiful, friendly, have initiative, have a sense of humor, tell right from wrong, ...
- Lady Lovelace’s Objection:
  - Babbage’s Analytical Engine can do whatever we know how to order it to perform
    - Machines not capable of doing something new, therefore incapable of learning

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Objections addressed by Turing

- The Nervous System Continuity Objection:
  - The nervous system is not a discrete-state machine
- The Informality of Behavior Objection:
  - It is not possible to produce a set of rules to describe what a human should do in every conceivable set of circumstances

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Turing's Learning Machines: Abstract Digital Computing Machine

- Has a limitless memory, and
- A scanner that moves back and forth through the memory, symbol by symbol, reading what it finds and writing further symbols
- Could learn from experience by altering its own instructions
  - The problem is mainly one of programming

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Turing's Learning Machines: Punishment, Reward, Randomness

- Teach a machine through punishments and rewards, by programming it to
  - Avoid repeating events leading to a punishment
  - Increase the probability of repeating events leading to a reward

- Rules with ephemeral validity can be changed

- A random element in a learning machine is useful when searching for a solution
  - E.g., find a number between 50 and 200 which is equal to the square of the sum of its digits

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Turing's Prediction and Wish

- Prediction: By 2000, it will be possible so that an average interrogator will not have more than 70% chance of making the right identification after 5 minutes of questioning
- Wish: Machines will eventually compete with humans in all purely intellectual fields
  - Which are the best intellectual fields to start with?
  - Many people think that a very abstract activity like the playing of chess would be best
- However, he states that his initial question "Can machines think?" is meaningless to deserve discussion!

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Turing's Concluding Sentence in his essay on Computing Machinery and Intelligence

We can see only a short distance ahead,

but we can see that much remains to be done.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Allen Newell and Herbert Simon, 10th Turing Lecture “Computer Science as Empirical Enquiry: Symbols and Search” (1976)

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.2482&rep=rep1&type=pdf

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Computer Science as Empirical Enquiry

As basic scientists we build machines and programs as a way of discovering new phenomena and analysing phenomena we already know about.

Symbolic Search – Heuristic Search

Have deep significance for understanding how information is processed and how intelligence is formed.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Symbol Processing

- Physical-Symbol System
  - Symbols lie at the root of intelligent action
- The Physical-Symbol System Hypothesis

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Physical-Symbol System

- It consists of a set of entities, which are called symbols, and are physical patterns.
- Symbols can occur as components of another type of entity, called an expression or symbol structure.
- The symbols that make up a symbol structure are related in some physical way, such as being next to each other.
- The system also has a number of processes, which operate on expressions and produce other expressions, such as creation, modification, reproduction, and destruction processes.

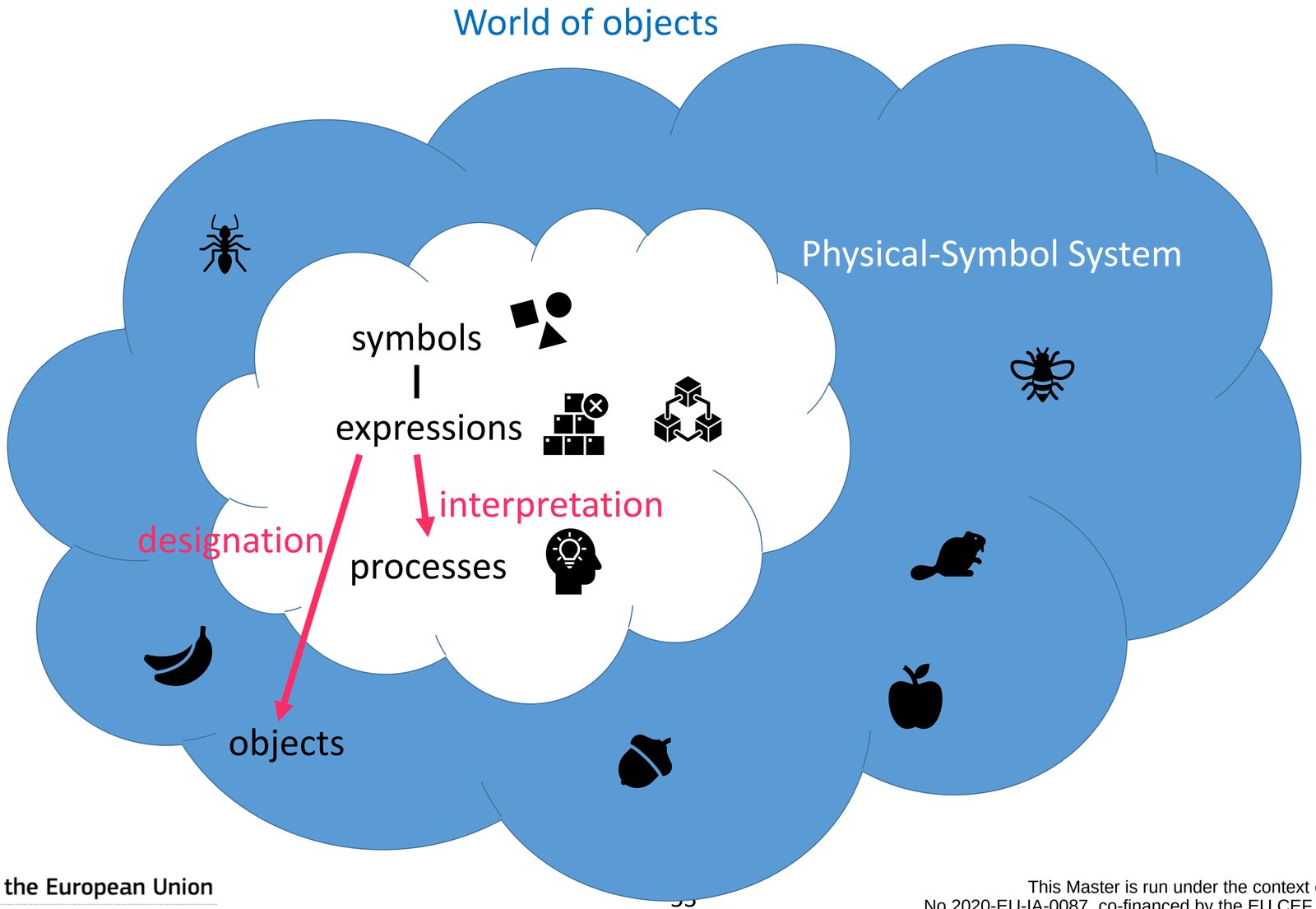
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Physical-Symbol System

- At any instant of time, the system contains a collection of such symbol structures.
- It is therefore a machine that produces through time an evolving collection of symbol structures.
- Such a system exists in a world of objects wider than just these symbolic expressions themselves.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Designation and Interpretation

- Designation:
  - Expressions designate objects and this way access to objects is provided
  - The system can either affect the object or behave in ways depending on the object
- Interpretation:
  - An expression designates a process for interpreting it
  - Given the expression, the system carries out the process

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Requirements of completeness and closure

- Symbols may dynamically designate any expression
  - Symbols and their relations determine which object is designated by a complex expression
- Every process is designated by some expression
- There exist processes for creating/modifying any expression in arbitrary ways
- Expressions continue to exist once created until explicitly modified or deleted
- The number of expressions that a system can hold is unbounded

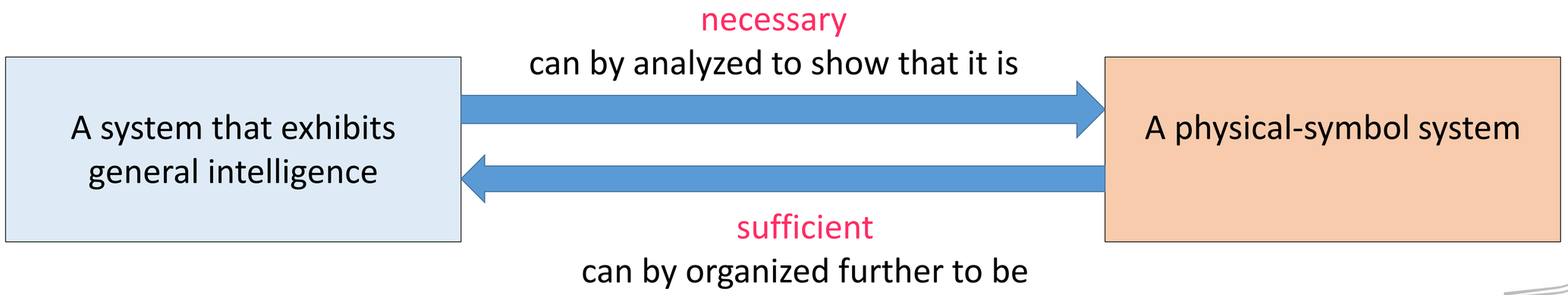
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

The Physical-Symbol System Hypothesis

A physical-symbol system has the necessary and sufficient means for general intelligent action

- It is an empirical hypothesis
- General intelligent action: same scope of intelligence as in human action



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The LISP language – devised by John McCarthy in 1958

- LISt Processing
  - Based on the evolving experience with list-processing
  - Arose out of the attempt to construct intelligent programs
- Symbols and s-expressions
  - Programs and data defined uniformly as s-expressions
- Functions, recursion, objects
- PROLOG (PROgramming in LOGic) came much afterwards

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

The role of search in intelligence

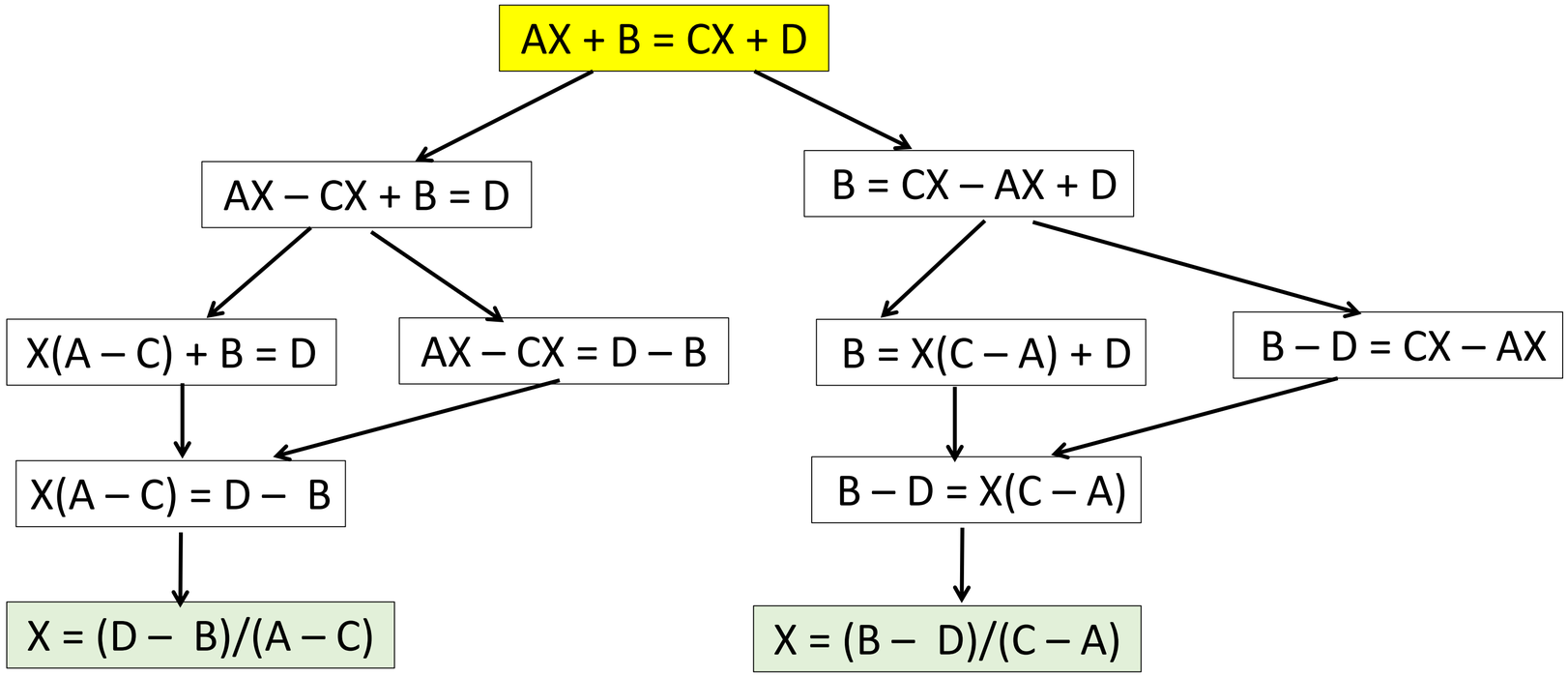
- Symbol systems solve problems through heuristic search
- Heuristic Search Hypothesis
  - Problem solutions are represented as symbolic structures
  - A Physical-Symbol System progressively generates and modifies symbol structures, through heuristic search, until it produces a solution structure
  - Heuristic search is used because the system computing resources are scarce relative to the problem solution complexity
- Problem space
  - A space of symbol structures, representing problem situations, including initial and goal situations
  - Move generators are processes for modifying one situation in the problem space into another
- During the first decade of AI research, the study of problem-solving was almost synonymous with the study of search processes

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: Given AX + B = CX + B, find the solution X = E such that AE + B = CE + D

- Non-intelligent approach: continuously produce numbers and test
- Intelligent approach:



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Avert exponential explosion of search

- This is the art of intelligence
- Generate only structures that show promise of being solutions or of being along the path towards a solution
  - Decrease rate of branching, not to prevent it entirely
  - Employ information-using techniques to guide search
    - Means-ends analysis; best-first search, ...

The Empirical Base

- Research in AI is concerned with how symbol systems must be organized in order to behave intelligently
- What to do next?

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Symbolic versus Connectionist Artificial Intelligence

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

If
the site of the culture is blood
the gram of the organism is neg
the morphology of the organism is rod
the burn of the patient is serious
then
there is weakly suggestive evidence (0.4) that
the identity of the organism is pseudomonas.

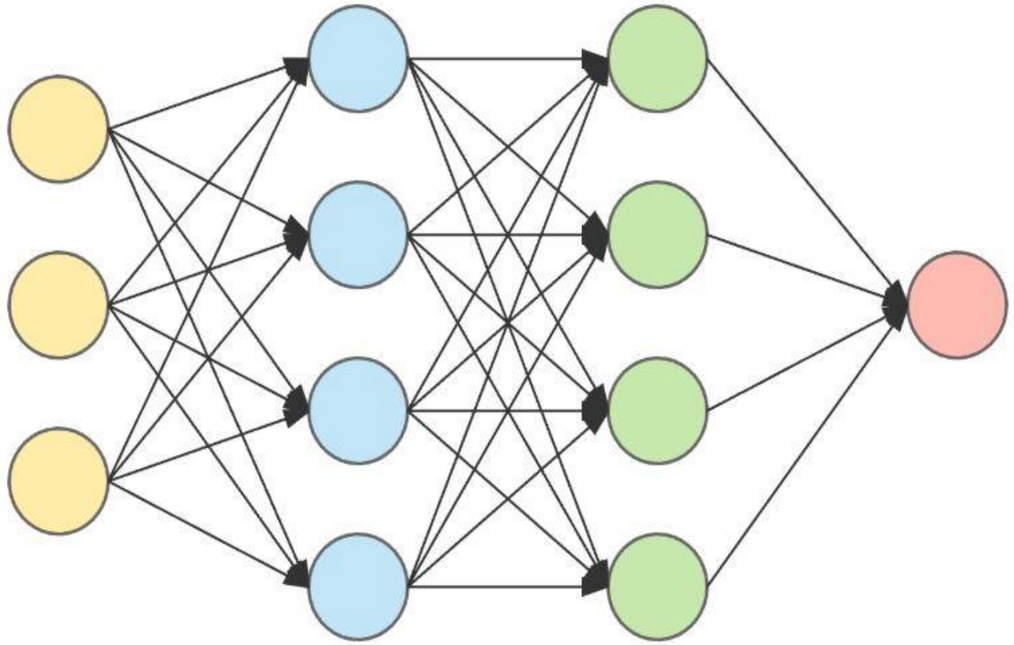
$$(\forall X)(sterile(X) \equiv (\forall Y)(bacterium(Y) \wedge in(Y,X) \supset dead(Y))),$$

$$(\forall XY)(hot(X) \wedge in(Y,X) \supset hot(Y)),$$

$$(\forall Y)(bacterium(Y) \wedge hot(Y) \supset dead(Y)),$$

and

$$hot(a).$$



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

EC High-Level Expert Group on Artificial Intelligence

https://www.aepd.es/sites/default/files/2019-12/ai-definition.pdf

- AI systems can either use symbolic rules or learn a numeric model
- Symbolic approach
  - Knowledge: modelling, representation, reasoning (search)
- Connectionist approach
  - Machine learning, neural networks, deep learning, decision trees, …..
  - Solve problems that cannot be precisely specified, or whose solution method cannot be described symbolically
  - ML techniques produce a numeric model (that is, a mathematical formula) used to compute the decision from the data
- Explainability is a property of those AI systems that can provide a form of explanation of their actions/decisions/recommendations

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

European Commission's Definition of AI

- Artificial Intelligence (AI) refers to systems that display intelligent behavior by analyzing their environment and taking actions – with some degree of autonomy – to achieve specific goals.
- AI systems can be purely software-based, acting in the virtual world (e.g., voice assistants, image analysis software, search engines, speech and face recognition systems) or AI can be embedded in hardware devices (e.g., advanced robots, autonomous cars, drones, or Internet of Things applications).

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Rationality rather than intelligence

- Intelligence (both in machines and in humans) is a vague concept
- Better use the notion of rationality: the ability to choose the best action to take in order to achieve a certain goal, given certain criteria to be optimized and the available resources; often only bounded rationality is achieved.
- Rationality, although not the only ingredient of intelligence, it is a significant part of it.

Embedding

Usually, AI systems are embedded as components of larger systems, rather than stand-alone systems

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Definition proposed by High-Level Expert Group

Artificial intelligence (AI) systems are software (and possibly also hardware) systems designed by humans that, given a complex goal, act in the physical or digital dimension by perceiving their environment through data acquisition, interpreting the collected structured or unstructured data, reasoning on the knowledge, or processing the information, derived from this data and deciding the best action(s) to take to achieve the given goal. AI systems can either use symbolic rules or learn a numeric model, and they can also adapt their behavior by analyzing how the environment is affected by their previous actions.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Summary

- Important landmarks in the history of AI
- Different meanings ascribed to the term “Artificial Intelligence”
- Alan Turing’s seminal thesis on computing machinery and his imitation game
- Allen Newell and Herbert Simon’s foundational work on symbols, search and the physical symbol system hypothesis
- Symbolic and connectionist AI

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September – December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Problem Solving through Search

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 2

Problem Solving through Search

CONTENTS

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on problem solving through search, students will be able:

1. To explain in general terms what a heuristic is and how heuristics could enhance purely algorithmic methods.
2. To explain how a problem is represented to be solved through search, that is, explain what the representation problem is.
3. To give and explain the algorithms for the "blind" methods, depth-first search, otherwise “gullible” search, and breadth-first search, otherwise “skeptical” search, and to explain the notions of combinatorial explosion, branching factor, admissibility and efficiency.
4. To give and explain the algorithm A* for search with heuristic guidance, and its variants branch-and-bound and best-first search.
5. To be able to design and implement an object-based generic search method and to extend it for solving a specific problem by search.
6. To explain at a high level what the frame problem is.
7. To be able to differentiate classification problems from synthesis problems.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Algorithms and Heuristics

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example

Which route to follow to go from some town X to some town Y, using a road map giving the direct connections between towns?

Algorithmic Solution: Use the shortest path

A purely algorithmic method is a step-by-step method that takes the problem input and computes the requested solution.

A heuristic method deploys “heuristics”

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristics

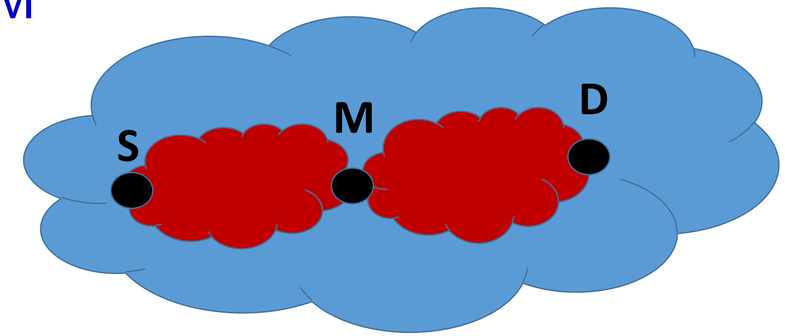
- Heuristics are rules of thumb, rules of good guessing; they are problem specific
- They do not guarantee success, hence they are not infallible, but they provide useful guidance in most cases of the problem, leading to satisfactory, or even optimal solutions, in a computationally effective/viable way.
- They do not negate the basic algorithmic approach in problem solving, in other words the step-by-step search for some solution but augment them, bestowing them “intelligence”.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example heuristics for the route problem

Rule 1:

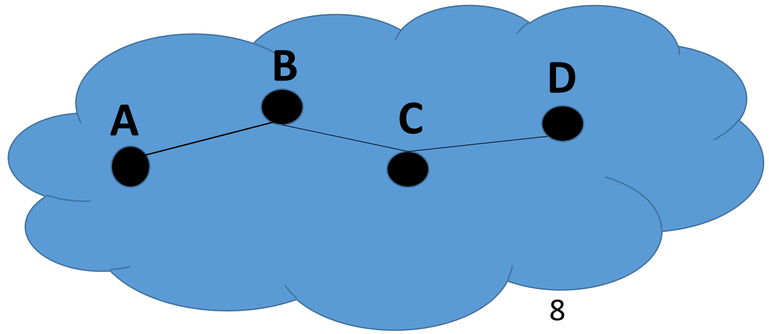
If the start is town-S and the destination is town-D and the day is Tuesday, do not consider routes through town-M



pruning heuristic

Rule 2:

If the start is town-A and the destination is town-D, use the route ‘town-A, town-B, town-C, town-D’



homing heuristic

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Highlights

- ❑ The rationale behind the given heuristics is not explicated
- ❑ These heuristics are very specific as they refer to actual towns by name
- ❑ A plethora of such very specific heuristics could surpass their utility

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Generalizing heuristic rule 1

If the passage through the CITY is not necessary and there is a high probability that there will be crowding there on the day of the passage, then block the CITY

Symbol CITY denotes any town, i.e., it is a variable and not a literal name

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Generalizing heuristic rule 1, gives rise to concepts “necessary passage” and “crowding”, that could also be denoted through symbolic rules as follows:

- If the CITY is the start or the destination, the passage through it is necessary
- If there is a public market going on, there is a high possibility of crowding
- If there is a road accident, there is a high possibility of crowding

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Representation Problem

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Solving a problem by search

1. Representation of the **states** of the problem; all possible states (problem instances) constitute the **state space**

2. Identification of **actions** (or action operators) leading from one state to another

3. Identification of the **navigation mechanism** in the state space

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Representation Problem

- This is a **meta-problem**
- It entails the representation of an (object) problem in a way that it can be solved by search
- A key issue is deciding the representation of the (object) problem states, which could be straightforward if there is only one choice, or not if there are alternative representation choices
  - The state transition operators and overall state space depend on the problem state representation
  - A problem state representation yielding a smaller state space is preferrable

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Representation problem for an object problem

- Specify state space – this constitutes the space in which the search takes place - search space
  - Structure for problem states – symbol structure
  - Operators for converting one state to another state
  - Initial and final/goal states
- Specify navigation/search method in the search space
  - 'Blind', systematic navigation
  - 'Guided' navigation – define heuristics

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: Representing the search space for a problem that involves playing a board game (chess, checkers, backgammon, etc.)

State space:	permitted board cconfigurations
State structure:	two-dimensional table
Initial state:	initial board configuration
Final stares:	the ones that represent win for one or the other opponent
Action operators:	the rules of the game

The great difficulty of the representation problem for such (object) problems is the identification of **powerful heuristics**, which can turn the computer into a powerful opponent.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

How big is a state space?

Chess: After both players move, 400 possible board setups exist. After the second pair of turns, there are 197,742 possible games, and after three moves, **121 million**.

Go: The state space is vast; the number of states is greater than the number of atoms in the universe!

Tic-tac-toe: The upper bound for the state space is $\mathbf{3^9 = 19,683}$; there are three situations for each cell and nine cells. This count includes many illegal positions, such as a position with five crosses and no noughts, or a position in which both players have a row of three.

8-puzzle: The state space is 9! = **362,880**

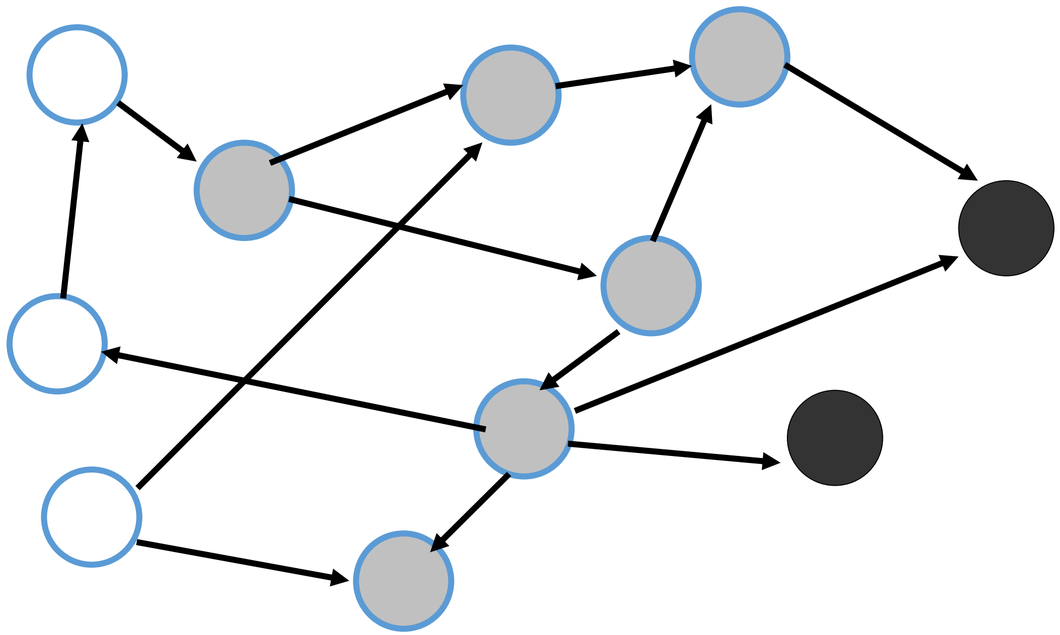
Jugs problem: The state space is (m+1)(n+1) where m and n are the respective capacities of the two jugs.

Brute force search methods are not viable in huge state spaces

Only part of a state space is explicated depending on the search method, and the power of the heuristics used – search tree

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



State Space

- Generally, it is a graph
  - Multiple ways to reach a state
- Nodes are states of the problem
  - initial, intermediate and final states
  - for specific instances of the problem, the initial state is given and possibly the final state(s) – goal state(s)
- The arcs represent actions

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Final States

- In many problems the final states cannot be specified in advance, or it is not appropriate to do so.
- The essence of the problem may be the recognition of the final state.
- If the final state can be fully defined in advance, the solution to the problem is to assemble a "satisfactorily good" path from the initial to the final state.

Objective

- To reach a (satisfactory) final state through a satisfactory route
- The solution is either the final state itself, the route itself, or both

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Navigation/Search Method

- To start with the initial state is defined, as well as the final state(s) if known
- Which part of the state space is subsequently explicated depends on the navigation method, based on which operators are selected and applied to the open states
- The part of the state space explicated forms a search tree where each node has one parent as we only need to remember the best way so far found of reaching any state

Operators

premise → action

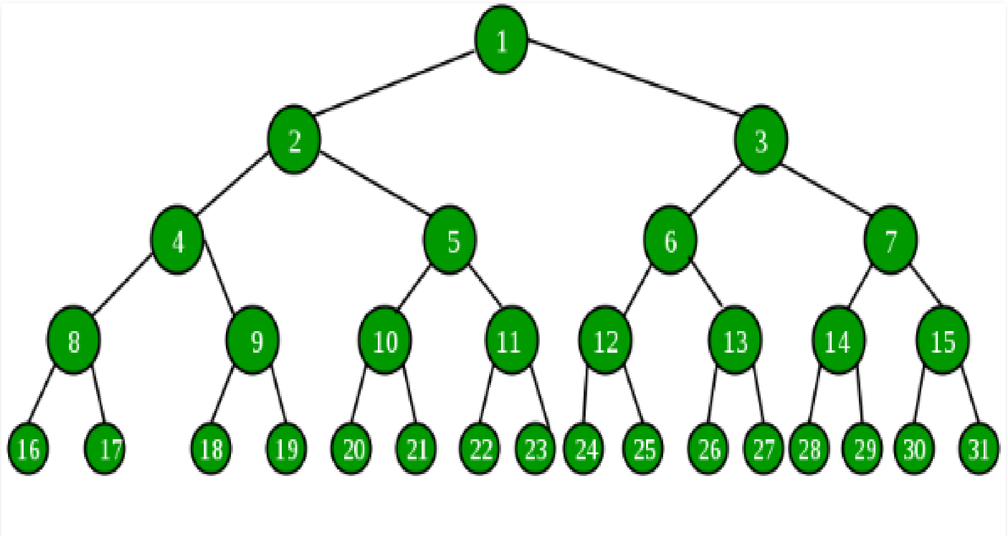
- The premise specifies a condition that must be satisfied by a state as a prerequisite for applying the action to the state
- The application of the action to the state leads to a new (successor) state
- Actions are summative (they only add elements to obtain new states) or are mutative (they could add and/or remove elements to obtain new states)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Combinatorial Explosion

- The search tree grows explosively with increasing depth
- A branching factor of **b** (average number of successors per search node) would give a search tree with $\mathbf{b^n}$ leaf nodes, when the search goes to depth **n**
- In chess:
  - b averages around 30
  - for a complete game, n will be around 100
  - Thus, exploring the whole tree will produce 30^{100} or around $\mathbf{10^{145}}$ nodes, which is considerably more than the number of atoms in the universe

uniform branching factor b = 2; n = 4



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Admissible Search Method

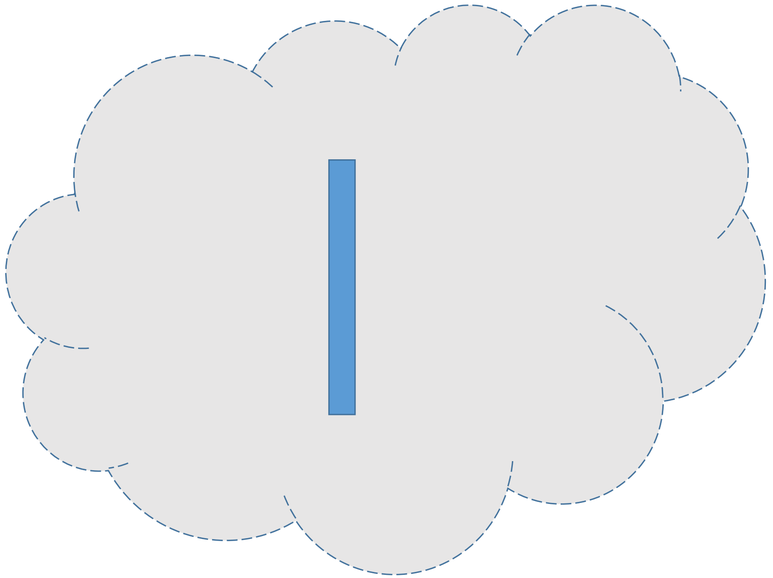
- Guaranteed to find the best solution, if one exists
- Theoretical property of the method

Efficiency of Search Method

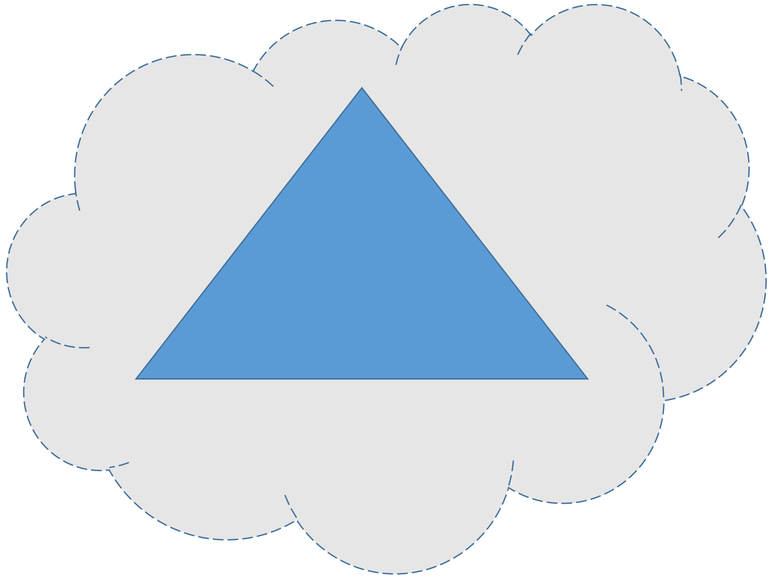
- How much effort is expended in finding a solution
- Efficiency = (number of nodes on solution path)/(number of nodes visited)
- Empirically measured for every instance of the search method performed
  - Get the average over a representative set of trials of the given problem
- An optimal search method would visit only the nodes which turn out to be on the solution path, yielding an efficiency of 1.0

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristics, efficiency and computational overhead



top efficiency; search tree is the solution path
(low memory demand and low time in expanding nodes visited)



low efficiency; search tree explicates a large part of the state space
(high memory demand and high time in expanding nodes visited)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristics, efficiency and computational overhead

- Heuristic deployment needs to strike a balance between the quality of guidance and the computational overhead associated with their use

- High computational overhead defeats the purpose of using heuristics
  - It could enhance the efficiency of search but at a high computational cost
  - Efficiency reflects how targeted the search is towards the problem goal
  - Could be more beneficial to have a higher number of visited nodes, in conjunction with the use of less effective heuristics, if the overall time in obtaining a problem solution is comparatively lower than when using highly effective heuristics

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

General Search Methods

- Depth-First search
- Breadth-First search
- Heuristic search – A*
  - Best-first search – greedy search
  - Branch-and-Bound search
- The minimum requirement is to avoid circular paths, as these lead to infinite search loops
  - There should be progress towards the sought solution

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Definitions

Open search node: non final state which has not been explored yet, or its re-exploring is indicated

Closed search node: a state that has been explored and presently it is of that status

Parent of search node s: a search node, s', leading directly, i.e., through the application of a single action, to search node s and at present the route from the initial search node to s, through s' is considered so far, the best

Successors of search node s: search nodes for which s is (presently) their parent; these could be a subset of the set of states descending directly from s through applicable actions (direct descenders of s which are not vetted as successors of s have other parents)

Deadlock: a non final state that has no successors, or all its successors lead to deadlock

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Search Assumptions

- The navigation starts from the initial problem state (the root node of the search tree) towards a (good enough and not necessarily optimal, if this is not predefined) final state (a leaf node of the search tree), piecing together a good enough route from the root node to the final leaf-node

- Initially there is only one open search node (giving the initial problem state) and there are no closed search nodes

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Depth-First and Breadth-First Search Methods

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Depth-First Search

- Attempts to **quickly penetrate deep** into the state space
- It can reach a final state in an **effective** way
- Its **memory requirement is not excessive**
- However, it **can get lost** in the state space, or reach a deadlock
- And it does not necessarily find an optimal solution - **not admissible**

"Gullible" method: It assumes that a deep-down search node is nearest to a goal state

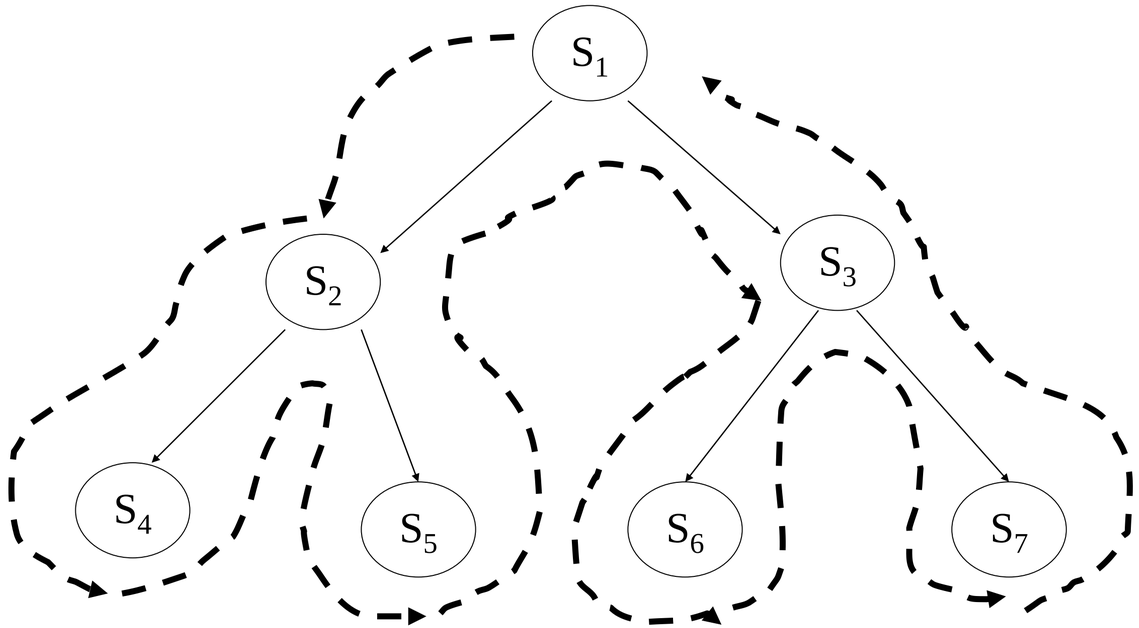
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Backtracking – when deadlock is reached, or the maximum path length is exceeded

- Revoking the last action choice
- The search goes back one step on the path under investigation (parent search node) and a new choice is made
- When all possible choices from the (parent) search node have been investigated, the backtracking continues with the immediately above search node (parent of the parent), etc.
- If the backtracking leads to the root search node (initial state) the entire search so far has been fruitless

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Depth-First Search



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Breadth-First Search

- It is possessed by **excessive skepticism**
- All routes of length N are thoroughly investigated, before a route of length N+1 is investigated, starting from routes of length 1
- It leads to an **optimal solution**, where optimal means ‘at the shortest distance from the initial state’

- The **admissibility** of the breadth-first search method is easily proven
  - Assume that the method finds a solution at depth N, but there is also a solution at depth M where M < N.
  - But all routes at depth M have been thoroughly investigated before moving deeper to depth N
  - Hence if a solution existed at depth M, it would have been discovered

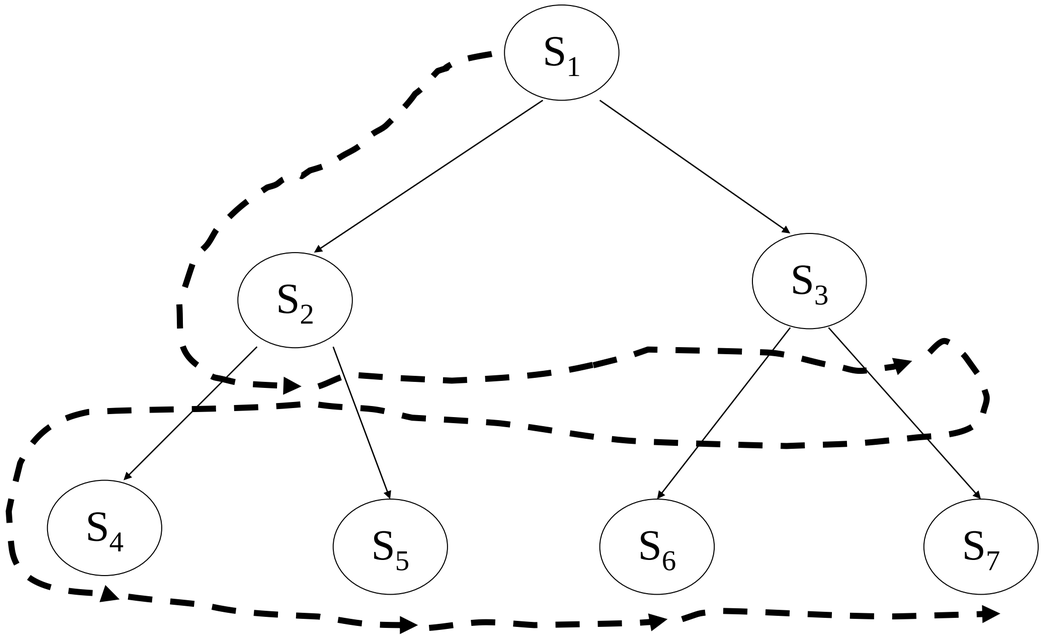
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Limitations of Breadth-First Search

- It leads to **combinatorial explosion**, with exponential demands both in memory space and computational time
- Hence it **cannot be realistically applied** to large state spaces

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Breadth-First Search



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Algorithm for Depth-First Search

1. OPEN := [s_o] , CLOSED := [ ]

2. If OPEN = [ ] terminate the search. There is no solution.

3. Remove the first search node, s_i, from OPEN and add to CLOSED.

4. Based on the actions that can be applied to s_i compute the successors of s_i, that are not already included in OPEN or CLOSED. Every successor node specifies s_i as its parent node.

5. If a successor node of s_i, say s_g, refers to the goal state, terminate the search and return as solution the route from s_o to s_g. Otherwise add the successors at the **start** of list OPEN and repeat from step 2.

It is noted that list OPEN is treated as a **stack**

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Algorithm for Breadth-First Search

1. OPEN := [s_o] , CLOSED := [ ]

2. If OPEN = [ ] terminate the search. There is no solution.

3. Remove the first search node, s_i, from OPEN and add to CLOSED.

4. Based on the actions that can be applied to s_i compute the successors of s_i, that are not already included in OPEN or CLOSED. Every successor node specifies s_i as its parent node.

5. If a successor node of s_i, say s_g, refers to the goal state, terminate the search and return as solution the route from s_o to s_g. Otherwise add the successors at the **end** of list OPEN and repeat from step 2.

It is noted that list OPEN is treated as a **queue**

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Depth-First and Breadth-First: How do they differ?

- In depth-first the list of OPEN search nodes is a **stack**, while in breadth-first it is a **queue**
- Hence in depth-first, a successor of the last search node explored, is the next search node to be explored, while in breadth-first the successors of the last search node explored will be explored once all search nodes, already in the OPEN list, are explored
- This difference results in the breadth-first search being unduly skeptical (nothing is overlooked) and in the depth-first search being unduly gullible (forcefully pursuing a single path until it succeeds or fails)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Search trees and exploring open search nodes

- At any time, before a goal state is reached, a search tree includes several paths:
  - Paths ending at CLOSED search nodes are not under consideration
    - If all paths are of this category, there is no solution
  - Paths ending at OPEN search nodes can be potentially expanded further

- In depth-first and breadth-first, the successors of an OPEN search node are ordered in a pre-determined way, depending on the order of the (applicable) actions
  - Why search nodes already OPEN or CLOSED are not included in the successors of the currently explored search node?

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Why in depth-first and breadth-first, search nodes already OPEN or CLOSED are not included in the successors of the currently explored search node?

- Recall that at any time, the search tree keeps the best path so far from the root node to some search node
- Breadth-first: if a successor, **s**, of the currently explored search node, **p**, is already OPEN or CLOSED, a new path from the root node to **s** via **p** is found. Is this new path better, i.e., shorter, than the presently kept path from the root to **s**? If the length of the present path is N and of the new path is M, could M < N? This is not possible as all paths of length smaller than N would have been investigated prior to investigating paths of length N.
- Depth-first: If the currently explored search node, **p**, is at depth N from the root node, no other OPEN search node is at a bigger depth. If a successor, **s**, of **p** is already OPEN at depth M, it follows that M ≤ N, and hence the new path to **s** via **p**, is worse as it has length N+1. If **s** is CLOSED, it is either on an abandoned path, or on the pursued path from the root to **p**, hence re-OPENing it with **p** as its parent would create a circular path.

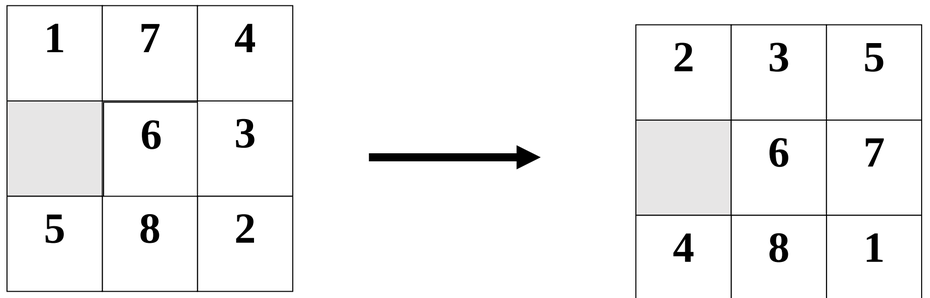
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: 8-puzzle

In the 8-puzzle game, eight tiles numbered from 1 to 8 are placed in a 3×3 board. Hence there is always a blank space.

The rules of the game allow the sliding of tiles one place up, down, left, or right, into the blank space.

The problem is to determine a sequence of sliding moves for converting a given initial configuration of the eight tiles (initial state), into some other given configuration (final or goal state). An instance of the problem is shown below:



initial state

final state

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Solving the representation problem for the 8-puzzle

- Define a structure for the problem states
  - 3×3 array of integers – 0 represents the blank space
- Define the operators (which are listed and evaluated in some order)
  - If tile 1 is above the blank space, slide it down into the blank space
  - If tile 1 is on the left of the blank space, slide it right into the blank space
  - If tile 1 is on the right of the blank space, slide it left into the blank space
  - If tile 1 is below the blank space, slide it up into the blank space
  - . . . . . .
  - If tile 8 is above the blank space, slide it down into the blank space
  - If tile 8 is on the left of the blank space, slide it right into the blank space
  - If tile 8 is on the right of the blank space, slide it left into the blank space
  - If tile 8 is below the blank space, slide it up into the blank space

Is it a good representation to have 32 very specific operators?

- The problem state structure and the operators define the state space
- Choose the search method

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Solving the representation problem for the 8-puzzle

- Define a structure for the problem states
  - 3×3 array of integers – 0 represents the blank space
- Define the operators (which are listed and evaluated in some order)
  - U: If there is a tile <T> above the blank space, slide tile <T> down into the blank space
  - D: If there is a tile <T> below the blank space, slide tile <T> up into the blank space
  - L: If there is a tile <T> on the left of the blank space, slide tile <T> right into the blank space
  - R: If there is a tile <T> on the right of the blank space, slide tile <T> left into the blank space

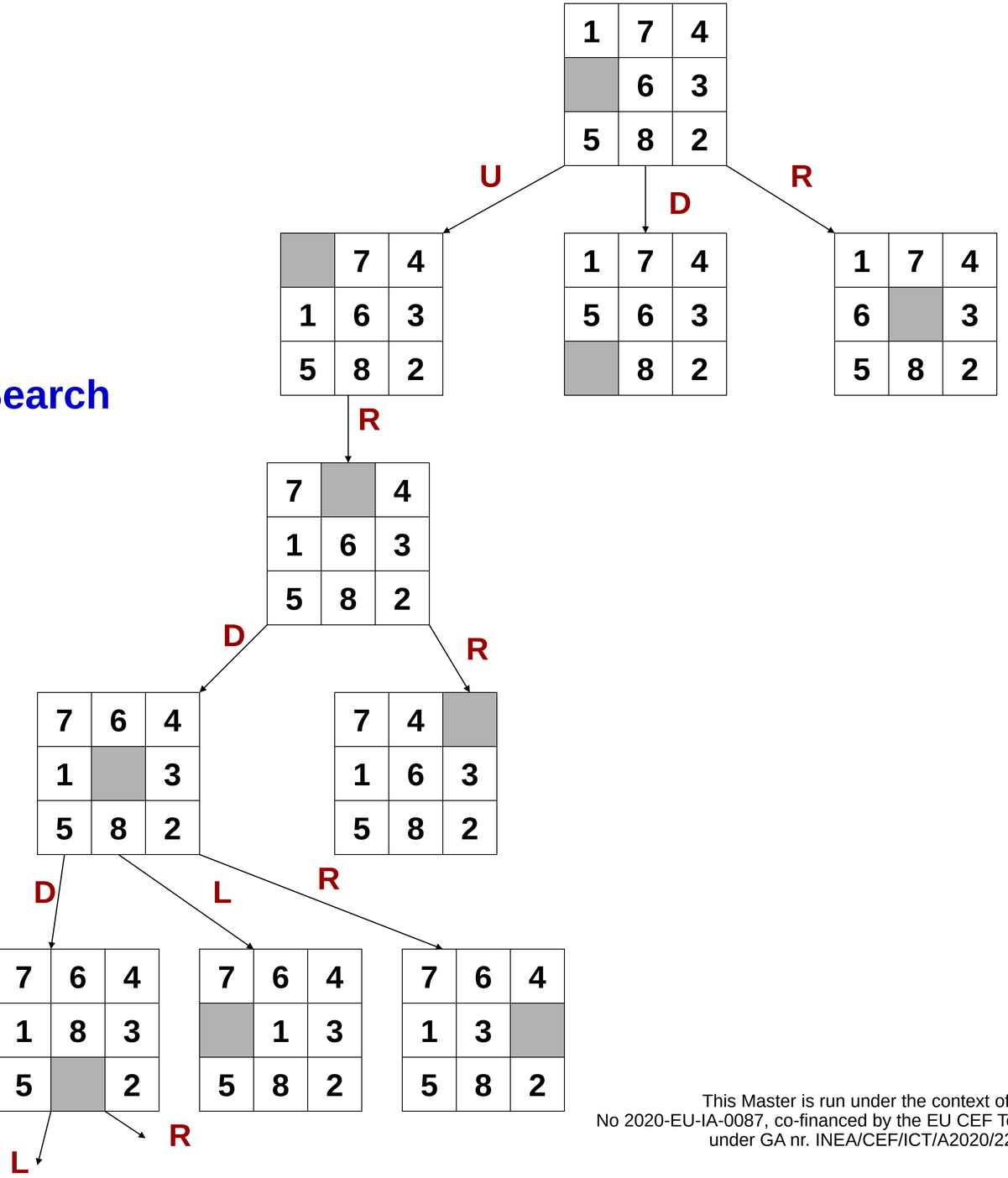
  - Only 4 general rules focusing on the position of the blank space:
    - <T> is a variable denoting any tile, that is appropriately instantiated when operators are applied

- The problem state structure and the operators define the state space
- Choose the search method
  - Depth-first or breadth-first
  - In either method the operators are ordered as U, D, L, R

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

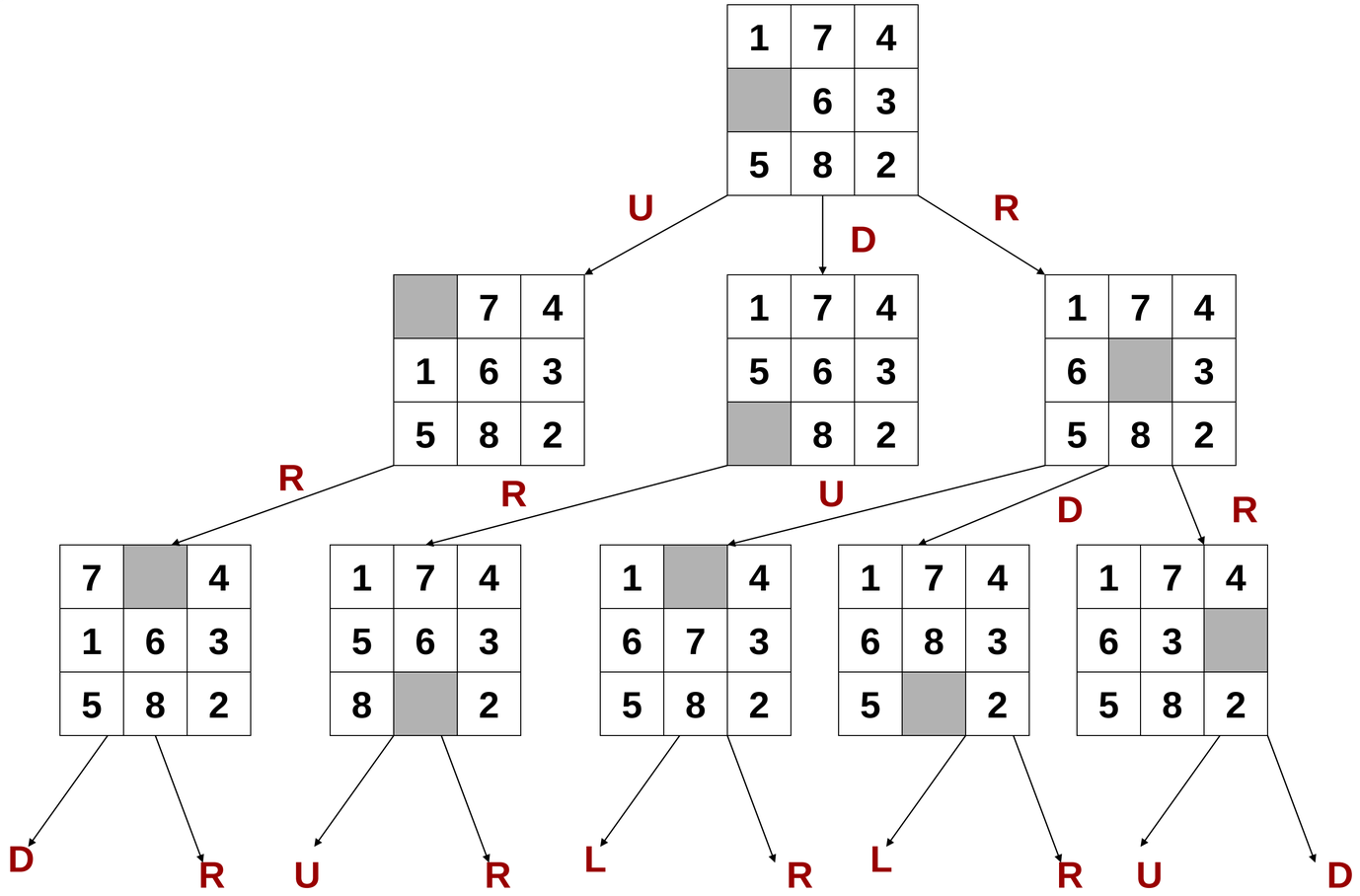
MAI4CAREU

Depth-First Search



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423



Breadth-First Search

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Some instances of the 8-puzzle cannot be solved

A pair of tiles in the initial state form an inversion if the values on the tiles are in reverse order of their appearance in the goal state.

If the number of inversions is even the given instance of the 8-puzzle is solvable.

1	8	2
	4	3
7	6	5

1	2	3
4	5	6
7	8	

solvable

10 inversions

1, 8, 2, 4, 3, 7, 6, 5 initial state flattening
1, 2, 3, 4, 5, 6, 7, 8 goal state flattening

Inversions
(8,2) (8,4) (8,3) (8,7) (8,6) (8,5)
(4,3)
(7,6) (7,5)
(6,5)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Some instances of the 8-puzzle cannot be solved

A pair of tiles in the initial state form an inversion if the values on the tiles are in reverse order of their appearance in the goal state.

If the number of inversions is odd the given instance of the 8-puzzle cannot be solved.

8	1	2
	4	3
7	6	5

1	2	3
4	5	6
7	8	

not solvable

11 inversions

8, 1, 2, 4, 3, 7, 6, 5 initial state flattening
1, 2, 3, 4, 5, 6, 7, 8 goal state flattening

Inversions
(8,1) (8,2) (8,4) (8,3) (8,7) (8,6) (8,5)
(4,3)
(7,6) (7,5)
(6,5)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Heuristic Search – Algorithm A* and its variants Branch-and-Bound and Best-First Search Methods

Co-financed by the European Union
Connecting Europe Facility

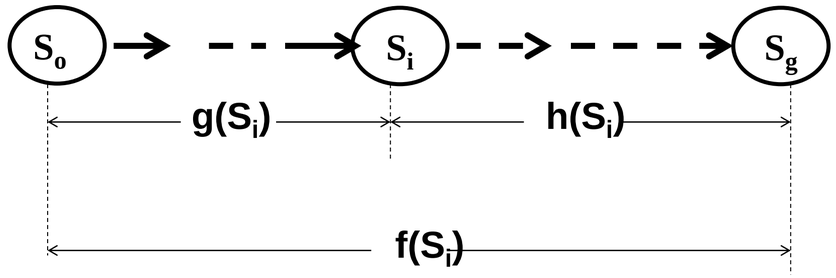
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristic Search

- Aims to combine the strengths of the depth-first and breadth-first methods and to alleviate their serious weaknesses
- The OPEN search nodes are “evaluated” and every time the search node investigated is the one that appears to be the most promising, i.e., the one that appears to be on the best path towards the solution
- Balances depth and breadth of search through heuristic guidance
  - Avoid getting into fruitless depth
  - Avoid combinatorial explosion
- How are OPEN search nodes evaluated?

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Search Node Evaluation



The evaluation function, f, is defined as:

$$f(S_i) = g(S_i) + h(S_i)$$

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Search Node Evaluation

- **Function g gives the real transition 'cost' from s_o to s_i:**
  - Every operator has a cost – this could be a unit cost for all operators
  - Since the best path so far from s_o to s_i is known, the real cost is known
  - $g(s_o) = 0$

- **The heuristic function, h, 'guesses' the transition cost from s_i to s_g:**
  - Hence, function h says how promising the problem state represented by search node s_i is, i.e., how appropriate it would be to search for a solution through s_i
  - The reliability of the evaluation function f depends on the reliability of the heuristic function h
  - If h overestimates the cost, there is a risk to ignore s_i
  - However, if the cost is underestimated, there is a risk to get jammed into a fruitless search
  - $h(s_g) = 0$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Admissible Search

- If for every problem state the heuristic function does not overestimate the real transition cost, its use guarantees optimal solution and hence the heuristic search is admissible
- For example, breadth-first search is admissible since it can be taken that it uses the heuristic function h(s) = 0, for any problem state s

Heuristic function, h

- Function h is a ‘black box’
- It could consist of a set of rules or some procedure, which when evaluated give back a number
- It is problem specific

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristic Search: Algorithm A*

1. OPEN := [s_o], CLOSED := [ ]
2. If OPEN = [ ] terminate. There is no solution.
3. Remove search node, s_i, from OPEN, for which $f(s_i) \leq f(s_j)$ for all other OPEN search nodes s_j and add it to CLOSED.
4. Compute the successors of s_i, and consider s_i, as the parent of each of them.
5. If s_g is amongst the successors of s_i, terminate and return the route from s_o to s_g.
6. Otherwise repeat for every successor, s_j, of s_i:
   - 6.1 Compute the value of $f(s_j)$.
   - 6.2 If no search node with the same problem state as s_j is in OPEN, or CLOSED, add s_j in OPEN.
   - 6.3 If a search node s_k with the same problem state as s_j is already in OPEN or CLOSED, compare $f(s_j)$ with $f(s_k)$. If $f(s_j) \geq f(s_k)$ discard search node s_j. Otherwise, change the parent of search node s_k to search node s_i and if s_k is CLOSED transfer it to OPEN.
7. Repeat from 2.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Remarks on A*

- OPEN search nodes are evaluated by function f and the search node with the smallest f value is selected for further exploration.
- For the root search node s_o representing the initial problem state, $f(s_o) = h(s_o)$ and for the search node s_g representing the goal state $f(s_g) = g(s_g)$.
- Lists OPEN and CLOSED have the same role as for depth-first and breadth-first. However, in A*, OPEN is an ordered list.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

When in A* the explored search node, s, could become the new parent of an existing node, n, in the search tree

- Is the path from the root node to n, via s, better than the path from the root node to n as currently depicted in the search tree?
  - Yes, if the evaluation function gives a smaller value for the new path to n, as a result of the smaller value of the g function; the value of the h function is independent of the path from the root to node n

- If the path via s is better, the parent of n should be changed to s

- Moreover, if n is CLOSED, now that a better path to it via s is found, it should be re-OPENed so that if its now higher promise leads to its (re)exploration, its own successors will be (recursively) reassessed, possibly reinstating n as the parent of some of its successors that currently are given other search nodes as parents

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

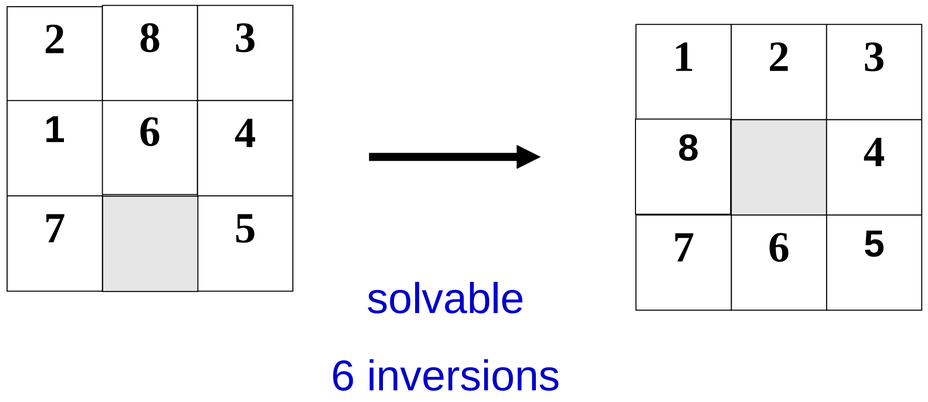
Branch-and-Bound and Best-First Search Methods

- Can be considered variants of A*
- Recall the evaluation function: $f(S_i) = g(S_i) + h(S_i)$
- Branch-and-Bound search
  - The heuristic function is 0 for all problem states: h(s) = 0
  - Only the g function is used in evaluating the OPEN search nodes
    - However, unlike in the breadth-first search where minimum depth is considered and hence all operators are taken to have unit cost, here operators can have varying costs and a solution with a minimum accumulated cost is sought, e.g., consider a map traversal problem where distances between cities represent the cost of traversal operators
    - Hence it can be considered the generalization of breadth-first
    - It is admissible
- Best-First search
  - The g function is 0 for all problem states: g(s) = 0
  - Only the heuristic function h is used in evaluating the OPEN search nodes
    - Hence the search node that is estimated to be nearest the goal state is selected
    - A non admissible, greedy method

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Solving a particular instance of the 8-puzzle with the various search methods



2, 8, 3, 1, 6, 4, 7, 5 initial state flattening
1, 2, 3, 8, 4, 7, 6, 5 goal state flattening

Inversions
(2,1)
(8,3) (8,1)
(3,1)
(6,4) (6,7)

- Every operator has unit cost
- The heuristic function counts the number of tiles out of place – mismatch between some problem state and the goal state
  - It underestimates the true cost making the A* search method using it admissible

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Breadth-First and Branch-and-Bound

Starting Search
==========
Search Succeeds
Efficiency 0.10344828
Solution Path
Node with state
2 8 3
1 6 4
7 0 5

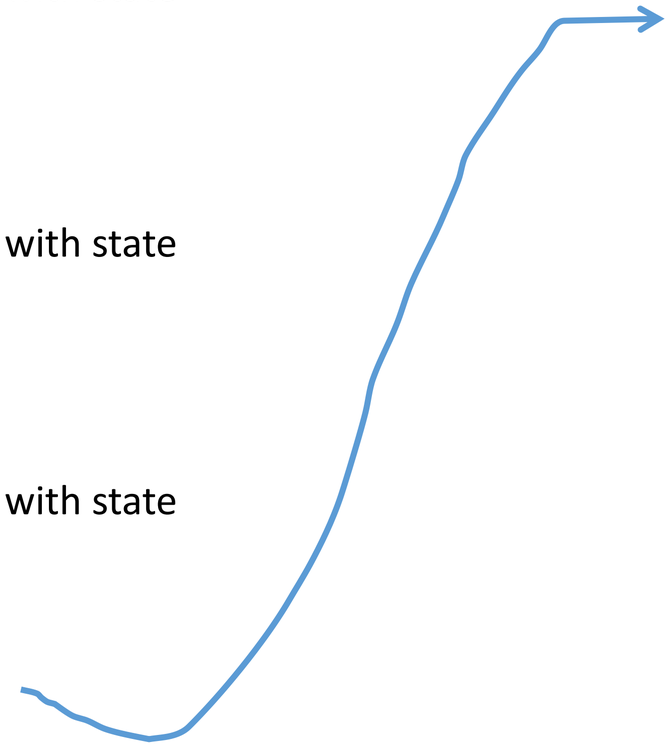
Node with state
2 8 3
1 0 4
7 6 5

Node with state
2 0 3
1 8 4
7 6 5

Node with state
0 2 3
1 8 4
7 6 5

Node with state
1 2 3
0 8 4
7 6 5

Node with state
1 2 3
8 0 4
7 6 5



Efficiency

The solution path has 6 search nodes and in total 58 search nodes were visited, i.e., expanded

Co-financed by the European Union
Connecting Europe Facility

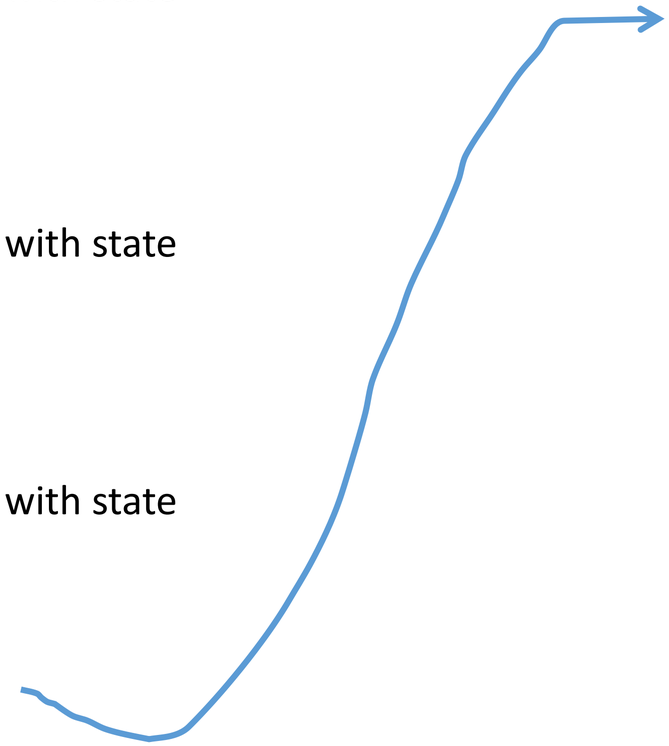
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

A* and Best-First

Starting Search
==========
Search Succeeds
Efficiency 0.85714287
Solution Path
Node with state
2 8 3
1 6 4
7 0 5

Node with state
2 8 3
1 0 4
7 6 5

Node with state
2 0 3
1 8 4
7 6 5



Node with state
0 2 3
1 8 4
7 6 5

Node with state
1 2 3
0 8 4
7 6 5

Node with state
1 2 3
8 0 4
7 6 5

Efficiency

The solution path has 6 search nodes and in total only 7 search nodes were visited, i.e., expanded – it is almost the perfect efficiency of 1.0

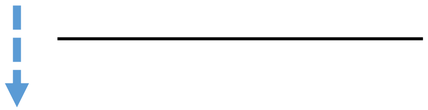
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Depth-First

Starting Search
==========
Search Succeeds
Efficiency 0.9555548
Nodes visited: 29317

Solution Path
Node with state
2 8 3
1 6 4
7 0 5



28012 intermediate states!!!

Node with state
1 2 3
8 0 4
7 6 5

Efficiency is high but solution is grossly away from optimality

The solution path has 28014 search nodes and in total 29317 search nodes were visited giving an efficiency of 0.9555548 that in this case does not reflect the quality of the search

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Object-Based Design for a Generic Search Method

(See Appendix for Java code)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Key Objects

- **Problem_State**
  - Abstract object capturing the possible situations of some problem, thus (implicitly) capturing a state space
  - Concrete extensions represent the states of given problems

- **Search_Node**
  - Concrete object representing the nodes of a search tree; a search tree explicates part of a state space in accordance with the deployed search method
  - It encapsulates a Problem_State

- **Search**
  - Abstract object capturing the generic features of the search methods discussed
  - Concrete extensions represent additional, problem specific elements

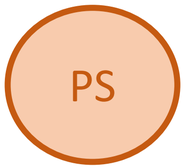
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Abstract Object Problem_State

Functionality through abstract methods, to be concretely defined in its extensions:

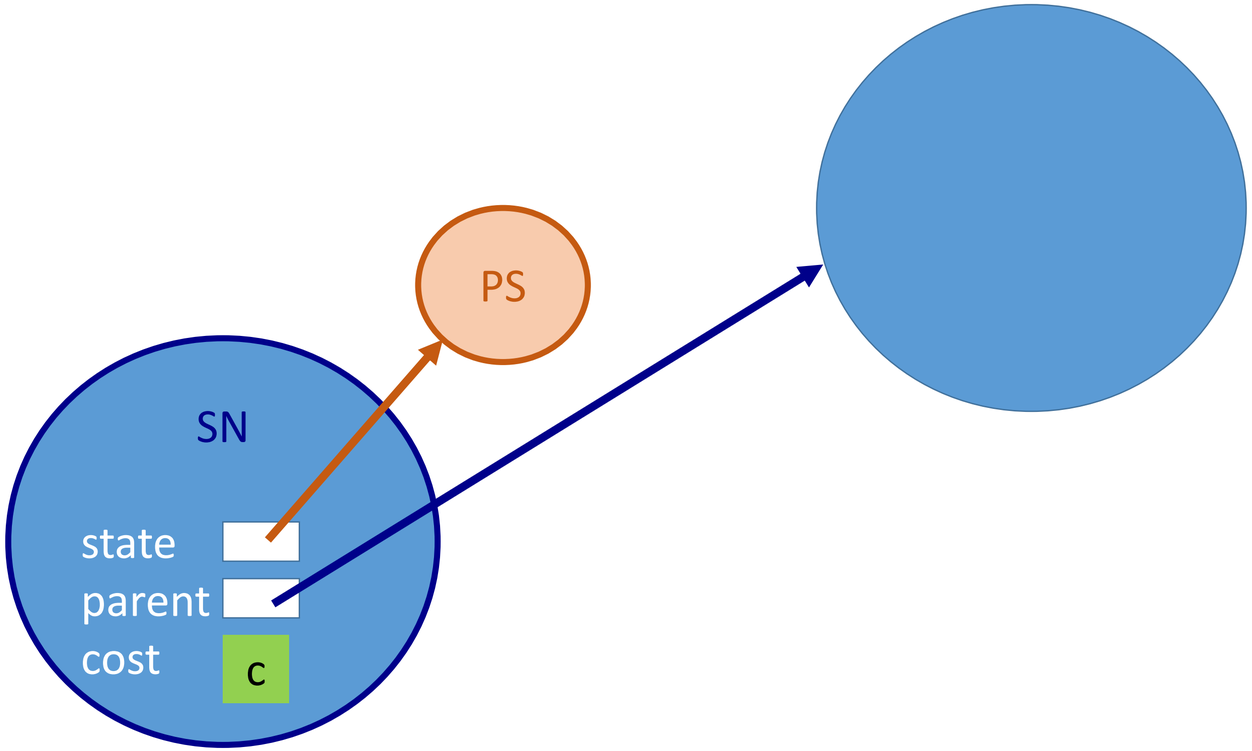
- ❑ Is it a goal state under a given Search?
- ❑ What are its successor Problem_States in a given Search (i.e., its direct descendants)?
- ❑ Is it the same as another Problem_State?
- ❑ What is the transition cost from a given Problem_State to this? (relevant operator cost)
- ❑ What is the estimated cost from this to a goal Problem_State in a given Search?



Co-financed by the European Union
Connecting Europe Facility

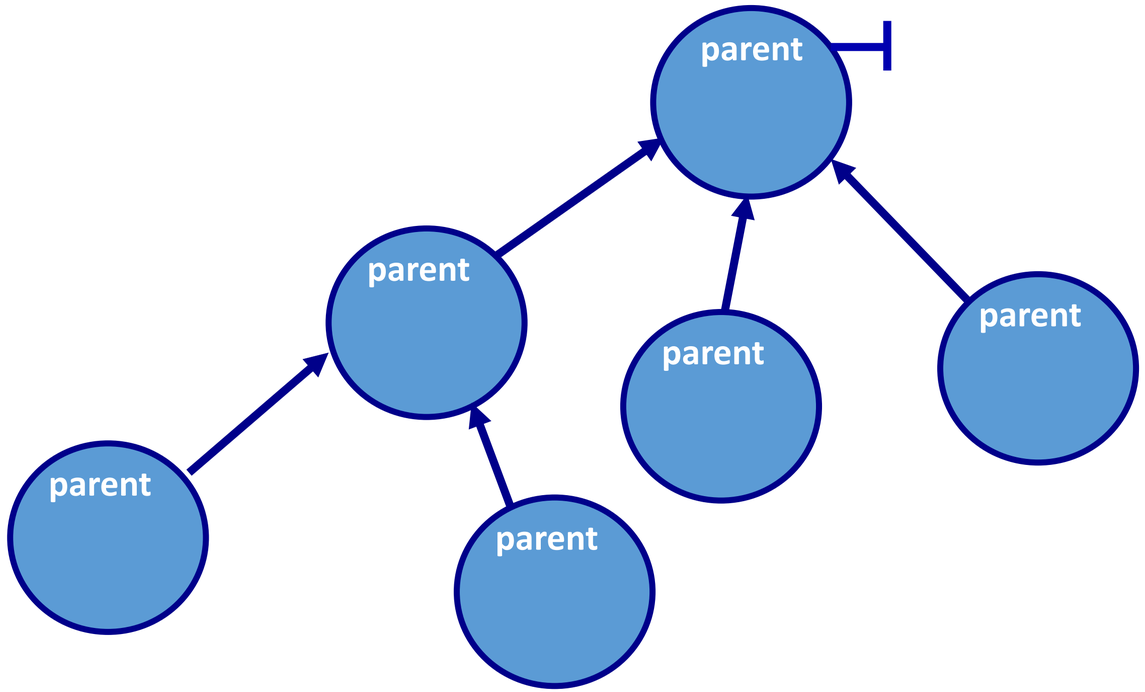
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Concrete Object Search_Node



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Linking together a search tree through ‘parents’



At any time, each node N keeps its ‘best’ parent, P, i.e., that node P on the best route (less costly) so far from the root node to N; hence the parent of a node can subsequently change as the search for a solution progresses

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Concrete Object Search_Node

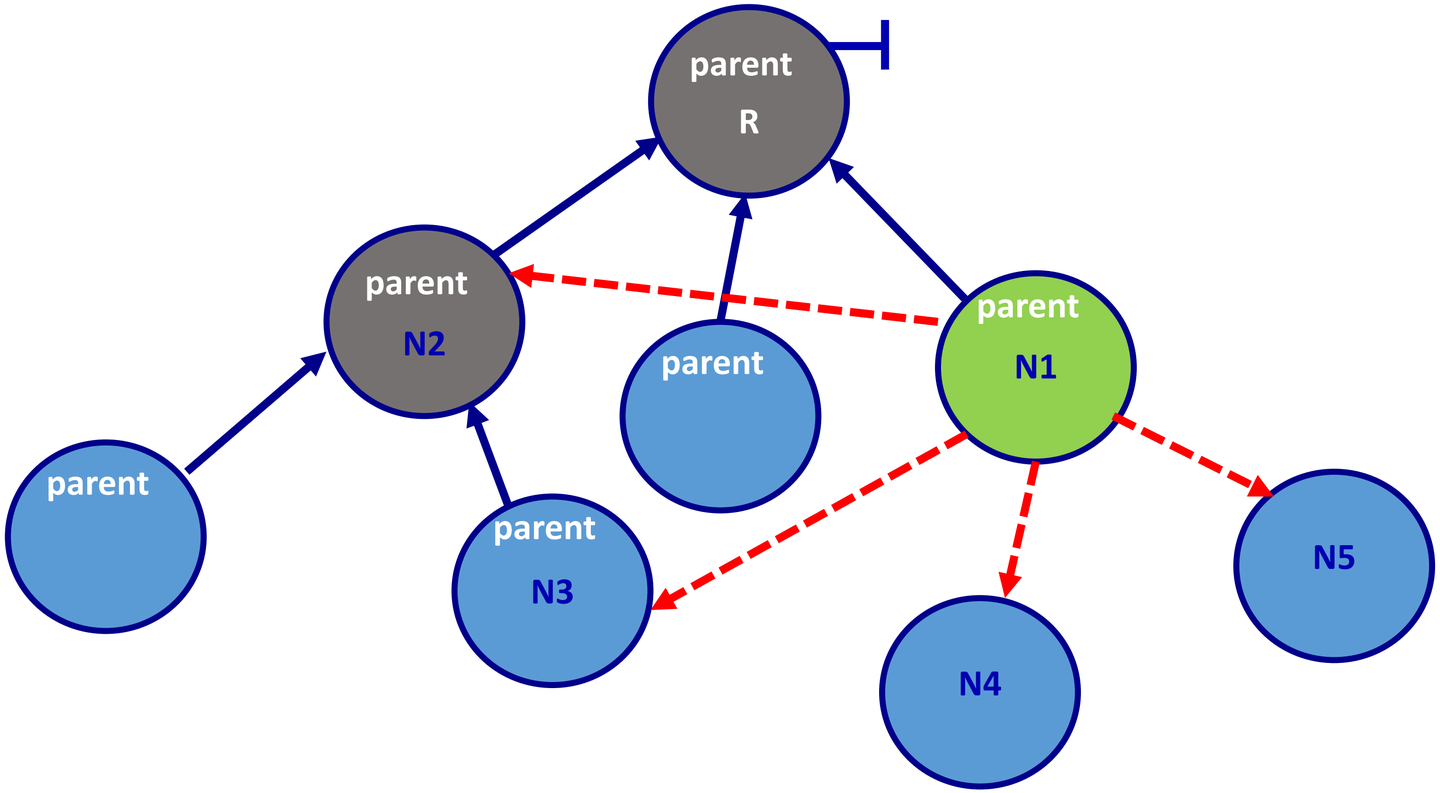
Functionality through concrete methods:

- ❑ What is its state, current parent, transition cost from its parent?
- ❑ Does it encapsulate the same state as another Search_Node?
- ❑ Change its parent to a new one
- ❑ Change the transition cost (i.e., when its parent changes)
- ❑ Does it encapsulate a goal Problem_State in a given Search?
- ❑ What is the (current) best path cost from it to the root Search_Node?
- ❑ What is the estimated transition cost from it to a goal Problem_State in a given Search?
- ❑ What is its overall promise in a given Search towards a solution?
- ❑ What are its successor Search_Nodes in a given Search (each of these designate it as its parent)
- ❑ How is it externally expressed?

Co-financed by the European Union
Connecting Europe Facility

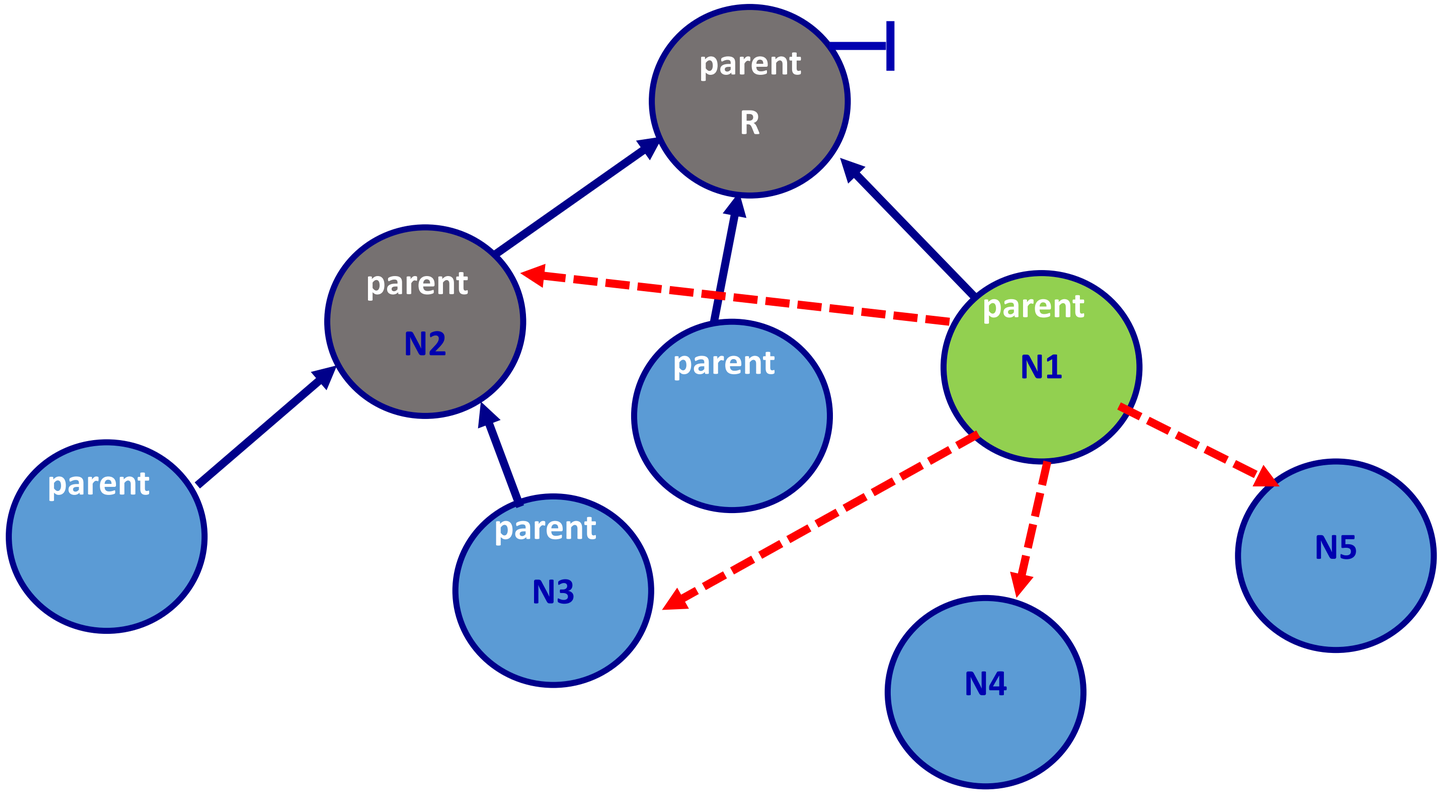
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Extending the search tree



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

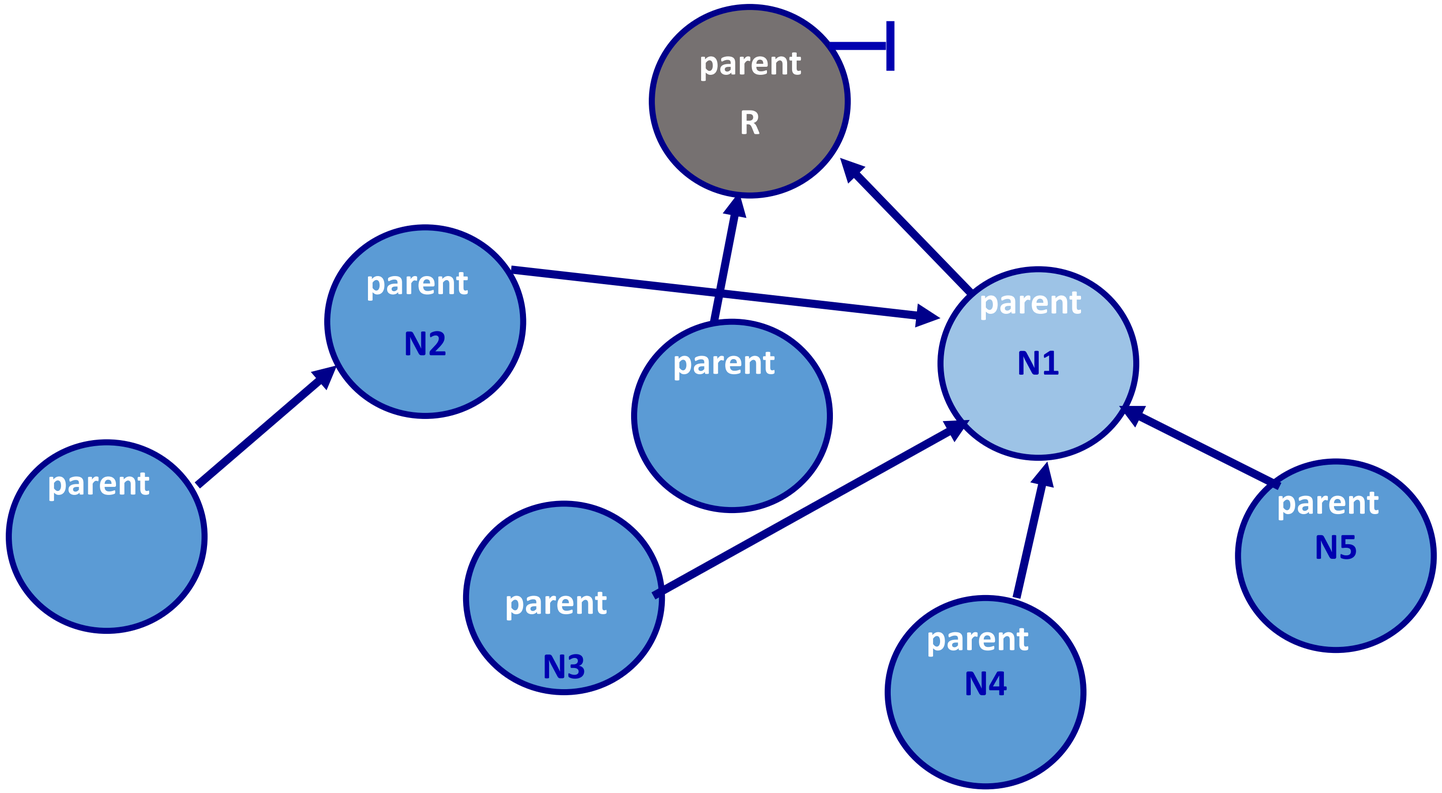
Extending the search tree



- OPEN node N1 is the current search node; its direct descendants (potential successors) are the existing CLOSED node N2, the existing OPEN node N3 and two new search nodes N4 and N5
- N4 and N5 will be added in the OPEN list and N1 will be designated as their parent; so far only one route to them from the root node, R, has been found and this is through N1
- Regarding N3 the question raised is whether its parent should be changed from N2 to N1; this would depend on whether the new route R -> N1 -> N3 is better than the existing route R -> N2 -> N3
- Regarding N2 again the question raised is whether the new route R -> N1 -> N2 is better that the existing (direct) route R -> N2 (given that operators can have varying costs, the direct transition could be a rough passage whilst the combined transitions R -> N1 and N1 -> N2 could be much smoother, faster and cheaper); if the answer is yes, N1 becomes the new parent of N2 and at the same time N2 is reOPENed, and this recursively could result in some direct descendants of N2 that currently do not have N2 as their parent in the search tree, to change their parent to N2

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Resulting extensions/changes



- Let's say that the new route R -> N1 -> N3 is better than the existing route R -> N2 -> N3; hence N2 loses N3 as a successor
- Let's also say that the new route R -> N1 -> N2 is better that the existing (direct) route R -> N2 and N2 becomes a successor of N1 and is reOPENed, for potential future investigation; Could this reinstate N3 as a successor of N2? In other words, could the route R -> N1 -> N2 -> N3 be better than the (existing route) R -> N1 -> N3?

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Abstract Object Search

Encapsulates

- The root Search_Node
- The current Search_Node and its successor Search_Nodes
- The OPEN and CLOSED Search_Nodes
- If known, the goal Problem_State

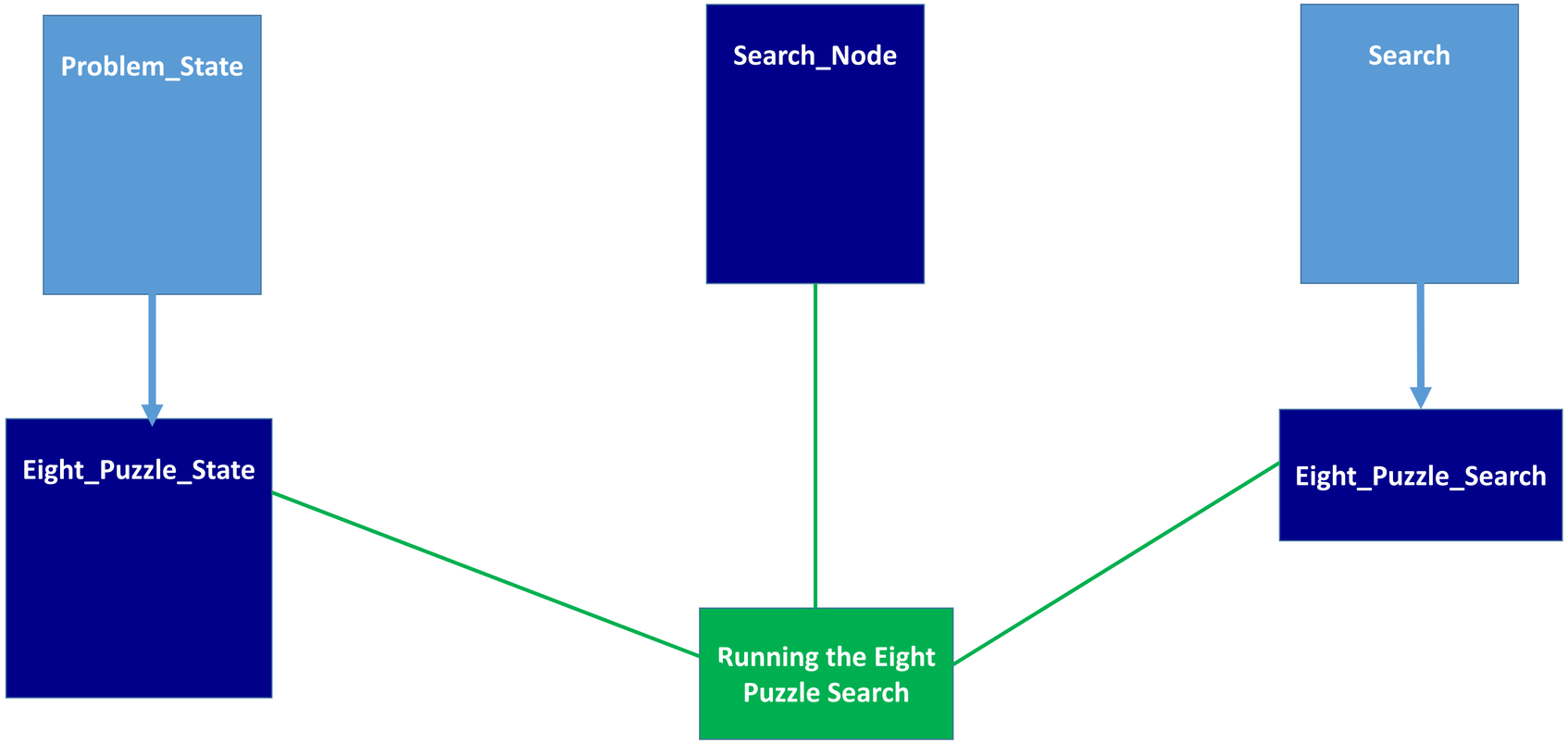
Functionality

- Running the Search given an initial Problem_State, and possibly a goal Problem_State, plus the name of the search method to be used (depth_first, breadth_first, A_star, …..), until either there are no OPEN Search_Nodes (**unsuccessful search**) or a Search_Node encapsulating a goal Problem_State is found (**successful search**) :
  - Selecting current Search_Node
  - Expanding current Search_Node that includes the vetting of its successors, and processing vetted successors
  - Reporting success or failure and in the case of success displaying the solution path from the root Search_Node to the (discovered) goal Search_Node

Co-financed by the European Union
Connecting Europe Facility

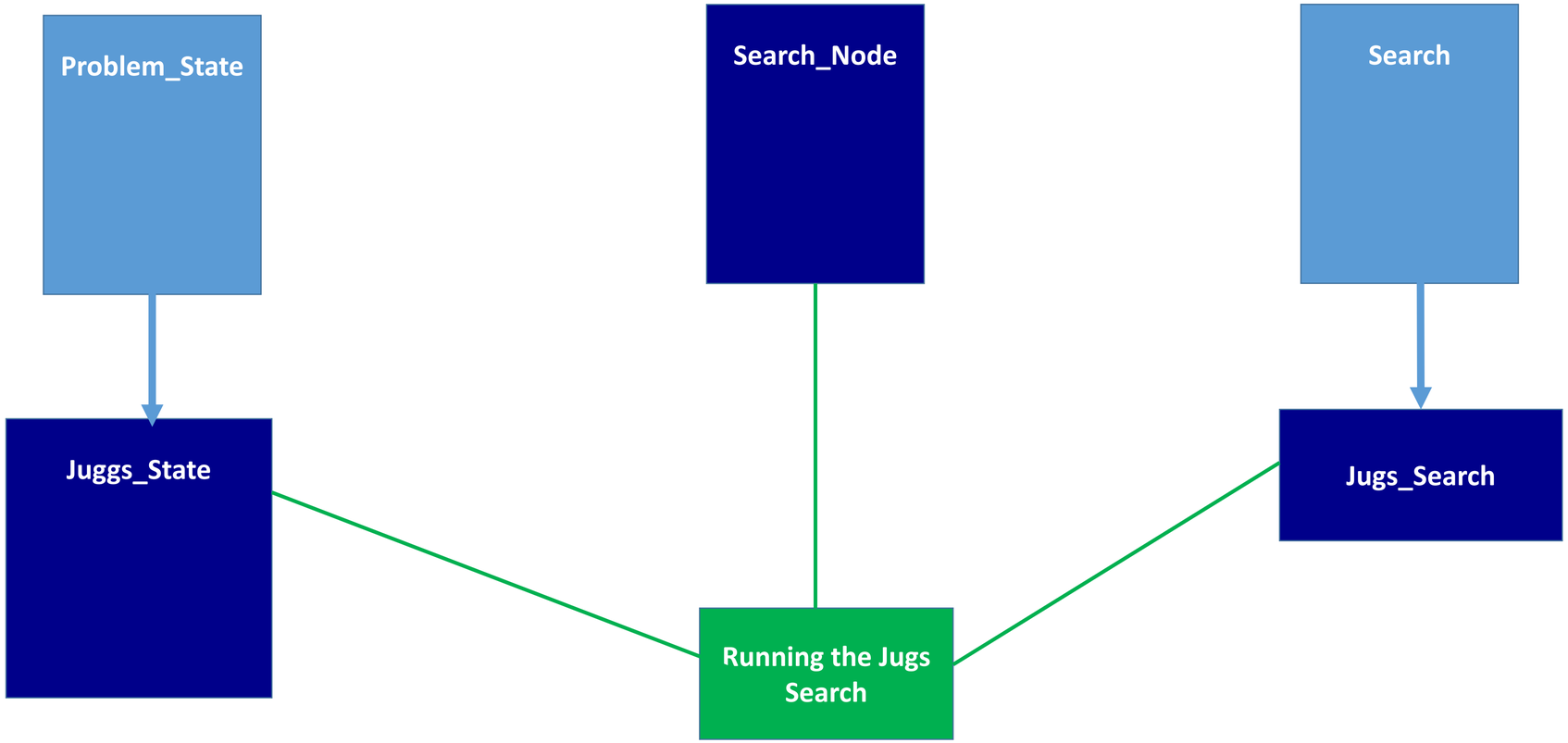
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Using above objects to solve specific problems



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Using above objects to solve specific problems



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Frame Problem

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Frame Problem – a term coined by McCarthy and Hays

- It arose in connection to **logic-based representations** of problem states/situations
- In such cases, deducing new sentences from the given axioms and sentences is a monotonic (additive) process where the new sentences are just added to a state/situation
- But in many problem domains, e.g., in planning problems or robotics, **actions are not monotonic**, i.e., their effects on a problem state are both additive and subtractive.
- The **frame problem** is how to generally describe the effects of such non-monotonic actions, that is how to describe the changes on a problem state from the application of a non-monotonic action
  - Adding and/or deleting sentences
  - Such changes tend to be very local as the bulk of the sentences composing a problem state generally remain unaffected; hence the challenge is how to represent the effects of actions in logic without having to represent explicitly many intuitively obvious non-effects

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Frame Problem – is this still a problem?

- The frame problem attracted much attention from philosophers and epistemologists who viewed it from a general, theoretical perspective regarding human cognition and theories of mind
  - Particularly those who had approached AI very skeptically/critically
  - And very much consider the frame problem as an **open problem** that needs to be solved for AI to have an epistemologically viable basis

- In AI, the frame problem has always been considered a technical problem, namely an issue concerning symbolic, logic-based problem representations, and how classical approaches can be extended/reformulated to deal with the non-monotonicity of actions
  - Many viable solutions have been proposed and it is now considered a **solved problem**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Classification and Synthesis Problems

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Classification and Synthesis Problems

- A **classification problem** is solved by selecting a solution from a predefined set of solutions or solution categories
  - The problem is recognized or classified as an instance of a given class
  - Hence classification problems are also known as **recognition problems**, as the essence is to derive/recognize a goal state as such a state is not known in advance
- In contrast the solution of a **synthesis problem** is constructed from simpler elements
  - Hence synthesis problems are also known as **constructive or configuration problems**
  - Often the essence is to piece together a plan of actions to generate a goal state which is predefined concretely or as a set of constraints

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

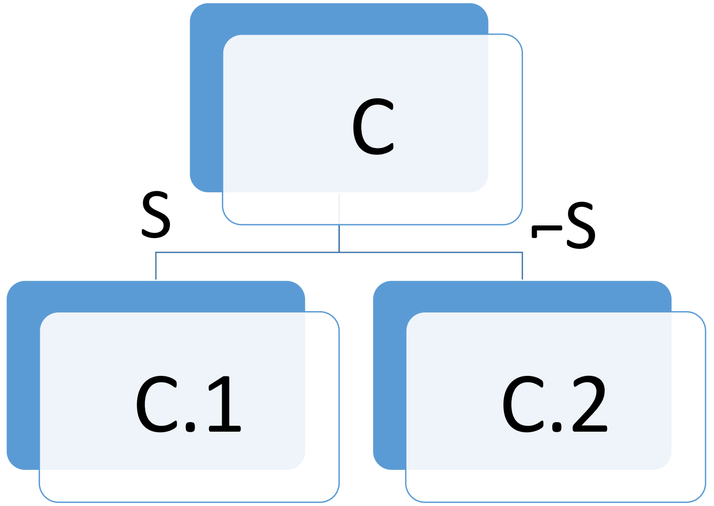
Classification Problems

- The **state space is hierarchical**, where the solution categories form the leaf problem states
  - The hierarchy needs to be specified in advance
- The **goal state is sought** amongst the leaf problem states
  - The goal state is not known in advance since the ultimate objective is to recognize it
- The general operator is the **differentiation operator**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Differentiation Operator

- If symptom S holds, replace C with C.1
- If symptom S does not hold, replace C with C.2



Symptom S constitutes the differentiation factor between the two subclasses

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Search Mechanism for Classification Problems

- The search mechanism for classification problems is referred to as **top-down refinement** or as **test and select**
  - Every non-terminal state represents a decision, which is taken based on some test expressed through the operators
  - E.g., decision-trees

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Synthesis Problems

- The problem solution is constructed or assembled, step by step, from simpler elements defined in advance
- The state space is not concretely defined
- The route from the initial to the final state is the solution as it represents the sequence of actions for constructing the final state
  - E.g., the 8-Puzzle belongs to this category of problems
- The solution construction is underlined by various **constraints**, such as temporal, spatial, or other constraints

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Combining Classification and Synthesis

- In real problems, classification and synthesis can be combined where part of the problem is solved through classification and part through synthesis
- For example, a high level (abstract) plan of action is selected through classification, and this is subsequently concretized through synthesis

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Summary

- Algorithms and Heuristics
- Representation Problem
- Depth-First and Breadth-First Search Methods – Blind Methods
- Heuristic Search – Algorithm A* and its variants Branch-and-Bound and Best-First Search Methods
- Generic, Object-Based Generic Method
- Frame Problem
- Classification and Synthesis Problems

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Appendix:

Java implementation of Problem_State, Search_Node and Search

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

```
import java.util.*;

public abstract class Problem_State {
   abstract boolean goalP(Search searcher);
   abstract ArrayList<Problem_State> get_Successors(Search searcher);
   abstract boolean same_State(Problem_State n2);
   abstract int cost_from (Problem_State from);
   abstract int difference (Problem_State goal);
}
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
import java.util.*;

public class Search_Node{

 private Problem_State state;
 private Search_Node parent;
 private int cost;
 public Problem_State get_State(){return state;}
 public Search_Node get_parent(){return parent;}
 public void put_parent(Search_Node p){parent = p;}
 public int get_cost(){return cost;}
 public void put_cost(int c){cost = c;}

 public Search_Node(Problem_State s, Search_Node p){
  state = s; parent = p; cost = 1;
 }

 public Search_Node(Problem_State s, Search_Node p, int c){
  state = s; parent = p; cost = c;
 }

 public boolean goalP(Search searcher){
    return state.goalP(searcher);
 }

 public int difference (Problem_State goal){
    return state.difference(goal);
 }

 public int best_path_cost(){
  int cos = 0;
  Search_Node n = this;
  while (n.parent != null){
    cos += n.cost;
    n = n.parent;
  }
  return cos;
 }

 public int evaluation_fn(Problem_State goal){
  return difference(goal) + best_path_cost();
 }

 public ArrayList<Search_Node> get_Successors(Search searcher){
  ArrayList<Problem_State> slis = state.get_Successors(searcher);
  ArrayList<Search_Node> nlis = new ArrayList<Search_Node>();
  for (Problem_State suc_state: slis){
        Search_Node n = new Search_Node(suc_state, searcher.get_current_node(),
                                suc_state.cost_from(searcher.get_current_node().get_State()));
        nlis.add(n);
  }
  return nlis;
 }

 public boolean same_State(Search_Node n2){
  return state.same_State(n2.get_State());
 }

 public String toString(){
  return "Node with state " + state.toString();
 }

}
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
import java.util.*;

public abstract class Search {

 protected Search_Node init_node;
 protected Search_Node current_Node;
 protected ArrayList<Search_Node> open;
 protected ArrayList<Search_Node> closed;
 protected ArrayList<Search_Node> successor_nodes;
 protected Problem_State goal_state;

 public Problem_State get_Goal(){return goal_state;}
 public void put_Goal(Problem_State goal){goal_state = goal;}
 public Search_Node get_current_node(){return current_Node;}


 public String run_Search(Problem_State init_state, Problem_State g_state,
                          String search_method){
   goal_state = g_state;
   return run_Search(init_state, search_method);
 }

 public String run_Search(Problem_State init_state, String search_method){
  init_node = new Search_Node(init_state, null);
  System.out.println("\nStarting Search");
  open = new ArrayList<Search_Node>();
  open.add(init_node);
  closed = new ArrayList<Search_Node>();
  int cnum = 1;
  while(!open.isEmpty()){
    select_Node(search_method); // puts value to current_Node
    if (current_Node.goalP(this)) return report_Success();//"Search Succeeds";
    expand(search_method); // successor_nodes of current_Node
    closed.add(current_Node);
    cnum++;
   }
   return "Search Fails";
 }
```

```
private void expand(String search_method){
    successor_nodes = current_Node.get_Successors(this);
    vet_Successors(search_method);
    switch (search_method){
     case "depth_first":
      for (Search_Node i : successor_nodes){open.add(0,i);}
      break;
     default:
      for (Search_Node i : successor_nodes){open.add(i);}
    }
  }

  private void vet_Successors(String search_method){
    ArrayList<Search_Node> vslis = new ArrayList<Search_Node>();
    switch (search_method){
     case "depth_first": case "breadth_first": case "best_first":
       for (Search_Node snode : successor_nodes){
         if (!(on_Closed(snode)) && !(on_Open(snode))) vslis.add(snode);
       };
       break;
     case "branch_and_bound":
       for (Search_Node snode : successor_nodes){
         if (!(on_Closed(snode)) && !(on_Open(snode))) vslis.add(snode);
         else if (on_Open(snode)){
                int i = pos_Open(snode);
                if (snode.best_path_cost() < open.get(i).best_path_cost()){
                     open.remove(i);
                     open.add(snode);
                     vslis.add(snode);
                 }
               }
          else if (on_Closed(snode)){
                 int i = pos_Closed(snode);
                 if (snode.best_path_cost() < closed.get(i).best_path_cost()){
                     closed.remove(i);
                     open.add(snode);
                     vslis.add(snode);
                  }
               }

         };
       break;
```

Co-financed by the European Union
Connecting Europe Facility

```
case "A_star":
        for (Search_Node snode : successor_nodes){
          if (!(on_Closed(snode)) && !(on_Open(snode))) vslis.add(snode);
          else if (on_Open(snode)){
                  int i = pos_Open(snode);
                  if (snode.evaluation_fn(goal_state) <
                      open.get(i).evaluation_fn(goal_state)){
                       open.remove(i);
                       open.add(snode);
                       vslis.add(snode);
                   }
                }
           else if (on_Closed(snode)){
                   int i = pos_Closed(snode);
                   if (snode.evaluation_fn(goal_state) <
                       closed.get(i).evaluation_fn(goal_state)){
                        closed.remove(i);
                        open.add(snode);
                        vslis.add(snode);
                    }
                }

         };

    }
    successor_nodes = vslis;
  }

  private boolean on_Closed(Search_Node new_node){
    boolean ans = false;
    for (Search_Node closed_node : closed){
      if (new_node.same_State(closed_node)) ans = true;
    }
    return ans;
  }

  private boolean on_Open(Search_Node new_node){
    boolean ans = false;
    for (Search_Node open_node : open){
      if (new_node.same_State(open_node)) ans = true;
    }
    return ans;
  }
```

```
private void select_Node(String search_method){
   int i = 0, min = 0;
   switch (search_method){
    case "depth_first":
      int osize = open.size();
      current_Node = open.get(0);// open is a stack; 0 element is last in
      open.remove(0);
      break;
   case "breadth_first":
      current_Node = open.get(0);// open is a queue; 0 element is first in
      open.remove(0);
      break;
   case "branch_and_bound":
      i = 0; min = open.get(0).best_path_cost();
      for (int j = 1; j < open.size(); j++){
        if (open.get(j).best_path_cost() < min){i = j; min = open.get(j).best_path_cost();}
      }
      current_Node = open.get(i);
      open.remove(i);
      break;
    case "best_first":
      i = 0; min = open.get(0).difference(goal_state);
      for (int j = 1; j < open.size(); j++){
        if (open.get(j).difference(goal_state) < min){i = j; min = open.get(j).difference(goal_state);}
      }
      current_Node = open.get(i);
      open.remove(i);
      break;
    case "A_star":
      i = 0; min = open.get(0).evaluation_fn(goal_state);
      for (int j = 1; j < open.size(); j++){
        if (open.get(j).evaluation_fn(goal_state) < min){i = j; min = open.get(j).evaluation_fn(goal_state);}
      }
      current_Node = open.get(i);
      open.remove(i);
   }
  }
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
private String report_Success(){
    Search_Node n = current_Node;
    StringBuffer buf = new StringBuffer(n.toString());
    int plen = 1;
    while (n.get_parent() != null){
      buf.insert(0,"\n");
      n = n.get_parent();
      buf.insert(0, n.toString());
      plen++;
    }
    System.out.println("================================");
    System.out.println("Search Succeeds");
    System.out.println("Efficiency " + ((float)plen/(closed.size()+1)));
    System.out.println("Nodes visited: " + (closed.size()+1));
    System.out.println("Solution Path");
    return buf.toString();
  }

  private int pos_Open(Search_Node new_node){
    for (int i = 0; i < open.size(); i++)
        if (new_node.same_State(open.get(i))) return i;
    return -1;
  }

  private int pos_Closed(Search_Node new_node){
    for (int i = 0; i < closed.size(); i++)
        if (new_node.same_State(closed.get(i))) return i;
    return -1;
  }
}
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September – December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Constraint Satisfaction Problems, Game Playing and Planning – search-based solutions

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 3

Constraint Satisfaction Problems, Game Playing and Planning –search-based solutions

CONTENTS

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on constraint satisfaction problems, game playing and planning, students will be able:

Regarding Constraint Satisfaction Problems:

1. To explain what a Constraint Satisfaction Problem (CSP) is and how a search-based solution for such a problem can be constructed.
2. To analyze the classical cryptarithmetic and N-Queens CSPs and implement solutions for such problems, as well as other CSPs, using depth-first search with backtracking.
3. To explain consistency checking and to distinguish between local and global constraints.
4. To discuss constraint propagation in temporal constraint graphs and explain how the propagation minimizes disjunctive constraints and detects conflicts where such conflicts exist.
5. To generalize temporal constraint graphs to variable constraint graphs and to explain the notions of arc, path, and K, consistency.
6. To give a high-level typology of CSPs and to outline heuristics for reducing the search time.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on constraint satisfaction problems, game playing and planning, students will be able:

Regarding Game Playing:

1. To explain the category of two player, perfect information, zero-sum games, which is the topic of discussion.
2. To analyze the representation problem regarding the above category of games.
3. To explain the minimax procedure as well as the alpha-beta procedure and the relevant pruning rules, and to be able to apply these on simple two-player games.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on constraint satisfaction problems, game playing and planning, students will be able:

Regarding Planning:

1. To give high-level definitions of a planning system and of a linear plan as a sequence of actions to accomplish a goal in a discrete, deterministic, static and fully observable environment.
2. To analyze the representation problem for the above category of planning systems.
3. To make a reference to the Planning Domain Definition Language (PDDL), the STRIPS model and the action schema.
4. To discuss search-based solutions for the construction of linear plans and to distinguish between forward (progression) search and backward (regression) search pointing strengths and weaknesses.
5. To outline the decomposability for non-interacting component goals of complex goals and to overview the classical means-ends-analysis search method associated with the General Problem Solver (GPS).
6. To discuss general heuristics, based on problem relaxation, for reducing the search time.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

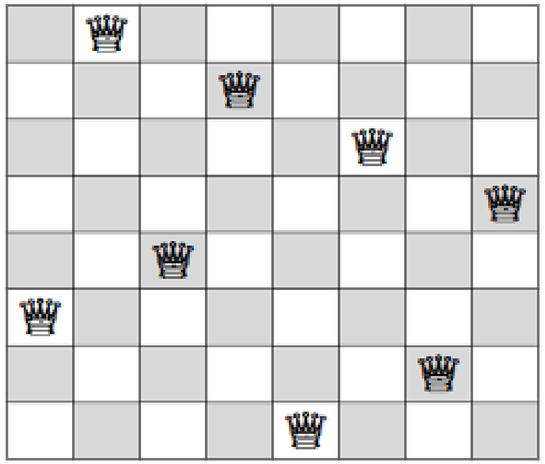
Master programmes in Artificial Intelligence 4 Careers in Europe

Constraint Satisfaction Problems (CSP)

Co-financed by the European Union
Connecting Europe Facility

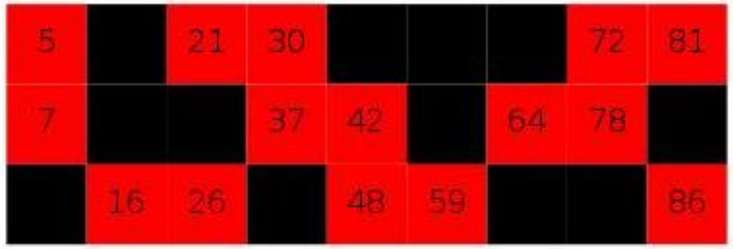
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Constraint Satisfaction Problems - Examples

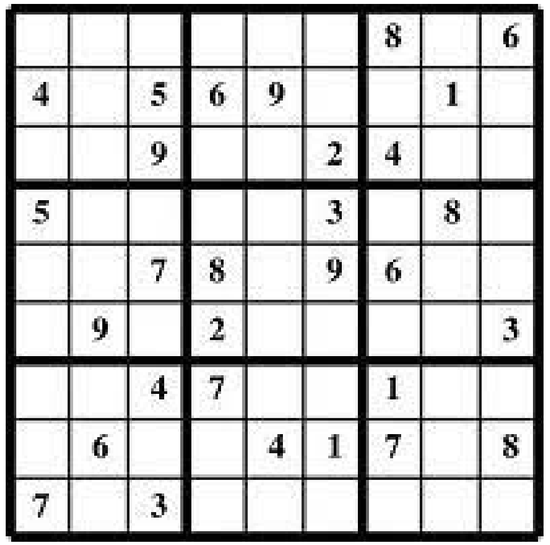


N-Queens Problem

Cryptarithmetic Puzzle



Constructing Bingo Cards



Sudoku

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Constraint Satisfaction Problem - Definition

- Set of Variables {X1, X2, …, Xn}
- Each variable Xi has a domain Di of possible values; often Di is discrete and finite
- Set of constraints {C1, C2, …, Cp}
- Each constraint Ck concerns a subset of the variables and specifies the permitted combinations of values for these variables

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Constraint Satisfaction Problem - Solution

- Set of Variables {X1, X2, …, Xn}
- Each variable Xi has a domain Di of possible values; often Di is discrete and finite
- Set of constraints {C1, C2, …, Cp}
- Each constraint Ck concerns a subset of the variables and specifies the permitted combinations of values for these variables

- Assign a value to each variable Vi, i = 1, .., n, such that all constraints Cj, j = 1, …, p, are satisfied

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: Cryptarithmetic Puzzles

```
  SEND
+MORE
-------
 MONEY
```

Variables:
D E N M O R S Y

Domains:
For D E N O R S Y the domain is {0,1,2,3,4,5,6,7,8,9}
For M S the domain is {1,2,3,4,5,6,7,8,9}

Constraints:
D E N M O R S Y are all different
X1 + X2 = X3, where
X1 = 10(10(10S + E) + N) + D
X2 = 10(10(10M + O) + R) + E
X3 = 10(10(10(10M + O) + N) + E) + Y

Solution:
D = 7, E = 5, N = 6, M = 1
O = 0, R = 8, S = 9, Y = 2

```
  9567
+ 1085
------
 10652
```

Alternative Solution if the domain of the leftmost letters (S, M) includes 0:
D = 9, E = 8, N = 4, M = 0
O = 6, R = 3, S = 5, Y = 7
5849 + 638 = 6487

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Other Cryptarithmetic Puzzles

```
   DONALD
 + GERALD
----------
   ROBERT
```

```
   CROSS
 + ROADS
----------
  DANGER
```

```
    BASE
  + BALL
----------
   GAMES
```

```
    TWO
  + TWO
----------
   FOUR
```

```
    SUN
  + FUN
----------
   SWIM
```

```
   MATH
 + MYTH
----------
   HARD
```

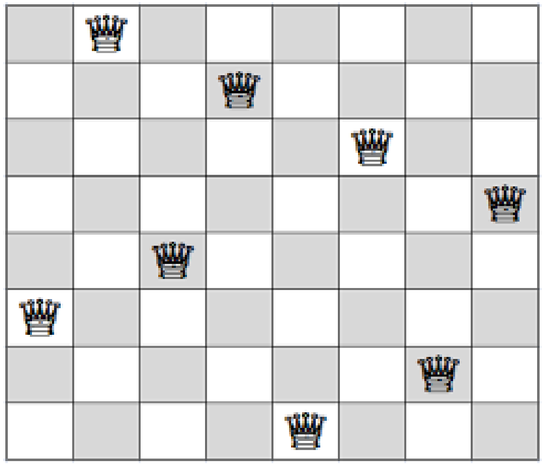
```
   TOUGH
 + DOUGH
----------
  RHYME
```

```
     WACKY
   x WACKY
------------
BICYCLISTS
```

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: N-Queens Problem



8-Queens

64^8 combinations!

Variables: Xij

Domain: {0,1}

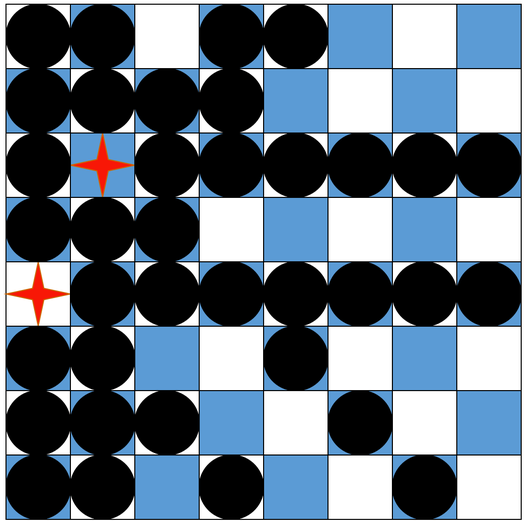
Constraints:

$$\forall i,j,k \quad Xij + Xik \leq 1$$
$$\forall i,j,k \quad Xij + Xkj \leq 1$$
$$\forall i,j,k \quad Xij + Xi'j' \leq 1, where\ i' = i + k, j' = j + k$$
$$\forall i,j,k \quad Xij + Xi'j' \leq 1, where\ i' = i + k, j' = j - k$$
$$\sum_{i,j} Xij = N$$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

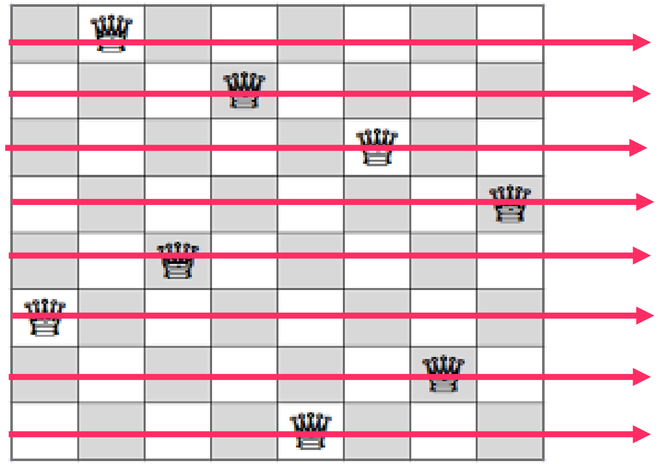
Example positioning of two Queens



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example: N-Queens Problem



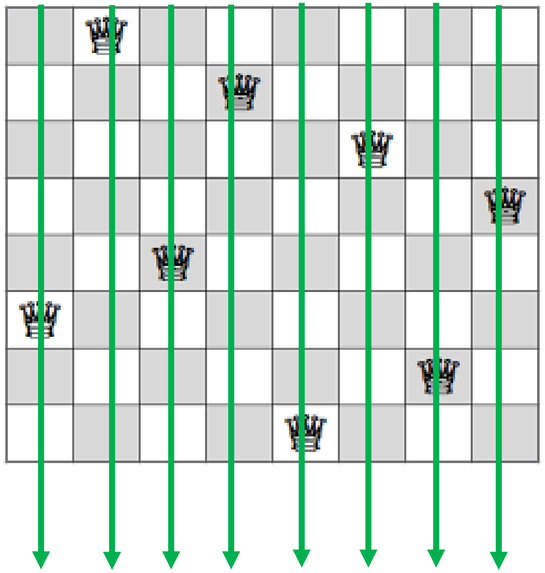
Restating the constraints

The sum of every row must be 1

The sum of all elements must equal N

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: N-Queens Problem



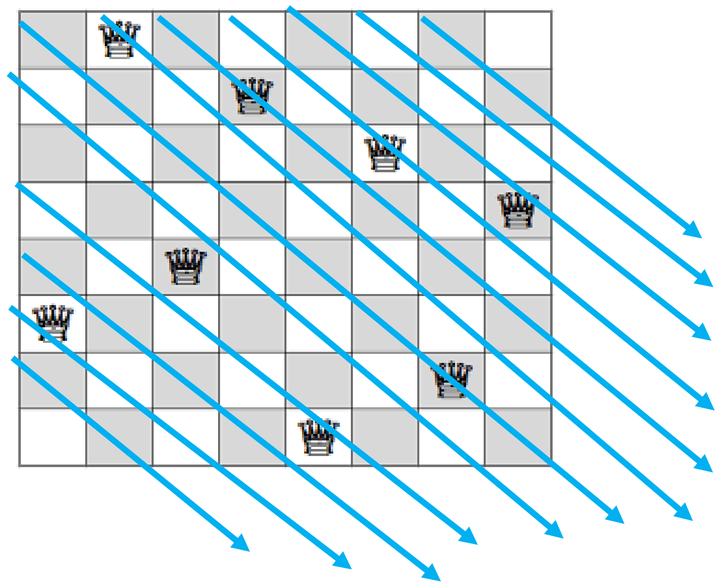
Restating the constraints

The sum of every column must be 1

The sum of all elements must equal N

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: N-Queens Problem



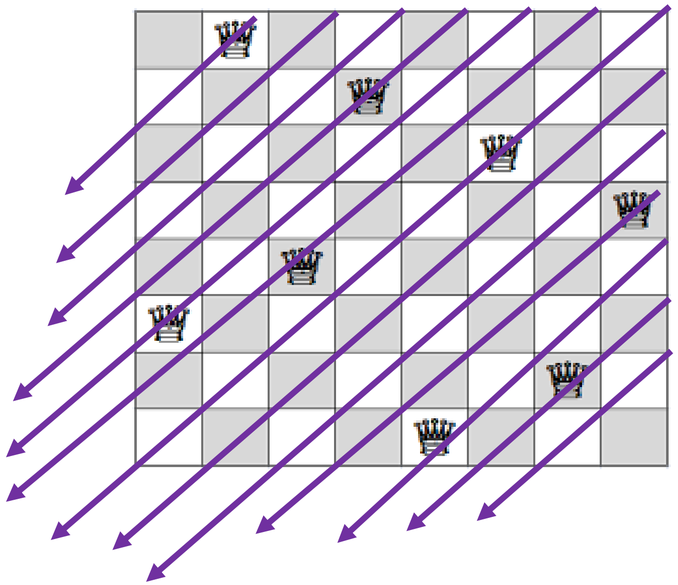
Restating the constraints

The sum of every diagonal cannot exceed 1

The sum of all elements must equal N

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: N-Queens Problem



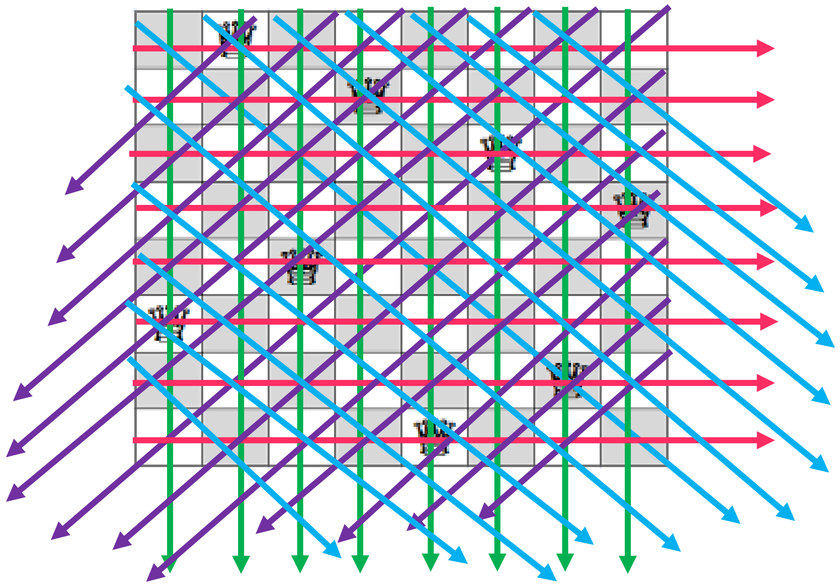
Restating the constraints

The sum of every anti-diagonal cannot exceed 1

The sum of all elements must equal N

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: N-Queens Problem



Restating the constraints

The sum of every row must be 1

The sum of every column must be 1

The sum of every diagonal cannot exceed 1

The sum of every anti-diagonal cannot exceed 1

The sum of all elements must equal N

Remodeling the variables

- There are N variables, Q1 Q2 Q3 . . . . Qn
- Qi represents the position of a Queen on row i
- Hence the domain of each variable is {1,2,3,..,N}
- N! combinations

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Solving Constraint Satisfaction Problems

- Some methods are:
  - Constraint propagation, with a view to eliminating inconsistencies and resulting in a minimal constraint graph where each variable has a specific value from its domain.
  - The simplex method (variable elimination) for linear programming
  - Special purpose solvers
  - General search methods
    - Which of the general search methods discussed would be appropriate?

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Search-based solutions to CSPs

- Which of the general methods discussed would be appropriate?
  - Recall that the search tree explicates part of the state space; a node of it encapsulate a problem state, or state for short, its parent node and the traversal cost from the parent state to its state – here the cost is of no interest
  - In a CSP, the initial state (given by the root node of the search tree) represents the situation where all variables are unbound, and a goal state giving a feasible solution to the CSP is when all variables are bound, i.e., assigned values from their domains and all constraints are satisfied
  - Operators assign values to variables; each step in the search determines the potential values for a single variable, i.e., the subset of values of its domain that do not (appear to) yield an inconsistency with preceding variable assignments on the given search path

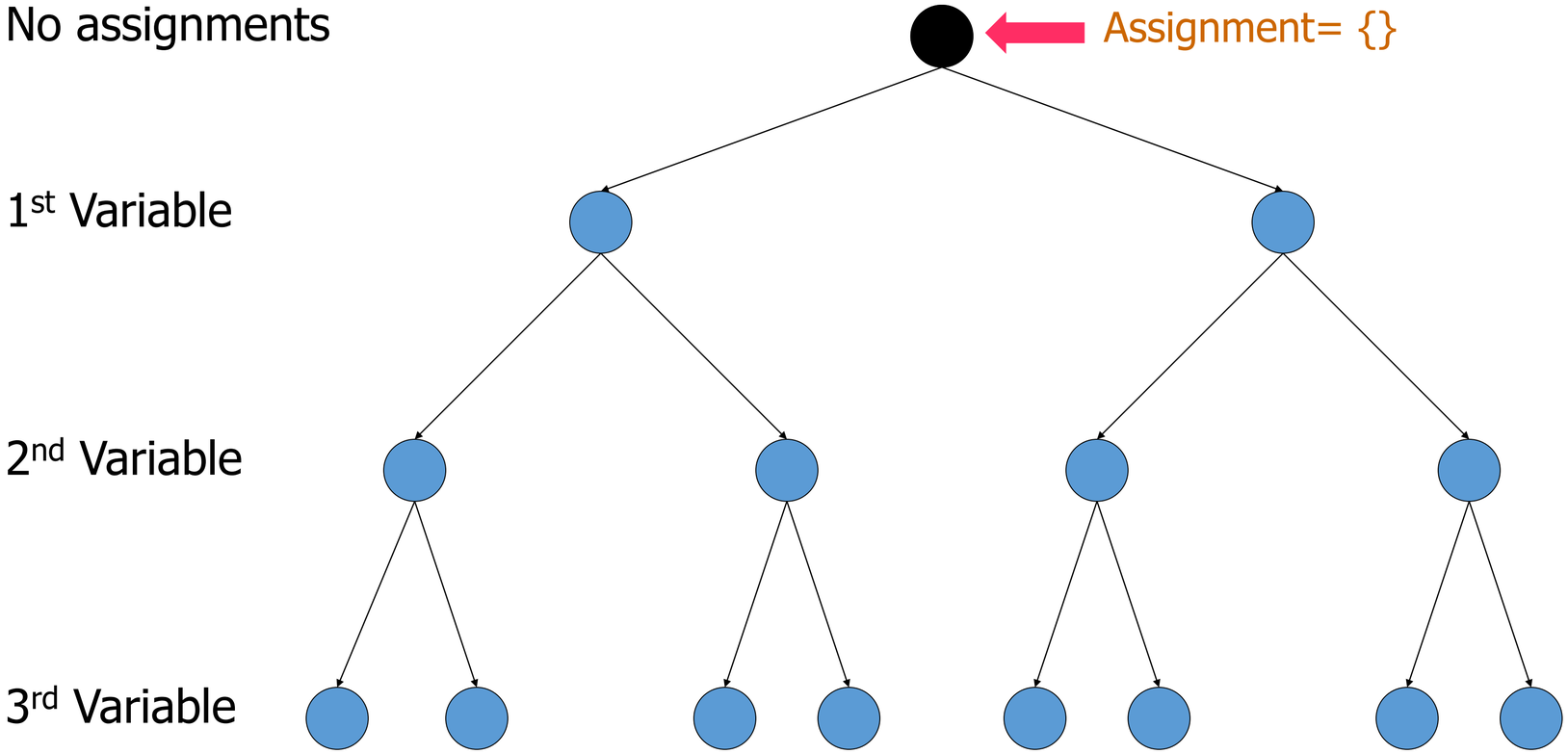
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Search-based solutions to CSPs

- Which of the general methods discussed would be appropriate?
  - The initial state is given but a goal state representing a feasible solution needs to be constructed
    - CSPs belong to the category of synthesis problems
  - Given N variables ordered as V_1, V_2, V_3, …, V_n, the first step selects possible values for V_1, the second step possible values for V_2, etc., and the last step possible values for V_n
    - The branching factor of a node representing a value assignment for variable V_i, is at most d, where d is the cardinality of the domain of V_{i+1}
    - Each path through the search tree has a depth of at most N; the depth of a path leading to a feasible solution is N
  - **Hence depth-first with backtracking as soon as a constraint violation is detected**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

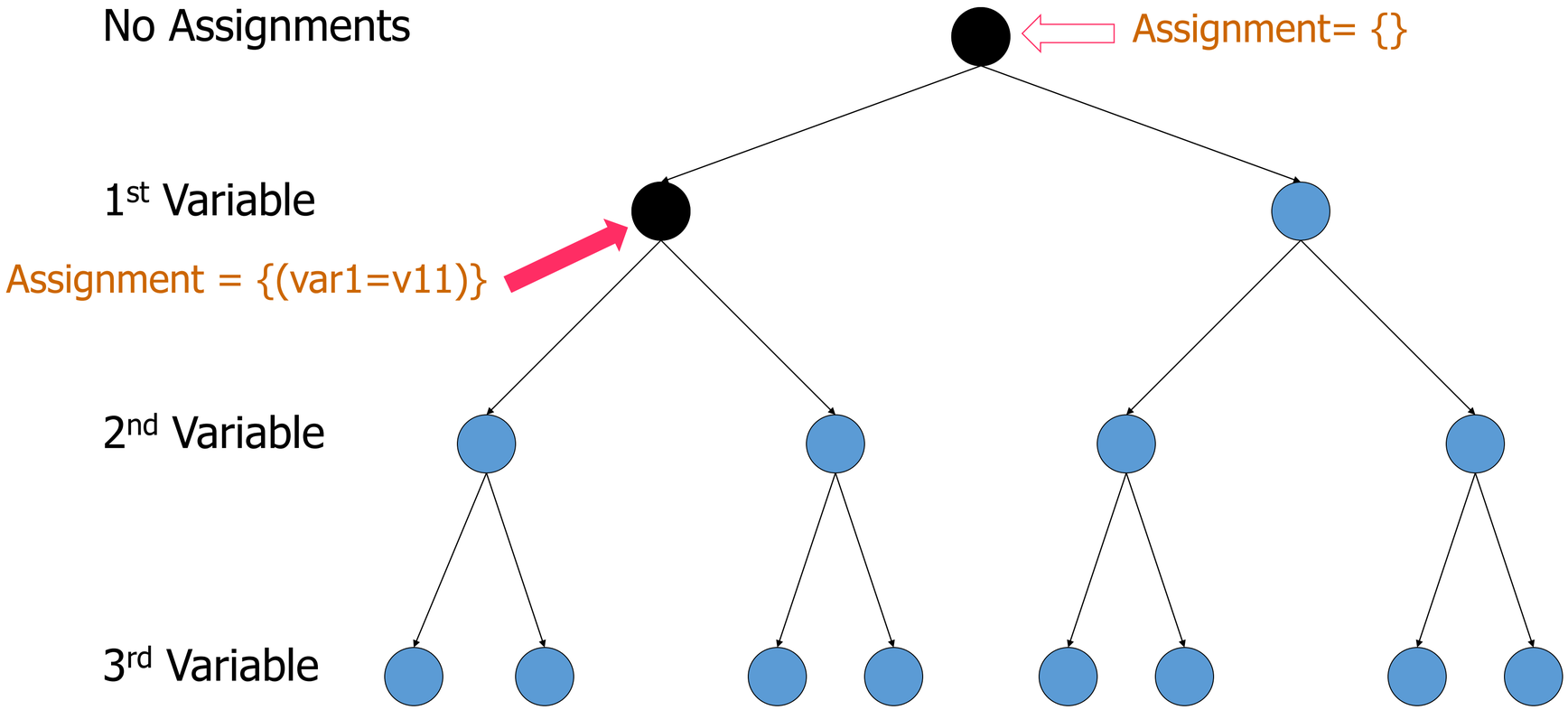
Depth-First with Backtracking



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

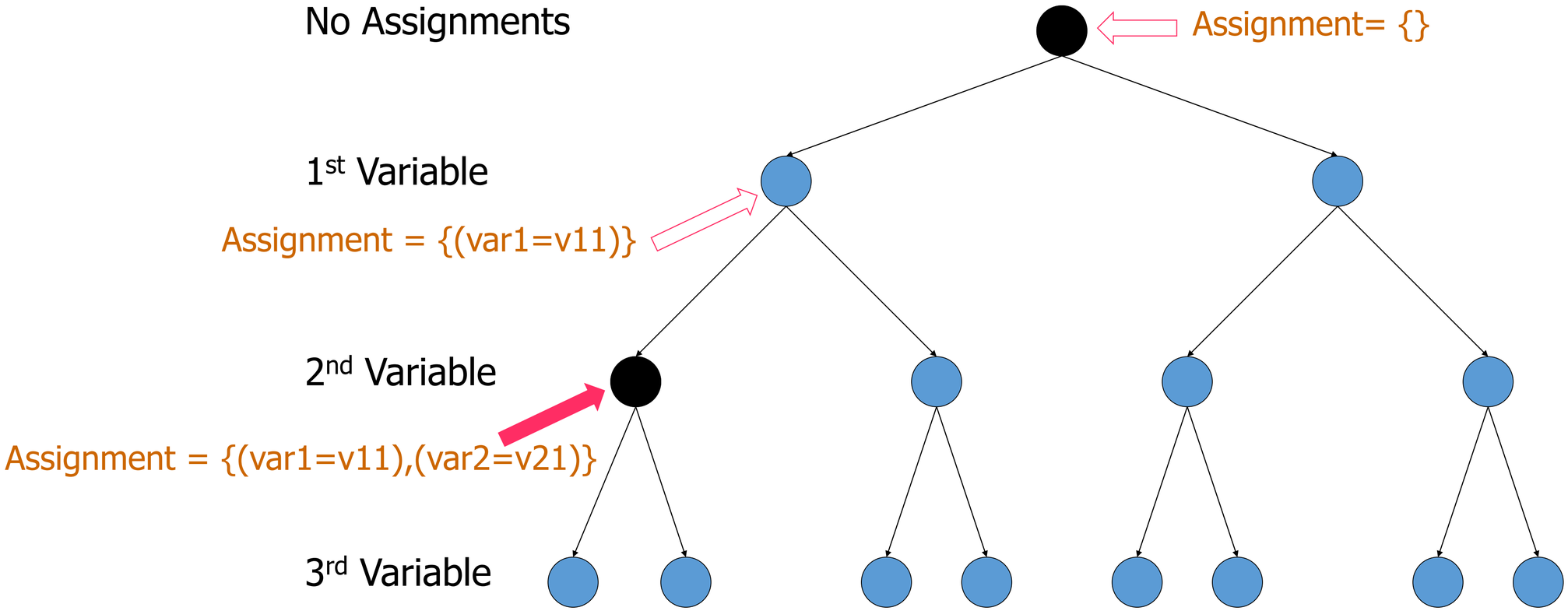
Depth-First with Backtracking



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

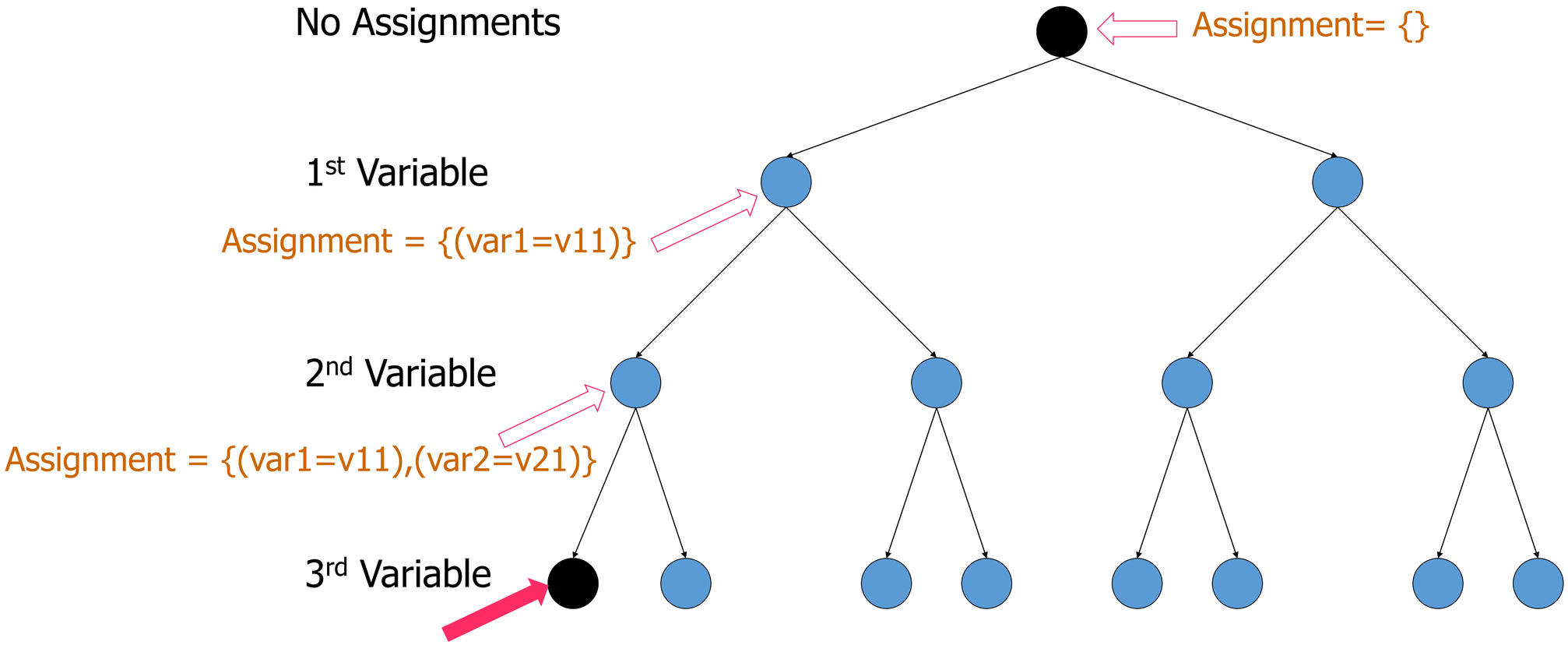
Depth-First with Backtracking



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

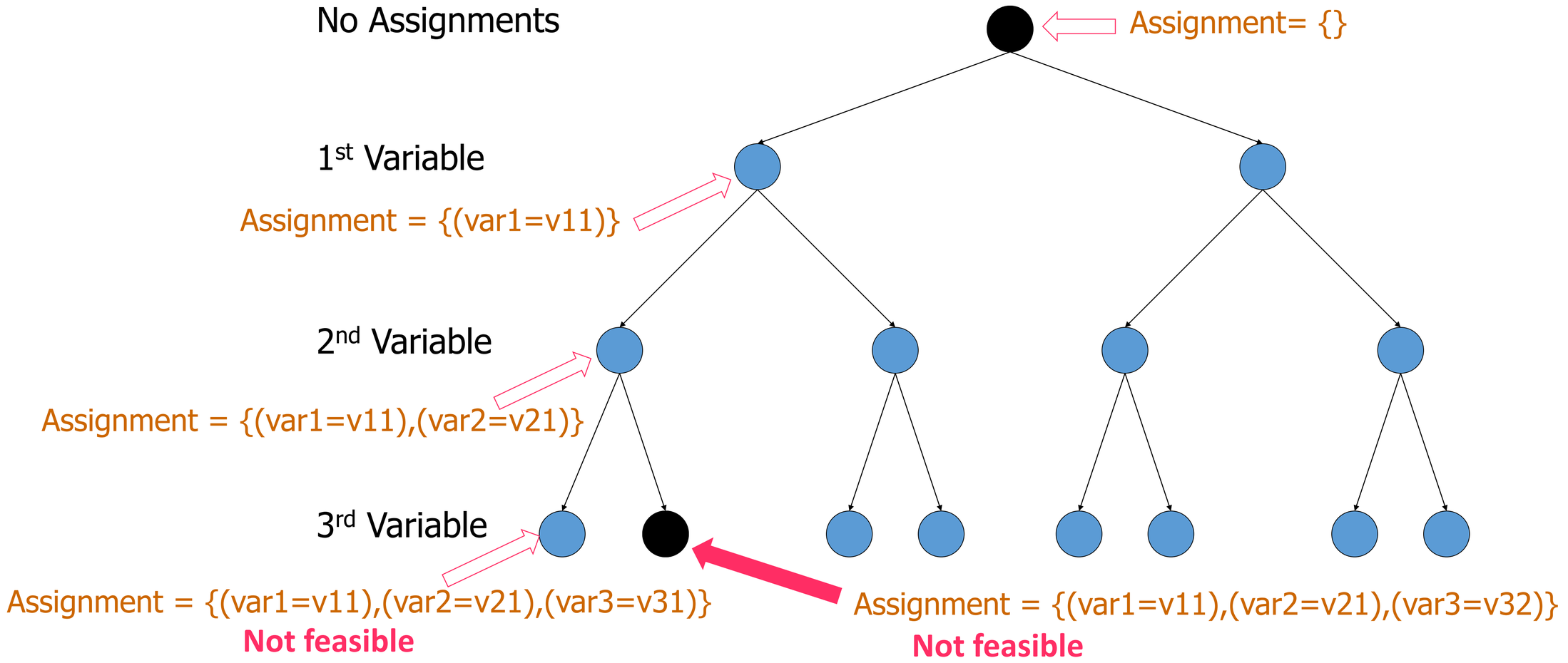
Depth-First with Backtracking



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

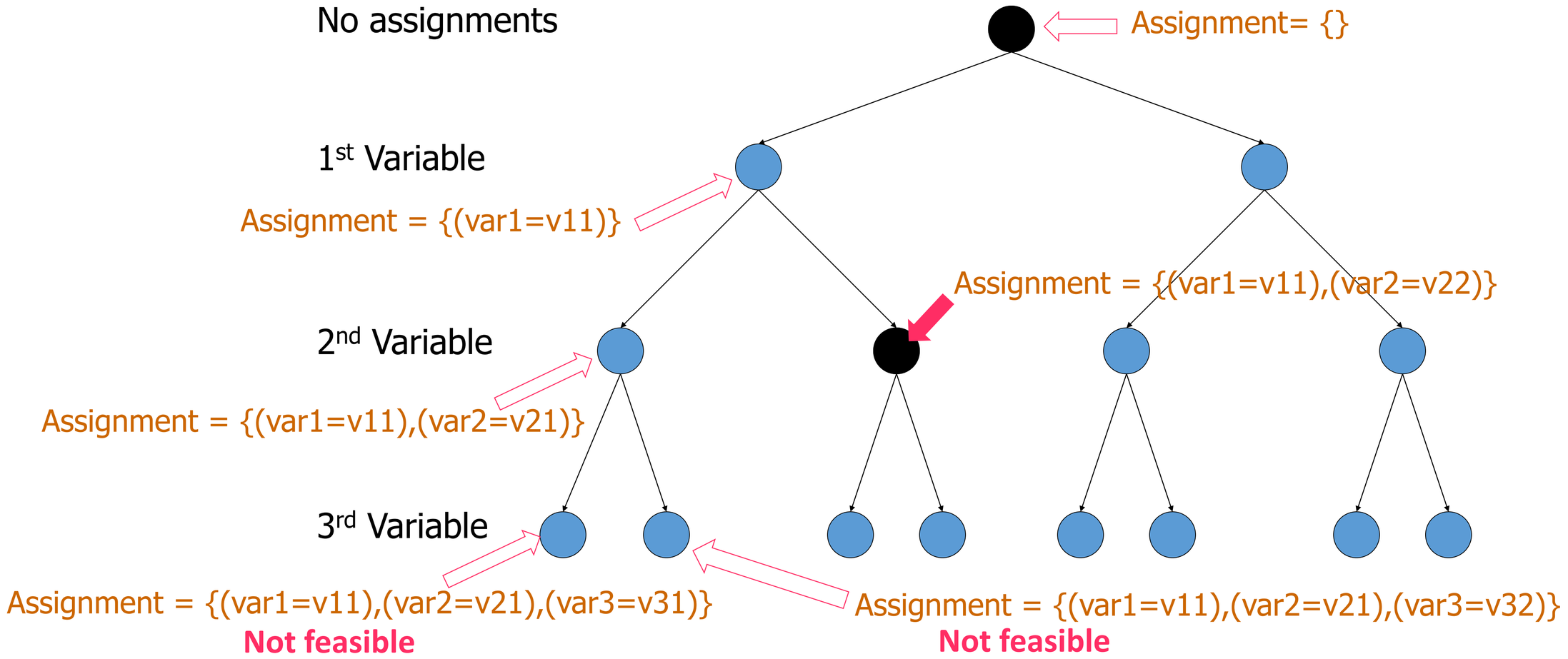
Depth-First with Backtracking



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

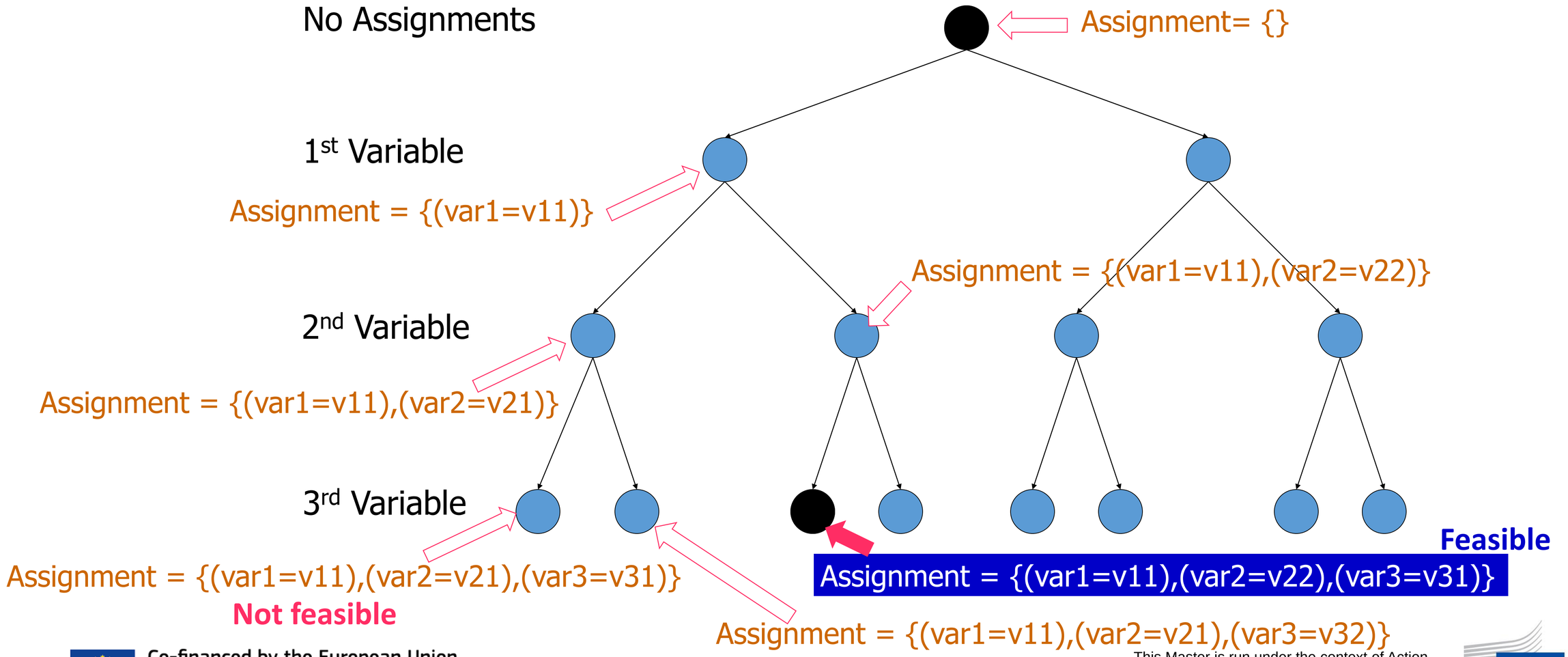
Depth-First with Backtracking



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Depth-First with Backtracking



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Consistency Checking

- The current search node, csn, is not a leaf node, and its state gives the value assignments to variables up to variable V_i
  - Its exploration generates successor nodes for the possible value assignments of the next variable V_{i+1}; each of these values should be consistent with the value assignments given in csn – **local consistency** (i.e., adherence to local constraints)
  - If no such value can be assigned to V_{i+1}, csn is a dead end and backtracking occurs
- The current search node, csn, is a leaf node
  - Its state gives value assignments for all variables, which are locally consistent
  - If csn's state also satisfies any remaining, global, constraints then it represents a feasible solution – **global consistency**
  - Otherwise, it is not a feasible solution and backtracking occurs

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Solving Cryptarithmetic Puzzles by Depth-First Search

- Recall the constraints:
  - **Local Constraints**
    - The domain of the leftmost letters is {1,2,..,9} and of the other letters is {0,1,2,…,9}
    - All letters are assigned different values
  - **Global Constraint**
    - Substituting the digits for the letters and doing the addition gives the correct sum
  - Local constraints are verified with each letter assignment, i.e., during the generation of the successor nodes of the current search node, while the global constraint is verified when all letters have been assigned values to see if the current search node is a feasible solution

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Solving Cryptarithmetic Puzzles by Depth-First Search

- The depth of the search is given by the number of different letters
- Hence for any cryptarithmetic puzzle the depth cannot be more than 10, as there can be 10 variables at most
  - Cryptarithmetic puzzles are very small scale CSPs
- Simple alphabetic ordering of letter assignments

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Representation Problem for Cryptarithmetic Puzzles

- Possible problem state representation

-1	-1	-1	10	10	-1	-1	-1	-1	-1	-1	-1	10	10	10	-1	-1	10	10	-1	-1	-1	-1	-1	10	-1
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

-1 denotes a nonexistent letter, and 10 means an existent but unbound letter – initial state

-1	-1	-1	7	5	-1	-1	-1	-1	-1	-1	-1	1	6	0	-1	-1	8	9	-1	-1	-1	-1	-1	2	-1
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

goal state

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Operator for Cryptarithmetic Puzzles

- There is one operator for computing successors of search nodes

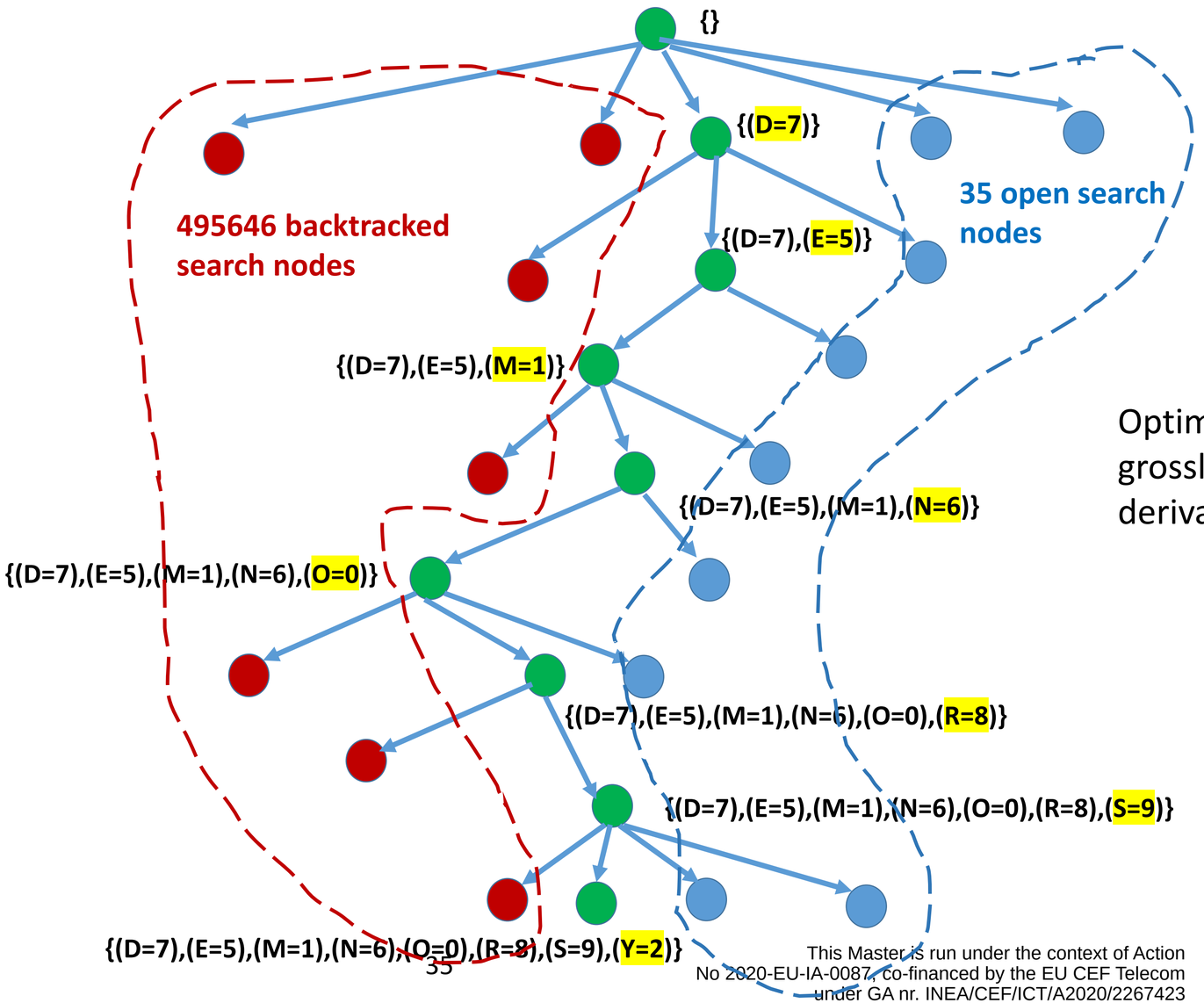
  If <L> is the next in order unbound letter and <D> is an available value from its domain then bound <L> to <D>

Search method for Cryptarithmetic Puzzles

- Depth-First search

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

```
$ java Run_Crypto_Search SEND MORE MONEY
Starting Search
==============================
Search Succeeds
Efficiency 1.815779E-5
Nodes visited: 495655
Solution Path
Node with state
Node with state
 D=7
Node with state
 D=7 E=5
Node with state
 D=7 E=5 M=1
Node with state
 D=7 E=5 M=1 N=6
Node with state
 D=7 E=5 M=1 N=6 O=0
Node with state
 D=7 E=5 M=1 N=6 O=0 R=8
Node with state
 D=7 E=5 M=1 N=6 O=0 R=8 S=9
Node with state
 D=7 E=5 M=1 N=6 O=0 R=8 S=9 Y=2
```

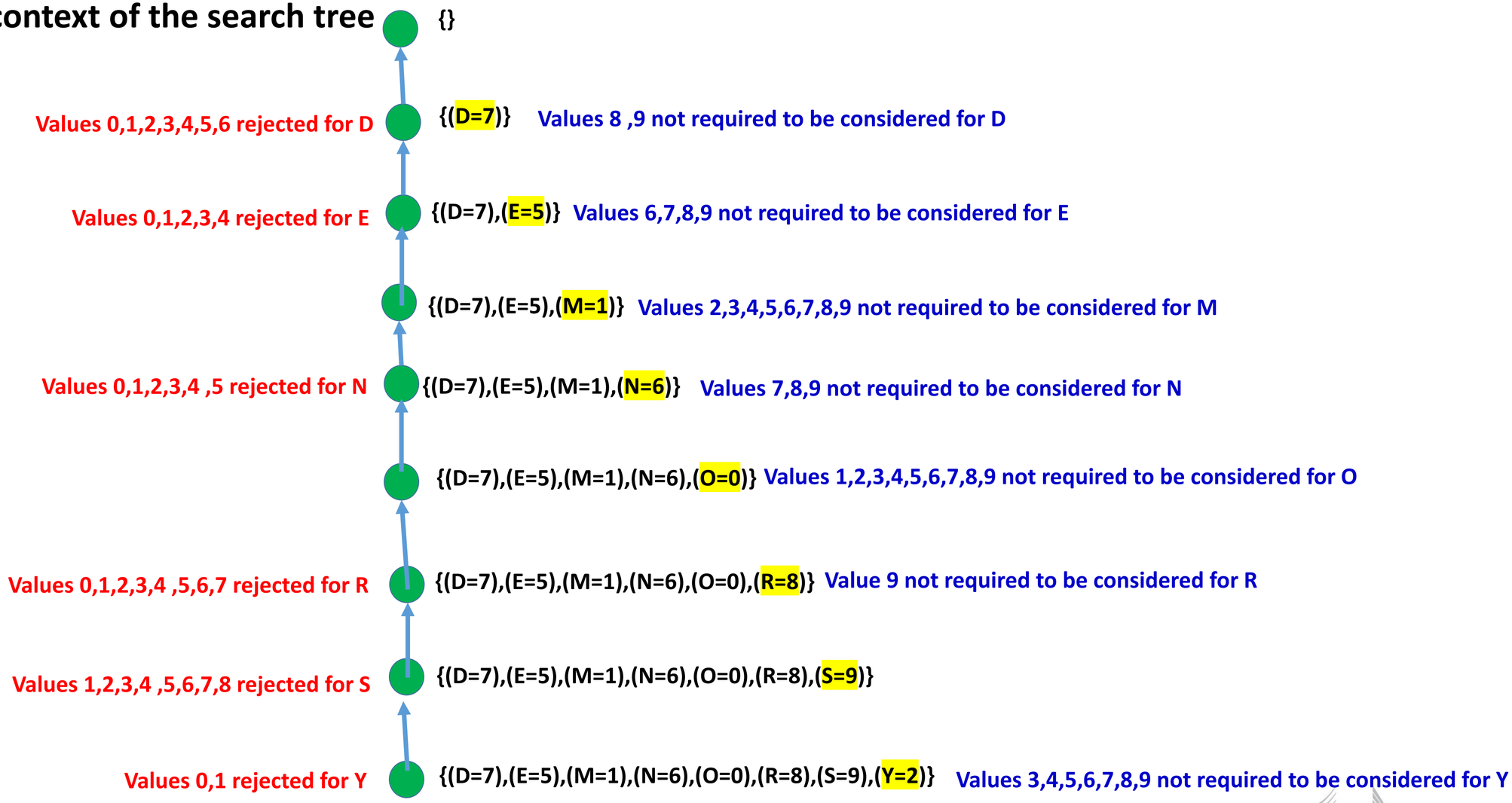


Optimal solution – but grossly inefficient derivation

Co-financed by the European Union
Connecting Europe Facility

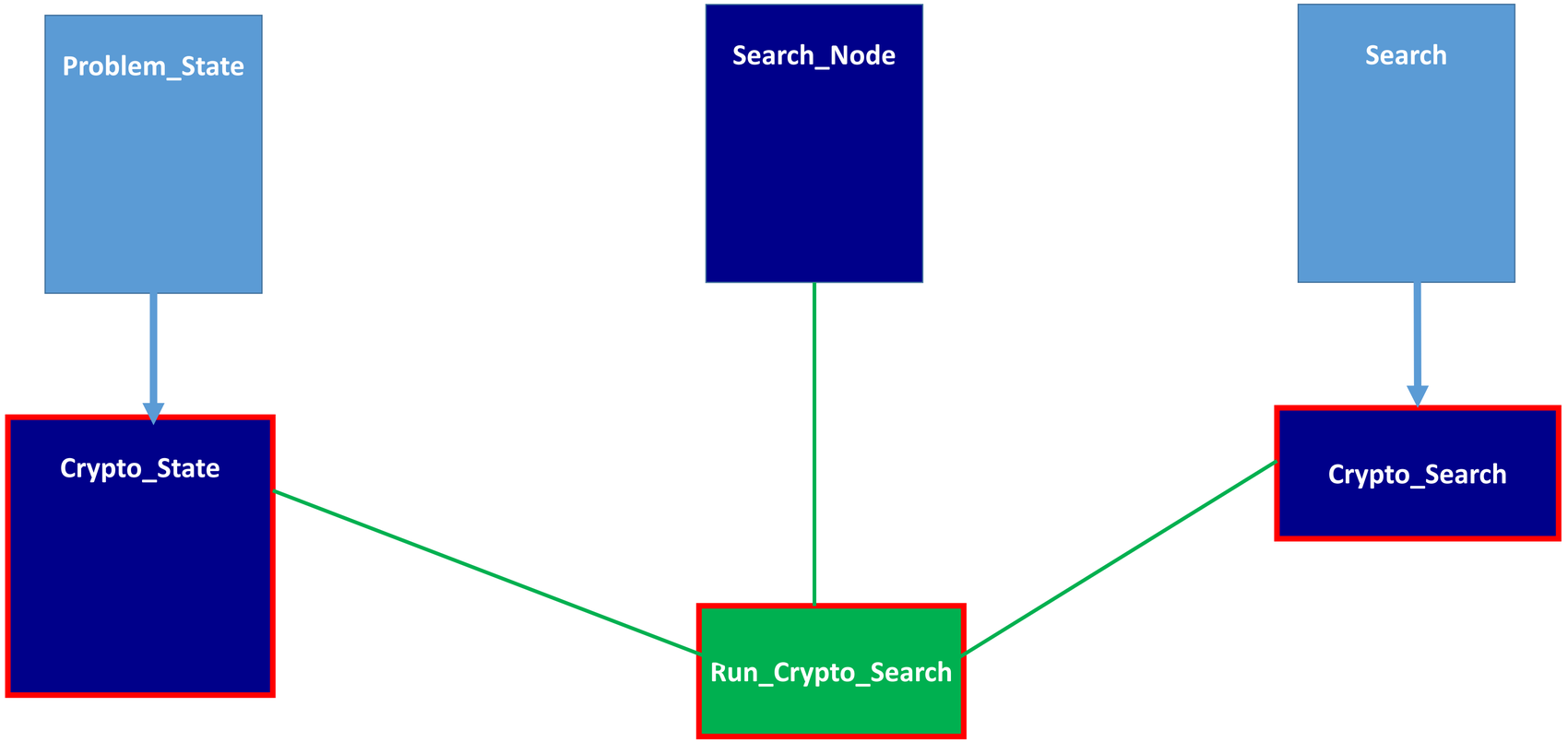
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Solution path in the context of the search tree



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Extending General Search to solve Cryptarithmetic Puzzles



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

```
import java.util.*;

public class Crypto_State extends Problem_State {

 private int[] conf;

 public Crypto_State(String word1, String word2, String sum){
  conf = new int[26];
  for (int i = 0; i < conf.length; i++) conf[i] = -1;
  for (int i = 0; i < word1.length(); i++) conf[word1.charAt(i) - 'A'] = 10;
  for (int i = 0; i < word2.length(); i++) conf[word2.charAt(i) - 'A'] = 10;
  for (int i = 0; i < sum.length(); i++) conf[sum.charAt(i) - 'A'] = 10;
 }

 public Crypto_State(Crypto_State cs){
   conf = new int[cs.conf.length];
   for (int i = 0; i < cs.conf.length; i++)
        conf[i] = cs.conf[i];
 }

 public int[] get_Conf(){return conf;}
 public int get_Elem(int i){return conf[i];}
 public int get_Elem(char c){return conf[c - 'A'];}
 public void put_Elem(int i, int n){conf[i] = n;}

 private int get_First_Unbound(){
   for (int i = 0; i < conf.length; i++)
     if (conf[i] == 10) return i;
   return -1;
 }
```

```
public int cost_from(Problem_State s){return 1;};

 public boolean same_State(Problem_State s2){
   Crypto_State cs = (Crypto_State) s2;
   for (int i = 0; i < conf.length; i++)
       if (conf[i] != cs.conf[i]) return false;
   return true;
 }

 private int convertNum(String word){
  int n = get_Elem(word.charAt(0));
  for (int i = 1; i < word.length(); i++)
     n = (n * 10) + get_Elem(word.charAt(i));
  return n;
 }

 public boolean goalP(Search searcher){
    if (get_First_Unbound() != -1) return false;
    Crypto_Search csearcher = (Crypto_Search) searcher;
    String word1 = csearcher.getWord1();
    String word2 = csearcher.getWord2();
    String sum = csearcher.getSum();
    return convertNum(word1) + convertNum(word2) == convertNum(sum);
 }

 private boolean committed (int n){
   for (int i = 0; i < conf.length; i++)
          if (n == conf[i]) return true;
   return false;
 }
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Crypto_State definition cont.

```
public ArrayList<Problem_State> get_Successors(Search searcher){
 ArrayList<Crypto_State> cslis = new ArrayList<Crypto_State>();
 ArrayList<Problem_State> slis = new ArrayList<Problem_State>();

 int pos = get_First_Unbound();
 if (pos != -1){
   int x = 0;
   Crypto_Search css = (Crypto_Search) searcher;
   String nz = css.getNonZero();
   char L = (char)(pos + 'A');
   for (int i = 0; i < nz.length(); i++)
     if (L == nz.charAt(i)) x = 1;
   for (int i = x; i <= 9; i++){
    if (!committed(i)){
     Crypto_State cs = new Crypto_State(this);
     cs.conf[pos] = i;
     cslis.add(cs);
    }
   }
 }
 for (Crypto_State cps : cslis)
  slis.add((Problem_State)cps);
 return slis;
}
```

```
public String toString(){
  String res = "\n";
  for (int i = 0; i < conf.length; i++){
     if (conf[i] != 10 && conf[i] != -1) res += " " + (char)(i + 'A') + "=" + conf[i];
  }
  return res;
 }
}
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
import java.util.*;

public class Crypto_Search extends Search{

 private String word1;
 private String word2;
 private String sum;
 private String nonZero;

 public Crypto_Search (String w1, String w2, String s){
   word1 = new String(w1);
   word2 = new String(w2);
   sum = new String(s);
   nonZero = "" + w1.charAt(0) + w2.charAt(0) + sum.charAt(0);
 }

 public String getWord1(){return word1;}
 public String getWord2(){return word2;}
 public String getSum(){return sum;}
 public String getNonZero(){return nonZero;}

}
```

```
import java.util.*;

public class Run_Crypto_Search{
 public static void main(String[] args){
   Crypto_Search searcher = new Crypto_Search(args[0],args[1],args[2]);
   Problem_State init_state = (Problem_State) new Crypto_State(args[0],args[1],args[2]);
   String res = searcher.run_Search(init_state, "depth_first");
   System.out.println(res);
 }
}
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Representation Problem for the N-Queens Problem

- Problem state representation – N x N binary matrix

Q1								
Q2								
Q3								
Q4								
Q5								
Q6								
Q7								
Q8								

There are N variables, Q1, Q2, .., Qn representing the positions of the N Queens on the N rows respectively, ordered in this sequence

Initial state: all cells are 0 meaning all variables are unbound

Goal state: One cell exactly on each row is 1 representing the positions of the N Queens and all other constraints are satisfied, i.e., each row adds up to 1, each column adds up to 1, and each diagonal/anti-diagonal cannot add up to more than 1

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Operator for N-Queens Problem

- There is one operator for computing successors of search nodes

  If <Q> is the next non-positioned Queen and it needs to be positioned on row <R>

  then bound <Q> to cell <C> of row <R> provided that the four sums corresponding to the row, column, diagonal and anti-diagonal do not exceed 1

Search method for N-Queens Problem

- Depth-First search

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Q1	0	0	0	0	0	0	0	0
Q2	0	0	0	0	0	0	0	0
Q3	0	0	0	0	0	0	0	0
Q4	0	0	0	0	0	0	0	0
Q5	0	0	0	0	0	0	0	0
Q6	0	0	0	0	0	0	0	0
Q7	0	0	0	0	0	0	0	0
Q8	0	0	0	0	0	0	0	0

Q1	1	0	0	0	0	0	0	0
Q2	0	0	0	0	1	0	0	0
Q3	0	0	0	0	0	0	0	1
Q4	0	0	0	0	0	1	0	0
Q5	0	0	0	0	0	0	0	0
Q6	0	0	0	0	0	0	0	0
Q7	0	0	0	0	0	0	0	0
Q8	0	0	0	0	0	0	0	0

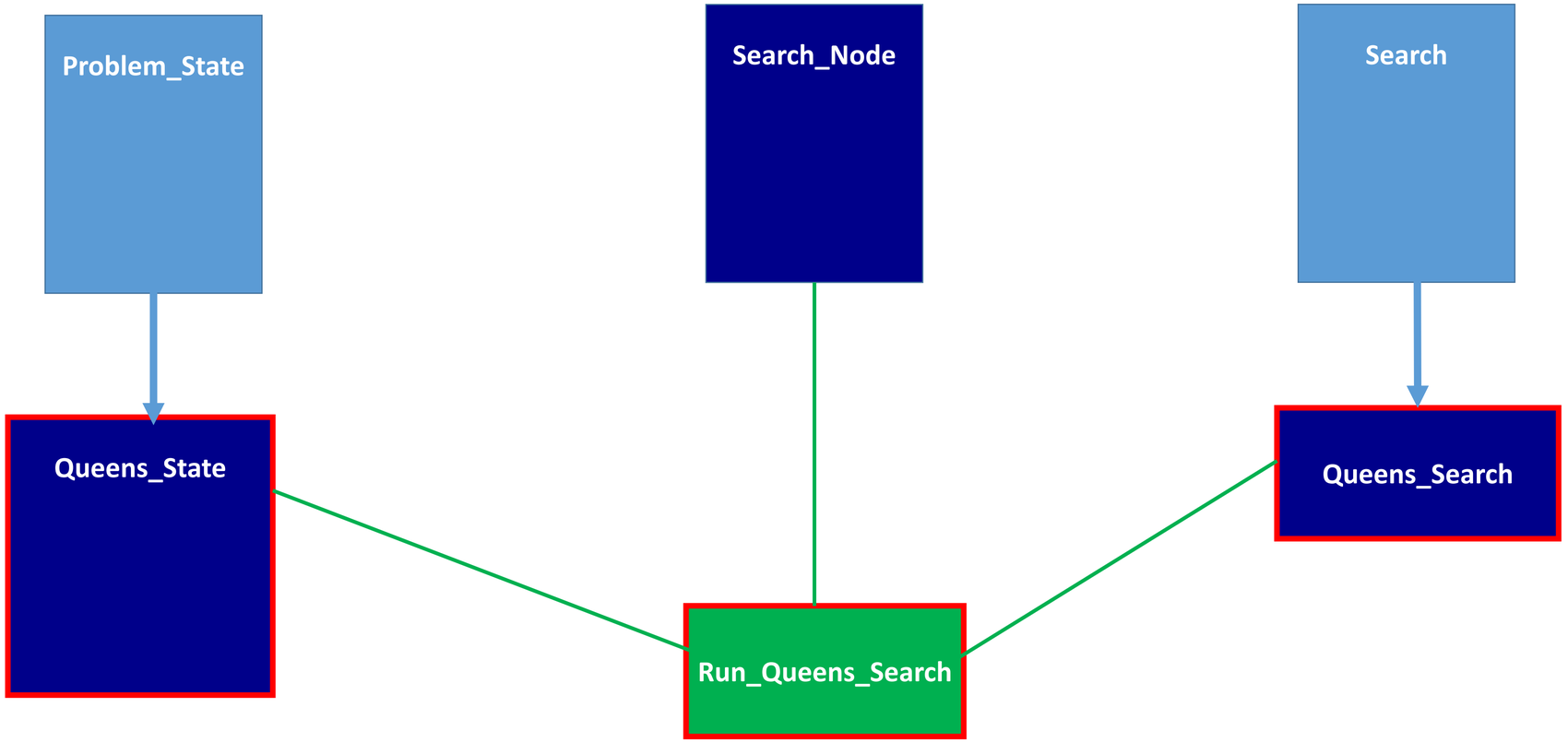
Q1	1	0	0	0	0	0	0	0
Q2	0	0	0	0	1	0	0	0
Q3	0	0	0	0	0	0	0	1
Q4	0	0	0	0	0	1	0	0
Q5	0	0	0	0	0	0	0	0
Q6	0	0	0	0	0	0	0	0
Q7	0	0	0	0	0	0	0	0
Q8	0	0	0	0	0	0	0	0

Local consistency: The four sums of the cell under consideration not to exceed 1

Global consistency: All row and column sums to be exactly 1 and all diagonal and anti-diagonal sums not to exceed 1, i.e., to be 0 or 1 – Global consistency follows from the satisfaction of all local consistencies

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Extending General Search to solve the N-Queens Problem



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

```
$ java Run_Queens_Search 3

Starting Search
Search Fails
```

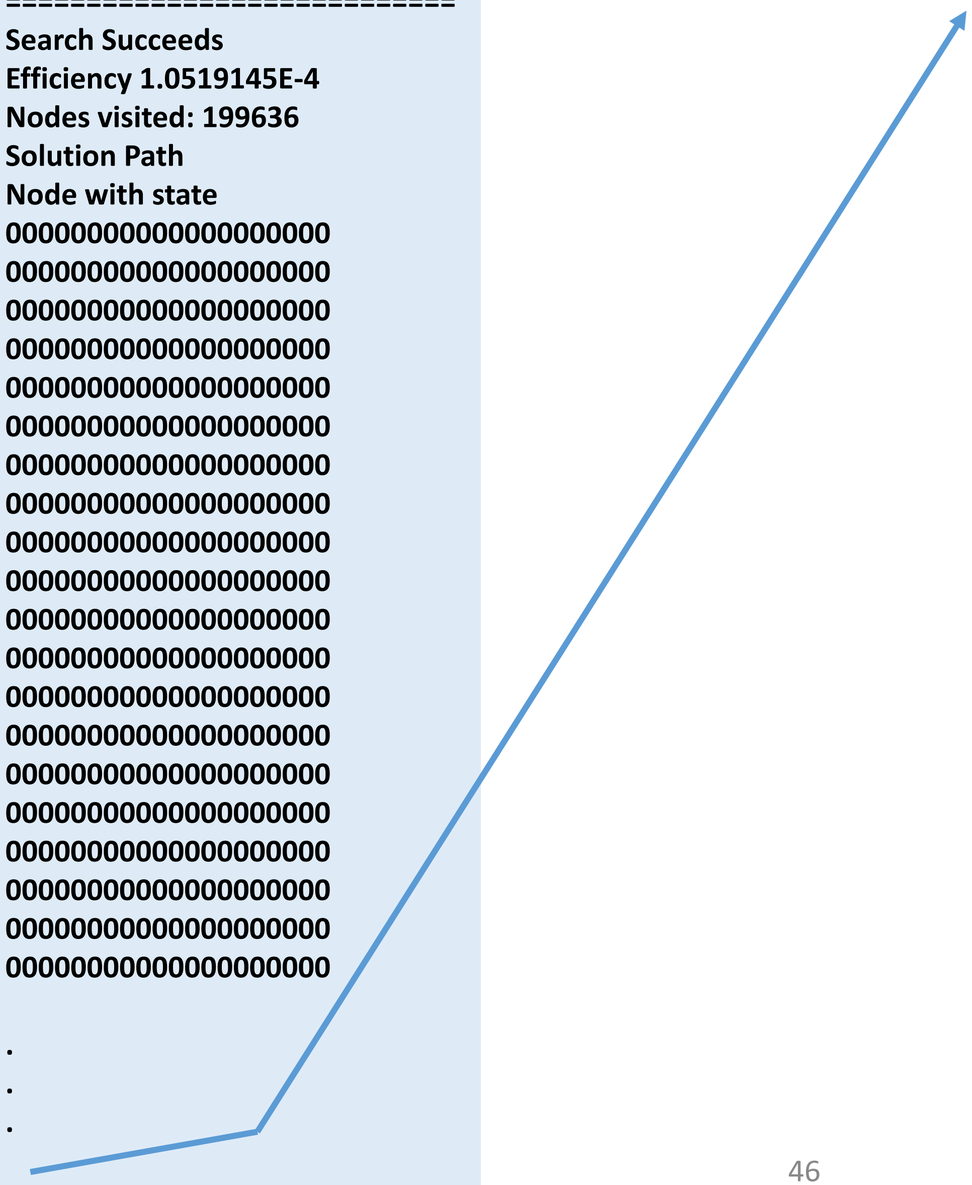
```
$java Run_Queens_Search 4

Starting Search
============================
Search Succeeds
Efficiency 0.5555556
Nodes visited: 9
Solution Path
Node with state
0000
0000
0000
0000

Node with state
0010
0000
0000
0000

Node with state
0010
1000
0000
0000

Node with state
0010
1000
0001
0000

Node with state
0010
1000
0001
0100
```

```
$ java Run_Queens_Search 8

Starting Search
============================
Search Succeeds
Efficiency 0.078947365
Nodes visited: 114
Solution Path
Node with state
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000

Node with state
00000001
00000000
00000000
00000000
00000000
00000000
00000000
00000000

Node with state
00000001
00010000
00000000
00000000
00000000
00000000
00000000
00000000

Node with state
00000001
00010000
10000000
00000000
00000000
00000000
00000000
00000000

Node with state
00000001
00010000
10000000
00100000
00000000
00000000
00000000
00000000

Node with state
00000001
00010000
10000000
00100000
00000100
00000000
00000000
00000000

Node with state
00000001
00010000
10000000
00100000
00000100
01000000
00000000
00000000

Node with state
00000001
00010000
10000000
00100000
00000100
01000000
00000010
00000000

Node with state
00000001
00010000
10000000
00100000
00000100
01000000
00000010
00001000
```

Co-financed by the Europ
Connecting Europe Facility

This Master is run under the context of Action 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

```
$ java Run_Queens_Search 20

Starting Search
============================
Search Succeeds
Efficiency 1.0519145E-4
Nodes visited: 199636
Solution Path
Node with state
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
.
.
.
```



```
Node with state
00000000000000000001
00000000000000000100
00000000000000010000
00000000000000000010
00000000000000001000
00000001000000000000
00000100000000000000
00000001000000000000
00100000000000000000
10000000000000000000
00010000000000000000
00000000000100000000
00001000000000000000
01000000000000000000
00000000000010000000
00000000001000000000
00000000000001000000
00000010000000000000
00000000000000100000
00000000010000000000
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
$ java NQueens 3
There is no solution for 3 Queens!!
$ java NQueens 4
0 1 0 0
0 0 0 1
1 0 0 0
0 0 1 0
$ java NQueens 5
1 0 0 0 0
0 0 1 0 0
0 0 0 0 1
0 1 0 0 0
0 0 0 1 0
$ java NQueens 6
0 1 0 0 0 0
0 0 0 1 0 0
0 0 0 0 0 1
1 0 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
$ java NQueens 7
1 0 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 0 1
0 1 0 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 0 1 0
```

```
$ java NQueens 8
1 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0

$ java NQueens 15
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
```

```
$ java NQueens 20
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
```

Alternative implementation displaying just the goal states

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

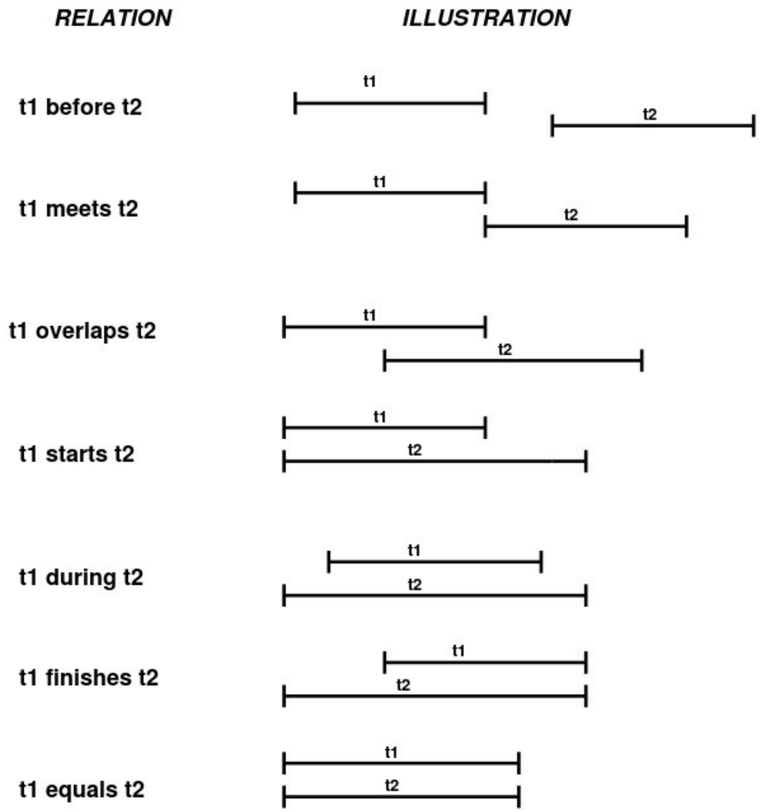
Another CSP Example: Temporal Constraints

- Temporal constraints occur in numerous situations: planning, scheduling, process modelling, etc.
- Time-interval based constraints
  - Task T3 contains task T1
  - Task T2 precedes task T1
  - Tasks T2 and T3 overlap
  - Task T4 follows task T1

- Are the above constraints consistent?
- What is the temporal relation between each pair of tasks?

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Allen's interval based temporal relations



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Task T3 contains task T1: Possible scenarios

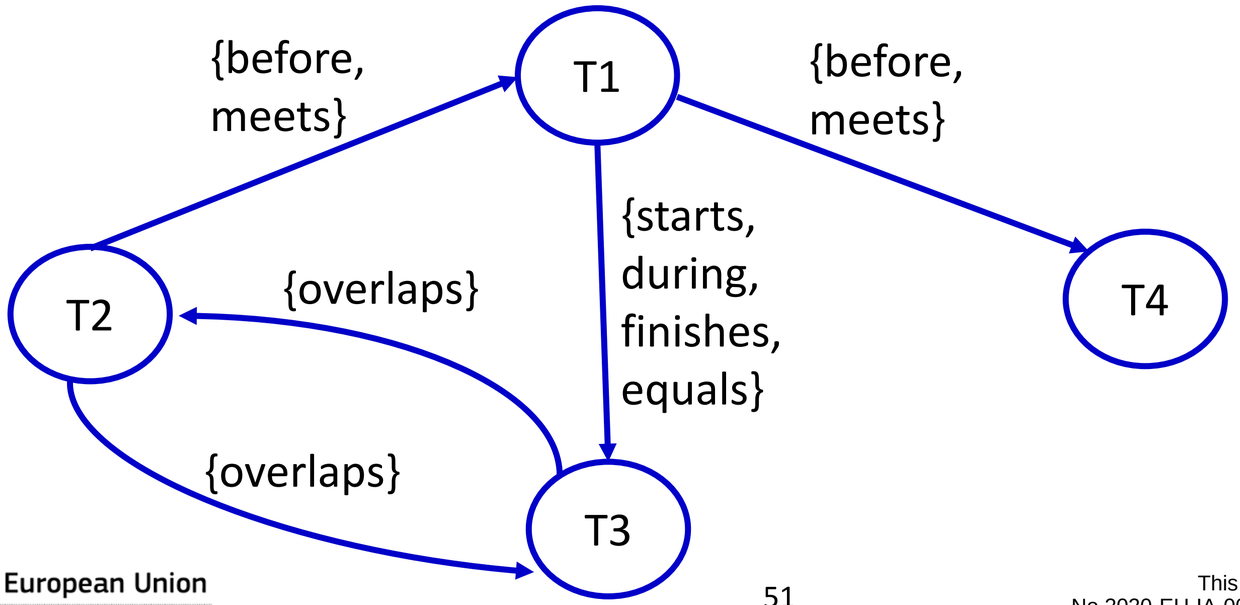
Task T2 precedes task T1: Possible scenarios

Tasks T2 and T3 overlap: Possible scenarios

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Task T4 follows task T1: Possible scenarios

Temporal Constraints Graph



Constraint propagation
to minimize disjunctions, if possible, and to derive temporal relations

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

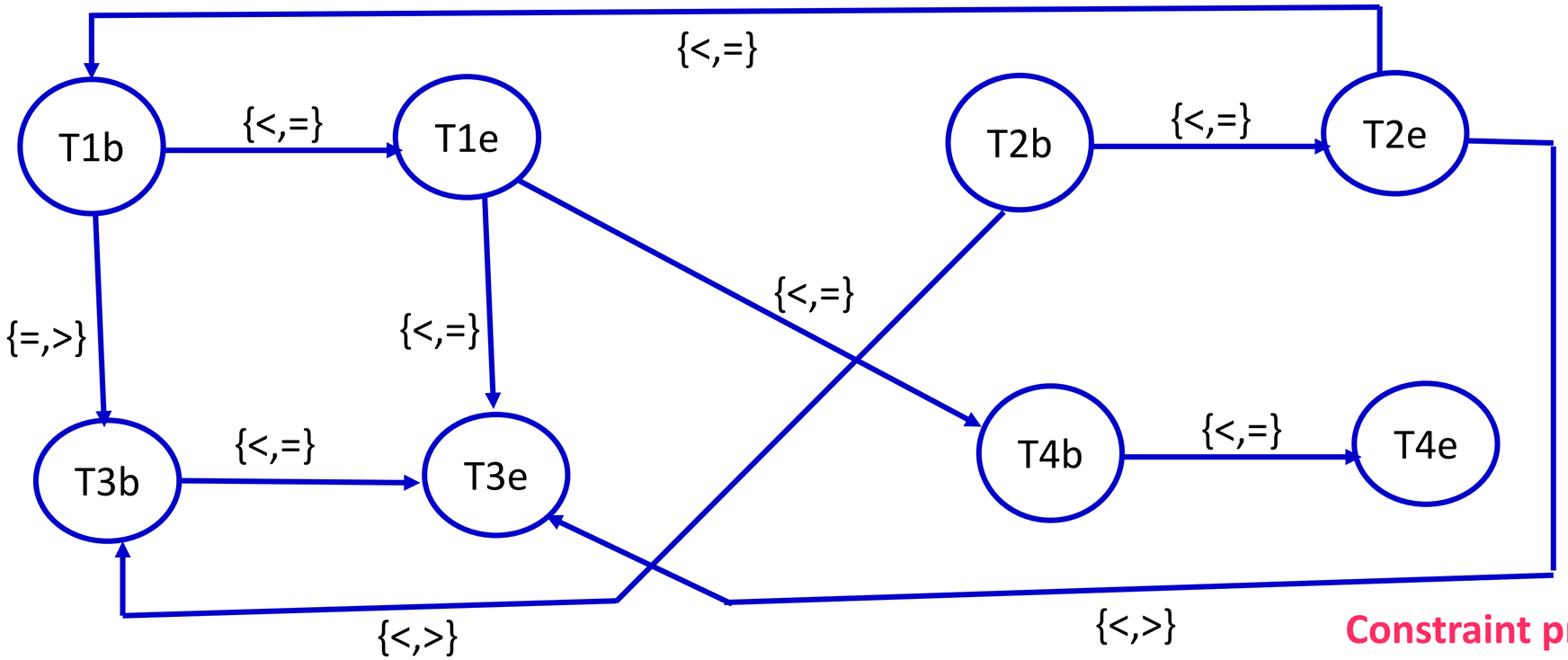
Alternative Representation: Time Point Constraints

- There are only three temporal relations:
  - Before (<)
  - Equal (=)
  - After (>)

- T1b the begin time-point of task T1, and T1e the end time-point of task T1
- Likewise, T2b and T2e for task T2, T3b and T3e for Task T3 and T4b and T4e for Task T4

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

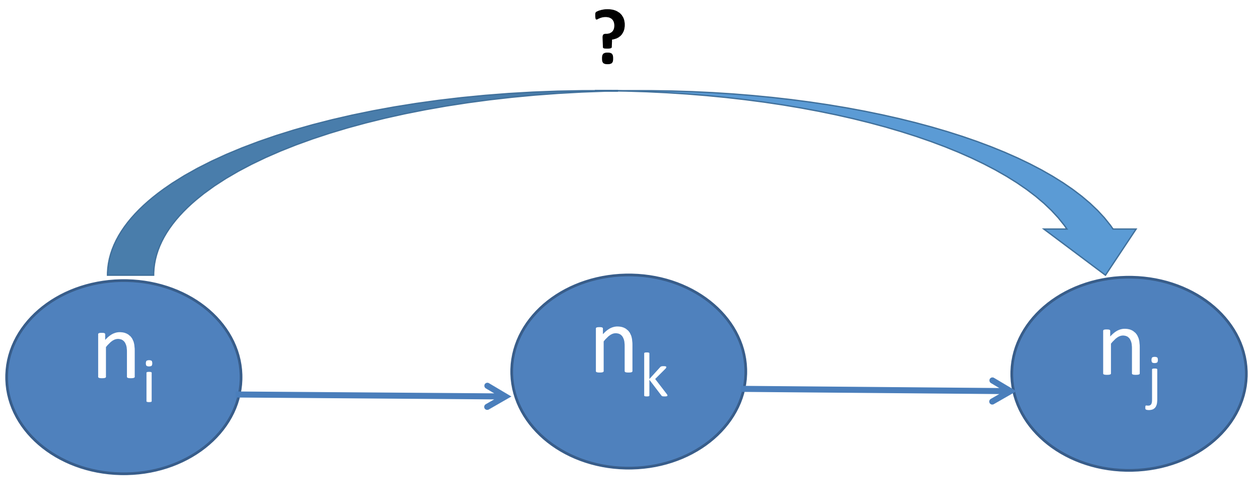
Temporal Constraints Graph



Constraint propagation to minimize disjunctions, if possible, and to derive temporal relations

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Propagating constraints



- The propagation aims to derive the possible temporal relations between each pair of nodes
- And, ideally to eliminate temporal uncertainty (relation disjunctions) producing a single possible relation between each pair of nodes – **minimization of the temporal constraints graph**
- The propagation **detects conflicts**: when all possible relations between ni and nj are refuted

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

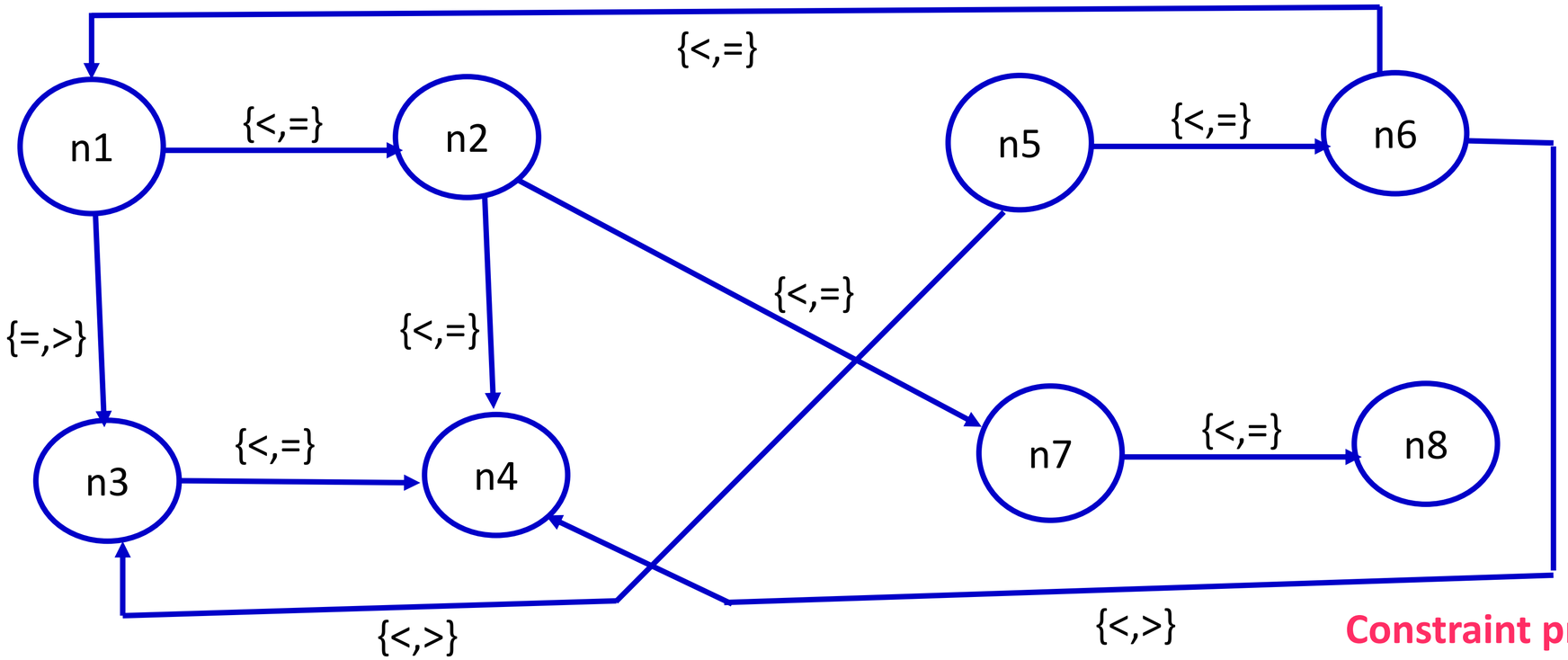
Transitivity Table driving the propagation

		C_{ik}		
	c_{ij}	<	=	>
C_{kj}	<	<	<	<, =, >
	=	<	=	>
	>	<, =, >	>	>

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Renaming nodes in the temporal graph



Constraint propagation to minimize disjunctions, if possible, and to derive temporal relations

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Original Temporal Constraints Graph

n1 ---> n2 { < = }
n1 ---> n3 { = > }
n2 ---> n4 { < = }
n2 ---> n7 { < = }
n3 ---> n4 { < = }
n5 ---> n3 { < > }
n5 ---> n6 { < = }
n6 ---> n1 { < = }
n6 ---> n4 { < > }
n7 ---> n8 { < = }

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Temporal Constraints Graph after Propagation

n1 ---> n2 { < = }
n1 ---> n3 { = > }
n1 ---> n4 { < = }
n1 ---> n5 { = > }
n1 ---> n6 { = > }
n1 ---> n7 { < = }
n1 ---> n8 { < = }
n2 ---> n3 { = > }
n2 ---> n4 { < = }
n2 ---> n5 { = > }
n2 ---> n6 { = > }
n2 ---> n7 { < = }
n2 ---> n8 { < = }

n3 ---> n4 { < = }
n3 ---> n5 { < > }
n3 ---> n6 { < = > }
n3 ---> n7 { < = }
n3 ---> n8 { < = }
n4 ---> n5 { > }
n4 ---> n6 { > }
n4 ---> n7 { < = > }
n4 ---> n8 { < = > }
n5 ---> n6 { < = }
n5 ---> n7 { < = }
n5 ---> n8 { < = }
n6 ---> n7 { < = }
n6 ---> n8 { < = }
n7 ---> n8 { < = }

Remarks:

- Some minimization was possible
  - In the original graph n6 is either before or after n4, while after propagation n4 is (definitely) after n6
- No conflicts are detected and hence in answering the initial question “all constraints are consistent”
- Some derived relations have no temporal uncertainty
  - n4 is after n5
- While others are completely temporally unconstrained
  - E.g., for the pairs (n3,n6), (n4,n7) and (n4,n8)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Minimization Algorithm

minimize (TemporalConstraintsGraph)

1. ConvertOrdered (TemporalGraph)
2. PropagateOrdered(TemporalGraph)

ConvertOrdered(TemporalConstraintsGraph)

/* Let node ordering be: n_1, n_2, …., n_m */
for i=1 to (m-1)
  for j=i+1 to m
   if nodes n_i to n_j are unconnected
      add an arc from n_i to n_j with label C /* all possible relations */
   else if there is only an arc from n_i to n_j
       do nothing
   else if there is only an arc from n_j to n_i
      add arc from n_i to n_j with label tc_{ji}^{-1}
      and delete the arc from n_j to n_i
   else $tc_{ij} \leftarrow match(tc_{ij}, tc_{ji}^{-1})$
      if tc_{ij} = {} a conflict is raised
      else delete arc from n_j to n_i

PropagateOrdered(TemporalConstraintsGraph)

/* Let node ordering be: n_1, n_2, …., n_m */
repeat
  /* for every intermediate node n_k */
  for k=2 to (m-1)
   /* for every incoming arc to n_k */
   for i=1 to (k-1)
    /* for every outgoing arc from n_k */
    for j=(k+1) to m
      /* critical step */
      $tc_{ij} \leftarrow match(tc_{ij}, propagate(tc_{ik}, tc_{kj}))$
      if tc_{ij} = {} a conflict is raised
until no arc label is reduced /* no change */

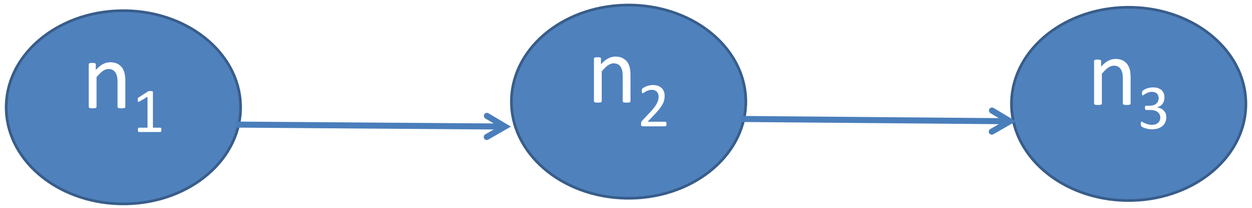
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Point of caution:

In the fully connected temporal constraints graph, there are at least two distinct paths between every pair of nodes and the propagation is bidirectional.

In the ordered temporal constraints graph the propagation is unidirectional and thus the labels of the basic arcs stay invariant under the propagation (a basic arc represents the sole path between a given pair of nodes)



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Critical Step: $tc_{ij} \leftarrow match(tc_{ij}, propagate(tc_{ik}, tc_{kj}))$

The critical step is executed $(m^3 - 3m^2 + 2m)/6$ times

$\mathcal{C}$ is the domain of binary temporal constraints.

- The elements of $\mathcal{C}$ are mutually exclusive;
- $\mathcal{C}$ is either a finite or an infinite set.

id checks whether the given constraints are identical
inverse gives the inverse of the given constraint
transit gives the disjunctive constraint from the transitivity table

$$id : \mathcal{C} \times \mathcal{C} \rightarrow \{true, false\}$$
$$inverse : \mathcal{C} \rightarrow \mathcal{C}$$
$$transit : \mathcal{C} \times \mathcal{C} \rightarrow 2^{\mathcal{C}}$$

$$match : 2^{\mathcal{C}} \times 2^{\mathcal{C}} \rightarrow 2^{\mathcal{C}}$$

match(C_i, C_j)
  $R \leftarrow \{\}$
  for $c_i \in C_i$ do
    for $c_j \in C_j$ do
      if $id(c_i, c_j)$ then $R \leftarrow R \cup \{c_j\}$
  return R

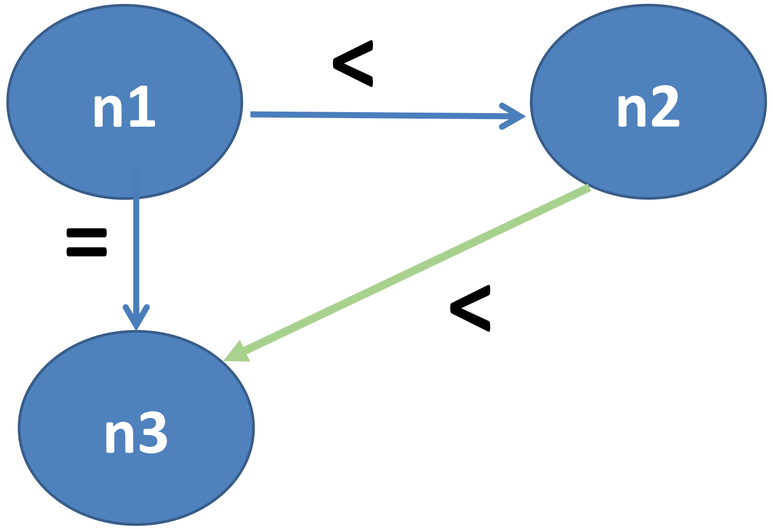
$$propagate : 2^{\mathcal{C}} \times 2^{\mathcal{C}} \rightarrow 2^{\mathcal{C}}$$

propagate(C_i, C_j)
  $R \leftarrow \{\}$
  for $c_i \in C_i$ do
    for $c_j \in C_j$ do
      $R \leftarrow R \cup transit(c_i, c_j)$
  return R

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

A simple example where a conflict occurs




```
n1 ---> n2   { < }
n1 ---> n3   { = }
n2 ---> n3   { < }
```

After propagation

```
n1 ---> n1   { = }
n1 ---> n2   { < }
n1 ---> n3   { = }
n2 ---> n1   { > }
n2 ---> n2   { = }
n2 ---> n3   { < }
n3 ---> n1   { = }
n3 ---> n2   { > }
n3 ---> n3   { = }
```

```
Conflict detected between
nodes n2 and n3
```

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

A Final Example

- Diagnosis reached (n_1)
- Discussion about therapy (n_2 – n_3)
- Consensus on best therapy regime (n_4)
- Patient agreed on proposed therapy (n_5)
- Application of therapy (n_6 – n_7)
- Adjustment of some therapy parameters (n_8)
- Patient monitoring (n_9 – n_{10})

Original Temporal Constraints Graph

```
n1  ---> n3   { <   = }
n2  ---> n1   { <   = }
n4  ---> n3   { =   > }
n5  ---> n4   { =   > }
n6  ---> n5   { =   > }
n8  ---> n6   { > }
n8  ---> n7   { < }
n9  ---> n2   { <   = }
n10 ---> n7   { > }
```

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Temporal Constraints after propagation

No conflicts detected and hence the conistency of the original temporal constraints is demonstrated

```
n1 ---> n2   { =  > }
n1 ---> n3   { <  = }
n1 ---> n4   { <  = }
n1 ---> n5   { <  = }
n1 ---> n6   { <  = }
n1 ---> n7   { < }
n1 ---> n8   { < }
n1 ---> n9   { =  > }
n1 ---> n10   { < }
n2 ---> n3   { <  = }
n2 ---> n4   { <  = }
n2 ---> n5   { <  = }
n2 ---> n6   { <  = }
n2 ---> n7   { < }
n2 ---> n8   { < }
n2 ---> n9   { =  > }
n2 ---> n10   { < }
```

```
n3 ---> n4   { <  = }
n3 ---> n5   { <  = }
n3 ---> n6   { <  = }
n3 ---> n7   { < }
n3 ---> n8   { < }
n3 ---> n9   { =  > }
n3 ---> n10   { < }
n4 ---> n5   { <  = }
n4 ---> n6   { <  = }
n4 ---> n7   { < }
n4 ---> n8   { < }
n4 ---> n9   { =  > }
n4 ---> n10   { < }
n5 ---> n6   { <  = }
n5 ---> n7   { < }
n5 ---> n8   { < }
n5 ---> n9   { =  > }
n5 ---> n10   { < }
```

```
n6 ---> n7   { < }
n6 ---> n8   { < }
n6 ---> n9   { =  > }
n6 ---> n10   { < }
n7 ---> n8   { > }
n7 ---> n9   { > }
n7 ---> n10   { < }
n8 ---> n9   { > }
n8 ---> n10   { < }
n9 ---> n10   { < }
```

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Generalizing temporal constraint graphs

- Temporal constraint graphs can be generalized to variable constraint graphs
  - Nodes represent variables and arcs represent constraints on the values of pairs of variables
- The discussed temporal constraint propagation procedure can be appropriately adapted based on the notions of arc, path and K, consistency
- **Arc consistency** between pairs of variables: A CSP (hence a constraint graph) has arc consistency, if for every pair of variables (Xi,Xj), for each value vi of variable Xi there exists a value vj for variable Xj such that the assignment Xi=vi, Xj=vj, satisfies all the constraints between Xi and Xj
- Arc consistency can be obtained by removing values from the domains of Xi and Xj (minimization)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Path consistency and K-Consistency

- **Path consistency** which entails three variables, is the extension of arc consistency; a CSP has path consistency if every assignment of values on a pair of variables Xi and Xj that satisfies the given constraints, can be extended to whichever third variable Xk
- **K-consistency** represents a higher degree of consistency
  - 1-Consisteny (Node Consistency): Each variable is checked separately in relation to unary constraints referring to it
  - 2-Consistency is arc consistency: Every consistent assignment to some variable can be extended in every other variable
  - 3-Consistency is path consistency: Any consistent pair assignment can be extended to any third variable
  - K-consistency: For every k variables, whichever k-1 consistent assignment can be extended to the kth variable
  - If on the basis of the above consistency checks, all values in the domain of some variable have been eliminated then an inconsistency has been detected

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Types of CSPs

- Discrete Variables
  - Finite Domains
    - A domain size of d means $O(d^n)$ possible assignments
    - NP-complete
  - Infinite Domains (e.g., integers)
    - E.g., job scheduling where variables correspond to the start time of each job
    - Linear constraints solvable, nonlinear undecidable
- Continuous Variables
  - E.g., start times of Hubble Telescope observations
  - Linear inequalities can be solved in polynomial time

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Real CSPs

- Course scheduling: e.g., which courses are offered when and in which rooms?
- Assignment of crews/aircraft on flights
- Gate assignment to airports
- Transportation scheduling
- Industrial production planning

- Many real problems include variables with real values . . .
- We are not addressing such problems in this unit; the problems addressed deal with discrete value variables

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Improvements for reducing search time

- Selection of variables/values
  - Which variable will be next in getting a value?
  - In which order to assign the values?
- Filtering/propagation
  - Could failed assignments be detected early?
    - Forward checking
- Structure
  - Could the problem structure be exploited?
  - Decompose problem into subproblems that can be solved independently

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Order of Selecting Variables

- Minimum-remaining-values MRV
  - Select the variable that has the smallest number of remaining values
- Most-constrained-variable heuristic
  - Select the variable that appears in most constraints with other unbound variables

Order of Selecting Values (for the selected variable)

- Least-constraining-value-heuristic
  - Select the value that would leave the biggest subset of values for the unbound variables

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Game Playing – search methods in two player, perfect information, zero-sum games

Adopted from Nils J. Nilsson’s classical book on Principles of Artificial Intelligence

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on constraint satisfaction problems, game playing and planning, students will be able:

Regarding Game Playing:

1. To explain the category of two player, perfect information, zero-sum games, which is the topic of discussion.
2. To analyze the representation problem regarding the above category of games.
3. To explain the minimax procedure as well as the alpha-beta procedure and the relevant pruning rules, and to be able to apply these on simple two-player games.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Two-player, perfect information, zero-sum games

- The two players play in turn
- Perfect information: everything is fully observable
- Zero-sum: what is good for one player is just as bad for the other
  - There is no “win-win” outcome

Example games of this category

- Checkers, tic-tac-toe, chess, go and nim

There are many other categories of games

- Multi-player games or partially observable games
- Games whose results are determined by chance, e.g., dice or most card games

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Representation Problem

For two-player, perfect information, zero-sum games

- A problem state gives a permitted configuration of the game and the player who is next to play, i.e., to apply an action (to make a move) on the given state
  - The players are usually referred to as MAX and MIN, e.g., MAX could be the computer and MIN the human player
- The operators are the rules of the game
  - These are applied in turn by MAX and MIN until a win state for the one or the other is reached or a draw between the two results – these are the terminal states
  - We assume that a (pursued) goal state represents a win for MAX
- A game tree depicts all possible scenarios for the two players
  - Constructing a game tree to termination is possible only for very simple games

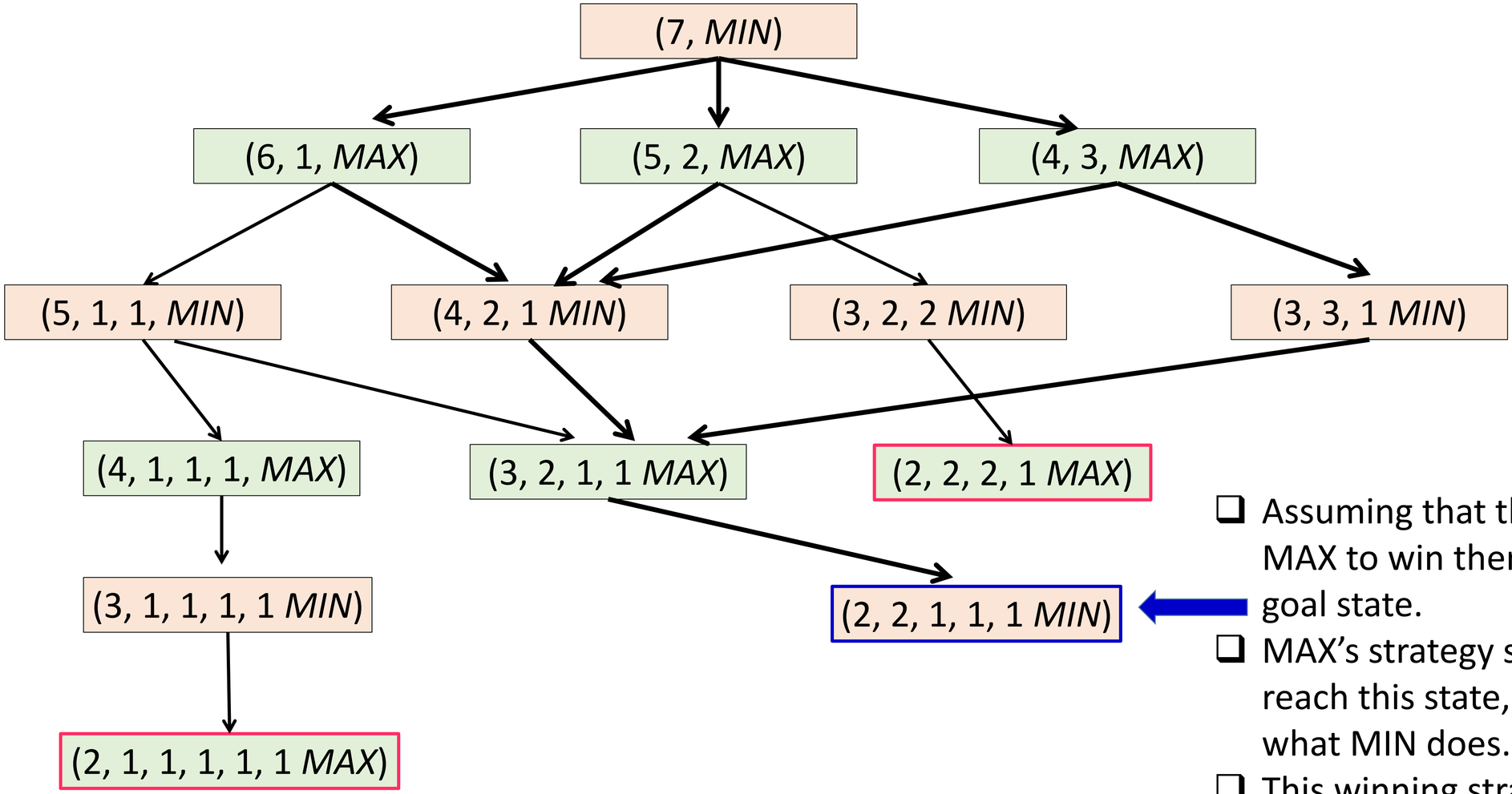
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

An example of such a simple game: “Grundy’s game”

- Rules of the game:
  - The two players have in front of them a single pile of objects, say a stack of Euro cents
  - The first player divides the original stack into two stacks that must be **unequal**
  - Each player in turn does the same to a **single** stack
  - The game comes to a stop when all stacks consist of either one cent or two cents
  - The player who first cannot play is the loser
- A problem state is a sequence of numbers giving the sizes of the stacks together with who is next to play, MIN or MAX
- A complete game tree for a specific instance of the game, where the original stack consists of 7 cents and the first to play is MIN is shown next

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

A game tree for an instance of Grundy's game

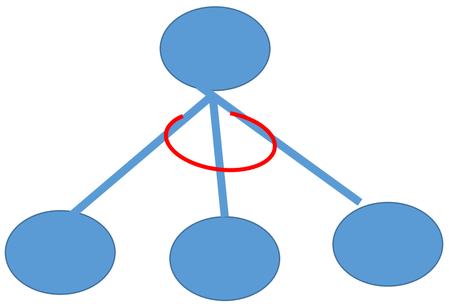


- Assuming that the aim is for MAX to win there is a single goal state.
- MAX's strategy should aim to reach this state, irrespective of what MIN does.
- This winning strategy for MAX is shown in bold arrows.

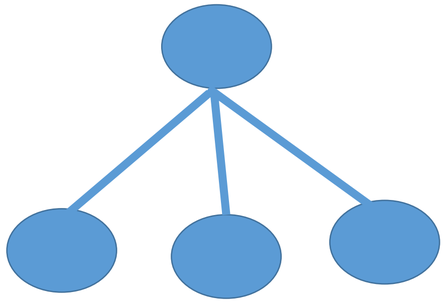
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Constructing a winning strategy for MAX – AND/OR states

- For every state from which it is MIN's move next , MAX should be able to win from every state resulting from whichever move MIN chooses to do
  - Thus, successor states of states from which it is MIN's move next are **AND states**
- For every state from which it is MAX's move next, it is only needed for MAX to win from just one of the states to which he might move to
  - Thus, successor states of states from which it is MAX's move next are **OR states**



AND



OR

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The Minimax Procedure: Assumptions and Objective

- For complex games, constructing a game tree to completion is completely out of the question
  - A complete game tree for checkers has approximately **10^{40} nodes** and the chess tree has approximately **10^{120} nodes**, and heuristic search techniques do not reduce the effective branching factor sufficiently to be of much help.
- **Assumptions**: MAX is always the first to play and both MAX and MIN are capable players playing as good as each other, i.e., none of the two players should underestimate its opponent.
- **Objective**: Find a “best” first move for MAX, and after each move by MIN find the “best” next move for MAX

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Steps of the Minimax Procedure

1. **Looking ahead**: Expand the game tree up to a given depth, using the depth-first or breadth-first search method; the start node is a MAX node.
2. **Evaluating leaf nodes**: Apply a static evaluation function to the leaf nodes to measure the "worth" of the game positions represented by them:
   - The evaluation function gives a positive/negative value for a game position favorable to MAX/MIN
   - $+\infty/-\infty$ denotes a win for MAX/MIN
3. **Backing-up evaluation values level by level from leaf nodes to start node**:
   - ❑ A MAX node gets the maximum value amongst its successor MIN nodes
   - ❑ A MIN node gets the minimum value amongst its successor MAX nodes
   - ❑ The successors of the start node are eventually assigned backed-up values; MAX should then choose as its first/next move the one corresponding to the successor having the largest backed-up value.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Utility of the Minimax Procedure

- The utility rests on the assumption that the backed-up values of the start node's successors are more reliable measures of the ultimate relative worth of these positions than are the values that would be obtained by directly applying the static evaluation function on the successors.
- Since the backed-up values are based on "looking ahead" in the game tree, they depend on features occurring nearer the end of the game; hence how deeply the looking ahead could go is of significance.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

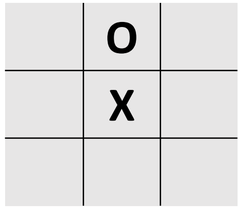
Illustrating the minimax procedure through the game of tic-tac-toe

- MAX marks crosses (X) and MIN marks circles (O)
- Looking ahead goes down to level 2; the start node (which is a MAX node) is at level 0
- The evaluation function, e(p), is defined as:
  - e(p) = +∞, if p is a winning position for MAX
  - e(p) = -∞, if p is a winning position for MIN
  - e(p) = (number of complete rows, columns, or diagonals still open for MAX) – (number of complete rows, columns, or diagonals still open for MIN), otherwise

Co-financed by the European Union
Connecting Europe Facility

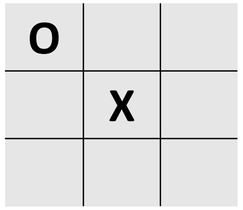
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

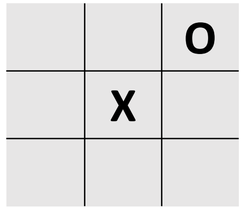
state p

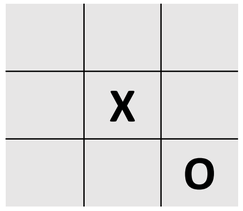


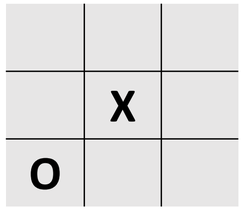
e(p) = 6 – 4 = 2

Symmetries can be used when generating successor positions, e.g. the following states are considered identical:





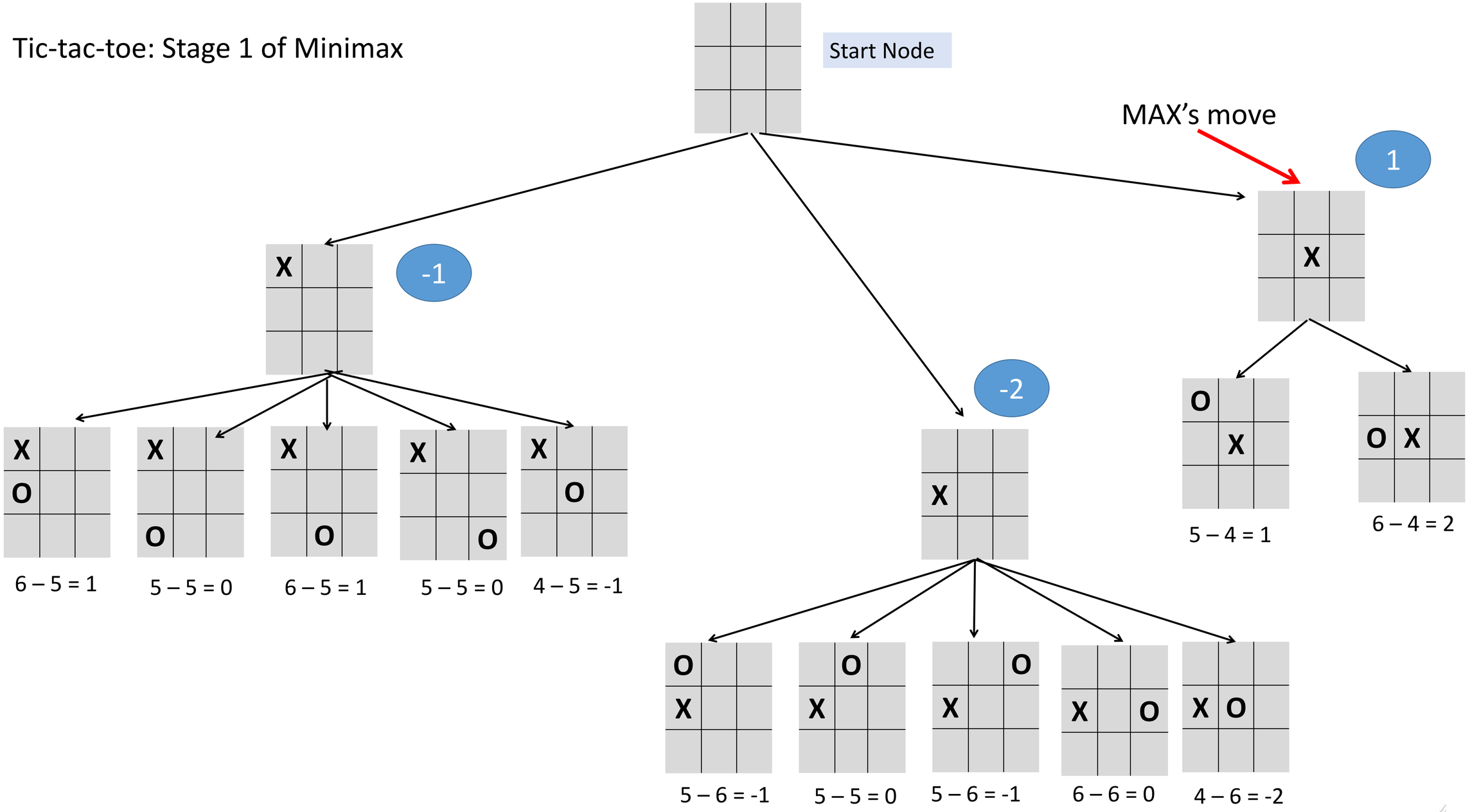




Early in the game the branching factor is kept small by symmetries; late in the game, it is kept small by the number of open spaces available.

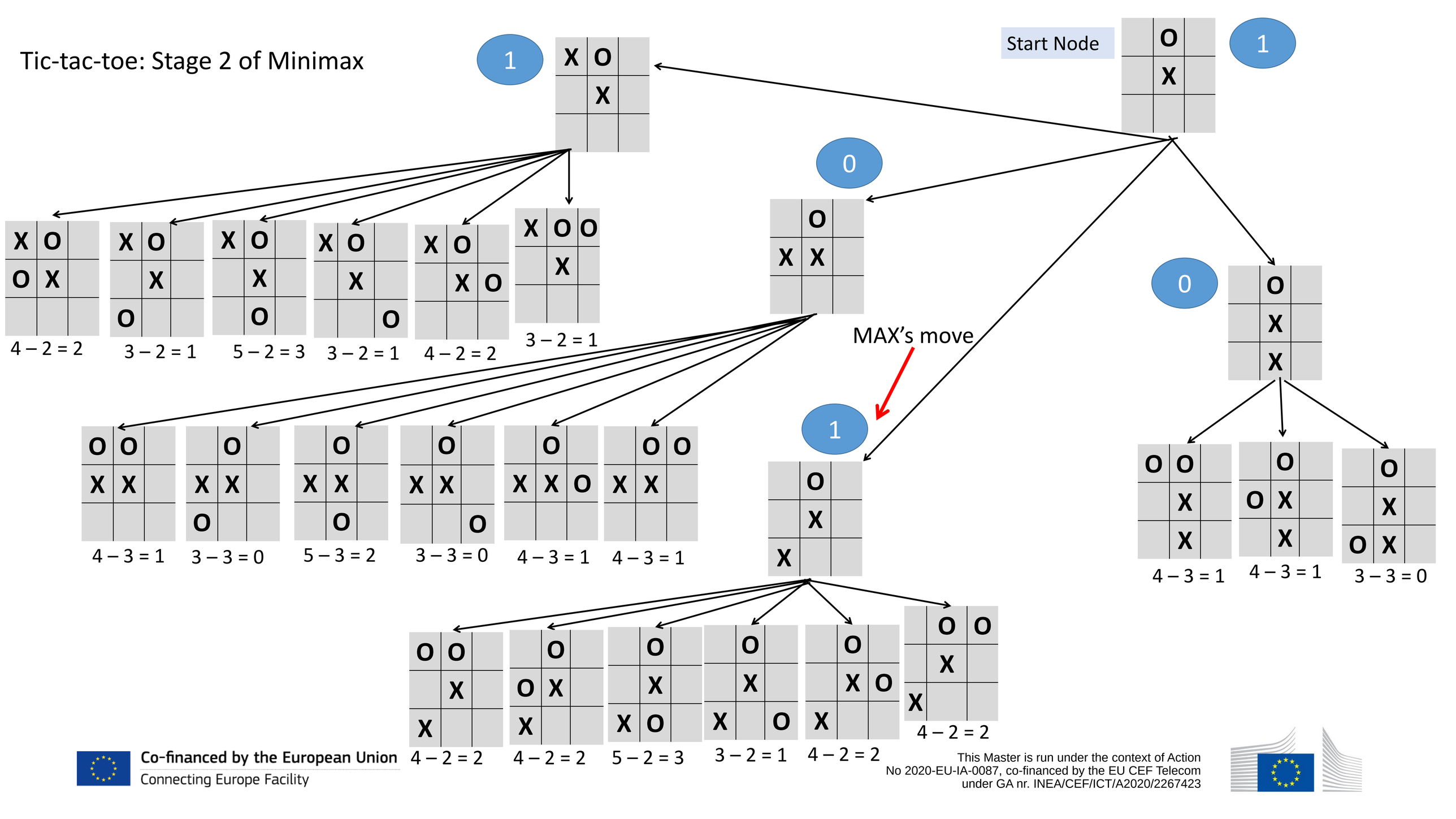
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Tic-tac-toe: Stage 1 of Minimax



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

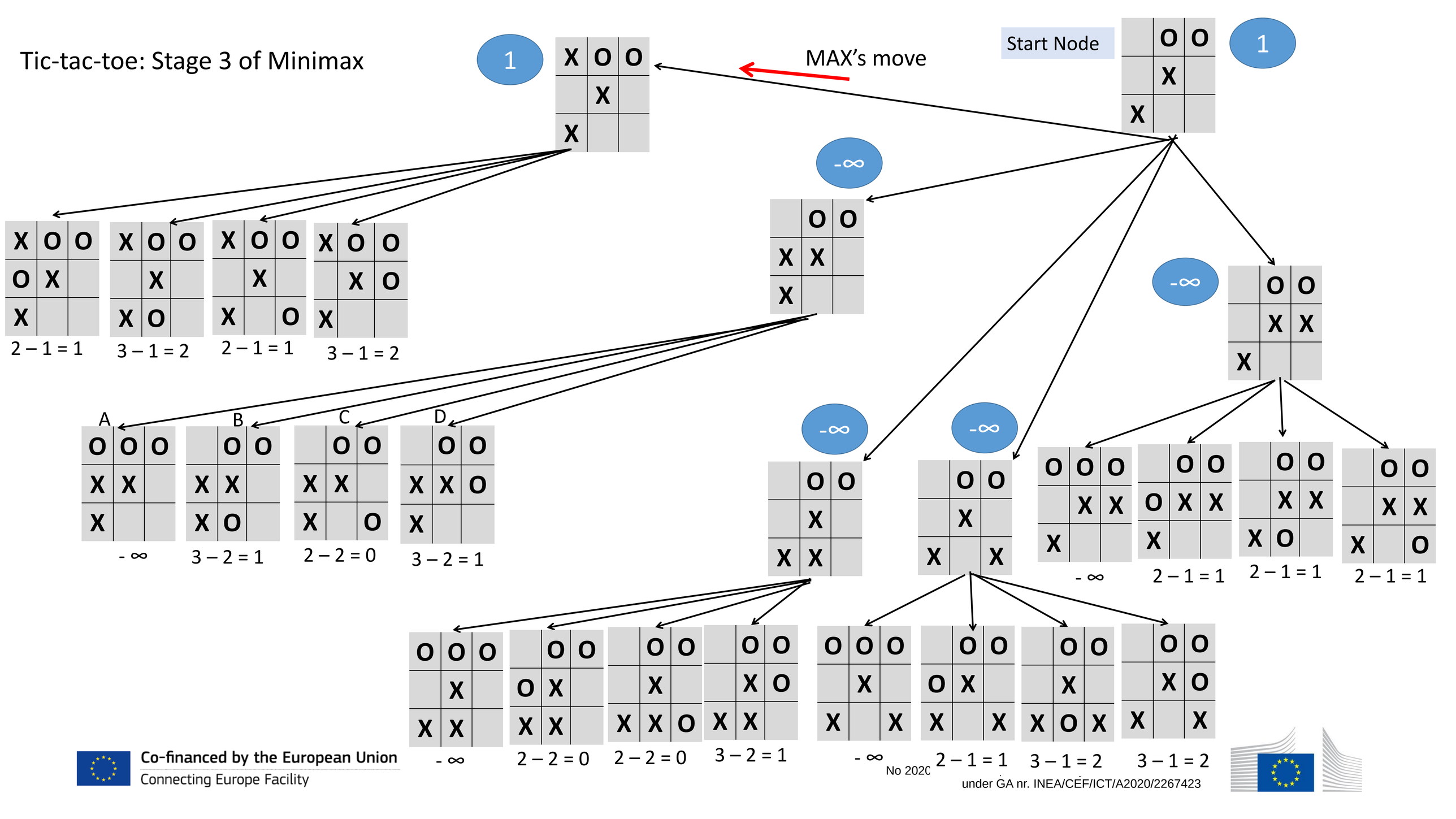
Tic-tac-toe: Stage 2 of Minimax



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Tic-tac-toe: Stage 3 of Minimax



Co-financed by the European Union
Connecting Europe Facility

No 2020
under GA nr. INEA/CEF/ICT/A2020/2267423

The Alpha-Beta Procedure

- In the discussed minimax procedure, the generation of the search tree (to the given look-ahead depth), and the evaluation of the leaf nodes together with the calculation of the back-up values are distinct processes
  - This separation results in a grossly inefficient strategy

- Remarkable reductions, possibly to many orders of magnitude, in the amount of search needed to discover an equally good move, can result if the leaf node evaluations and the calculation of back-up values is interleaved with the tree generation
  - The reductions are greater when searching to greater depths

Co-financed by the European Union
Connecting Europe Facility

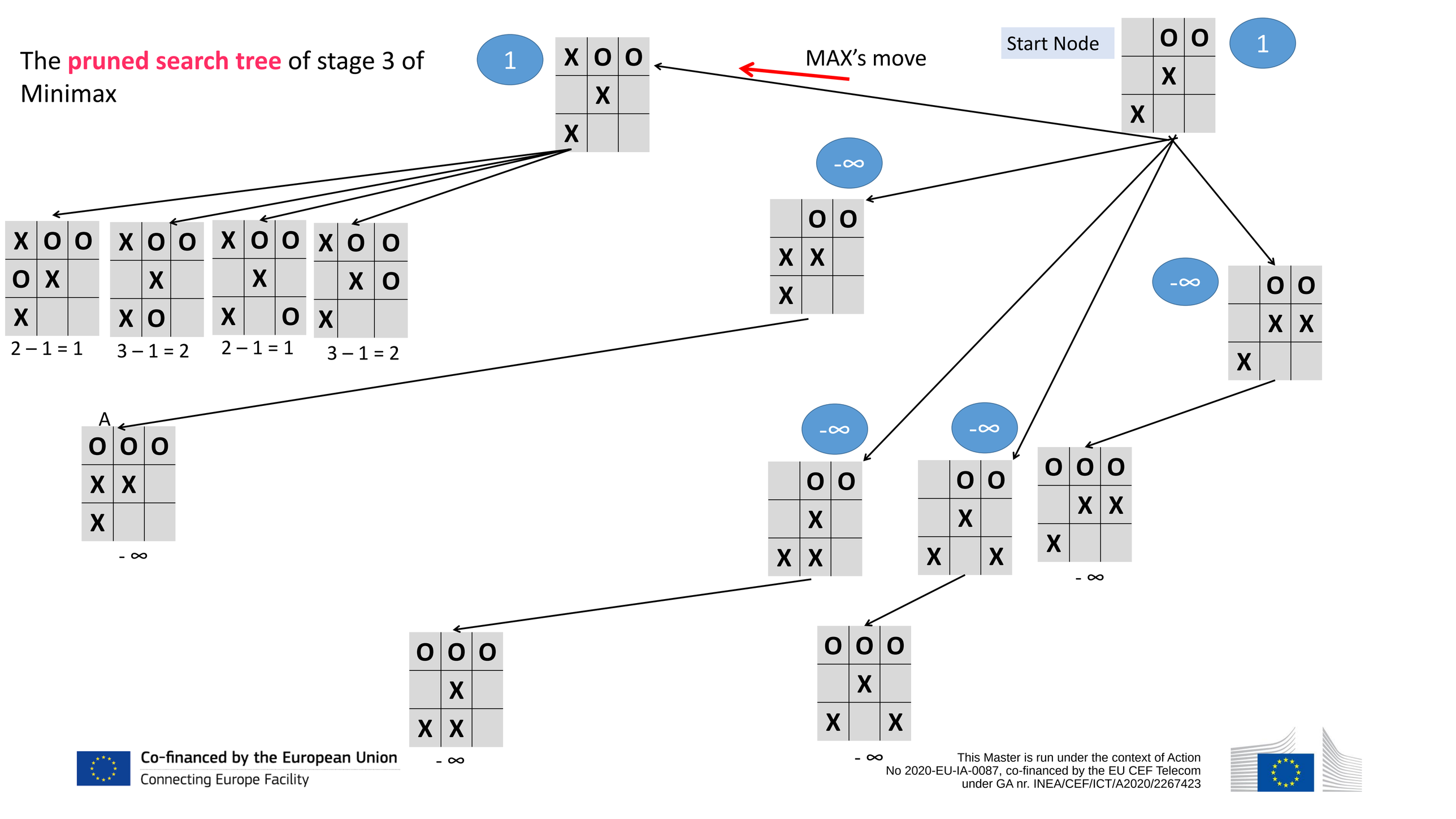
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Consider the game tree of stage 3 of the tic-tac-toe example

- Suppose that each leaf node is evaluated as soon as it is generated.
- When node A is generated and evaluated, that represents a win for MIN, there is no point in generating and evaluating nodes B, C and D.
  - MIN will certainly choose A
  - Hence A's parent can be assigned the backed-up value of -∞ and proceed with the search
  - Failing to generate nodes B, C and D can in no way affect what will turn out to be MAX's best move

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

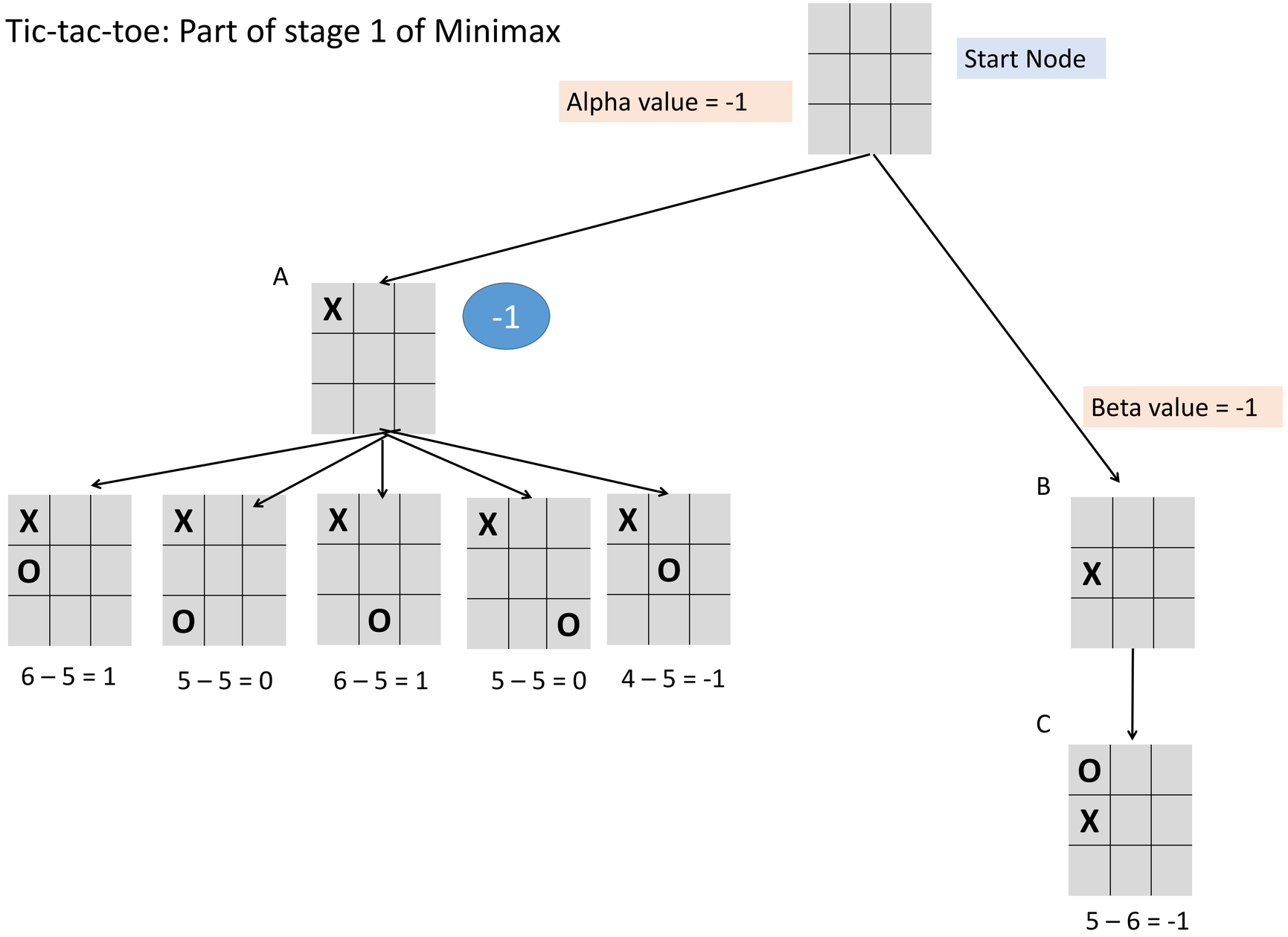
The **pruned search tree** of stage 3 of Minimax



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Tic-tac-toe: Part of stage 1 of Minimax

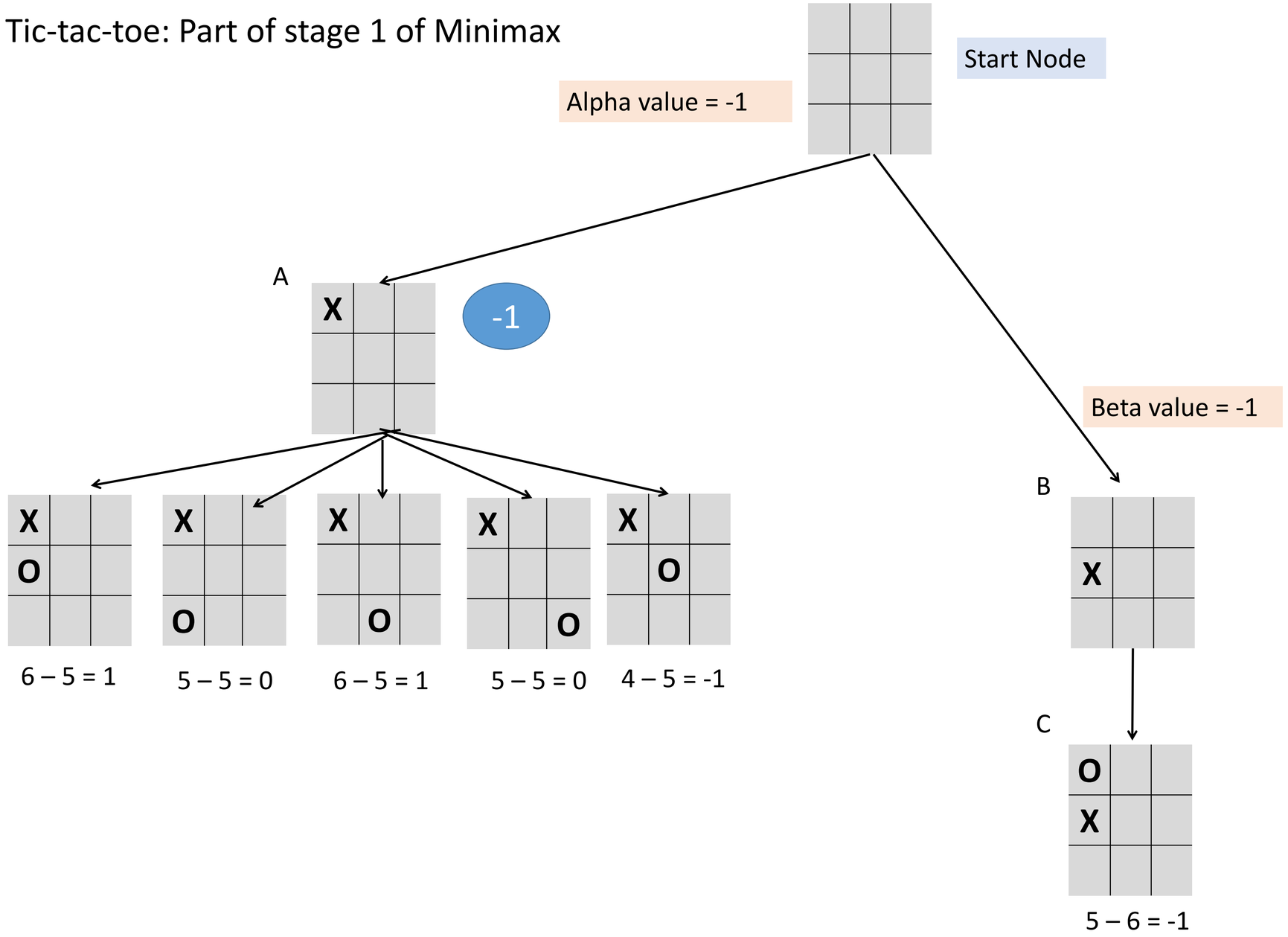


As soon as node A gets its backed-up value of -1, we know that the backed-up value of the (MAX) start node is bounded from below by -1. As further successors of the start node are generated and get their backed-up values, the final backed-up value of the start node can increase, but it cannot get below -1. This lower bound is referred to as the **alpha value** for the start node.

When node C is generated and evaluated, we know that the backed-up value of its parent, i.e., (MIN) node B, is bounded from above by -1. It can get lower when other successors of B are generated but can never get higher. This upper bound is referred to as the **beta value** for node B.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Tic-tac-toe: Part of stage 1 of Minimax



Given that the backed-up value of (MAX) start node cannot get lower than -1, and the backed-up value of (MIN) node B cannot get higher than -1, there is no point in continuing the search further below node B, since node B will never make a better choice than node A for the start node.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Alpha-Beta Pruning of the Search Tree

- As the successors of a node are given backed-up values, the bounds on backed-up values can be revised, noting though:
  - The alpha values of MAX nodes (including the start node) can never decrease, and
  - The beta values of MIN nodes can never increase

- **Pruning Rules:**
  1. Discontinue search below a MIN node that has a beta value less than or equal to the alpha value of any of its MAX node ancestors. The final backed-up value of this MIN node can then be set to its beta value; this is referred to as an **alpha cutoff**.
  2. Discontinue search below a MAX node that has an alpha value greater than or equal to the beta value of any of its MIN node ancestors. The final backed-up value of this MAX node can then be set to its alpha value; this is referred to as a **beta cutoff**.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

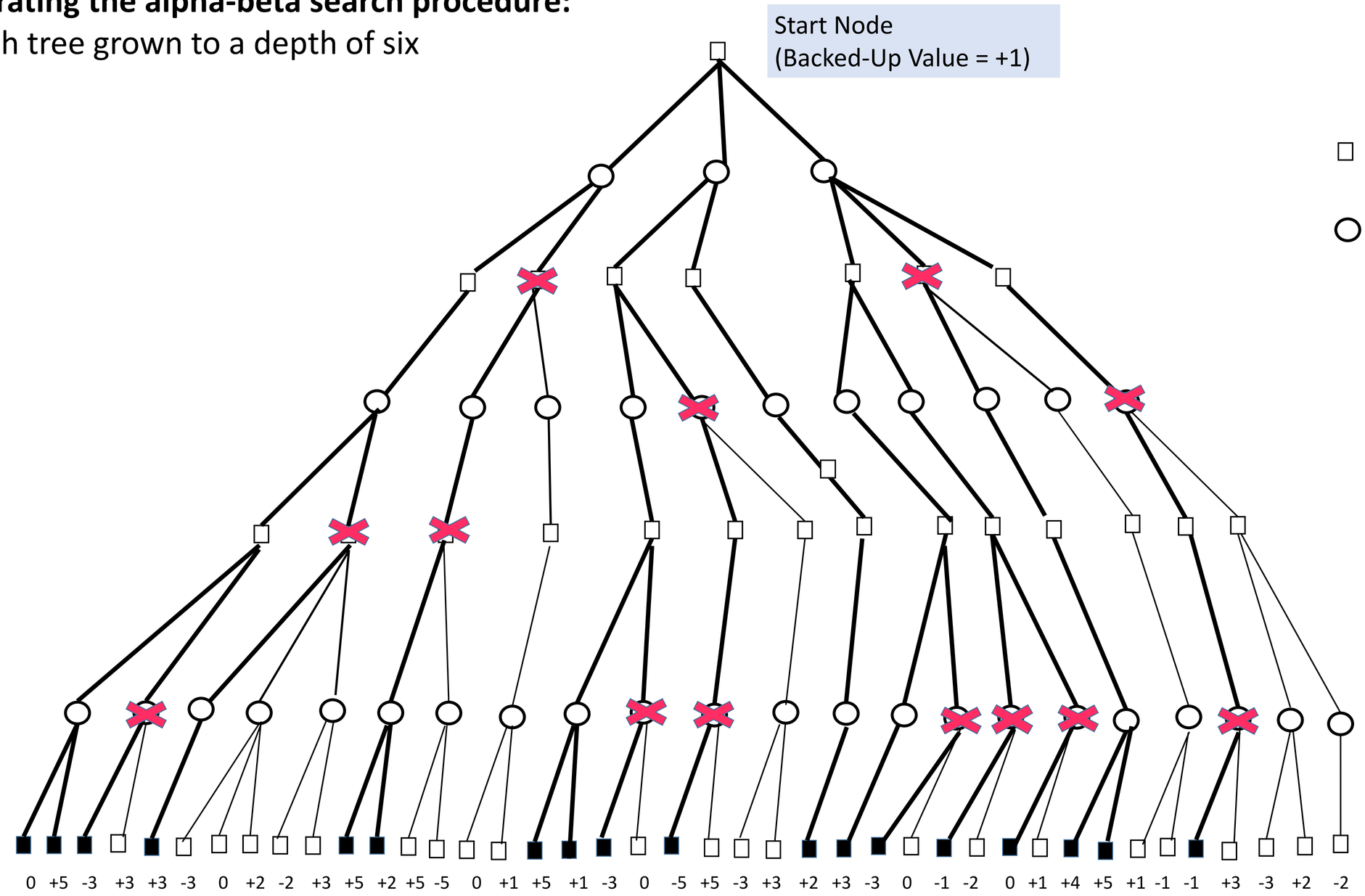
Computing alpha and beta values and procedure termination

- The alpha value of a MAX node is set equal to the current largest final backed-up value of its successors.
- The beta value of a MIN node is set equal to the current smallest final backed-up value of its successors.

- The alpha-beta procedure terminates when all the successors of the start node have been given final backed-up values, and the best move is the one creating that successor having the highest backed-up value.
- Minimaxing with alpha-beta pruning always results in finding, usually after much less search, a move that is equally as good as the move that would have been found by the simple minimax method searching to the same depth.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Illustrating the alpha-beta search procedure:
Search tree grown to a depth of six



The subtree generated through alpha-beta pruning is shown in bold lines; only 18 of the original 41 leaf nodes had to be generated and evaluated; depth-first search is usually employed.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Search efficiency of the alpha-beta procedure

- The number of leaf nodes of depth D that would be generated by optimal alpha-beta search (maximizes the number of cutoffs and minimizes the number of leaf nodes generated) is about the same as the number of leaf nodes that would have been generated at depth D/2 without alpha-beta.
- Hence alpha-beta can double the look-ahead depth, giving MAX higher game-playing abilities!

Comment: Deep Blue used custom VLSI chips to parallelize the **alpha-beta search algorithm**, an example of GOFAI (Good Old-Fashioned Artificial Intelligence). The system derived its playing strength mainly from brute force computing power.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Planning

Finding a sequence of actions to accomplish a goal in a discrete, deterministic, static, fully observable environment – classical planning

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on constraint satisfaction problems, game playing and planning, students will be able:

Regarding Planning:

1. To give high-level definitions of a planning system and of a linear plan as a sequence of actions to accomplish a goal in a discrete, deterministic, static and fully observable environment.
2. To analyze the representation problem for the above category of planning systems.
3. To make a reference to the Planning Domain Definition Language (PDDL), the STRIPS model and the action schema.
4. To discuss search-based solutions for the construction of linear plans and to distinguish between forward (progression) search and backward (regression) search pointing strengths and weaknesses.
5. To outline the decomposability for non-interacting component goals of complex goals and to overview the classical means-ends-analysis search method associated with the General Problem Solver (GPS).
6. To discuss general heuristics, based on problem relaxation, for reducing the search time.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Classical Planning

- Belongs to the category of synthesis or constructive problems as the objective is to construct a sequence of actions which when executed in the specified order will convert a given initial state to a given goal state – e.g., the 8-puzzle
- Plan is the sequence of actions and planner is the program embodying a search method that derives the plan.
- Classical search methods tend to be insufficient for real planning problems due to
  - Combinatorial explosion
  - Inability to define heuristics
  - Inability to decompose a planning problem into sub-problems
- An important issue is the specification of a definition language to support the application of appropriate algorithms

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

PDDL – Planning Domain Definition Language

- The representation problem applies to planning problems:
  - Initial state, goals, available actions

- PDDL is currently the basic language for describing planning problems
  - There are several versions, differing on their power of expression
  - More expressive versions demand more complex algorithms
  - The simplest subset of PDDL is the STRIPS language

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The STRIPS model

- **S**tanford **R**esearch **I**nstitute **P**lanning **S**ystem
  - Proposed in 1971 by Fikes and Nilsson for a small robot named Shakey for executing simple actions
  - Simple model using elements of propositional logic
  - Suitable for problems where no uncertainty is involved
  - Initially it did not support the representation of temporal or other constraints

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

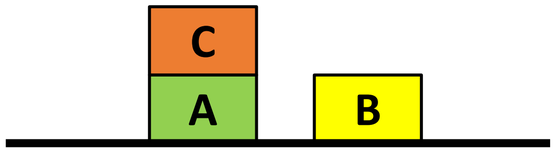
Assumptions

- Indivisible actions
  - No concern about the world state during the execution of an action
- Deterministic effects of actions
- Omniscience
  - The planning system has full knowledge of the current world state and of its own abilities
- Closed-world assumption
  - Whatever has not been declared as true in the initial state, is considered to be false
- Static world
  - The world changes only through the actions of the planning system

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

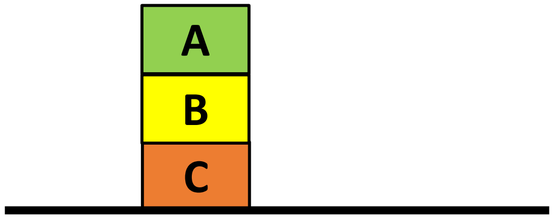
States in PDDL

- A state is a conjunction of ground, atomic, fluents:
  - Ground: has no variables
  - Fluent: can change over time (it is true and then becomes false, or vice versa)
  - Ground atomic: there is a single predicate, and it there are arguments, they are constant
- Hence states are defined as sets of specific facts (or statements) that are true
  - Example



initial state

Init(Block(A) ∧ Block(B) ∧ Block(C) ∧ On(C,A) ∧
On(A,Table) ∧ On(B,Table) ∧ Clear(B) ∧ Clear(C))



goal state

Goal(On(B,C) ∧ On(A,B))

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Action Schemas in PDDL

- An action schema represents a family of ground actions
- Examples

Action(**Pickup**(x),
PRECOND: On(x,Table) ∧ Clear(x) ∧ Handempty
EFFECT: ~Handempty ∧ ~ On(x,Table) ∧ ~Clear(x) ∧ Holding(x))

Action(**Putdown**(x),
PRECOND: Holding(x)
EFFECT: ~Holding(x) ∧ On(x,Table) ∧ Clear(x) ∧ Handempty)

Action(**Stack**(x,y),
PRECOND: Holding(x) ∧ Clear(y)
EFFECT: ~Holding(x) ∧ ~Clear(y) ∧ Handempty ∧ On(x,y) ∧ Clear(x))

Action(**Unstack**(x,y),
PRECOND: Handempty ∧ Clear(x) ∧ On(x,y)
EFFECT: ~Handempty ∧ ~Clear(x) ∧ ~On(x,y) ∧ Holding(x) ∧ Clear(y))

Co-financed by the European Union
Connecting Europe Facility

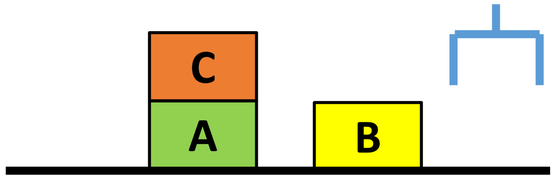
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

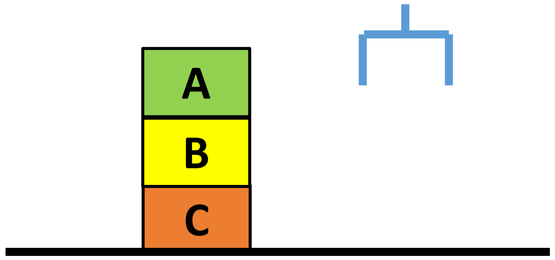
Applicable action instances

- A ground action, or an action instance, is obtained from the action schema by replacing variables with constants
- A ground action, **a**, is applicable in state **s**, if **s** entails the precondition of **a**:
  - Every positive literal in the precondition is in s and every negative literal is not
- The effect of **a** can be divided into
  - a delete list, or DEL(a), consisting of the negative literals
  - an add list, or ADD(a), consisting of the positive literals
- The result of executing **a** in **s** is a new state **s'** obtained by removing from **s** the fluents that appear in DEL(a) and adding the fluents that appear in ADD(a)
  - RESULT(s, a) = (s – DEL(a)) U ADD(a)
  - Recall the frame problem: specifying which formulae in a state description should change and which should not

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

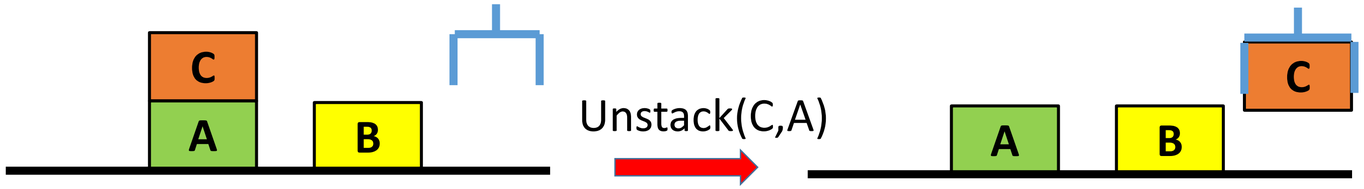
Example Linear Plan

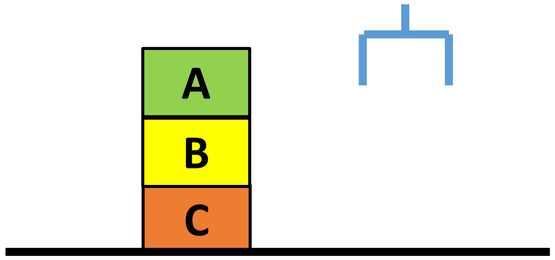




This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

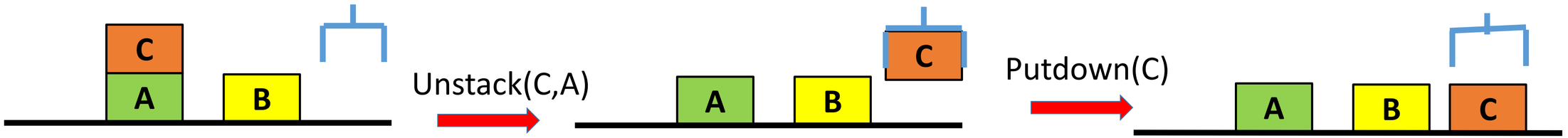
Example Linear Plan

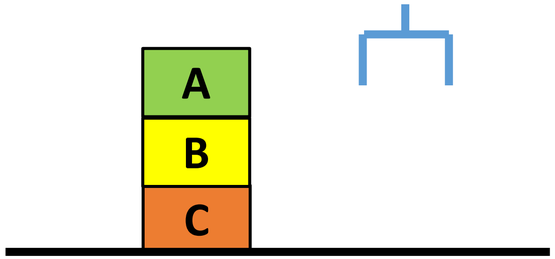




This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

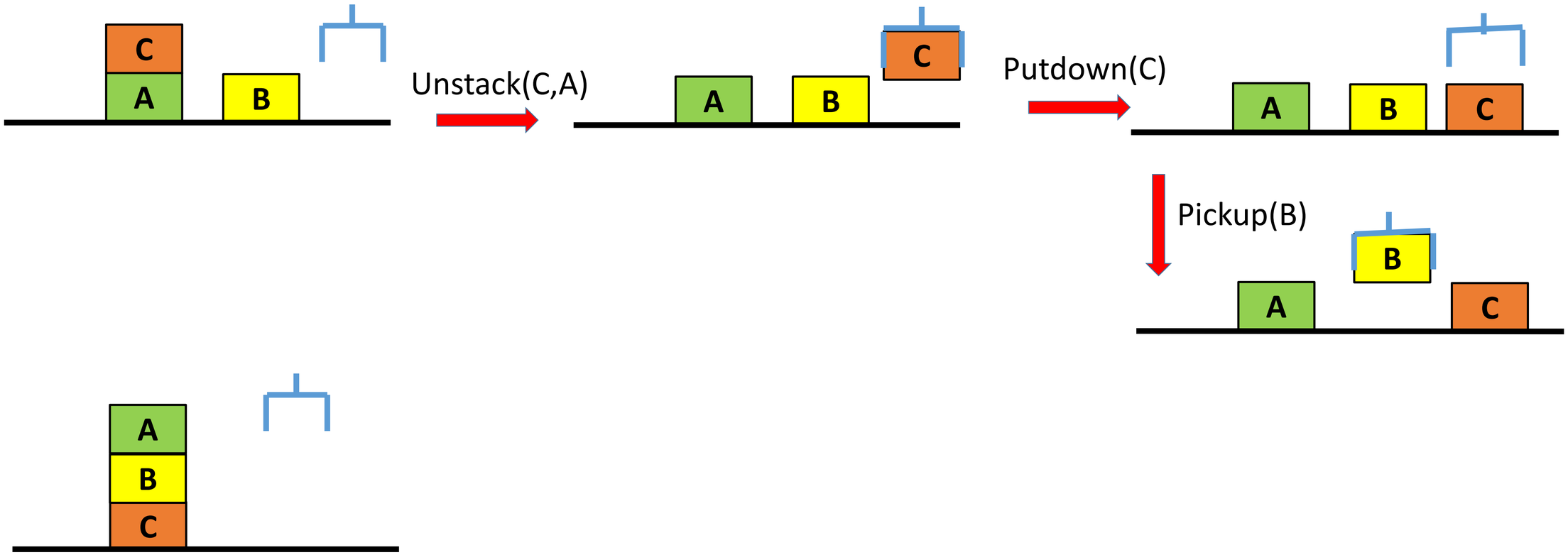
Example Linear Plan





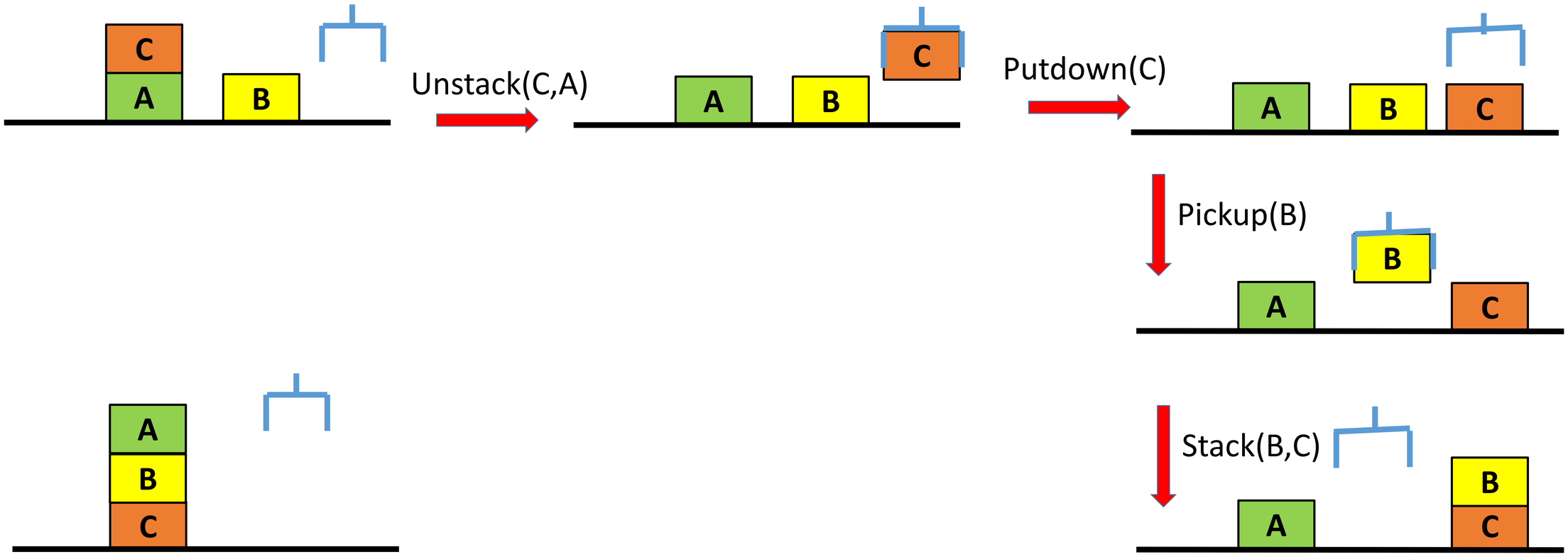
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example Linear Plan



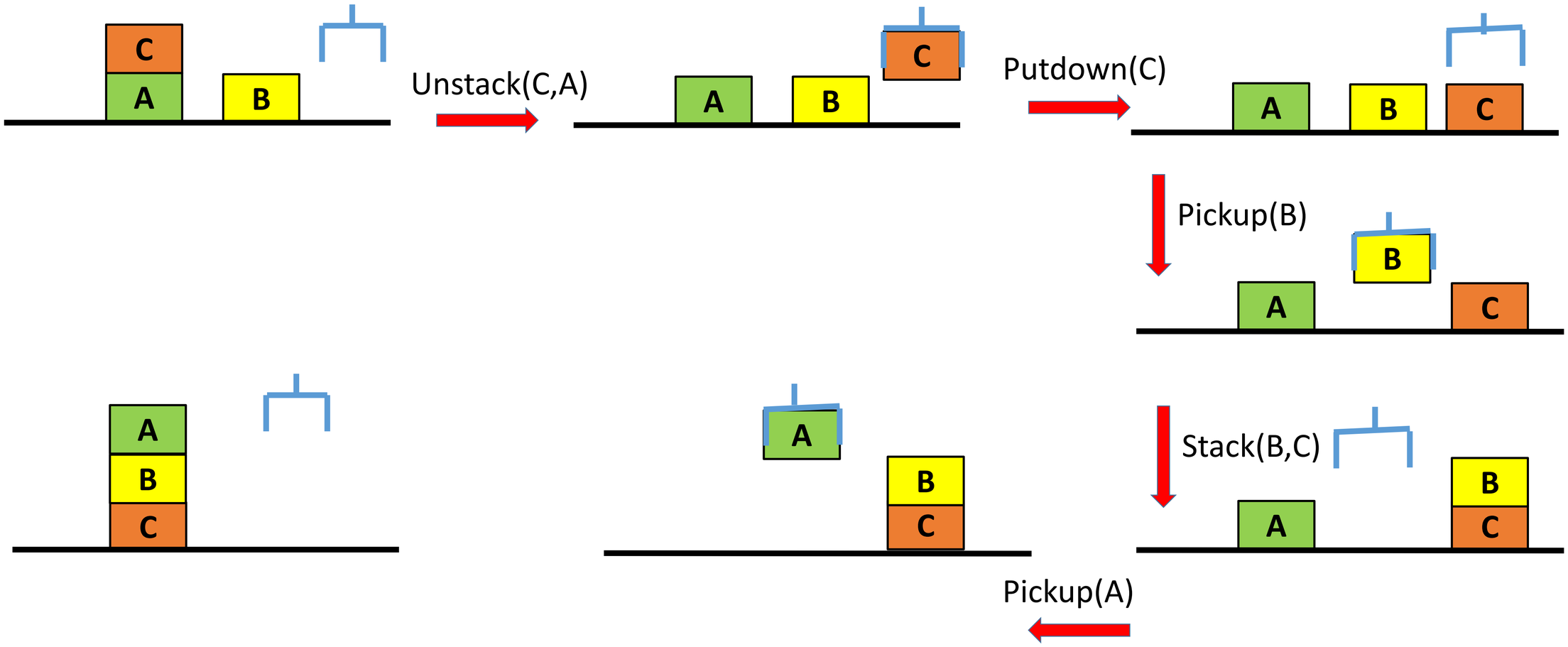
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example Linear Plan



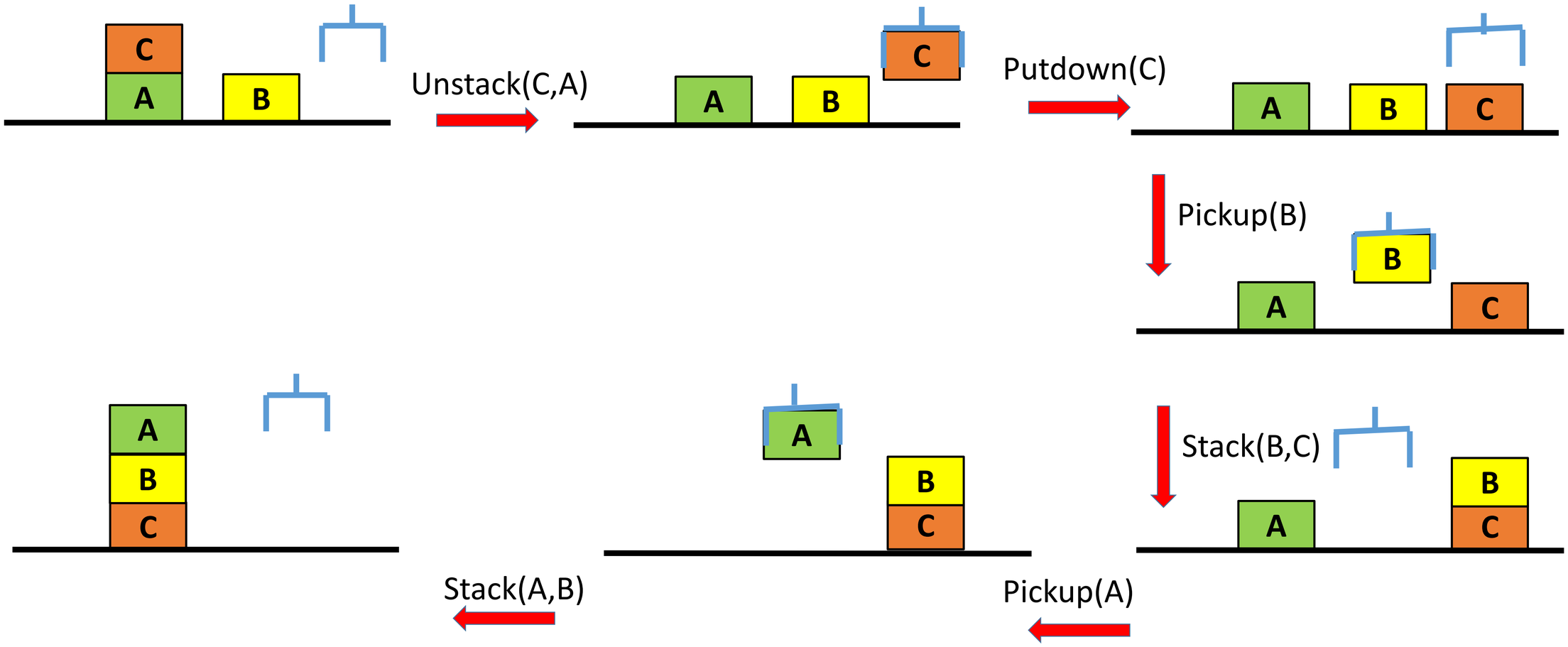
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example Linear Plan



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

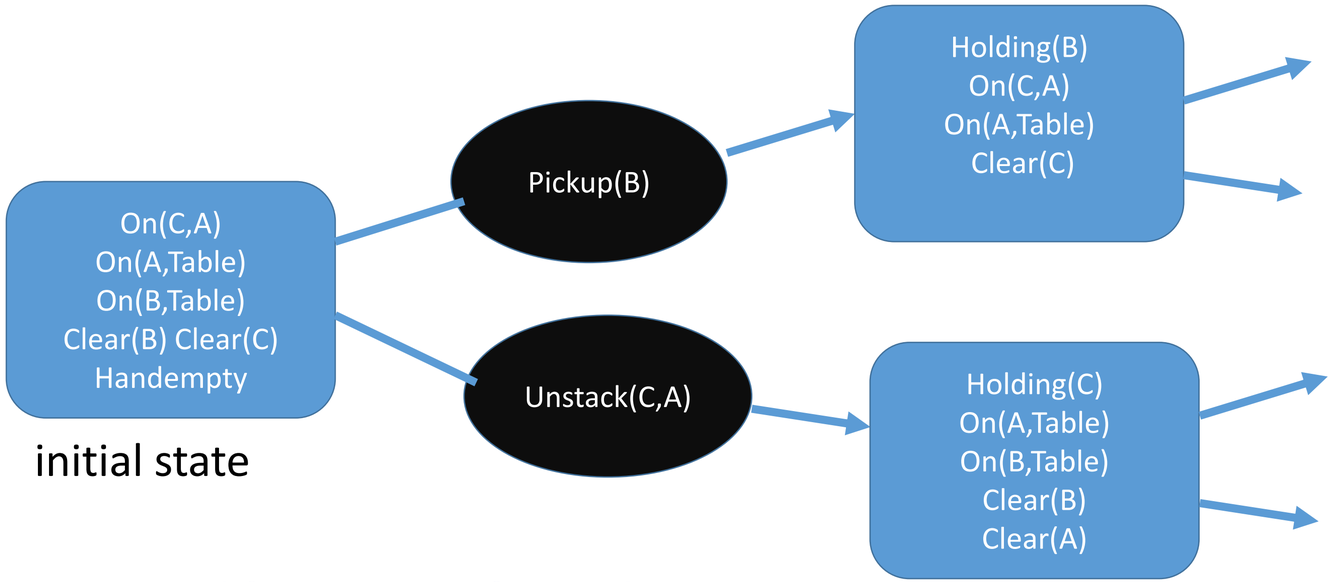
Example Linear Plan



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

How is a plan like the following constructed?

Unstack(C,A)
Putdown(C)
Pickup(B)
Stack(B,C)
Pickup(A)
Stack(A,B)



initial state

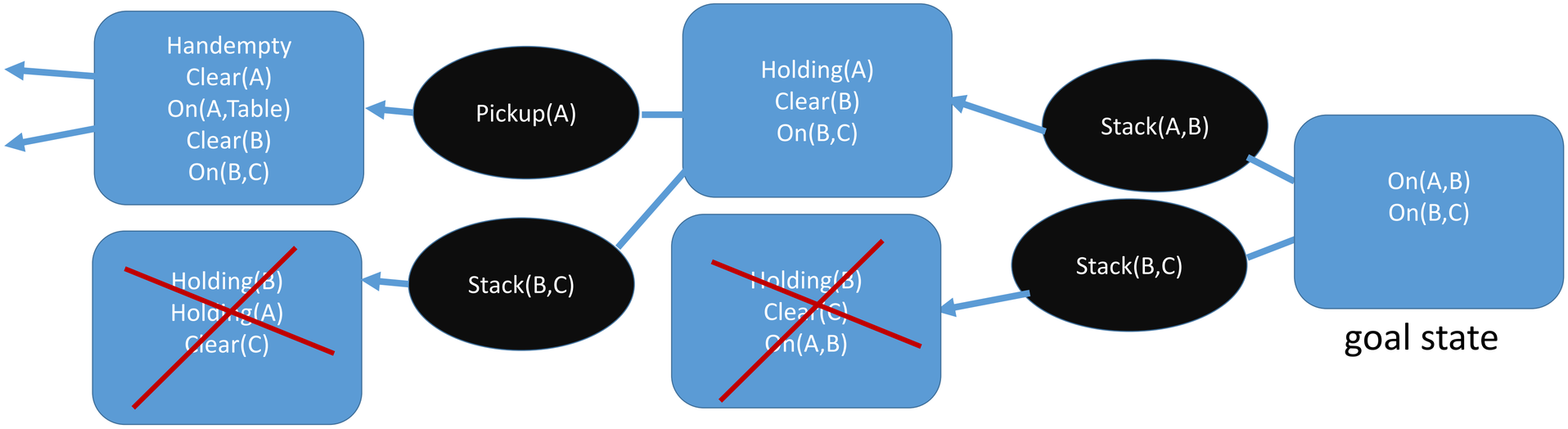
Forward (progression) search

- From a given state s there could be multiple applicable ground actions, some could be from the same action schema
- In the above example applicable action Pickup(B) leads to redundant steps
- The progression continues until a state that matches the goal description is reached

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

How is a plan like the following constructed?

Unstack(C,A)
Putdown(C)
Pickup(B)
Stack(B,C)
Pickup(A)
Stack(A,B)



Backward (regression) search

- Using the inverse of the actions, consider **relevant** actions, i.e., actions with an effect that unifies with one of the goal literals, but with no effect that negates any part of the goal
- The regression terminates successfully when a subgoal description that is matched by the facts in the initial state, is produced
- Considerable reductions in branching factor could accrue for domains having many possible actions

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Regression: Applying an action in a backwards direction

- Given a goal **g** and an action **a**, the regression from **g** over **a** gives a state **g'** as follows:
  - POS(g') = (POS(g) – ADD(a)) U POS(PRECOND(a))
  - NEG(g') = (NEG(g) – DEL(a)) U NEG(PRECOND(a))

- These equations are straightforward for ground literals; if there are variables in **g** and **a**, care is required.
  - Although backward search keeps the branching factor lower than forward search, the presence of variables in states makes it hard to produce good heuristics. This is the main reason why most systems favor forward search.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Decomposability for Non-Interacting Component Goals

- Decompose a compound goal into component goals which are solved separately (even in parallel), and their individual solutions are joined to form a plan for the compound goal.
- This is possible only if the component goals do not interact
- Interaction means that solving one goal undoes an independently derived solution to the other
  - E.g., the example compound goal **On(B,C)** $\wedge$ **On(A,B)**
  - In such cases component goals cannot be isolated and separately solved and the redundancy of multiple solutions to the same goal component in different subgoals cannot be avoided

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Means-Ends Analysis

- The means-ends analysis method that was used in the General Problem Solver (GPS), one of the early AI systems, has similarities with the STRIPS regression method; in fact, the design of STRIPS was motivated by GPS
- The means-ends-analysis method attempted to calculate the difference between a state S and a goal state G
  - The calculation function was domain specific and thus had to be written especially for each domain of application
  - A domain-specific difference table was also needed that listed “relevant differences” and the rules (or operators) were associated with the differences that their actions could reduce
  - A rule was selected as relevant to removing a difference, and then GPS worked recursively on the preconditions of the selected rule

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

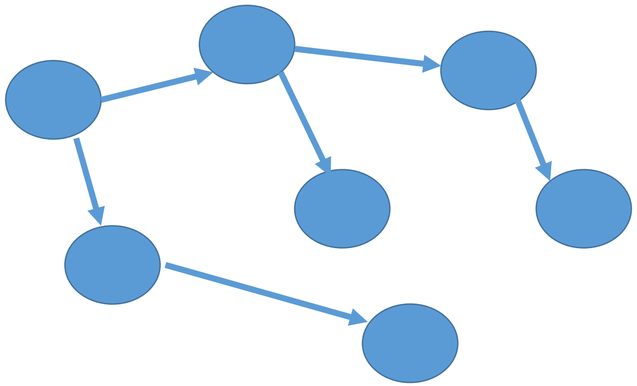
Heuristics for Planning

- Admissible heuristics can be derived by defining a relaxed problem that is easier to solve
  - Recall that admissible heuristics do not overestimate the traversal cost from a node to a goal node
- Two types of relaxations:
  - Adding more edges to the state graph
  - Grouping multiple nodes together

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

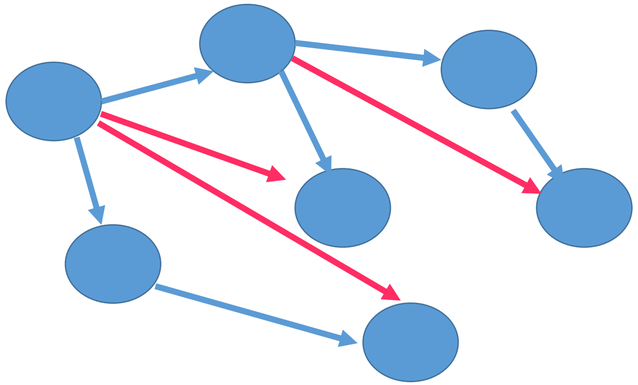
Heuristics for Planning

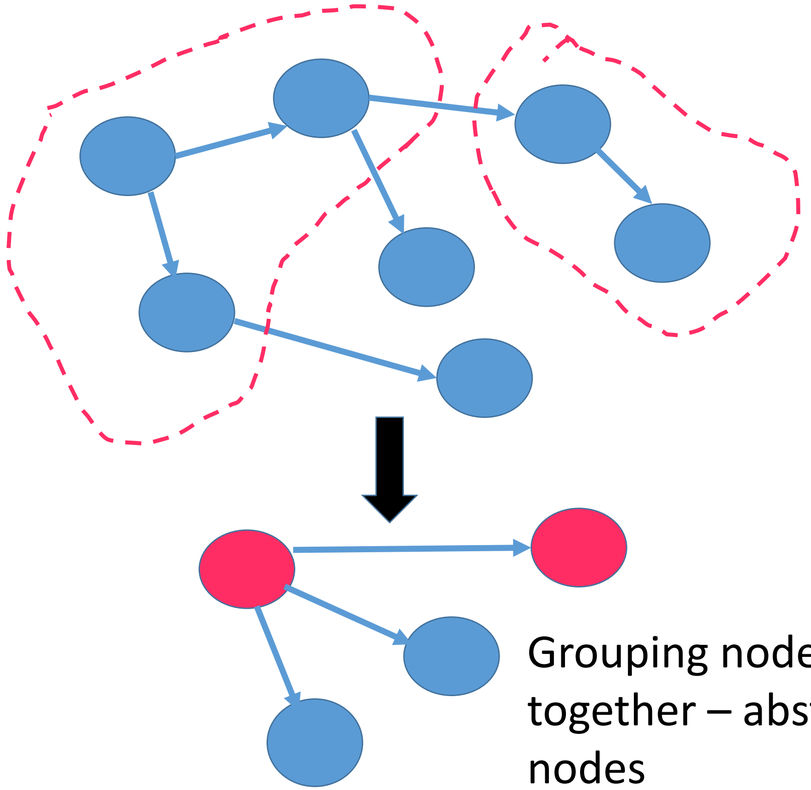
- Two types of relaxations:
  - Adding more edges to the state graph
  - Grouping multiple nodes together



Original state graph

Adding more edges – abstract links





Grouping nodes together – abstract nodes

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example Heuristics

- Adding more edges to the state graph
  - Ignore-preconditions heuristic:
    - Admissible heuristic
    - Drops all preconditions from actions, hence every action becomes applicable in every state, and any single goal fluent can be achieved in one step (if there are any applicable actions, otherwise the problem is impossible)
    - It can be said that the number of steps required to solve the relaxed problem is the number of unsatisfied goals; however, this is not necessarily so as some actions achieve multiple goals, or undo the effects of others
    - Variant: Do not drop the preconditions that are literals in the goal

  - Recall the 8-puzzle:

    Action(Slide(t,s1,s2),

    PRECOND: $On(t,s1) \wedge Tile(t) \wedge Blank(s2) \wedge Adjacent(s1,s2)$

    EFFECT: $On(t,s2) \wedge Blank(s1) \wedge \sim On(t,s1) \wedge \sim Blank(s2)$)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example Heuristics

- If preconditions Blank(s2) and Adjacent(s1,s2) are dropped, this corresponds to the heuristic that counts the number of wrongly positioned tiles
- If only the precondition Blank(s2) is dropped, the Manhattan-distance heuristic is obtained

- Dropping the delete list of all actions
  - Ignore-delete-lists heuristic
    - Monotonic progress towards the goal – no action will ever undo progress made by another action
    - Obtaining an optimal solution continues to be NP-hard
    - Obtaining non optimal solutions is easier – polynomial time
    - It is the basis of many modern planning systems such as LAMA

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Summary

- Constraint Satisfaction Problems (CSPs)
- Search-based solutions to CSPs
- Constraint Propagation and Consistency Checking
- Two-player, perfect information, zero-sum games
- Minimax and alpha-beta procedures for game playing
- Constructing linear plans for accomplishing goals – STRIPS model
- Forward (progression) and backward (regression) search methods in planning domains - PDDL

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September – December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Intelligent Agents and Multiagent Systems

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 4

Intelligent Agents and Multiagent Systems

CONTENTS

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on intelligent agents and multiagent systems, students will be able:

Regarding intelligent agents:

1. To define and explain what an intelligent (autonomous) agent is and to discuss its characteristics.
2. To analyze the notion of rationality with respect to intelligent agents and to overview the relation between information gathering, autonomy and learning.
3. To list the properties of an environment where an agent is situated.
4. To discuss abstract architectures for intelligent agents and distinguish the category of agents with internal state.
5. To discuss concrete architectures for intelligent agents, namely a logic-based architecture, a reactive architecture, the “Belief-Desire-Intention” architecture and layered architectures and to point out strengths and weaknesses of these architectures.
6. To outline a learning agent and to overview its learning element.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on intelligent agents and multiagent systems, students will be able:

Regarding multiagent systems:

1. To explain how the need for multiagent systems arise and to list the characteristics of the environments of multiagent systems.
2. To analyze communication and interaction between agents, how agents may be coordinated to achieve coherence, what the dimensions of the meaning associated with communication are, and what the formal communication elements are.
3. To discuss the role of the agents in a dialogue and to give the types of messages between agents.
4. To list the principal communication protocols and overview the Knowledge Query and Manipulation Language (KQMF), its protocol, and the Knowledge Interchange Format (KIF).
5. To explain the Cooperation protocol and the steps in the Contract Net protocol which is a main cooperation protocol.
6. To discuss the Blackboard system.
7. To outline the Negotiation protocol and to list the principal features of a society of agents.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Intelligent Agents

Largely adapted from M. Wooldridge's chapter in G. Weiss (ed.) Multiagent Systems, The MIT Press, 2013

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The general term “Agent”

- A person or thing that acts or makes an effort
- Some physical force that acts on a case-by-case basis causing some results
- A person (or company) authorized to do a job for someone else
- A person acting on behalf of another to achieve a legitimate relationship between that other and a third party

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Definition of the term “Agent” given by FIPA (Foundation for Intelligent Physical Agents)

The Foundation for Intelligent Physical Agents (FIPA) is **a body for developing and setting computer software standards for heterogeneous and interacting agents and agent-based systems**.

An **agent** is the fundamental unit of action (actor) in an area. It combines one or more service capabilities into a unified, integral execution model, which may include access to external software, users, and communication mechanisms.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

AI-based specification of the term “Agent”

- Many applications require systems that can decide autonomously what they need to do to meet their design goals
- These computer systems are referred to as agents

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Intelligent or Autonomous Agent

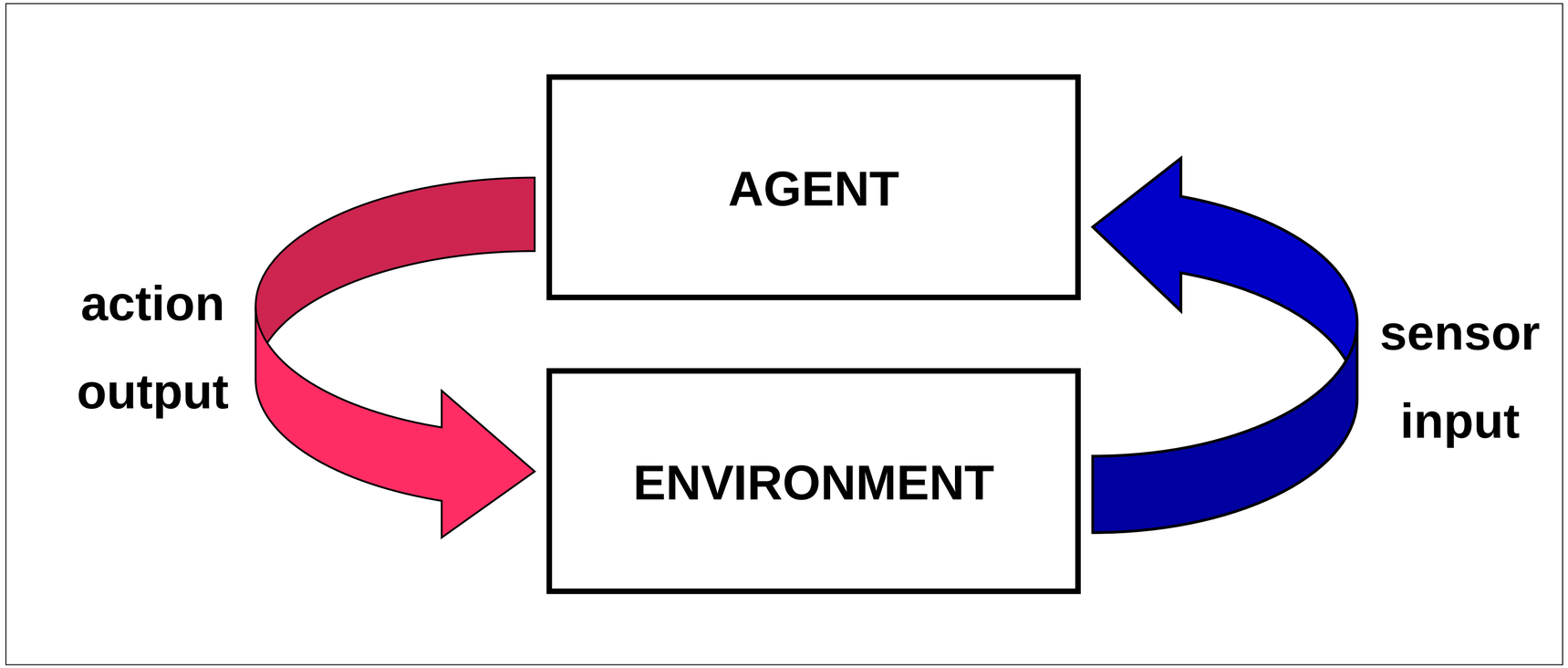
Operates with resistance to rapidly changing, unpredictable or open environments, where there is a high probability of failure.

Example Applications of Intelligent Agents

- Navigation of spacecraft from Earth to space
  - Greater autonomy in making immediate decisions under unforeseen circumstances
- Search for information on the internet

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agent and its Environment



Often the interaction is ongoing and non-terminating

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

An agent ….

….. is a computer system that is situated in some environment and that is capable to act autonomously in this environment in order to achieve its delegated objectives.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Environment and Agent

- At best the agent has some control over the environment, in the sense that it can influence it

- The same action in seemingly identical situations may have different effects or may fail to lead to the desired effect
  - Non-deterministic environments
  - Ability to handle failure
  - Actions have preconditions

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Classification of Environment Properties

- Accessible vs inaccessible
  - Does the agent have complete, accurate and up-to-date information on the state of the environment?
- Deterministic vs non-deterministic
  - Can the state in which an action leads be determined with certainty?
- Episodic vs non-episodic (sequential)
  - Can the agent decide its current action solely based on the current episode (and not its effects on future episodes)? If yes, the agent does not need to think ahead.
  - In a sequential environment short-term actions can have long-term consequences.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Classification of Environment Properties (cont.)

- Static vs dynamic
  - Does it remain stable or not beyond the effects of the agent's actions?
  - A dynamic environment can change while the agent is deliberating.
- Discrete vs continuous
  - Is the set of actions and percepts specific and finite?
  - “Percept” means the content an agent’s sensors are perceiving

- If an agent’s actions need to depend on the entire percept sequence, the agent will have to remember the percepts.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The most complex environment category

- Inaccessible
- Non-deterministic
- Non-episodic
- Dynamic
- Continuous

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Examples of Agents

- Any control system, e.g., a thermostat (inhabits a physical environment)
  - too cold → heating on
  - temperature OK → heating off
- Most software daemons (e.g., the background processes in the Unix operating system) which monitor a software environment and perform actions to modify it
  - e.g., the xbiff program (inhabits a software environment)
- The above are not intelligent agents!

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Intelligent Agents

- Capable of **flexible** autonomous operation to achieve its delegated objectives
- Meanings of flexibility
  - **Reactivity**: Ability to perceive the environment and respond to changes in a timely manner in order to satisfy their delegated objectives
  - **Pro-activeness**: Ability to exhibit goal-driven behavior by taking the initiative to satisfy their delegated objectives
  - **Social ability**: Ability to interact with other agents (and possibly humans) in order to satisfy their design objectives

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Intelligent Agents must be Rational

- A rational agent is one that behaves as well as possible, i.e., one that does the “right” thing
  - How well an agent behaves depends on the nature of the environment
  - Evaluate an agent’s behavior on its consequences
    - If an agent’s actions cause the environment to go into desirable states, then the agent performs well
- Defining a rational agent:
  - A rational agent, for each possible percept, should select an action that is expected to maximize its performance measure, given by the evidence provided by the percept sequence and whatever built-in knowledge the agent has
  - Hence rationality maximizes expected performance where a rational choice depends on the percept sequence to date.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Information gathering, Autonomy and Learning

- Information gathering is an important part of rationality
  - Doing actions in order to modify future percepts
  - Learn as much as possible from what it perceives; through experiential learning, the agent's prior knowledge of the environment may be modified and augmented
- A rational agent should be autonomous
  - A rational agent, through its ability to learn what it can, compensates for partial or incorrect prior knowledge
  - An agent relying entirely on prior knowledge rather than its own percepts and learning process, lacks autonomy

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agents and Objects

- An object executes the requested method. Instead, an agent decides for itself whether to execute the request it receives
  - Objects do it for free; agents do it for money
  - It is understood that agents can be implemented using object-oriented techniques
- The standard object-oriented model does not refer to autonomous, flexible action; objects are simple passive service providers, incapable of reactive, proactive or social behavior
- A system of multiple agents is "multi-threaded" since each agent has its own thread of control and is continually executing

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agents and Expert Systems

- Expert Systems were the most important AI technology of the 1980s
  - An expert system can solve problems or give advice in some knowledge-rich domain
- An expert system usually does not have direct interaction with any environment, hence expert systems are inherently disembodied
- It does not receive its data through sensors but through a user
- It does not act on any environment, but provides feedback or advice to a third party
- It is not generally required for an expert system to co-operate with other agents
- Exceptions are expert systems that perform real-time control tasks

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Abstract Architectures for Intelligent Agents

- **Environment States**
  - $S = \{s_1, s_2, \ldots\}$
  - At all times, the environment is in one of these states
- **Agent Actions**
  - $A = \{\alpha_1, \alpha_2, \ldots\}$
- **Abstract view of a standard agent**
  - $action: S^* \rightarrow A$
  - as a function that maps sequences of environmental states to actions

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Environment Behavior

- ***env: S × A → Power(S)***
- If in each case env(s, a) gives a unique state, then the behaviour of the environment is deterministic. Otherwise, it is non-deterministic where the possible successive states from the execution of a given action in a given state are identified.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agent – Environment Interaction History

- The sequence

$$h : s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} s_3 \xrightarrow{\alpha_3} \ldots \xrightarrow{\alpha_{u-1}} s_u \xrightarrow{\alpha_u} \ldots$$

- is a possible history of the agent's interaction with the environment if and only if the following conditions are true:
  - $\forall u \in N, \alpha_u = action((s_0,s_1,\ldots,s_u))$
  - $\forall u \in N$ s.t. $u > 0, s_u \in env(s_{u-1},\alpha_{u-1})$

.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Characteristic Behavior of an Agent

- It is the sum of all possible interaction histories
  - hist(agent, environment)
- If a property φ is valid in all possible histories, then this property is an invariant property of the agent in the environment
- If and only if hist(ag1, env) = hist(ag2, env), agents ag1 and ag2 have equivalent behavior in environment env. If this is the case for any environment, then the two agents have equivalent behavior
- Usually, an agent's interaction with its environment never ends and therefore its history is infinite

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Purely Reactive Agents

- They decide to act without reference to their history - every decision is based solely on the present without any reference to the past
  - action: S → A
- Example: the thermostat
- For every "purely reactive agent" there is an equivalent "standard agent", but not the opposite

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Perception

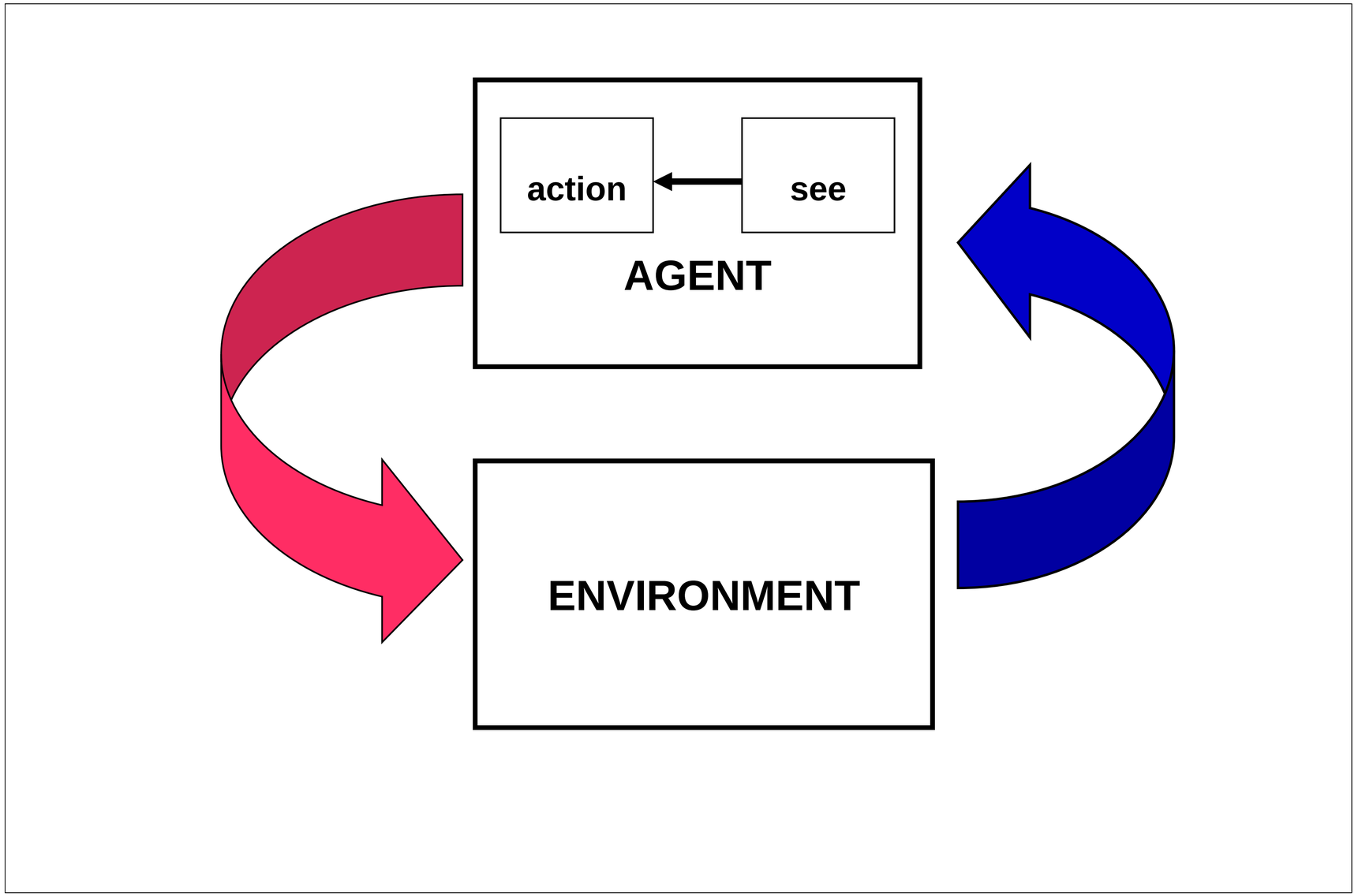
- An agent must have the ability to observe/perceive its environment:
  - see: S → P, where P is the set of observations/percepts
- It is implemented in hardware in the case of agents that are integrated in a physical world (e.g., video camera, infra-red sensors for mobile robots)
- In the case of "software agents" it consists of commands to extract information about the "software environment"

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Respecifying function “action”

- action: P * → A
  - The action function represents the agent's decision-making process
  - In the revised version it maps sequences of "percepts" to actions
- Let s1 ∈ S and s2 ∈ S, where s1 ≠ s2. If see (s1) = see (s2) from the agent’s perspective the two environment states s1 and s2 are indistinguishable from each other

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Distinguishing Environment States

- Let s and s'∈ S. The relation ≡ means see(s) = see(s'). This relation therefore divides S into mutually indistinguishable sets of states from the agent's perspective.

- If |≡| = |S| then the agent can distinguish every state of the environment and therefore has a perfect perception of the environment (omniscient agent).

- If |≡| = 1 then the agent has no perception at all since it is not able to distinguish any state - from the agent's perspective all states are the same.

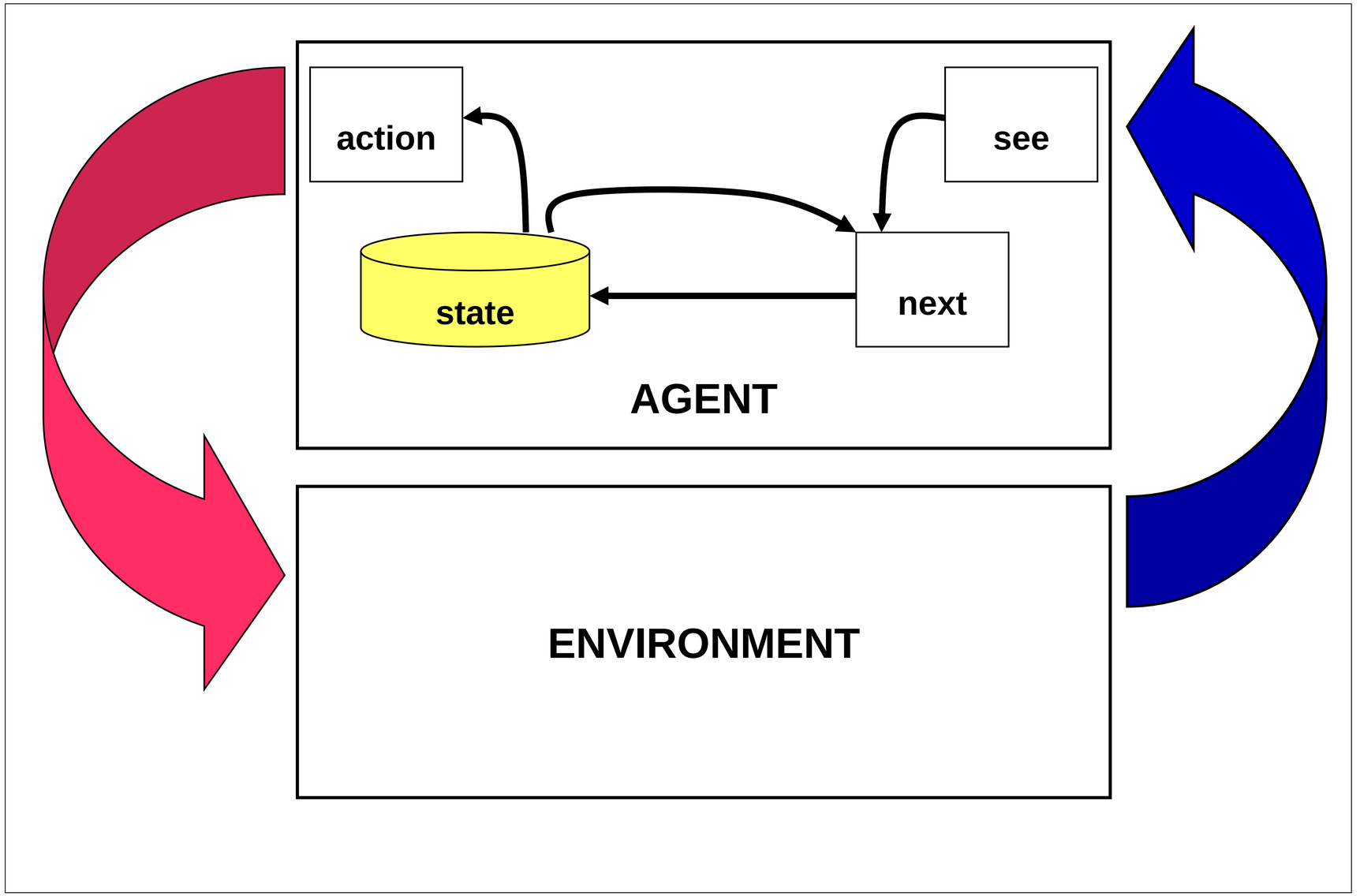
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agents with internal State

- They have an internal data structure that records information about the state of the environment and history
  - I is the set of the agent's internal states
  - see: $S \rightarrow P$
  - action: $I \rightarrow A$
  - next: $I \times P \rightarrow I$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Agents with internal State

- The agent has some initial internal state, i_0
- It then observes the state of environment s and generates its perceptions - **see(s)**
- What follows is the update of the agent's internal state - **next(i_0, see (s))**
- Then the action is decided - **action(next(i_0, see (s)))**
- The action is executed and the "**see – next – action**" cycle is repeated

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agents with State versus Standard Agents

- They are just as powerful
- Their power of expressiveness is identical
- Any agent with internal state can be turned into a standard agent with equivalent behavior

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Concrete Architectures for Intelligent Agents

- **Logic-based architectures**
  - Decision making is done through logical deduction
- **Reactive Architectures**
  - Decision making is done through the direct mapping of state to action
- **“Belief-Desire-Intention” Architectures**
  - Decision making is based on the processing of data structures that represent the agent's beliefs, wishes and intentions
- **Layered Architectures**
  - Decision making is based on different levels of software that perform reasoning in relation to the environment from different levels of abstraction

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Logic-Based Architectures

Logic-based agents may be referred to as **deliberate agents**: Such agents maintain an internal database of predicate logic sentences, representing in symbolic form the information they have about their environment, just like the beliefs in humans.

- L, the set of predicate logic sentences
- $D = P(L)$, the set of the internal states of the agent ($\Delta, \Delta_1, \ldots, \in D$) – its database
- ρ, the deductive rules of inference
- $\Delta \vdash_\rho \varphi$, formula φ can be proved from the database of sentences Δ using only the rules of inference ρ
- see : $S \rightarrow P$
- next : $D \times P \rightarrow D$
- action : $D \rightarrow A$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Action Selection in Deliberate Agents

function *action* (Δ, *ρ*) returns an action

begin

  for each α ∈ *A* do

    if Δ $\vdash_{\rho}$ *Do(α)* then return α

  for each α ∈ *A* do

    if Δ $\nvdash_{\rho}$ *~Do(α)* then return α

  return *null*

end function *action*

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Logic-Based Architectures

- **Strengths:**
  - The "elegance" and clean semantics of logic
- **Serious weaknesses:**
  - The computational complexity of proving theorems calls into question the effectiveness of this approach in environments where time is limited
  - The assumption of "calculative rationality" that the world will not change substantially while the agent ponders what to do, and that an action that is rational when reasoning begins will continue to be rational when reasoning is complete does not always apply

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Reactive Architectures

- They are based on the rejection of any symbolic representation and its management mechanisms
- They are rooted on the belief that intelligent behavior can result from the combination of many basic actions and the way an agent interacts with its environment – hence intelligent, rational behavior is seen as innately linked to the environment an agent occupies

- Reactive: reacting to an environment without reasoning about it
- Alternative terms: behavioral (developing and combining individual behaviors) or situated (in some environment rather than being disembodied from it)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Basic Characteristics of Reactive Agents

- Simplicity and interaction in a basic way, but from which complex behavior emerges
- It consists of various components that operate autonomously and are responsible for specific tasks
- The communication between these components is the minimum possible and quite low level
- Operates with the processing of elementary representations, such as data coming from various sensors

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Rodney Brooks' Subsumption Architecture

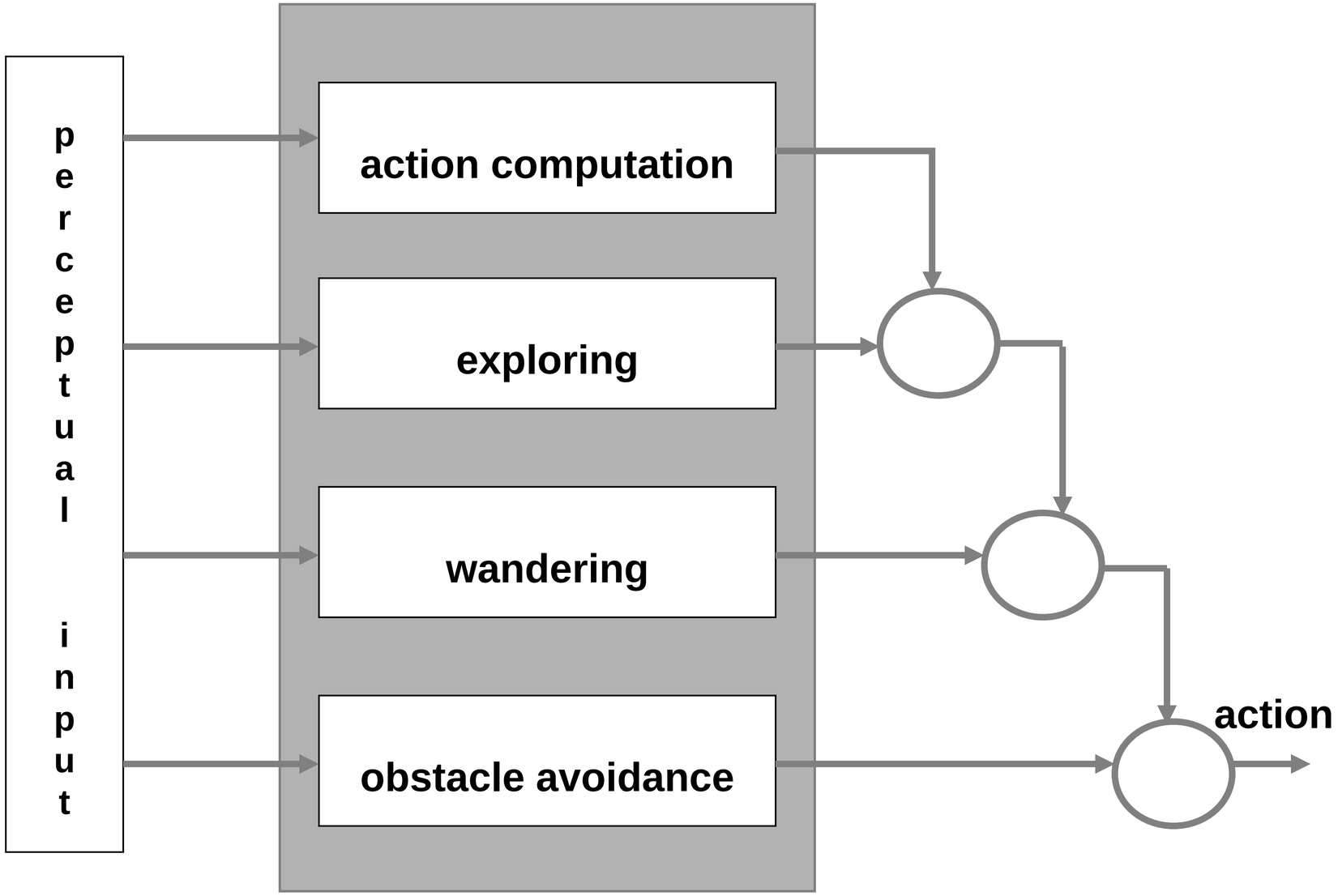
- An agent's decision making is realized through a set of task accomplishing behaviors (implemented as finite-state machines)
  - Each behavior is an individual action selection process, which continually takes perceptual input and maps it to an action to perform
- Many behaviors can "fire" simultaneously
  - Behaviors are arranged in layers forming a subsumption hierarchy, where lower layers in the hierarchy can inhibit higher layers – the lower a layer is the higher is its priority

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Subsumption Architecture

(best known reactive agent architecture)



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Behavior Rules

- Set of behaviors in the form of rules "situation → action"
  - $Beh = \{(c,\alpha) \mid c \subseteq P \text{ and } \alpha \in A\}$
  - A behavior (c,α) can be activated when the environment is in state $s \in S$, if and only if $see(s) \in c$

- "inhibition" relation (¥)
  - $¥ \subseteq R \times R$, where $R \subseteq Beh$
  - $(b_1,b_2) \in ¥$, means "b_1 inhibits b_2", i.e., b_1 is at a lower layer than b_2 and hence it has priority

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Action Selection in the Subsumption Architecture

function *action (p : Percept) returns an action*

var *fired* : *Power(R)*

begin

  fired := {(*c*,*α*) | (*c*,*α*) ∈ *R* and *p* ∈ *c* }

  for each *(c,α)* ∈ *fired* do

    if ~(∃ *(c',α')* ∈ *fired* s.t. *(c',α')* ¥ *(c,α)*

    then return *α*

  return *null*

end function *action*

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Advantages of Reactive Architectures

- Simplicity, economy, computational tractability, robustness against failure, elegance

Fundamental weaknesses

- As agents do not employ models of their environment, they must have sufficient local information to determine acceptable actions
- Since non-local information cannot be considered, the decision making has an inherently short-term view
- There is no principled methodology for building such agents – experimentation, trial and error
  - The essence of the architecture suggests that the relationship between individual behaviors, environment and overall behavior is not understandable, given that the claim is that overall behavior “emerges” from component behaviors
- Hard to build agents with many layers as the behavior interaction dynamic become too complex

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

“Belief-Desire-Intention” Architectures

- They are based on the principles of practical reasoning, i.e., the decision-making, at all times, focuses on the action to be taken to advance the pursued goals
- Two important processes are involved
  - Deliberation: decide the goals to be achieved
  - “Means-ends“ reasoning: decide how to achieve these goals

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Intentions play a crucial role

- They drive the "means-ends“ reasoning
- Constrain future "deliberation"
- They persist
- They influence the beliefs on which future practical reasoning is based

Reconsidering Intentions

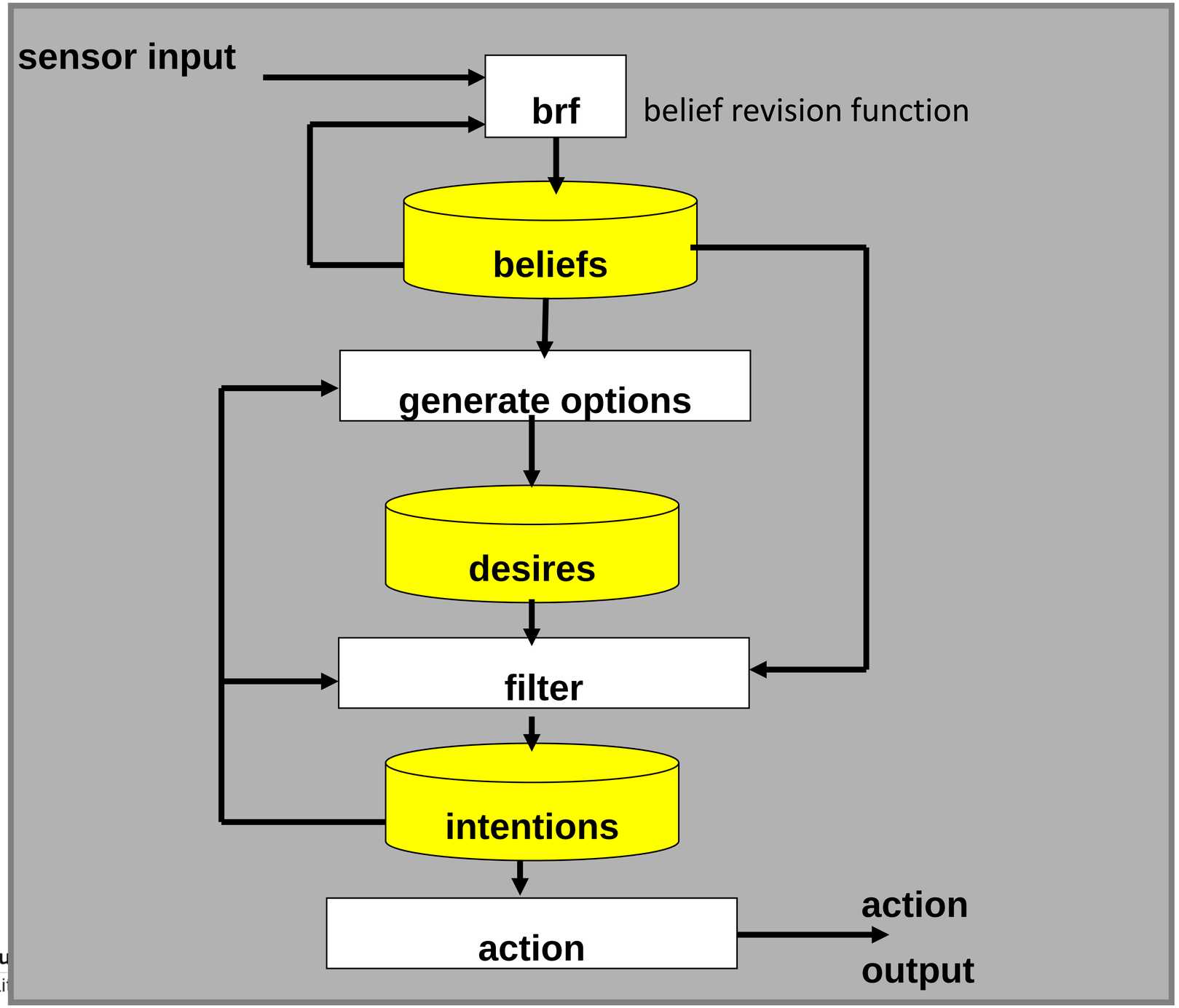
- If the agent does not reconsider its intentions regularly, there is a risk that it will continue to pursue unrealistic intentions or intentions for which there is no longer any reason to achieve
- If the agent constantly reconsiders its intentions, there is a risk that there will not be enough time to pursue them and as a result they can never be achieved

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Reasoning Strategies

- They must consider the type of environment
- In a static non-changing environment, proactive goal-driven reasoning is enough
- In a dynamic environment, the ability to react to change and modify intentions is required
- The dilemma is how to balance proactive (goal-driven) and reactive (event-driven) behavior
- Bold agents never stop to reconsider
- Cautious agents constantly stop to reconsider
- Bold agents outperform cautious ones in environments that do not change quickly, while cautious agents outperform bold ones in environments that change frequently

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



Co-financed by the Eu
Connecting Europe Facili

Agent decision function

function *action (p : Percept) returns an action*

begin

B := brf(B,p)

D := options(B,I)

I := filter(B,D,I)

return *execute(I)*

end function *action*

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

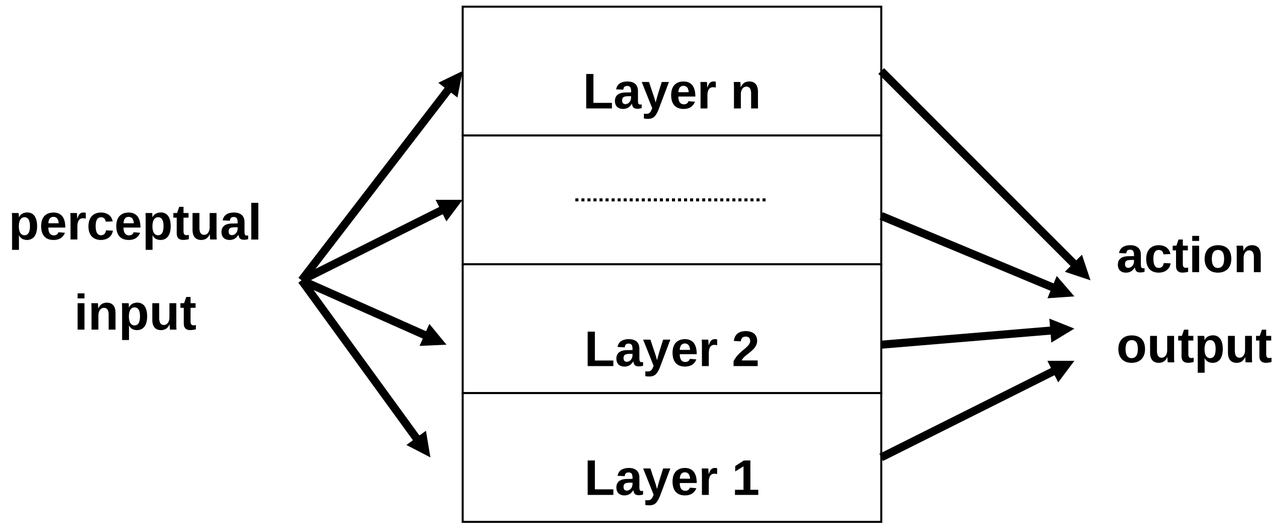
Layered Architectures

- Requirement: an agent is capable of reactive and proactive behavior
- There are two types of control flow within layered architectures, horizontal layering and vertical layering
- Horizontal Layering
  - Each layer receives direct "sensory input" and produces "action output", i.e., each layer acts as an agent producing suggestions for the next action – hence in effect layers are competing with one another
- Vertical Layering
  - "Sensory input" and "action output" are each dealt with by at most one layer each

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

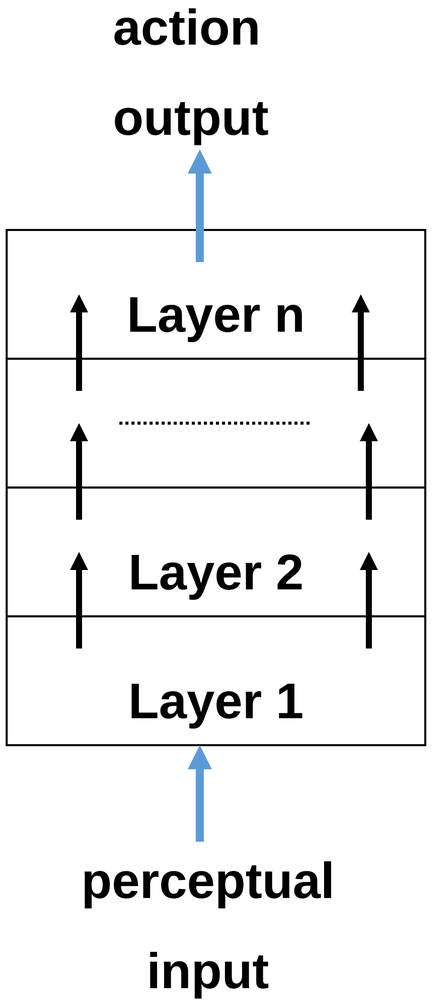
Advantages: conceptual simplicity
Weaknesses: the inherent competition between the layers may lead to non-coherent overall behavior of the agent
"Solution": to obtain consistency include a mediator function to decide which layer has "control" of the agent at any given time; however, this creates design difficulties and introduces a bottleneck into the agent's decision making



Horizontal layering

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Control flows sequentially through each layer, until the final layer generates action output.

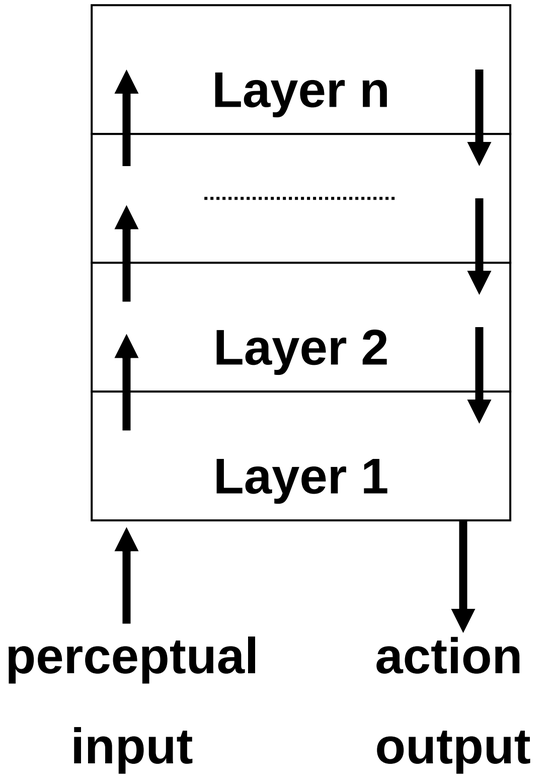


Vertical layering

(one pass control)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Information flows up the architecture (the first pass) and control then flows back down – clear analogy with the way organizations work, with information flowing up to the highest levels of the organization, and commands then flowing down



Vertical layering

(two pass control)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Horizontally versus Vertically Layered Architectures

- Vertically layered architectures aim to alleviate the interaction bottlenecks of the horizontally layered architectures
- However, the simplicity of the vertically layered architectures comes at the cost of some flexibility
  - A vertically layered architecture does not have fault-tolerance since in order to make a decision, control must pass between each layer and consequently a faulty layer can seriously hinder the performance of the agent

Co-financed by the European Union
Connecting Europe Facility

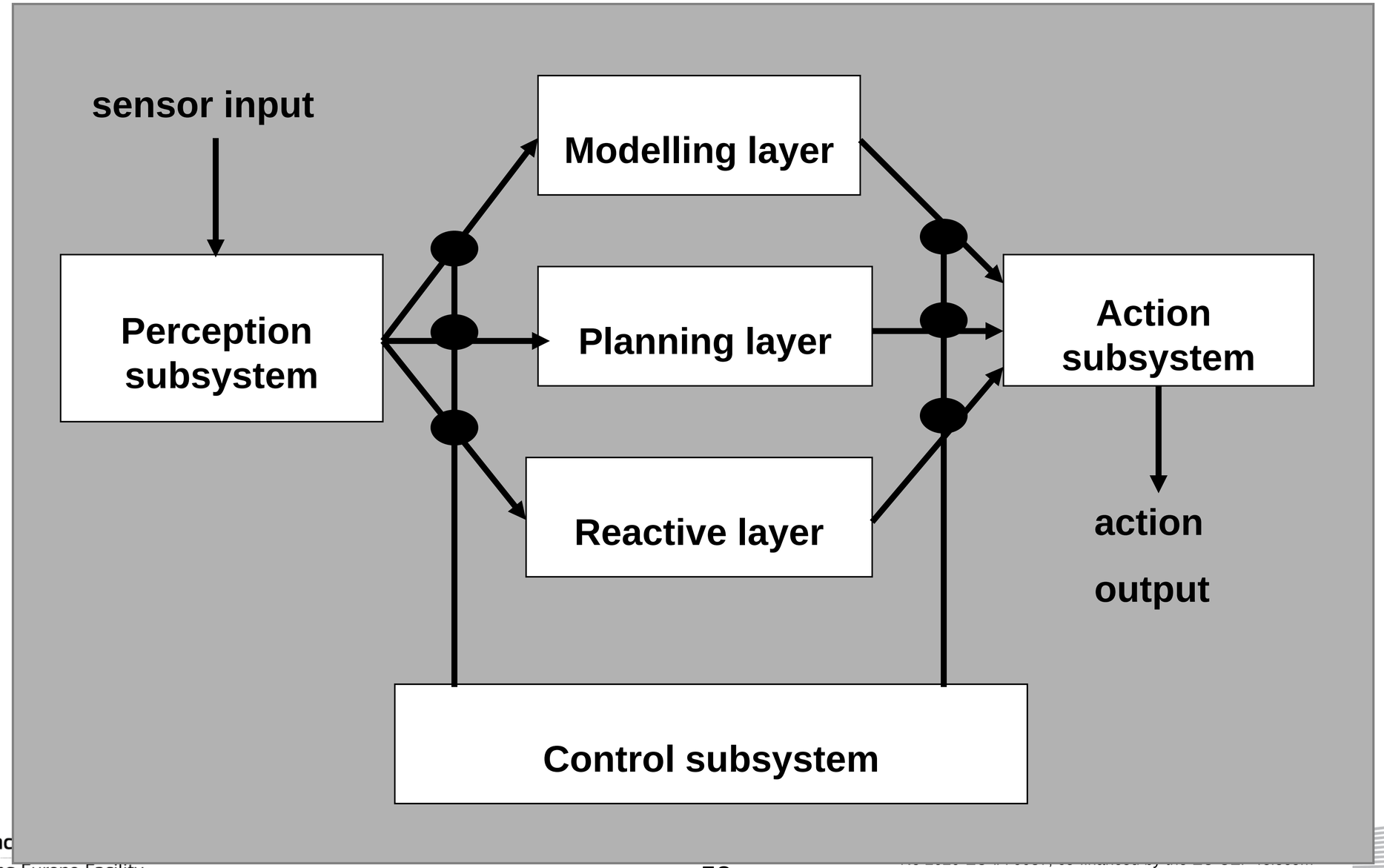
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Two Examples of Layered Architectures

- Innes Ferguson's TOURINGMACHINES – horizontally layered architecture
  - Three activity producing layers
    - Reactive layer: responds immediately to environment changes (implemented as a set of situation-action rules
    - Planning layer: deals with the agent's proactive behavior; it does not create plans from scratch but uses a library of plan "skeletons" called schemas (hierarchically structured plans)
    - Modelling layer: has a representation of itself and other agents and uses it to predict conflicts between agents and generates new goals to resolve these conflicts
  - The control layer decides which of the activity layers should have control over the agent; it is implemented as a set of control rules that either suppress sensor information or censor action outputs

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

TOURINGMACHINES Horizontal layering



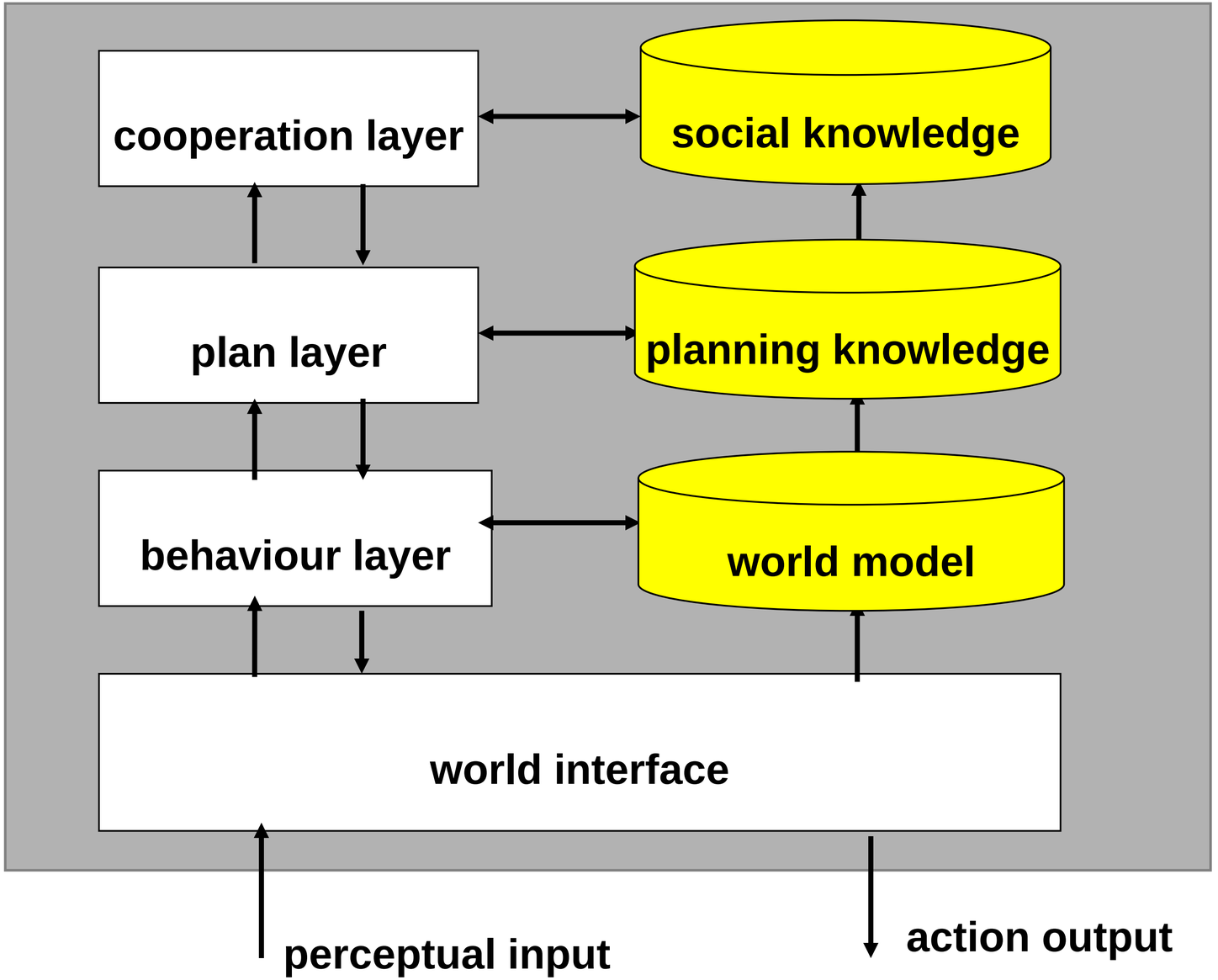
under GA nr. INEA/CEF/ICT/A2020/2267423

Two Examples of Layered Architectures

- Jörg Müller's INTERRAP – a two-pass vertically layered architecture
  - Each of the three layers has a knowledge base representing the agent and its environment at different levels of abstraction:
    - Behavior layer: deals with reactive behavior; its knowledge base represents "raw" data about the environment
    - Plan layer: deals with proactive behavior, i.e., it does everyday planning to meet the agent's goals; its knowledge base represents the plans and actions of the agent itself
    - Cooperation layer: deals with social interactions (cooperative planning); its knowledge base represents the plans and actions of other agents in the environment
  - The explicit use of knowledge bases distinguishes TOURINGMACHINES from INTERRAP. In addition, the INTERRAP layers interact with each other to achieve the same end
    - Bottom-up activation: a lower layer passes control to a higher layer
    - To-down execution: a higher layer uses facilities provided by a lower layer to achieve one of its goals
    - Each layer has two general functions: a situation recognition and goal activation function (analogous to the options function in BDI architecture) and a planning and scheduling function

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

InteRRaP (two-pass) Vertical layering



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Learning Agents

- A. Turing was the first to introduce the term “learning machine”
- The ability to learn improves the overall performance of the agent and bestows it with autonomy
  - Initial knowledge: an agent does not need to have complete autonomy from the start; initially it would have to act randomly
  - Learning ability: after sufficient experience of its environment, the behavior of a rational agent can become effectively independent of its prior knowledge
- The incorporation of learning allows a single rational agent to succeed in a vast variety of environments

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Learning Element

- Uses feedback from a critic function on how the agent is doing and determines how the performance should be modified to do better in the future
  - Part of the incoming percept is distinguished as a reward (or penalty) that provides direct feedback on the quality of the agent's behavior
  - Reinforcement learning: how certain behaviors are encouraged, and others discouraged. Behaviors are reinforced through rewards which are gained through experiences with the environment
- Hence learning in intelligent agents is the process of modification of each component of the agent, thereby improving the overall performance of the agent

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Multiagent Systems

- ❑ Characteristics of multiagent environments
- ❑ Protocols for communication, coordination, cooperation and negotiation between agents
- ❑ Agent societies

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on intelligent agents and multiagent systems, students will be able:

Regarding multiagent systems:

1. To explain how the need for multiagent systems arise and to list the characteristics of the environments of multiagent systems.
2. To analyze communication and interaction between agents, how agents may be coordinated to achieve coherence, what the dimensions of the meaning associated with communication are, and what the formal communication elements are.
3. To discuss the role of the agents in a dialogue and to give the types of messages between agents.
4. To list the principal communication protocols and overview the Knowledge Query and Manipulation Language (KQMF), its protocol, and the Knowledge Interchange Format (KIF).
5. To explain the Cooperation protocol and the steps in the Contract Net protocol which is a main cooperation protocol.
6. To discuss the Blackboard system.
7. To outline the Negotiation protocol and to list the principal features of a society of agents.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Why are multiagent systems needed?

There are many and important reasons for interconnecting multiple computational agents and expert systems, including:

- ❑Collaboration in problem solving
- ❑The transfer of expertise
- ❑The parallel processing of problems that are shared
- ❑Modular development and implementation
- ❑Fault tolerance through redundancy
- ❑The depiction of multiple viewpoints and the knowledge of multiple experts
- ❑Reusability

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Communication and Interaction

- The environments in which agents exist must provide the computing infrastructure required for agents to operate efficiently and interact productively with each other
- Specifically, protocols are required for
  - Communication between agents
  - Interaction between agents

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Communication and Interaction Protocols

Communication Protocols

They allow agents to exchange and understand messages

Interaction Protocols

They allow agents to have conversations, i.e., structured message exchanges

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Communication Protocol: Examples of Messages Types

- Propose a course of action
- Accept a course of action
- Reject a course of action
- Retract a course of action
- Disagree with a proposed course of action
- Counterpropose a course of action

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Interaction Protocol for Negotiation: Example

- Agent-1 proposes some course of action to Agent-2

  Agent-2 evaluates the proposal and
- Sends acceptance to Agent-1 or
- Sends counterproposal to Agent-1 or
- Sends disagreement to Agent-1 or
- Sends rejection to Agent-1

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Characteristics of multiagent environments

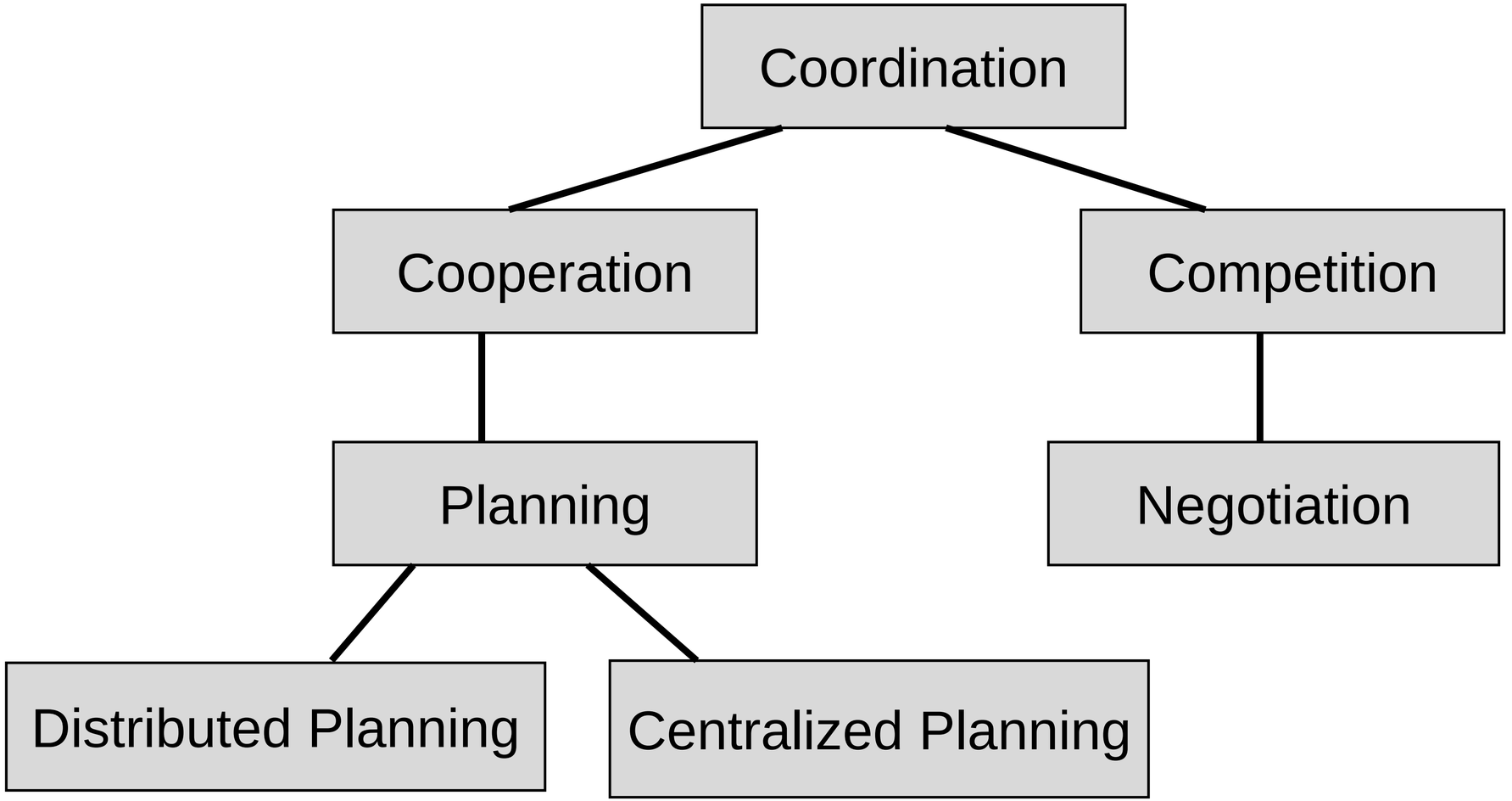
- They provide an infrastructure specifying communication and interaction protocols
- They are usually open environments without a central design
- They contain agents that are autonomous and distributed and that can be self-interested or cooperative

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agent Coordination

- Communication allows agents to coordinate their actions and behaviors, thus achieving systems that are more "connected" and "coherent"
- Coordination is required when the environment is shared:
  - Competition for resources is reduced and unnecessary activity is avoided
  - Dead ends are avoided
  - Applicable security levels are maintained

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



A taxonomy of different forms of coordination

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Coherence

- How well a system behaves as a unit
- The problem for a multiagent system is how to achieve coherence at the global level without explicitly having global control. This entails the ability on the part of agents to:
  - decide on the goals they share with other agents
  - decide on common goals
  - avoid unnecessary conflicts
  - pool knowledge and evidence
- It is facilitated if there is some organizational structure between the agents
- Social commitments can be the means for achieving the desired coherence

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Formal communication elements

- **Syntax**
  - How the symbols of communication are structured
- **Semantics**
  - What do the symbols denote?
- **Pragmatics**
  - How are the symbols interpreted?

- Meaning is a combination of semantics and pragmatics
- Agents communicate to understand and to be understood

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Dimensions of meaning associated with communication (1)

- **Descriptive vs. Prescriptive**: messages describe phenomena or prescribe behavior
- **Personal vs. Conventional Meaning**: opt for conventional meaning since multiagent systems are typically open in which new agents might be introduced at any time
- **Subjective vs. Objective Meaning**: the effect of a message on the environment may be understood differently internally, i.e., subjectively by the sender or receiver of the message, and externally to them, i.e., objectively
- **Speaker's vs. Hearer's vs. Society's Perspective**: independent of the conventional or objective meaning of a message, the message can be expressed according to the viewpoint of the speaker, or hearer or other observers
- **Semantics vs. Pragmatics**: pragmatics are concerned with how communicators use the communication which include considerations about mental states of communicators and the environment, that are external to the syntax and semantics of the communication

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Dimensions of meaning associated with communication (2)

- **Contextuality**: messages are not understood in isolation, but in the context of the mental states of agents, the present state of the environment and the environment history
- **Coverage**: the communication language should enable an agent to convey the meanings it intends – power of expression
- **Identity**: the identities and roles of agents involved in a communication, or their specifications, impinge on the meaning of the communication; a message can be sent to a particular agent or to any agent satisfying a specified criterion
- **Cardinality**: a message sent privately to one agent would be understood differently when the same message is broadcasted publicly

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The role of an agent in a dialogue

- Active
  - Master
- Passive
  - Slave
- Both roles
  - Peer

- An agent can send or receive messages over a communication network. There are two main categories of messages:
  - Assertions
  - Queries

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agent capabilities

	Basic Agent	Passive Agent	Active Agent	Peer Agent
Receives assertions	●	●	●	●
Receives queries		●		●
Sends assertions		●	●	●
Sends queries			●	●

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Interagent message types

Communicative Action	Illocutionary Force	Expected Result
Assertion	Inform	Acceptance
Query	Question	Reply
Reply	Inform	Acceptance
Request	Request	
Explanation	Inform	Agreement
Command	Request	
Permission	Inform	Acceptance
Refusal	Inform	Acceptance
Offer/Bid	Inform	Acceptance
Acceptance		
Agreement		
Proposal	Inform	Offer/Bid
Confirmation		
Retraction		
Denial		

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Communication Protocol Levels

- **Lower level**
  - Specifies the interface method
- **Medium level**
  - Specifies the syntax of the transferred information
- **Higher level**
  - Determines the semantics of the information, both in relation to the substance of the message and in relation to the type of the message

Types of Communication Protocols

- **Binary protocols**: One sender and one recipient
- **N-ary, broadcast or multicast protocols**: One sender, multiple recipients

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Data structure for communication protocols

1. Sender
2. Recipient or recipients
3. Protocol language
4. Encoding and decoding functions
5. Actions to be taken by the recipient(s).

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Query and Manipulation Language (KQML)

- Separation between
  - semantics of the communication protocol
    - it must be domain independent
  - and the semantics of the enclosed message
    - it may depend on the domain

- The communication protocol must be universally understood by all agents
  - Must be concise and have only a limited number of primitive communication acts

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
(KQML-performative
        :sender      <word>
        :receiver    <word>
        :language    <word>
        :ontology    <word>
        :content     <expression>
)
```

KQML: Communication Protocol among both agents and application programs

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
(tell
        :sender       Agent1
        :receiver     Agent2
        :language     KIF
        :ontology     Blocks-World
        :content      (AND (Block A) (Block B) (On A B))
)
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

KQML Protocol

- Agents that communicate via the KQML protocol present themselves to each other as clients or servers
- Communications are synchronous or asynchronous
  - Synchronous communication
    - The sender is waiting for the reply
  - Asynchronous communication
    - The sender continues its process, which may be interrupted at a later stage when the replies arrive

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

```
(forward
      :from         Agent1
      :to           Agent2
      :sender       Agent1
      :receiver     Agent3
      :language     KQML
      :ontology     kqml-ontology
      :content      (tell
                        :sender         Agent1
                        :receiver       Agent2
                        :language       KIF
                        :ontology       Blocks-World
                        :content        (AND (Block A) (Block B) (On A B))
                    )
)
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
(advertise
      :sender       Agent2
      :receiver     Agent1
      :language     KQML
      :ontology     kqml-ontology
      :content      (ask-all
                          :sender          Agent1
                          :receiver        Agent2
                          :in-reply-to     id1
                          :language        Prolog
                          :ontology        Blocks-World
                          :content          “on(X,Y)”
                    )
)
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
(ask-all
                    :sender          Agent1
                    :receiver        Agent2
                    :in-reply-to     id1
                    :reply-with      id2
                    :language        Prolog
                    :ontology        Blocks-World
                    :content          “on(X,Y)”)
```

```
(tell
                    :sender          Agent2
                    :receiver        Agent1
                    :in-reply-to     id2
                    :language        Prolog
                    :ontology        Blocks-World
                    :content          “[on(a,b),on(c,d)]”)
```

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Basic Categories of “KQML Performatives”

- Basic query performatives (**evaluate, ask-one, ask-all**, ...)
- Multiresponse query performatives (**stream-in, stream-all**, …)
- Response performatives (**reply, sorry**, …)
- Generic informational performatives (**tell, achieve, cancel, untell, unachieve**, …)
- Generator performatives (**standby, ready, next, rest**, …)
- Capability-definition performatives (**advertise, subscribe, monitor**, …)
- Networking performatives (**register, unregister, forward, broadcast**, …)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Matters arising

- The sender and receiver must understand the language of communication used
  - The ontology must be created and be accessible to all communicating agents

- KQML must operate within a communication infrastructure (not part of KQML) that allows agents to locate each other
  - Systems may use "on-premise" programs referred to as routers or facilitators

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Interchange Format (KIF)

- Logic-based language
- Recommended for expert system applications, databases, intelligent agents, etc
- It was designed specifically as an “interlingua”, i.e., as an intermediate language for translating other languages

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example Sentences/Expressions in KIF

```
(salary 015-46-3946 widgets 72000)
(salary 026-40-9152 grommets 36000)

(> (* (width chip1) (length chip1))
   (* (width chip2) (length chipt2)))

(=> (and (real-number ?x) (even-number ?n))
    (> (expt ?x ?n) 0))

(interested joe '(salary ,?x ,?y ,?z))

(progn (fresh-line t)
       (print "Hello!")
       (fresh-line t))
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Ontologies

An **ontology** specifies the objects, concepts, and relationships in a domain

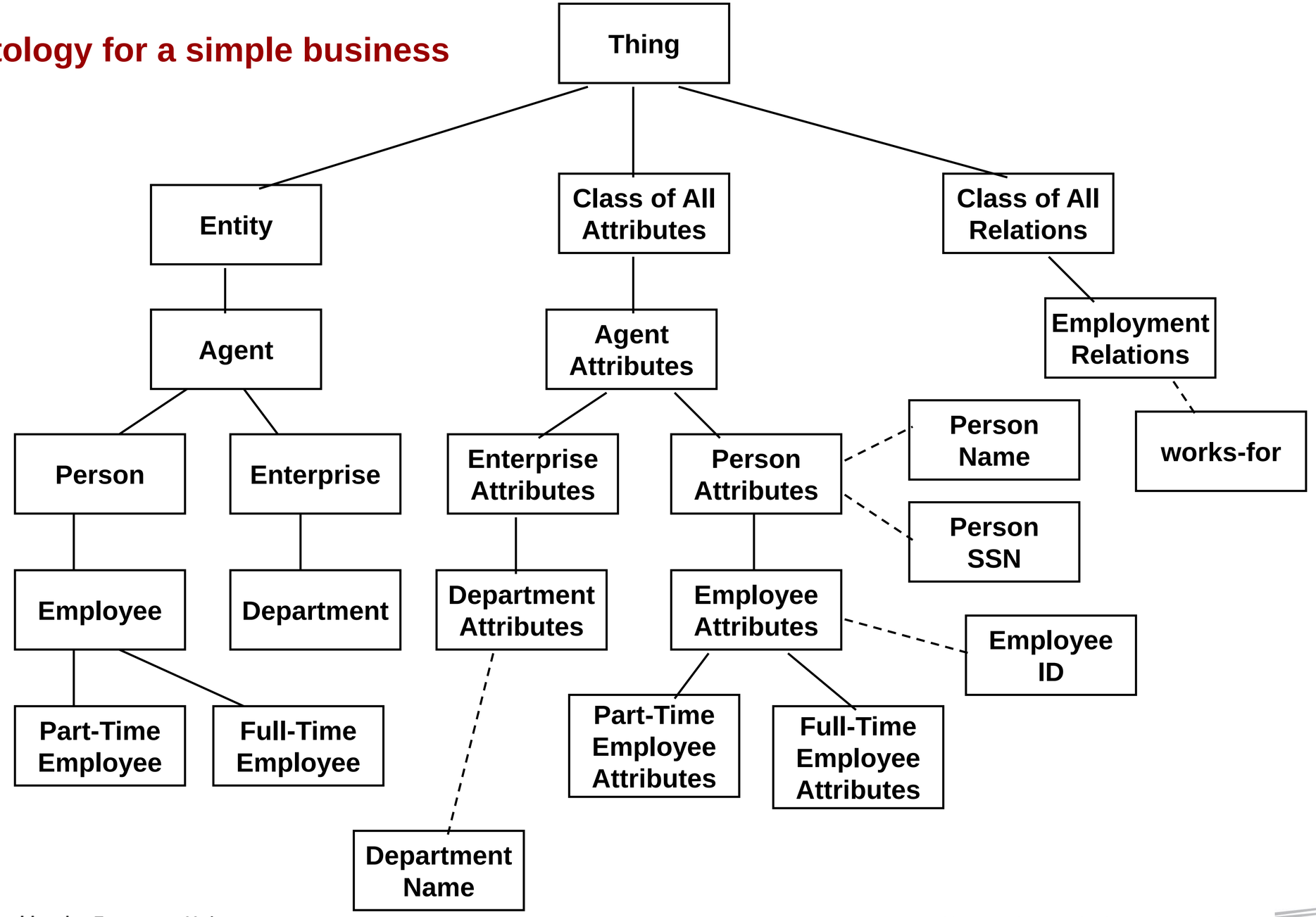
For example, the “Blocks-World” ontology could contain:

- ∀x (Block x) => (PhysicalObject x)
- (class Block)
- (class PhysicalObject)
- (subclassOf Block PhysicalObject)
- ∀x,y,z (instanceOf x y) ^ (subclassOf y x) => (instanceOf x z)
- (domain On PhysicalObject)
- (range On PhysicalObject)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example ontology for a simple business



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agent Interaction Protocols

- Coordination Protocols
- Cooperation Protocols
- Contract Net
- Blackboard Systems
- Negotiation
- Multiagent Belief Maintenance
- Market Mechanisms

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Activities that require coordination

- Define goals graph (classical AND/OR tree)
  - Identify and classify dependencies
- Assign particular regions of the goals graph to appropriate agents
- Control decisions regarding which areas of the graph to explore
- Graph navigation
- Confirmation that a successful navigation is reported

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Agent structures for coordination

- **Commitments**
  - Confirmations for taking specific actions
  - They provide some degree of predictability so that agents can take future actions of other agents into account when dealing with interagent dependencies, global constraints, or resource utilization conflicts
- **Conventions**
  - They provide the means to manage commitments in changing circumstances
  - Maintaining, modifying or abandoning commitments
- Commitments and conventions are the cornerstones of coordination:
  - Commitments provide the necessary structure for predictable interactions
  - Social conventions provide the necessary degree of mutual support

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

LIMITED-BANDWIDTH SOCIAL CONVENTION

INVOKE WHEN

Local commitment dropped

Local commitment satisfied

ACTIONS

RULE1: IF Local commitment satisfied

THEN inform all related commitments

RULE2: IF Local commitments dropped because unattainable

or motivation not present

THEN inform all strongly related commitments

RULE3: IF Local commitments dropped because unattainable
or motivation not present
AND communication resources not overburdened
THEN inform all weakly related commitments

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
BASIC JOINT-ACTION CONVENTION

INVOKE WHEN
        Status of commitment to joint action changes
        Status of commitment to attaining joint action in
                present team context changes
        Status of joint commitment of a team member changes

ACTIONS
RULE1:  IF Status of commitment to joint action changes OR
        IF Status of commitment to present team context changes
        THEN inform all other team member of these changes

RULE2:  IF Status of joint commitment of a team member changes
        THEN determine whether joint commitment still viable
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Cooperation Protocols

- Basic cooperation protocol strategy: Breakdown and distribute tasks
- Thus, if there are alternative ways of breaking down, a choice is needed
- Additionally, there may be dependencies between subtasks or conflicts between agents
- Distribution criteria
  - Avoid overloading critical resources
  - Assign tasks to agents who have the required capabilities
  - Ask a broad-perception agent to delegate work to other agents
  - Assigning overlapping responsibilities to agents to achieve coherence
  - Assigning highly interdependent tasks to agents that are spatially or semantically close
    - Minimizes communication and synchronization costs
  - Redistribute tasks if necessary to complete urgent tasks

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Task Distribution Mechanisms

- Market mechanisms
  - Tasks are paired with agents by general agreement or mutual selection
- Contract net
  - announcement, bid and award
- Planning agents are responsible for assigning tasks
- Organizational structure
  - Agents have predefined responsibilities for specific tasks

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Contract Net Protocol

- The most widely used collaboration protocol
- It is based on the contract mechanism implemented for the exchange of products and services
- Connection problem
  - how to find a suitable agent to carry out a particular task
  - manager
    - the agent wanting the task solved
  - contractors
    - agents who may be able to undertake the task

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Manager

Announces the task that wishes to be carried out

Receives and evaluates bids from potential contractors

“Signs” a contract with an appropriate contractor

Receives and synthesizes results

Contractor

Receives task announcements

Evaluates its capability to carry out the task

Replies (decline, bid)

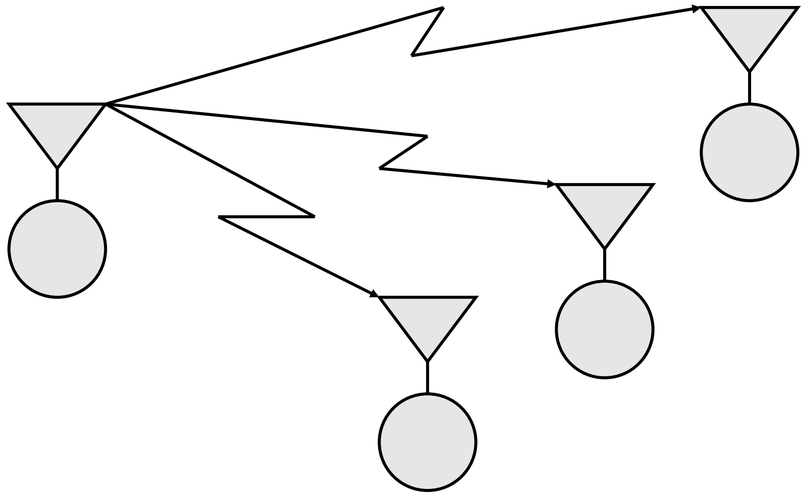
Carries out the task if it is given a contract

Sends the results

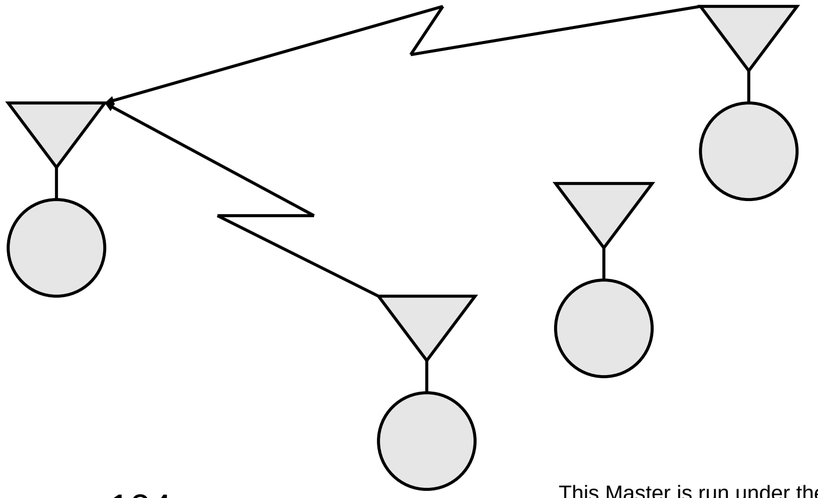
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Basic Steps of Contract Net

1. The manager announces the existence of tasks

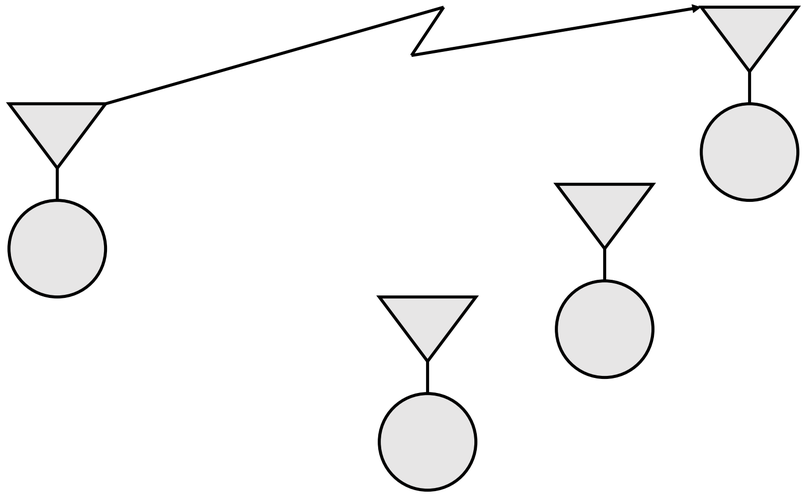


2. The agents evaluate the announcement, and some submit bids



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

3. The manager assigns the contract to the most appropriate agent, and the two of them communicate privately as required



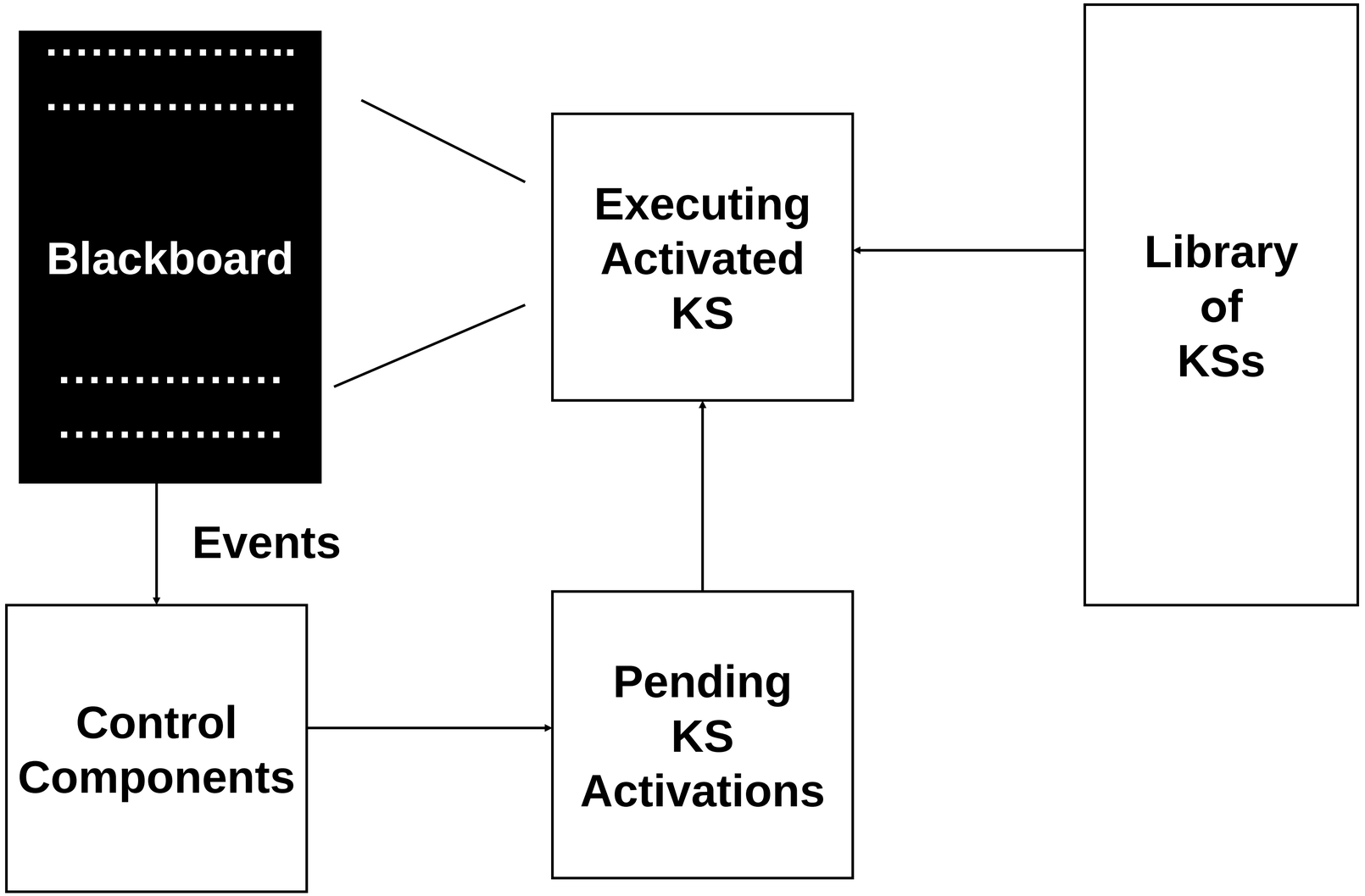
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Blackboard Systems

- **Independence of expertise**
  - The specialists (or Knowledge Sources – KSs) are experts on some aspects of the problem and can contribute to the solution independently of the particular mix of other specialists
- **Diversity of problem-solving techniques**
  - The knowledge and inference of each KS are hidden
- **Flexible representation of blackboard information**
  - No prior restrictions are placed on what information can be placed on the blackboard
- **Common interaction language**
  - KSs must be able to communicate and interpret the information recorded on the blackboard by other KSs
- **Event-based activation**
  - Each KS informs the blackboard system about the kind of events it is interested
- **Need for control**
  - A control component, separate from the KSs, is responsible for managing the course of problem solving
- **Incremental solution generation**
  - KSs contribute to the solution as appropriate

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Architecture of a basic blackboard system



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Negotiation

- A frequent form of interaction between agents with different goals
- The aim of the negotiation process is to reach a joint decision between two or more agents, each of whom seeks to achieve some individual goal or purpose

Key Negotiation Elements

- The language used by the participating agents
- The protocol followed during the negotiation
- The decision-making process each agent applies to decide its positions, points of departure, and criteria for reaching an agreement

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Negotiation Techniques

- **Focus on the environment**
  - How can the rules of the environment be designed so that agents, regardless of background, abilities or intentions, are able to interact productively and fairly?

- **Focus on agent**
  - Given the environment in which this agent must operate, what is the best strategy for it?

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Societies of Agents

- Agents, in a multi-agent environment, need to be social
- They act and coordinate to achieve both their own goals and the goals of their society
- They rely on other goals to learn things, so they don't have to know everything themselves

Social Commitments

- Commitments of an agent to another
- They are distinguished from individual commitments
- When an agent becomes a member of a group, it assumes certain roles and these roles entail given social commitments
- An agent joins a social group voluntarily and acts autonomously, but its actions are limited by its social commitments

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Summary

- Intelligent agents and their environment
- Rationality, autonomy, information gathering, learning
- Perception
- Purely reactive agents and agents with internal state
- Abstract and concrete agent architectures
- Multiagent systems and characteristics of multiagent environments
- Protocols of communication, coordination, cooperation and negotiation between agents
- Agent societies

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September - December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Knowledge Representation and Reasoning: Predicate Logic and Semantic Networks

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 5

Knowledge Representation and Reasoning: Predicate Logic and Semantic Networks

CONTENTS

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on knowledge representation and reasoning, predicate logic and semantic networks, students will be able:

Regarding Knowledge Representation:

1. Distinguish between data, information and knowledge.
2. List the knowledge types comprising some expertise.
3. Give the practical and theoretical properties of some knowledge representation.
4. Point out the key dispute elements regarding the declarative/procedural controversy.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on knowledge representation and reasoning, predicate logic and semantic networks, students will be able:

Regarding Predicate Logic:

1. Explain the syntax and semantics of predicate logic.
2. Discuss Conjunctive Normal Form (CNF), Disjunctive Normal Form (DNF) and clausal form and present and apply the algorithm for converting sentences to CNF.
3. Explain sentence equivalence (giving examples) and sentence unification.
4. Present the basic forms of deductive reasoning, namely modus ponens, universal specialization and resolution.
5. Discuss the procedure of resolution refutation, the use of heuristics, and its extension for deriving specific answers to questions.
6. Explain Horn clauses, negation as failure and the closed world assumption.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on knowledge representation and reasoning, predicate logic and semantic networks, students will be able:

Regarding Semantic Networks:

1. Explain the key elements of semantic networks as a knowledge representation formalism.
2. Discuss the hierarchical relations isa (is-a-kind-of) and part_of (is-a-part-of) through which taxonomies and meronomies of concepts/objects can be formed and outline how inheritance of properties is supported through these structures.
3. Explain how the formalism can be extended to partitioned semantic networks.
4. Outline the reasoning method of intersection search applied to semantic networks.
5. Draw a comparison between these two declarative formalisms (predicate logic, semantic networks).

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Knowledge Representation

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge

- Is the central element in symbolic AI
- Intelligent behavior and thinking are clearly conditioned by knowledge
- The automation/mechanization of knowledge in the context of intelligent computing systems (knowledge-based systems or expert systems) entails its representation in a formal, symbolic way
- The development of logics, or more generally formalisms, for the representation and reasoning with knowledge, is part of basic research in AI
- Representation formalisms are developed independently of some class of problems or applications

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Representation Languages

- Several of the representation formalisms have been implemented in the form of programming languages, which are called knowledge representation languages, e.g., OPS5, CLIPS, etc.
- Such languages provide higher abstraction than general programming languages
- They are primarily declarative and not procedural
- They incorporate some knowledge representation formalism and provide the relevant reasoning mechanisms

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

PROLOG and LISP

- The PROLOG language satisfies the above requirements and can therefore be considered a knowledge representation language
- The LISP language, although specifically designed to support AI applications, does not meet all the requirements because it does not provide a reasoning mechanism
  - The basic element of LISP is the symbolic expression
  - This makes it particularly suitable for implementing knowledge representation languages
  - Specifically, OPS5 (Official Production System) has been implemented in LISP and was used in the successful expert system R1/XCON
  - The syntax of OPS5 as well as CLIPS derives from LISP; in fact, CLIPS was implemented in C by NASA and stands for the “C Language Integrated Production System”

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Basic Knowledge Representation Formalisms

- Predicate Logic – declarative
- Semantic Networks – declarative
- Frames – declarative with procedural elements
- Production Rules - declarative

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Data, Information and Knowledge

- **Data** concern specific situations or entities, specific events, etc., e.g., John had a bad case of the flu in January 2020
- **Information** is generated dynamically from the current data set, e.g., in January 2020 there were three times as many cases of influenza as in December 2019:
  - Information is a kind of summary of data
  - So, it is at a higher level of abstractness than data
  - However, it does not cease to be specific
- **Knowledge** consists of generalizations, covering existing relevant data, as well as future data, e.g. 'If there is a sharp drop in temperature, then there is a high probability that the conditions will be created for a flu epidemic'
  - The scope of the truth of knowledge must be universal, and not just a particular set of data

Co-financed by the European Union
Connecting Europe Facility

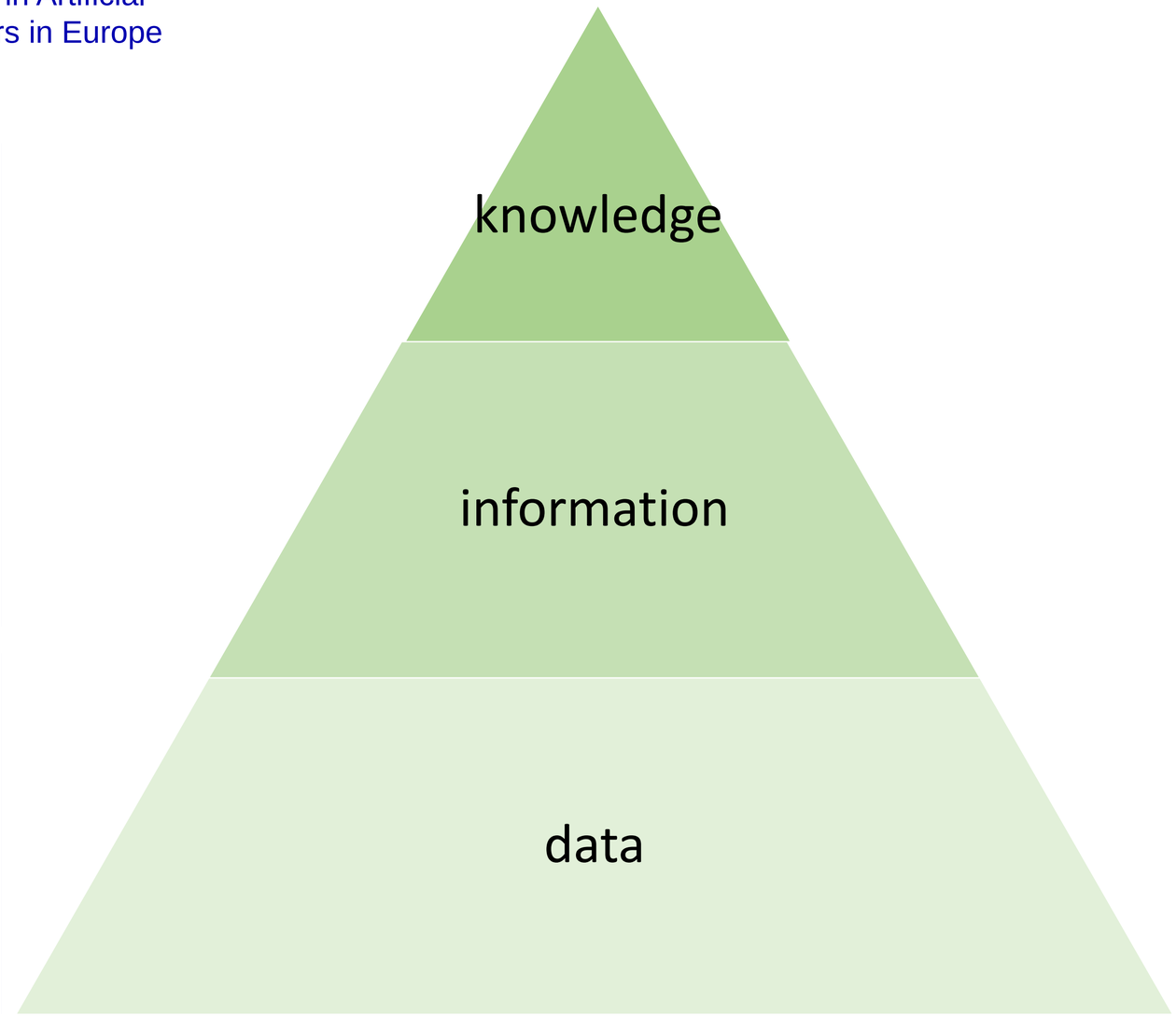
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Information is generated from data, while knowledge explains the data

or

From data comes information, from which knowledge can be born (beliefs or universal truths)

Knowledge can be acquired from various sources, e.g., domain experts or it can be automatically induced from data, e.g., through machine learning



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Databases vs Knowledge bases

- Databases:
  - much larger in size
  - the semantic distinctions that need to be made in a database are relatively simple, generally speaking

- Knowledge Bases:
  - much richer in structure
  - in a knowledge base there are many and complex semantic distinctions

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

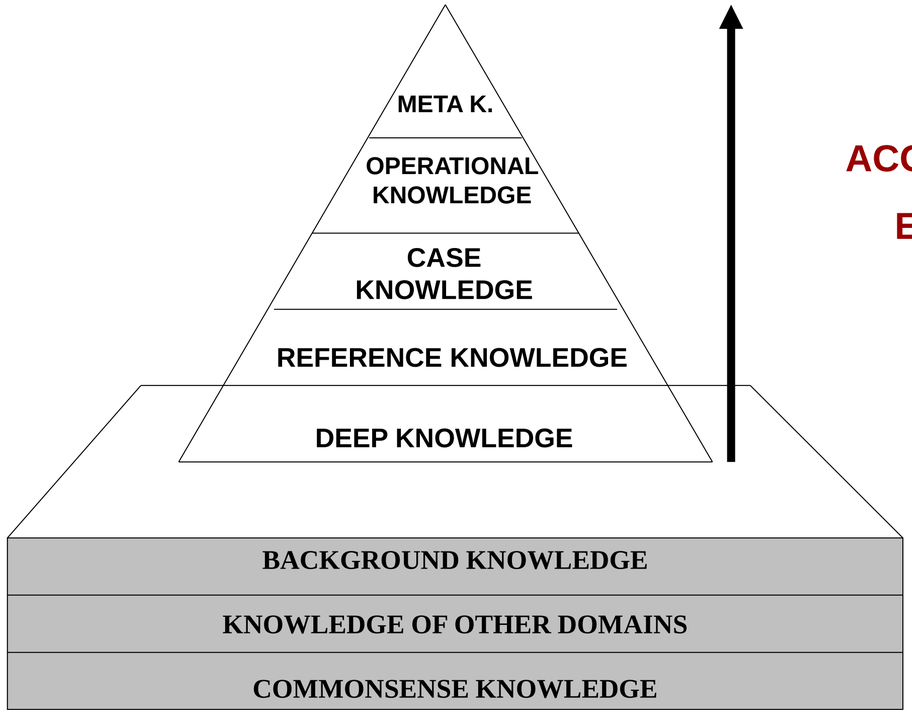
Expertise: Types of Knowledge

- **Theoretical Knowledge**: Structural and Relational Models, Taxonomies and Meronomies of Concepts - **Deep Knowledge**
- **Knowledge from reference sources**
- **Practical or actionable knowledge**: Knowledge that is directly related to problem solving, i.e., operational knowledge
- **Case Knowledge**: Expertise is acquired through experience and therefore an expert has extensive involvement in problem solving; he has recorded in his memory a multitude of previous relevant incidents
- **Meta-knowledge**: 'Knowledge about knowledge', which exists at various levels and has various views principally the strategic view and the reflective view
- **Background knowledge**, which every practitioner of the specific profession should know
- **Knowledge of other domains**
- **Commonsense knowledge**, i.e., knowledge of the world in general; the breadth of commonsense knowledge (and commonsense reasoning) is such that automating it independently of some goal or task is very difficult
- **Knowledge regarding cause and effect**: the causal relation concerns all domains where change, and therefore reasoning about change, matters

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

An expert possesses several, interacting, knowledge layers



ACQUISITION OF EXPERTISE

Knowledge Engineering methodologies aim at the elicitation of a model of expertise – types of knowledge and interactions between them

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Desirable Properties of some Knowledge Representation

Knowledge Representation Formalism

A coupling between a symbolic way of expressing knowledge (static part) and a set of reasoning mechanisms (dynamic part)

Power of Expression

of a language is:

- the kinds of sentences that the language allows us to express
- the semantic distinctions it allows us to articulate
- and ultimately what the language allows us to express

Efficiency of Reasoning

- A body of knowledge is represented so that reasoning can be conducted with the goal of solving relevant problems
- The ability to reason efficiently and effectively is as important as the power of expression

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Relation between Power of Expression and Reasoning Efficiency

Usually there is an inverse relationship between the power of expression and the reasoning efficiency – high expressive power implies complex reasoning mechanisms, while simple expression is associated with simple yet efficient reasoning mechanisms

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Practical Properties of a Knowledge Representation

- Logical adequacy
- Notational convenience or acquisitional efficiency
- Heuristic adequacy

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

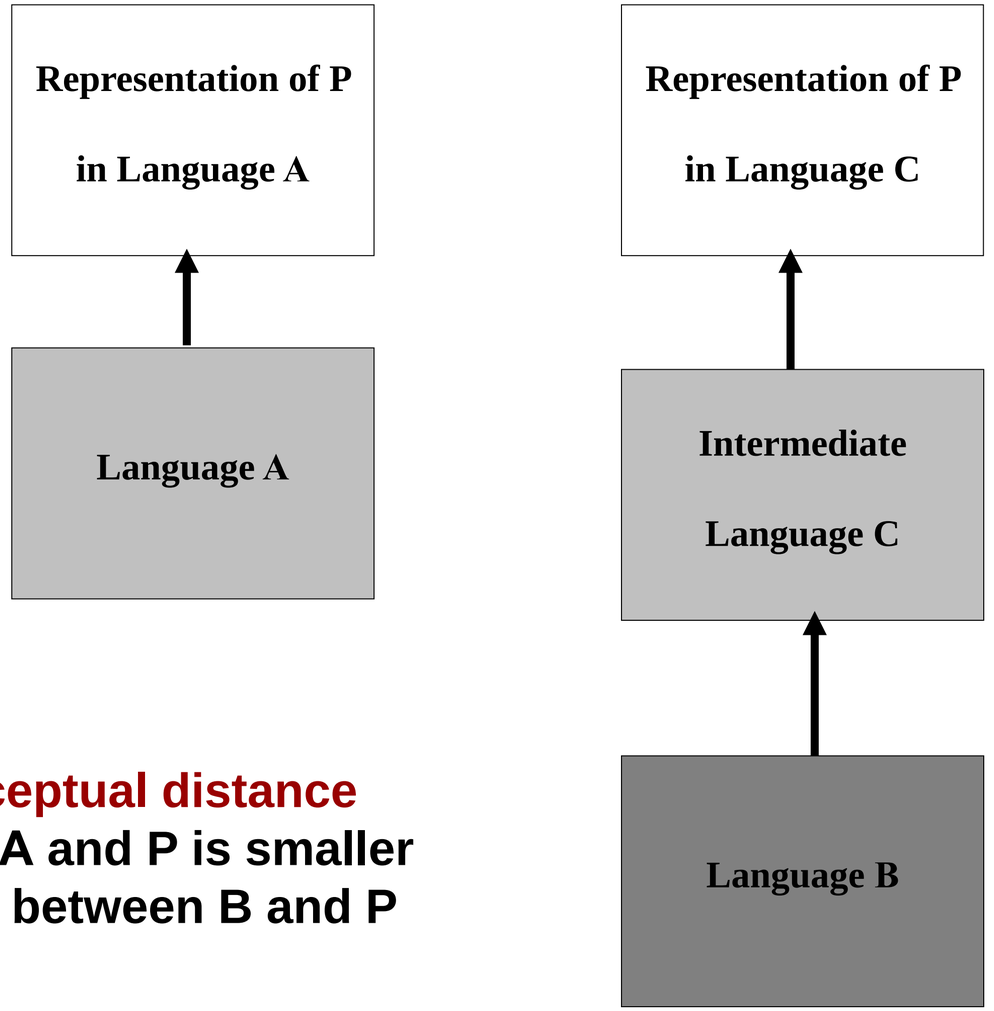
Logical Adequacy

The power of expression is adequate, i.e., all necessary distinctions are possible

Languages that have logical adequacy with respect to one class of problems do not necessarily facilitate expressivity equally

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

For example, languages A and B have logical adequacy with respect to problem P



The conceptual distance between A and P is smaller than that between B and P

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Notational Convenience or Acquisitional Efficiency

Usually, the amount of knowledge involved is large, and so it stands to reason that the closer the level of representation is to the level of knowledge, the easier it is to acquire the knowledge and express it in that notation.

Also, it is important that the representation is easy for people to understand and allows inspection of the knowledge (in this form), even if one does not know how it will be used.

Logical adequacy and acquisitional efficiency are concerned with the creation and maintenance of a knowledge base. The third desired practical property concerns the use of the knowledge base.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristic Adequacy

The ability to recall from an (extensive) knowledge base, the pieces of this knowledge that are relevant to the problem under consideration, so that reasoning regarding the production of the solution to the problem is efficient.

Simplicity of expression is associated with high heuristic adequacy, while heuristic adequacy decreases as the power of expression rises.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Theoretical Properties of a Knowledge Representation

- Theoretical properties concern the dynamic part of the representation, i.e., the reasoning mechanism, or the inference rules
- The knowledge base represents a 'micro-world', which is taken to be consistent, i.e., it does not contain contradictions
- The theoretical properties are:
  - Soundness
  - Completeness
  - Decidability

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Soundness

- The conclusions drawn are valid (they correspond to the 'reality' of the micro-world represented by the knowledge base)
- Therefore, two conflicting conclusions cannot be drawn at the same time, e.g., p and the negation of p (~p)
- In other words, a sound reasoning mechanism cannot lead to false conclusions from a true knowledge base

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Completeness

- For anything that is true in said 'reality', the reasoning mechanism is able to prove that it is indeed so, without the use of extraneous factors

Decidability

- The inference engine can answer any verification query, positively or negatively
- If the mechanism does not have completeness, it cannot have decidability

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Theoretical Properties

- The properties of soundness and completeness are the minimum desirable properties to guarantee the 'correctness' of reasoning.

- The property of decidability is also highly desirable, especially from a computational point of view. In general, however, achieving decidability is not possible. E.g., predicate logic does not have decidability, but semi-decidability.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The Declarative/Procedural Controversy

Arguments in favor of a Declarative Representation

- Emphasis must be placed on the static aspect of knowledge, what we know about various objects, events, and their relationships, and in general the states of the 'world'
- What we know about the 'world' must be described independently of how that knowledge can be used - multiple uses of the same body of knowledge
- Each piece of knowledge is stored only once and can be used in many different contexts – making it easy to update the knowledge base
- A declarative representation has high transparency as to what it represents, which is very positive for the desirable property of acquisitional efficiency

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The Declarative/Procedural Controversy

Arguments in favor of a Procedural Representation

- A 'cognitive' system is one that knows how to use its knowledge, how to refer to the relevant pieces of knowledge for a specific problem, how to draw its conclusions, etc.
- Therefore, cognitive behavior is best represented as processes, which articulate how knowledge is used, not what knowledge is
- It is usually easier to express how someone does something ('how-to' knowledge), for a given task goal, than to express what they know about something, at an abstract level
- Procedural knowledge is 'actionable' knowledge, directly related to the specific task, while descriptive knowledge is not immediately actionable, and it can be seen in retrospect that much of it was not needed
- Heuristics, default hypotheses, and probabilistic reasoning can all be more easily expressed through procedures
- Implementing a procedural representation using compilation rather than interpretation results in faster processing

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The Declarative/Procedural Controversy

In retrospect it transpires that the weaknesses of the procedural representation outweigh its strengths, because the declarative representation is the one that has prevailed, mainly for reasons of transparency.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Predicate Logic

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Predicate Logic

- Predicate logic is an eminently declarative representation.
- Knowledge represented in this way can be processed using general rules of reasoning.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Syntax and Semantics

The validity of syllogisms depends on their form, not their content. For example, the validity of the following syllogisms

All men are mortal. Socrates is a man.
Hence Socrates is mortal.

All squares are red. This is a square.
Hence this is red.

depends on the following syllogism form:

Every x that satisfies property Φ satisfies property Ψ.
A satisfies property Φ.
Hence A satisfies property Ψ.

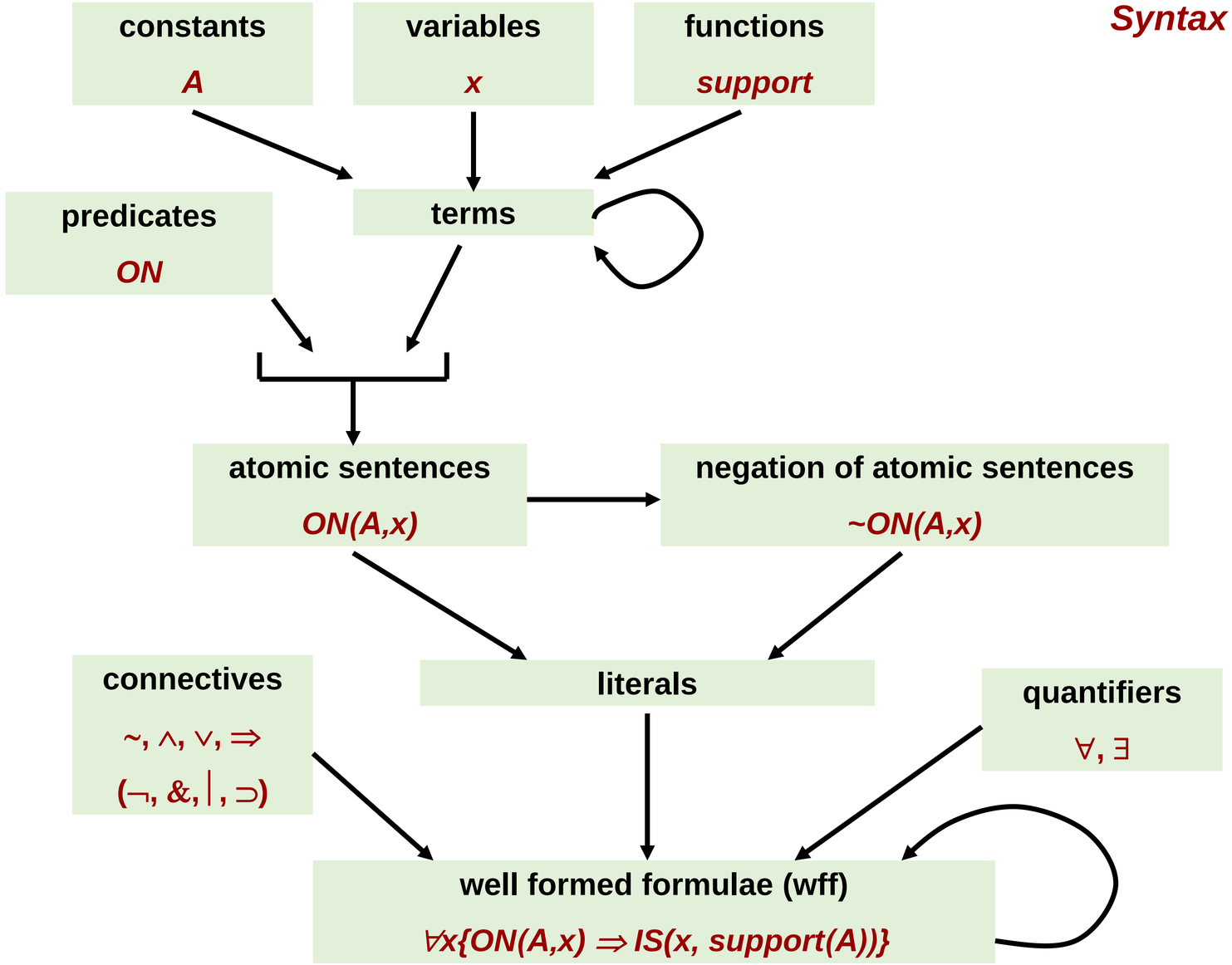
$\forall x\ \Phi(x) \Rightarrow \Psi(x)$
$\Phi(A)$
$\therefore\ \Psi(A)$

known as modus ponens.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Syntax of Predicate Logic



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

well formed formulae – wff:

1. Every atomic sentence is a wff.

2. Let A be a wff. Then, ~A, $\forall$x A, and $\exists$x A, are also wff.

3. Let A and B be wff. Then, A $\wedge$ B, A $\vee$ B, and A $\Rightarrow$ B, are also wff.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Semantics

- **Constants** represent objects or entities
- **Variables** represent domains of constants
- **Functions** represent mappings between the domains of constants
- **Predicates without arguments** (0-place predicates) are simple logical sentences (propositions) which are true or false
- **Predicates with at least one argument** (n-place predicates, where n≥1) represent relations

Remarks:

- Predicates represent relations between objects, while functions return objects.
- There is a relation amongst the arguments of a function and the object returned, e.g., sum(2,3) = 5, could be represented as SUM(2,3,5)
- The opposite does not hold, since only one-to-one or many-to-one relations can be represented as functions

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Predicate	Function
FATHER(x,y)	father(x)
RED(x)	?
BROTHER(x,y)	brother(x) **What happens if there are more than one brothers?**
SUM(x,y,z)	sum(x,y)
PRIME_NUMBER(x)	?

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Computable Predicates

Predicates that are implemented as functions, e.g., PRIME_NUMBER

Sentence Representation

One sentence, for example ‘The house is yellow’, can be represented in several ways such as

```
YELLOW(HOUSE_1)
COLOR(HOUSE_1, YELLOW)
VALUE(COLOR, HOUSE_1, YELLOW)
```

Wff semantics

wff are true, false, or unknown

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Quantifiers

- Universal quantifier, ∀, that allows us to express general sentences, such as ‘All elephants are gray’,

  $\forall x\ \text{ELEPHANT}(x) \Rightarrow \text{GRAY}(x)$

- Existential quantifier, ∃, that allows us to express non-universal sentences, for example the existence of certain objects, entities, etc., such us ‘There is someone who wrote Chess on the Computer’,

  $\exists x\ \text{WROTE}(x, \text{CHESS\_ON\_COMPUTER})$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

∀x A ή ∃x A

- Scope of quantifier: the wff A
- Variable x is quantified over, and is considered a bound variable
- Free variables: variables in a sentence that are not quantified over
- Semantically correct are the wff that do not contain free variables

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Normal Forms

- The logical connectives ($\sim$, $\wedge$, $\vee$, $\Rightarrow$) allow the expression of complex sentences of any degree of complexity
- For computation purposes though, simplicity of expression is highly advisable
- Normal forms provide the sought simplicity without compromising the power of expression. These are:
  - Conjunctive Normal Form – CNF
  - Disjunctive Normal Form – DNF
  - Clausal Form - CNF disguised

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Converting wff to normal forms

- Any wff can be converted to any of the three normal forms
- For each of these, the resulting simplification allows the application of only one inference rule, namely that of resolution

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Conjunctive Normal Form

$A_1 \wedge A_2 \wedge \ldots \wedge A_n$, where $n \geq 1$ and
each A is of the form $B_1 \vee B_2 \vee \ldots \vee B_m$, where $m \geq 1$ and
each B is a literal

Disjunctive Normal Form

$A_1 \vee A_2 \vee \ldots \vee A_n$, where $n \geq 1$ and
each A is of the form $B_1 \wedge B_2 \wedge \ldots \wedge B_m$, where $m \geq 1$ and
each B is a literal

Clausal Form

$A_1 \vee A_2 \vee \ldots \vee A_n \Leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_m$, where $n \geq 0$, $m \geq 0$
and each A and B are atomic sentences

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Clausal Form

In this form there is no negation.

Facts B_i constitute hypotheses, while facts A_j conclusions.

The semantics is: If all hypotheses hold, the given conclusions can be drawn.

Clausal Form is a direct conversion of CNF, aiming to eliminate negation:

$$A_1 \vee A_2 \vee \ldots \vee A_n \Leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_m \equiv$$
$$A_1 \vee A_2 \vee \ldots \vee A_n \vee \sim B_1 \vee \sim B_2 \vee \ldots \vee \sim B_m$$

Note: $A \Rightarrow B \equiv \sim A \vee B$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Special clauses

$\Leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_m$	**No conclusion follows from the (complex) hypothesis. This sentence is inconsistent.**
$A_1 \vee A_2 \vee \ldots \vee A_n \Leftarrow$	**No hypothesis is needed for the proposition to be valid. It is always true.**
$\Leftarrow \equiv \square$	**This is the empty sentence ($\square$) which is always untrue.**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Sentence Equivalences

Two wff are equivalent, if for any possible interpretation they have the same truth value.

$$A \Rightarrow B \equiv \sim A \vee B$$
$$\sim(\sim A) \equiv A$$

De Morgan

$$\sim(A \wedge B) \equiv \sim A \vee \sim B$$
$$\sim(A \vee B) \equiv \sim A \wedge \sim B$$

Distributive Laws

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$
$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$$

Commutative Laws

$$A \wedge B \equiv B \wedge A$$
$$A \vee B \equiv B \vee A$$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Sentence Equivalences

Associative Laws

$$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$$
$$(A \vee B) \vee C \equiv A \vee (B \vee C)$$

Contrapositive Law – modus tollens

$$A \Rightarrow B \equiv \sim B \Rightarrow \sim A$$

Quantifier equivalences

$$\sim \exists x\, A \equiv \forall x \sim A$$
$$\sim \forall x\, A \equiv \exists x \sim A$$
$$\forall x \{A \wedge B\} \equiv \forall x\, A \wedge \forall x\, B$$
$$\exists x \{A \vee B\} \equiv \exists x\, A \vee \exists x\, B$$

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Inference Rules

New wff can be derived from existing ones using inference rules.

Modus Ponens

From $\forall x \{\Phi(x) \Rightarrow \Psi(x)\}$ and $\Phi(A)$ we can derive $\Psi(A)$

Universal Specialization

From $\forall x\ \Phi(x)$ we can derive $\Phi(A)$

Resolution

From $A \vee B$ and $\sim A \vee C$ we can derive $B \vee C$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Converting Sentences to CNF

$$\forall x\,[BRICK(x) \Rightarrow (\exists y\,[ON(x,y) \wedge \sim PYRAMID(y)] \wedge \sim\exists y\,[ON(x,y) \wedge ON(y,x)] \wedge \forall y\,[\sim BRICK(y) \Rightarrow \sim SAME(x,y)])]$$

The sentence in natural language:

Every object which is a brick can be placed on top of another object that cannot be a pyramid. In addition, the object on which the brick is placed, cannot itself be placed on the brick. Moreover, a brick cannot be the same with an object that is not a brick.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Conversion Algorithm

1. Eliminate the implication connective (⇒) using the equivalence A ⇒ B ≡ ~A ∨ B

$$\forall x\,[\sim BRICK(x) \vee (\exists y\,[ON(x,y) \wedge \sim PYRAMID(y)] \wedge \sim\exists y\,[ON(x,y) \wedge ON(y,x)] \wedge \forall y\,[\sim(\sim BRICK(y)) \vee \sim SAME(x,y)])]$$

2. Move negations to the level of atomic sentences

$$\forall x\,[\sim BRICK(x) \vee (\exists y\,[ON(x,y) \wedge \sim PYRAMID(y)] \wedge \forall y\,[\sim ON(x,y) \vee \sim ON(y,x)] \wedge \forall y\,[BRICK(y) \vee \sim SAME(x,y)])]$$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

3. Eliminate existential quantifiers

E.g., $\exists x\ P(x)$, is converted to P(A) – variable x, is replaced by a dummy constant A.

What happens, though, with the sentence $\forall x\ \exists y\ P(x,y)$?

Variable y depends on variable x.

Hence, the conversion $\forall x\ P(x,A)$ is not logically equivalent.

The correct conversion is: $\forall x\ P(x, g(x))$. Function g is called a Skolem function.

<u>General Case</u>

$\forall x_1\ \forall x_2 \ldots . \forall x_n \ldots . \exists y \ldots .$, where $n \geq 0$.
Variable y is replaced with a function with n arguments, say $g(x_1, x_2, \ldots, x_n)$.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

$$\forall x\ [\sim BRICK(x) \vee$$

$$((ON(x,support(x)) \wedge \sim PYRAMID(support(x))) \wedge$$

$$\forall y\ [\sim ON(x,y) \vee \sim ON(y,x)] \wedge$$

$$\forall y\ [BRICK(y) \vee \sim SAME(x,y)])]$$

4. Rename (the universally quantified) variables, such that each variable has a different name

$$\forall x\ [\sim BRICK(x) \vee$$

$$((ON(x,support(x)) \wedge \sim PYRAMID(support(x))) \wedge$$

$$\forall y\ [\sim ON(x,y) \vee \sim ON(y,x)] \wedge$$

$$\forall z\ [BRICK(z) \vee \sim SAME(x,z)])]$$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

5. Move the universal quantifiers ∀ to the left

Prenex form: all universal quantifiers appear at the beginning of the sentence and constitute the prefix while the rest of the sentence is called the matrix

$$\forall x\ \forall y\ \forall z\ [\sim BRICK(x) \vee ((ON(x,support(x)) \wedge \sim PYRAMID(support(x))) \wedge [\sim ON(x,y) \vee \sim ON(y,x)] \wedge [BRICK(z) \vee \sim SAME(x,z)])]$$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

6. Move the disjunctions at the level of literals, using the Distributive Laws

$$\forall x \ \forall y \ \forall z \ [(\sim BRICK(x) \vee ON(x,support(x))) \ \wedge$$

$$(\sim BRICK(x) \vee \sim PYRAMID(support(x))) \wedge$$

$$(\sim BRICK(x) \vee \sim ON(x,y) \vee \sim ON(y,x)) \wedge$$

$$(\sim BRICK(x) \vee BRICK(z) \vee \sim SAME(x,z))]$$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

7. Eliminate $\forall$ and $\wedge$, and standardize variables apart so that each name appears in only one disjunctive sentence (clause)

$\sim BRICK(x) \vee ON(x,support(x))$

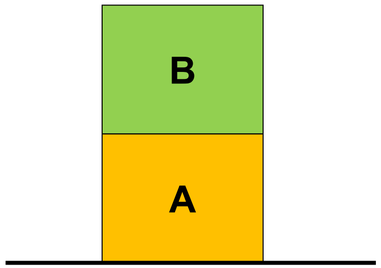
$\sim BRICK(w) \vee \sim PYRAMID(support(w))$

$\sim BRICK(u) \vee \sim ON(u,y) \vee \sim ON(y,u)$

$\sim BRICK(v) \vee BRICK(z) \vee \sim SAME(v,z)$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Resolution Refutation Procedure



ON(A, TABLE)
ON(B,A)
FREE(B)

Axioms

$\forall x\ [FREE(x) \Rightarrow \sim\{\exists y\ ON(y, x)\}]$

$\forall x\ \forall y\ [ON(x,y) \Rightarrow ABOVE(x,y)]$

$\forall x\ \forall y\ \forall z\ [ABOVE(x,y) \wedge ABOVE(y,z) \Rightarrow ABOVE(x,z)]$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Resolution Refutation Procedure

- For any valid reasoning there is a consistent set of sentences. E.g., the set of sentences $\{\forall x\ \Phi(x) \Rightarrow \Psi(x), \Phi(A), \Psi(A)\}$ is consistent - sentence $\Psi(A)$ follows from the other two (modus ponens rule).

- Hence, we can prove that the reasoning from the sentences $P_1, \ldots, P_n$ to the conclusion C is valid, by proving that the set of sentences $\{P_1, \ldots, P_n, C\}$ is consistent.

- Usually, it is easier to prove that the set of sentences $\{P_1, \ldots, P_n, \sim C\}$ is inconsistent. In this case, the set of sentences $\{P_1, \ldots, P_n, C\}$ is consistent and the reasoning '$P_1, \ldots, P_n$ hence C' is valid.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Sentence Unification

Replacement of variables with terms, provided that a term does not contain the variable, in order for the two sentences to become identical, e.g.:

- ON(x,y) and ON(A, support(A)) can be unified – the replacements are A/x and support(A)/y
- Q(x,g(y)) και Q(B,y) cannot be unified. The replacement B/x is permitted, but not the replacement g(y)/y.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Applying Resolution on Disjunctive Sentences

P(x,f(y,z)) ∨ ~Q(A,y) ∨ R(B,g(x))
S(v,m) ∨ Q(w,u) ∨ N(w,u)

A clause is a set of literals:

{P(x,f(y,z)), ~Q(A,y), R(B,g(x))}
{S(v,m), Q(w,u), N(w,u)}

Variables have been normalized, i.e., they have been renamed so that each has a different name.

Resolution can be applied to subclauses {~Q(A,y)} and {Q(w,u)}, since the sentences {~Q(A,y)} and {Q(w,u)} can be unified using the substitutions A/w και y/u. The result is the new clause:

{P(x,f(y,z)), R(B,g(x)), S(v,m), N(A,y)}

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Resolution Procedure

In order to prove that some wff, say W, is logically implied from a set, S, of wff,we prove that the set S∪{~W} is inconsistent.

A set is inconsistent if at the same time it implies both some sentence and the given sentence's negation.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example

We want to prove that from the sentences

1. Anyone who can read is literate.

$$\forall x\,[R(x) \Rightarrow L(x)]$$

2. Seals are not literate.

$$\forall x\,[S(x) \Rightarrow \sim L(x)]$$

3. Some seals have intelligence.

$$\exists x\,[S(x) \wedge I(x)]$$

it can be deduced that

4. Some who have intelligence cannot read.

$$\exists x\,[I(x) \wedge \sim R(x)]$$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

It will be proved, by proving that the set of sentences {1, 2, 3, ~4} is inconsistent.

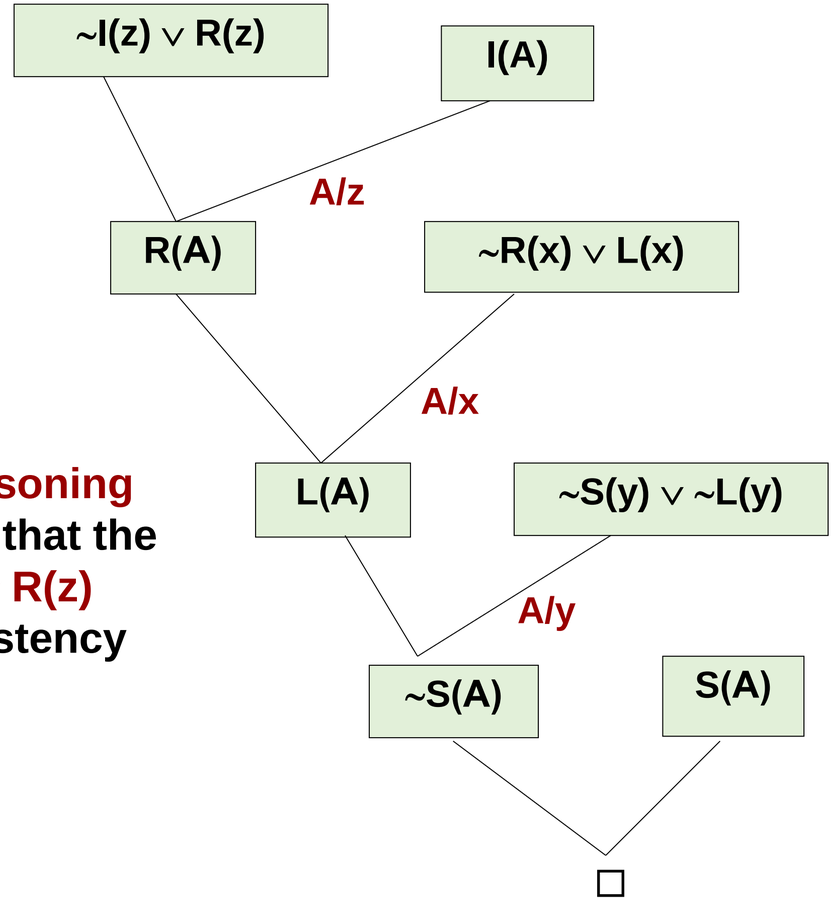
Converting sentences 1-3 and ~4 to CNF:

1. ~R(x) ∨ L(x)
2. ~S(y) ∨ ~L(y)
3α. S(A)
3β. I(A)
~4. ~I(z) ∨ R(z)

All variables have been normalized.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Resolution Refutation Tree



Goal-driven reasoning aiming to show that the sentence **~I(z) ∨ R(z)** creates inconsistency

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Resolution Refutation Procedure

1. DISJUNCTIVE_SENTENCES ← S

2. Repeat

2.1 Choose from the DISJUNCTIVE_SENTENCES two distinct sentences, say c_i and c_j, on which resolution can be applied

2.2 Derive the successor sentence, say r_{ij}, from the parent sentences c_i and c_j

2.3 Add r_{ij} in the DISJUNCTIVE_SENTENCES

Until the empty sentence is a member of the DISJUNCTIVE_SENTENCES

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Detecting the inconsistency is yet another search problem

- A **problem state** consists of a set of disjunctive sentences
- The **operator** is resolution
- The **initial state** consists of the original set of disjunctive sentences and the negation of the goal-sentence
- The **final state** contains the empty sentence

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Using Heuristics

A given search node can have multiple successor nodes, one for each different pair of potential parent sentences, i.e., the resolution operator can have multiple applications from one search node.

Hence, in step 2.1, the use of appropriate heuristics is called for in order to guide the selection of the parent sentences.

If the selection is done in a systematic way, the search procedure will eventually detect the inconsistency, if indeed there is an inconsistency.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Simplifying Disjunctive Sentences

The set of DISJUNCTIVE_ SENTENCES can be simplified as follows:

- **Eliminate tautologies, e.g.:**
  P(x) ∨ B(y) ∨ ~B(y)
  P(f(A)) ∨ ~P(f(A))

- **Eliminate sentences which are subsumed by other sentences, e.g.:**
  P(x) subsumes P(y) ∨ Q(z)
  P(x) subsumes P(A)
  P(x) ∨ Q(A) subsumes P(f(A)) ∨ Q(A) ∨ R(y)

- **Evaluate literals, e.g., the sentence EQUALS(add_1(2), 3) ∨ Φ(A) can be eliminated since the evaluation of the literal EQUALS(..) converts the sentence to a tautology. Likewise, the sentence SMALLER(6,3) ∨ Φ(A) can be simplified to Φ(A).**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristics

Set of support:

Using this heuristic gives rise to backwards reasoning, i.e., goal-driven reasoning

Unit preference:

The successor sentence to be shorter in length than its parent sentences

Both heuristics aim to guide the search nearer to the empty sentence, and hence nearer to the goal state, and can be used in combination.

Their use touches the heuristic adequacy of the representation of disjunctive sentences.

There could be several routes leading to the sought inconsistency. If the goal is to detect the inconsistency, we are interested on the shorter route. However, for answer extraction, it may be necessary to construct all routes.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Answer Extraction

- A specific question-sentence, can be answered with a yes, or a no
- A more general question, e.g., ‘What food does Yiannis like?’ can be answered either negatively (Yiannis does not like any food) or the useful thing would be to give specific positive answers, for example Yiannis likes oranges

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example

1. $\forall x\ \forall y\ [ON(x,y) \Rightarrow ABOVE(x,y)]$

2. $\forall x\ \forall y\ \forall z\ [ABOVE(x,y) \wedge ABOVE(y,z) \Rightarrow ABOVE(x,z)]$

3. ON(A, TABLE)

4. ON(B, A)

Question: ‘Cube B is above which objects?’

5. $\exists x\ [ABOVE(B,x)]$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Converting sentences to CNF:

1.~ON(u,v) $\vee$ ABOVE(u,v)

2.~ABOVE(x,y) $\vee$ ~ABOVE(y,z) $\vee$ ABOVE(x,z)

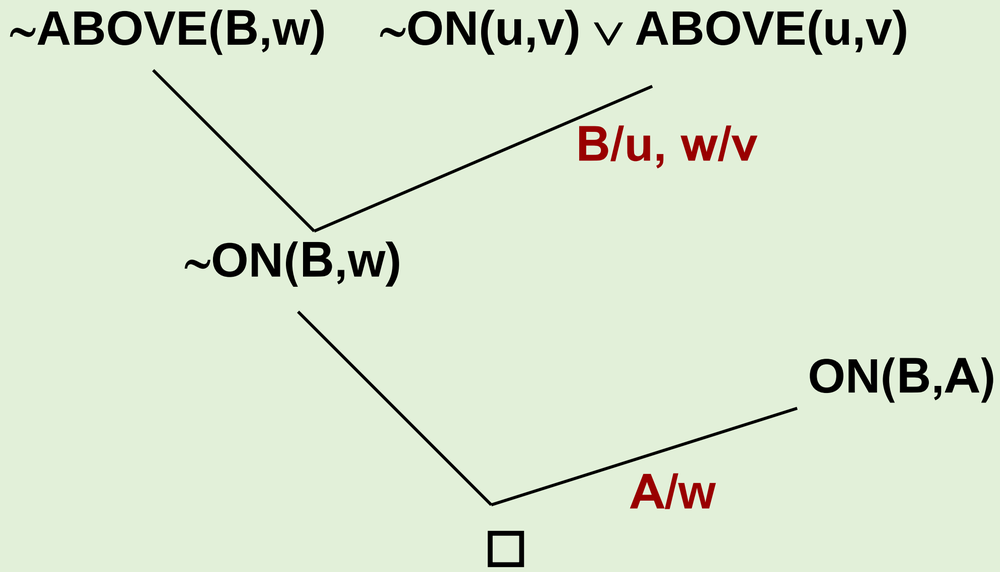
3.ON(A, TABLE)

4.ON(B, A)

~5. ~ABOVE(B, w)

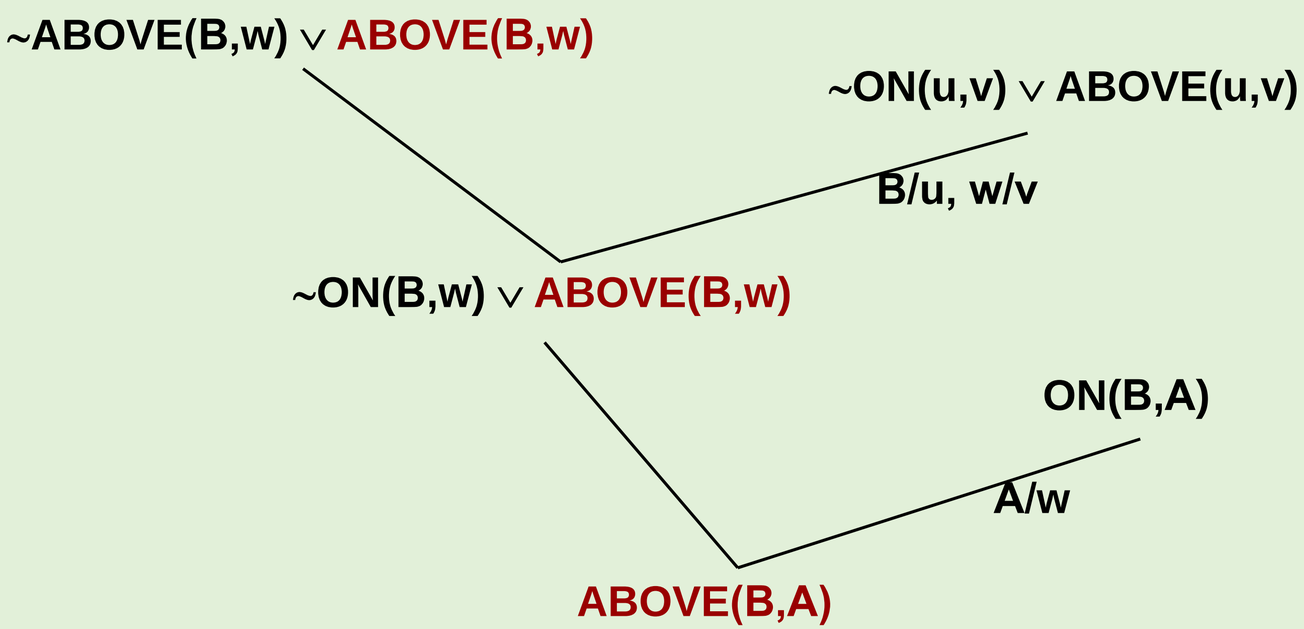
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Step One: Should the question be answered positively?



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Step Two: What is the specific answer?



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Horn Clauses and Negation as Failure

Clausal Form

$A_1 \vee A_2 \vee \ldots \vee A_n \Leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_m$, where $n \geq 0$, $m \geq 0$.

Negation does not appear at all.

Horn clauses are a subcategory of Clausal Form, where $0 \leq n \leq 1$, i.e., it is not allowed to have more than one atomic sentence in the left part.

The following sentences are Horn clauses:

$A \Leftarrow B_1 \wedge B_2$

$A \Leftarrow$

$\square \Leftarrow B$

$\square \Leftarrow$

But the ones below are not:

$A_1 \vee A_2 \Leftarrow B$

$A_1 \vee A_2 \Leftarrow$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Horn clauses

- The important constraint, in relation to the whole Clausal Form, is that it is not possible to have disjunctive conclusions
- As such the power of expression, and consequently the logical adequacy, are reduced
- The reduction is justified on the ground that the elimination of disjunctive conclusions facilitates greatly the computability

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Negation as Failure

- Negation is introduced, but with a rather peculiar interpretation
- The sentence that every mammal is of male or female gender

  $GENDER\_MALE(x) \vee GENDER\_FEMALE(x) \Leftarrow MAMMAL(x)$

  in logic programming will be translated as follows

  $GENDER\_FEMALE(x) \Leftarrow MAMMAL(x) \wedge \sim GENDER\_MALE(x)$ or

  $GENDER\_MALE\ (x) \Leftarrow MAMMAL(x) \wedge \sim GENDER\_FEMALE(x)$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

How Negation is Interpreted

- A fact (atomic sentence) is considered untrue, if it cannot be proven that it is true
- The classic interpretation is that some sentence is either true or false, but it does not mean that we know, or are able to prove which is the correct truth value

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Closed World Assumption

Either we know that an atomic sentence is true, or we can prove that it is true, otherwise we take it to be false.

This means that we know all relevant (basic) facts, and our knowledge of the given domain is complete.

Consider the following implications:

$$\text{FEVER}(x) \Leftarrow \text{FLU}(x)$$
$$\text{FEVER}(x) \Leftarrow \text{SMALLBOX}(x)$$

Every sentence on its own suffices to prove the presence of fever. However, can it be said that collectively these sentences constitute the necessary conditions regarding the truth value of predicate FEVER?

Obviously not. Several other causes of fever may be listed, such as cold, food poisoning, measles, and lots of others.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Semantic Networks

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on knowledge representation and reasoning, predicate logic and semantic networks, students will be able:

Regarding Semantic Networks:

1. Explain the key elements of semantic networks as a knowledge representation formalism.
2. Discuss the hierarchical relations isa (is-a-kind-of) and part_of (is-a-part-of) through which taxonomies and meronomies of concepts/objects can be formed and outline how inheritance of properties is supported through these structures.
3. Explain how the formalism can be extended to partitioned semantic networks.
4. Outline the reasoning method of intersection search applied to semantic networks.
5. Draw a comparison between these two declarative formalisms (predicate logic, semantic networks).

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Semantic Networks

- A **semantic network** is a knowledge structure that depicts how concepts are related to one another and illustrates how they interconnect.
- The aim is to represent all knowledge about some concept or object, giving emphasis to the **structure** of knowledge.
- The formalism is rooted on research findings about the understanding of natural language, specifically how semantic distinctions and relationships between concepts are depicted.
- The term **associative networks** is used synonymously or as a generalization of semantic networks, influenced from findings regarding the nature of human memory as an **associative memory**.
- The formalism of **frames**, to be discussed at the next unit, also emphasizes the structuring of knowledge and has many commonalities with semantic/associative networks.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Structural Elements of Semantic Networks

A semantic network consists of nodes and arcs (relationships).

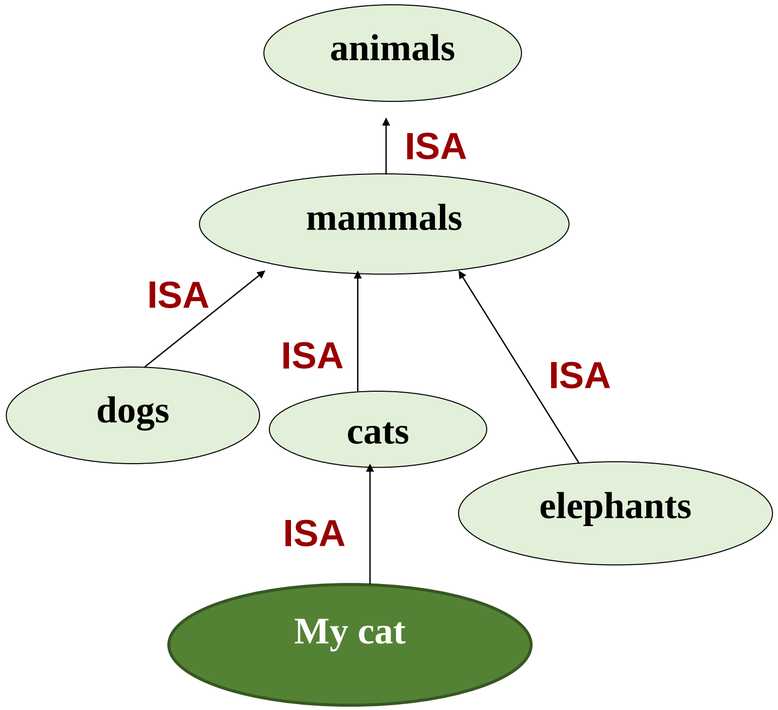
In the basic version, the nodes represent ‘inseparable’ entities that cannot be broken down to finer components.

In the extended version of partitioned semantic (associative) networks, a node can be a partition, consisting of a semantic network at a lower level.

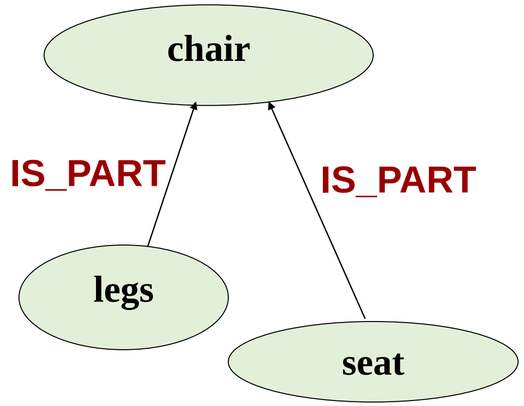
Two important, general purpose, relations, are the taxonomic και meronomic relations, respectively ‘ISA’ και ‘IS_PART’.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Taxonomy



Meronomy



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Taxonomies and Meronomies of Concepts

- A **taxonomy** is an organization of categories and subcategories
  - 'ISA' (is-a-kind-of (isa))
- A **meronomy** is an organization of constituent parts/components
  - 'IS_PART' (is_part)
- Relations 'ISA' and 'IS_PART' are transitory and hence the hierarchies formed from these relations are hereditary (support the **inheritance** of properties)
- The top-down flow of information, in some taxonomy, is **not monotonic,** i.e., exceptions should be supported

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Flow of Reasoning

- In taxonomies, top-down reasoning concretizes, while bottom-up reasoning generalizes/ broadens
- In meronomies, top-down reasoning breaks down the object into constituent elements, while in the opposite direction it synthesizes the object from the elements that make it up

Example Rules of Inference

- If a category is eliminated, all its subcategories are eliminated; beware, though, of non-strict taxonomies, where a subcategory can belong to several categories
- If at least one instance of a subcategory exists, all categories to which the subcategory belongs, exist
- If a complex object is ok, all its components are ok; again, there may be exceptions to this rule, e.g., a four-legged stool may still be ok if one of its legs is broken

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Semantic Network Structure

A semantic network consists of nodes and arcs.

Nodes represent concepts, objects and events, while arcs represent binary relations.

Example: ‘Yiannis gave the book to Mary’

In predicate logic the sentence could be expressed as

GIVE(YIANNIS, MARY, BOOK_1)

Alternatively, it could be expressed as the conjunction

ISA(GIVE_1, GIVE) ∧
GIVER(GIVE_1, YIANNIS) ∧
RECIPIENT(GIVE_1, MARY) ∧
OBJECT(GIVE_1, BOOK_1)

The central concept is the event, which from a (three argument) relation is converted to an object

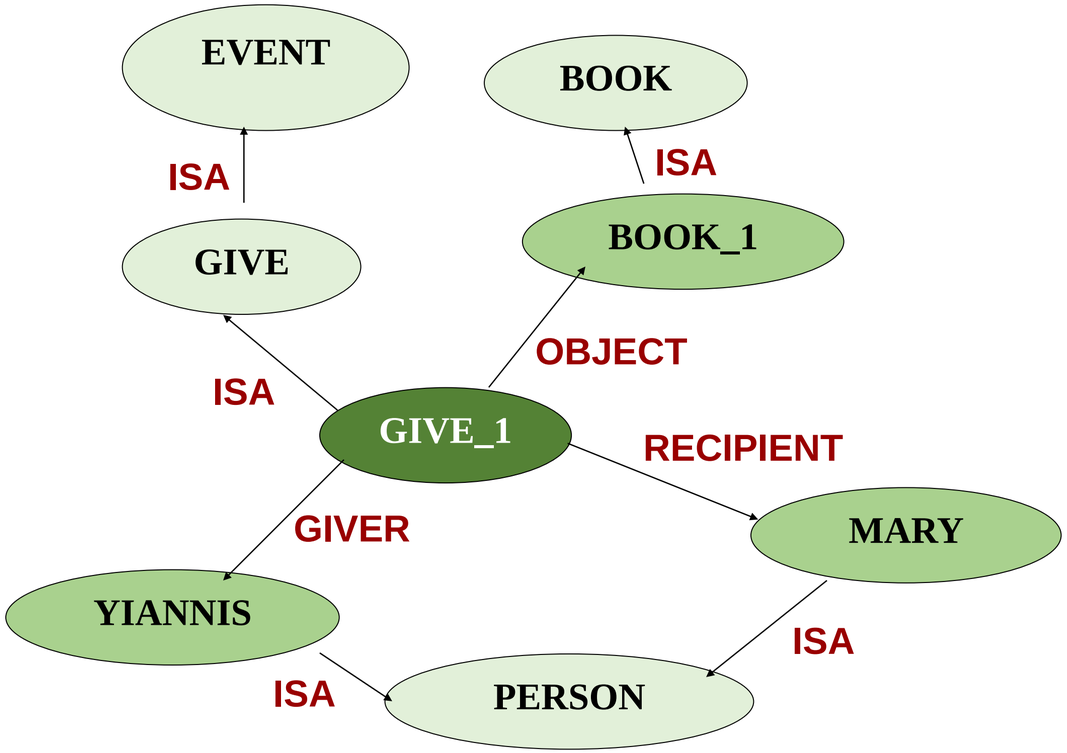
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Corresponding Semantic Network

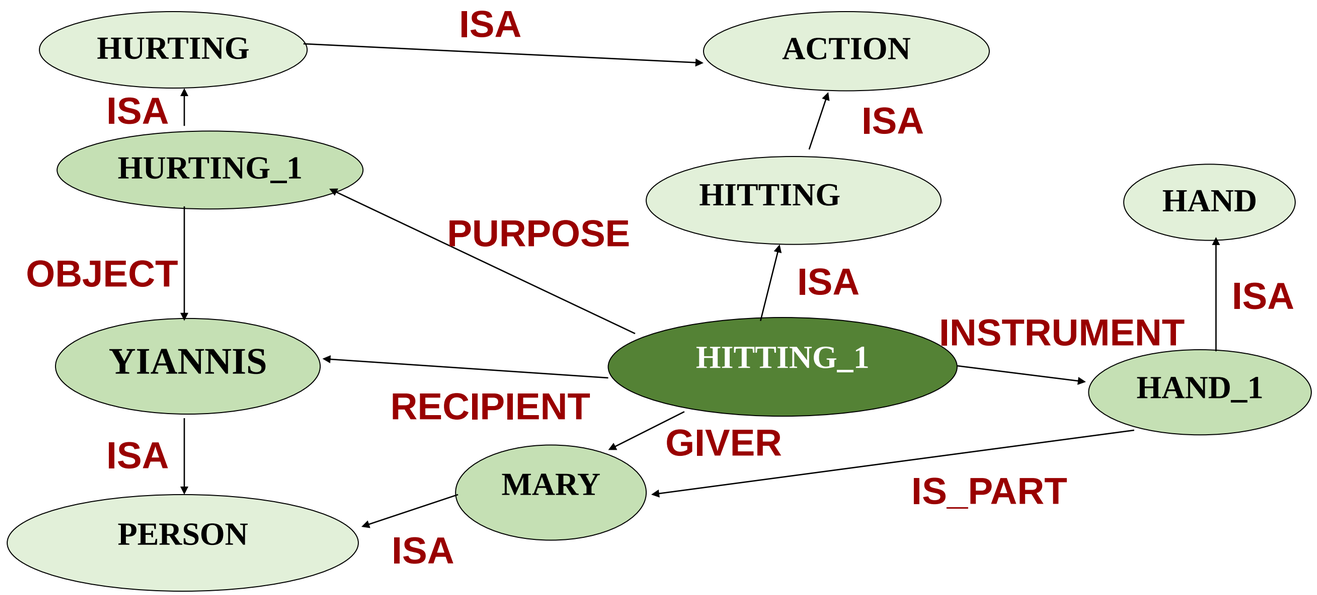
Relevant Questions:

- What did Yiannis give Mary?
- What did Mary receive from Yiannis?
- Did Mary receive something?
- Etc.



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

What is the following semantic network saying;



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Intersection Search

- Starting from the nodes representing the objects, **activation is spread** to other objects through the arcs that join them (activation is bidirectional), at the same time checking at which points the **activations intersect**
- The method exploits the key strength of semantic networks, that the knowledge organization is based on entities

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Partitioned Semantic Networks

- In the basic version of the formalism, the nodes do not have internal structure, and everything is expressed at the same level
- This version has lower logical adequacy than categorical logic
  - For example, universally quantified expressions cannot be represented

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Examples of universally quantified expressions that cannot be represented under the basic semantic networks formalism

E1: Every Computer Science student has attended a programming course.

E2: Every Computer Science student has attended the Digital Design course.

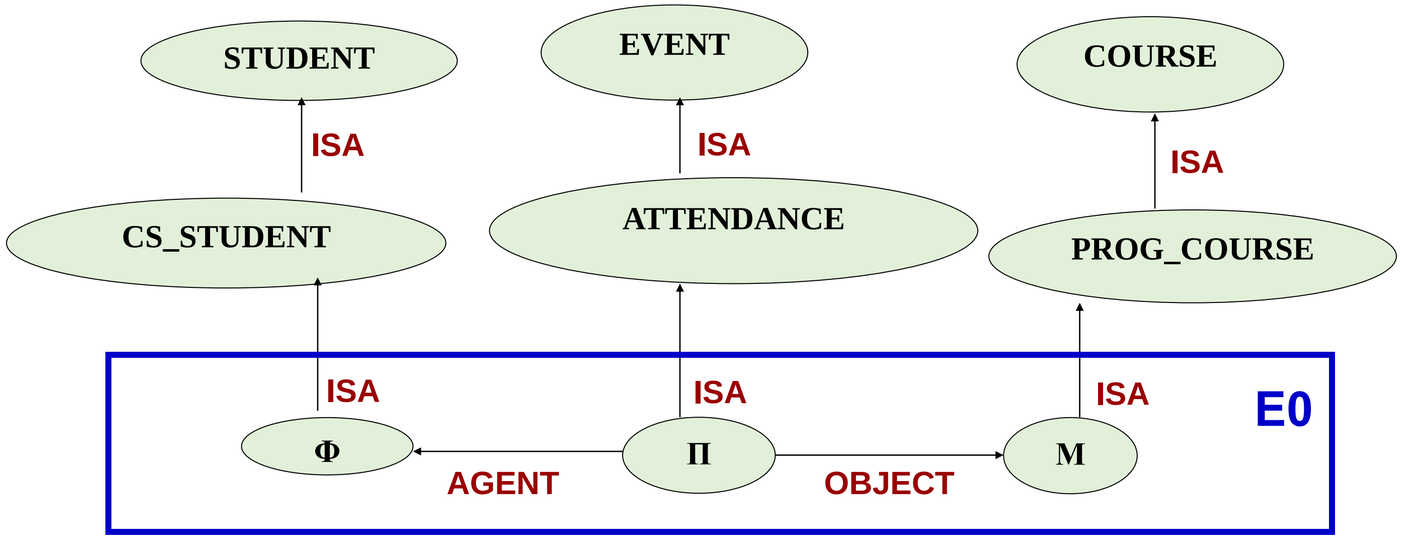
E3: Every Computer Science Student has attended all mandatory courses of his curriculum.

As such the basic formalism has been extended with the notion of a **partition**.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Sentence: ‘The Computer Science student attended the programming course’

$\exists x \exists y \; CS\_STUDENT(x) \wedge PROG\_COURSE(y) \wedge ATTENDANCE(x,y)$



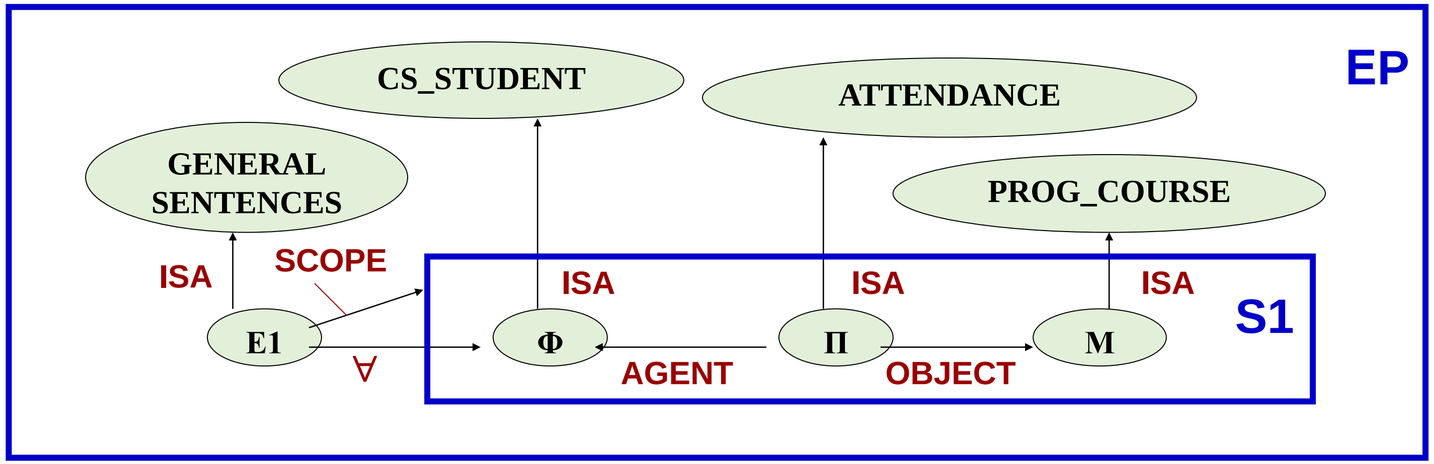
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Sentence S1: **Every Computer Science student has attended a programming course**

$$\forall x \{CS\_STUDENT(x) \Rightarrow \exists y\ PROG\_COURSE(y) \wedge ATTENDANCE(x,y)\}$$

Since variable y is in the scope x, the central predicate can be converted to

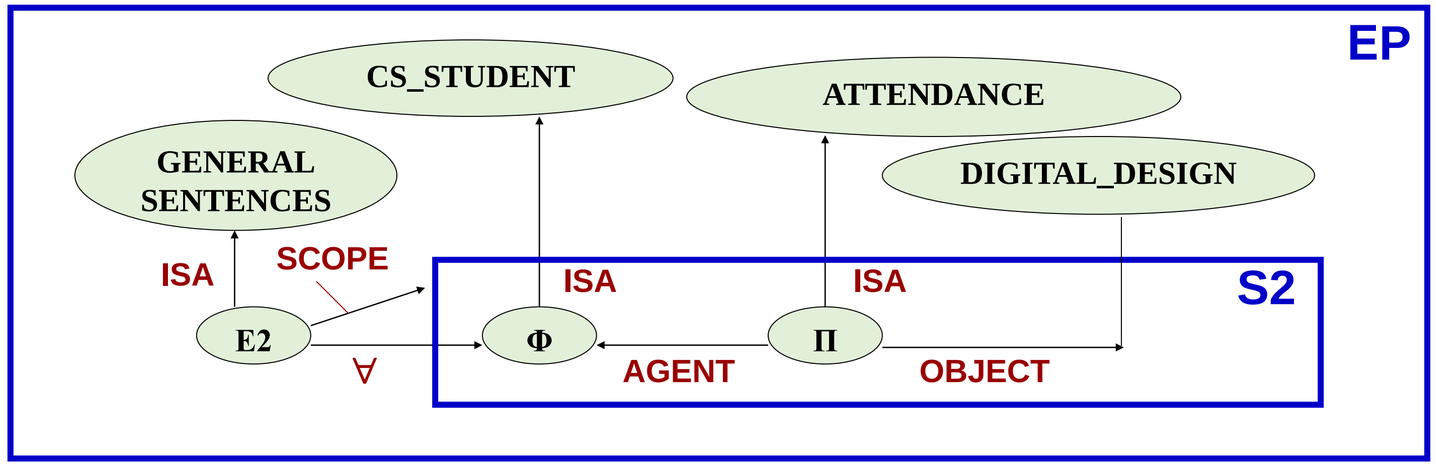
ATTENDANCE(x,f(x))



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Sentence S2: **Every Computer Science student has attended the Digital Design course**

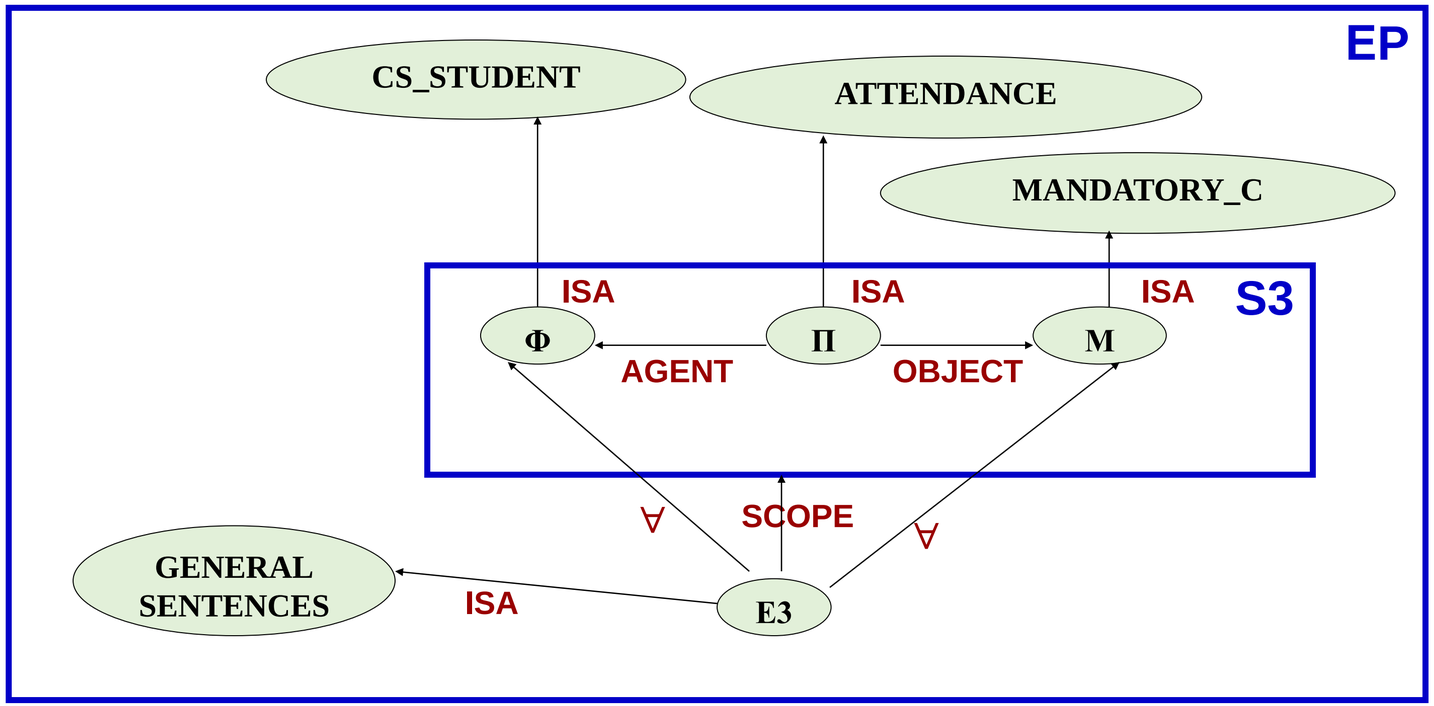
$$\forall x \{CS\_STUDENT(x) \Rightarrow ATTENDANCE(x,DIGITAL\_DESIGN)\}$$



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Sentence S3: **Every Computer Science student has attended all mandatory courses of his curriculum.**

$$\forall x \{CS\_STUDENT(x) \Rightarrow \forall y\ MANDATORY\_C(y) \wedge ATTENDANCE(x,y)\}$$



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Visibility Rules

- Access to some partition is governed by the **visibility rules**. The search finds nodes and arcs contained in the space from which the search has started or in other spaces that also contain the starting points.
- Access to some internal partition is permitted only in special cases, for example when the search traverses an arc labelled ‘SCOPE’.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Semantic Networks and Predicate Logic

- Both formalisms are descriptive (declarative), where knowledge is expressed independently of its use
- The extended version of semantic networks with partitions, provides the same power of expression as with predicate logic
- The reasoning mechanisms applied are completely different in the two formalisms
  - Semantic networks emphasize the structuring of knowledge, particularly from the perspective of the concepts of ‘hierarchy’ and ‘event’
- It can also be said that the notational convenience of semantic networks (as a more visual formalism) is higher to that of predicate logic

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Summary

- Knowledge Representation – formalisms – desirable properties
- Data – Information – Knowledge
- Knowledge types in some expertise
- Predicate Logic – syntax/semantics – normal forms – resolution refutation
- Semantic Networks – associative memory – structural elements – hierarchical structures – intersection search – partitions

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September - December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Knowledge Representation Formalisms of Frames and Production Rules

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 6

Knowledge Representation Formalisms of Frames and Production Rules

CONTENTS

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on the knowledge representation formalisms of frames and production rules, students will be able:

Regarding Frames:

1. Explain what a frame is and discuss the key structural features of this representation formalism, namely slots and facets.
2. Discuss procedural attachment and the use of active procedures or demons.
3. Explain what a frame system is, how reasoning is performed in such a system of control and object frames and how object frames can be instantiated (filling their slots).
4. Discuss inheritance and give algorithms for simple inheritance applied to strict hierarchies and multiple inheritance applied to non-strict hierarchies.
5. Analyze how the formalism of frames combines declarative and procedural representations.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on the knowledge representation formalisms of frames and production rules, students will be able:

Regarding Production Rules:

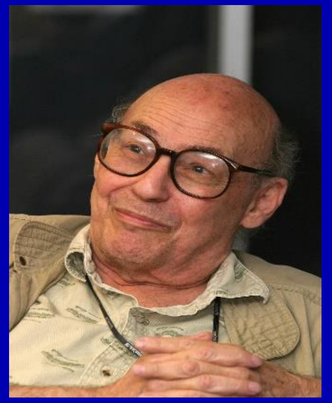
1. Explain what a production rule is and discuss the architecture of a production system in terms of its key components (production memory, working memory and control structure).
2. Present the two basic ways of applying production rules, namely forwards and backwards chaining, and outline the key concepts of the Rete algorithm for forward chaining.
3. Distinguish between chaining and reasoning (goal-driven, or backwards, and event-driven, or forwards, reasoning).
4. Discuss the control structure of a production system, present performance criteria and global strategies, and explain meta-rules for local control.
5. Explain an inference network (or AND/OR tree), the indexing of the production memory, and give algorithms for the procedures, Findout for objects and Monitor for rules that collectively implement backward chaining.
6. Discuss explanations why and how.
7. Present in more detail (than in Unit 4) the blackboard model and its underlining strategy of opportunistic search.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Frames

Marvin Minsky, A Framework for Representing Knowledge, MIT, 1974 (https://dspace.mit.edu/bitstream/handle/1721.1/6089/AIM-306.pdf?%2520sequence%3D2)



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MASSACHUSETTS INSTITUTE OF TECHNOGY
A.I. LABORATORY

June 1974

Artificial Intelligence
Memo No. 306

A FRAMEWORK FOR REPRESENTING KNOWLEDGE*

MARVIN MINSKY

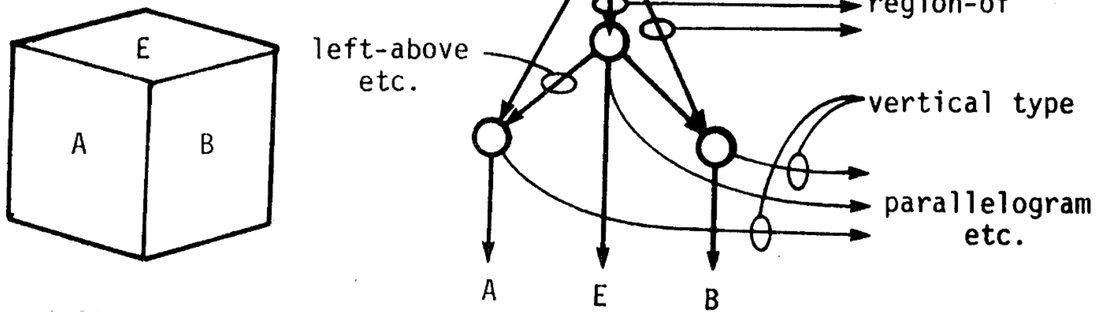
ABSTRACT

This is a partial theory of thinking, combining a number of classical and modern concepts from psychology, linguistics, and AI. Whenever one encounters a new situation (or makes a substantial change in one's viewpoint) he selects from memory a structure called a frame: a remembered framework to be adapted to fit reality by changing details as necessary.

A frame is a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed.

The "top levels" of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many "slots" that must be filled by specific instances or data. Collections of related frames are linked together into frame-systems. The effects of important actions are mirrored by transformations between the frames of a system. These are used to make certain kinds of calculations economical, to represent changes of emphasis and attention, and to account for the effectiveness of "imagery."

In Vision, the different frames of a system describe the scene from different viewpoints, and the transformations between one frame and another represent the effects of moving from place to place. Other kinds of frame-systems can represent actions, cause-effect relations, or changes in conceptual viewpoint. The paper applies the frame-system idea also to problems of linguistic understanding: memory, acquisition and retrieval of knowledge; and a variety of ways to reason by analogy and jump to conclusions based on partial similarity matching.



Frames

- The concept of **frame** is due to Marvin Minsky who formulated his proposal in relation to research in machine vision.
- A frame gives the **'totality'** of knowledge about the (abstract or concrete) entity it represents:
  - Part of the knowledge describes the entity, while another is about its use.
  - For example, a frame may describe a **stereotypical situation**, such as going to a restaurant. Part of the knowledge is related to what is going to happen when someone goes to a restaurant, as well as what to do if the expected does not happen, i.e., any exceptions and the actions involved.
- The formalism combines declarative and procedural representation.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Slots and Facets

- The key structural elements of a frame are the **slots** and their **facets**
- Slots represent the characteristics of the entity
- Slots are **“filled”** (obtain values) based on various constraints expressed in the facets
- A slot could be **single-valued** (only one of its domain values can be assigned to it) or **multi-valued** (can be assigned more than one value)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Typical Facets

❑Values

- Permitted set of values or conditions that need to be satisfied for any filler of the slot
- This is the only facet for the slots of **frame instances**
- Hence, the **frame instantiation procedure** involves the assignment of values to facet ‘values’ of the relevant slots
  - This is referred to as **‘filling’ the slots**

❑Default

- Value that can be assumed to hold in the absence of any contra-indications

❑Procedural attachments or ‘demons’

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Procedural attachments or ‘good demons’

- These facets represent reasoning knowledge in the form of procedures
- The given ‘good demons’ are activated automatically based on specific conditions:
  - **if-added**: Actions that need to take place when the slot is filled or when new information has been added to the slot
  - **if-removed**: Actions that need to take place when the filler of the slot is removed
  - **if-needed**: Procedure for computing the value of the slot. Its use is underlined by various conditions safeguarding the availability of the necessary information for the computation

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Consistency control by ‘demons’

- Demons ‘if-needed’ and ‘if-removed’ aim to maintain the **consistency** of information
- Every procedure provides control at **local level**
- The initial activation of some demon could lead to the activation of a whole chain of other such procedures involving several slots; these activations collectively achieve consistency at **global level**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: Frame for the abstract concept ‘chair’

Chair

Isa: furniture

Use:

- **Values:** for-sitting, for-standing, for-support
- **Default:** for-sitting
- **If-needed:** ask

Number-of-legs:

- **Values:** ‘up-to-four’
- **Default:** 4

Status:

- **Values:** ok, broken, partially-hidden
- **If-needed:** Number-of-legs = 4 → ok
  Number-of-legs = 1 → broken
  . . . . . .

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Reasoning with Frames

A frame is defined in order to allow reasoning about the concept it represents.

A comprehensive frame for the concept of chair could support reasoning about chairs such as:

- **Walking into a room, which objects are chairs?**
- **Can someone climb on a chair to reach a high shelf?**
- **When can one sit on a chair? when in a museum? when in a dollhouse? when already in use?**
- **Etc.**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

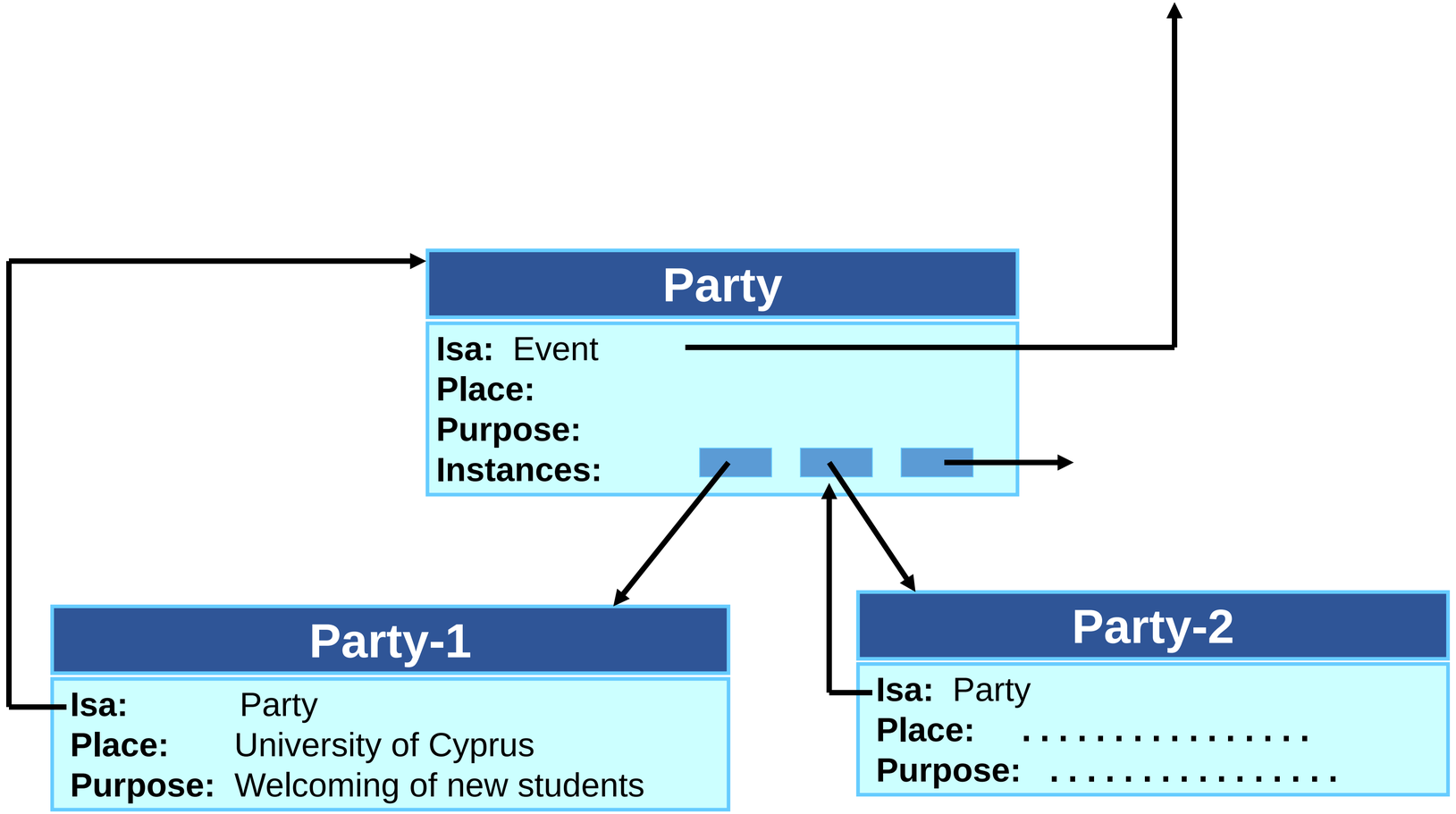
Key Question: **Can frames be formulated 'completely' in advance, to support common-sense reasoning or specialized reasoning?**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Frame Instantiation

- The **instantiation** of a frame that represents an abstract concept, refers to the filling of the slots to create specific cases (**instances**) of the concept
- The specific frames have the general frame as their direct predecessor in the hierarchy

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Frame System

- In a **frame system**, the different frames are interrelated in various ways
- Frames fall into two general categories
  - **Object frames**
  - **Control frames** - define how other frames are used

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Frame System

A frame system is an organized base of control frames and object frames, for a given purpose.

It is a search space, where the goal is to create instances of object frames that collectively provide the solution to the problem.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Critical Points of Reasoning Process:

- The initial focus on the search space
- Further navigation in this space especially if the expectations do not correspond to reality

Based on some incomplete, uncertain and vague information, an attempt is made to identify an external condition, whether it refers to a visible object, such as a chair, or to some non-visible condition, such as the internal dysfunction of a patient.

A basic mechanism is the matching of expectations, as they are formulated in the relevant frame, with reality, as it appears (partially) through incomplete information.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

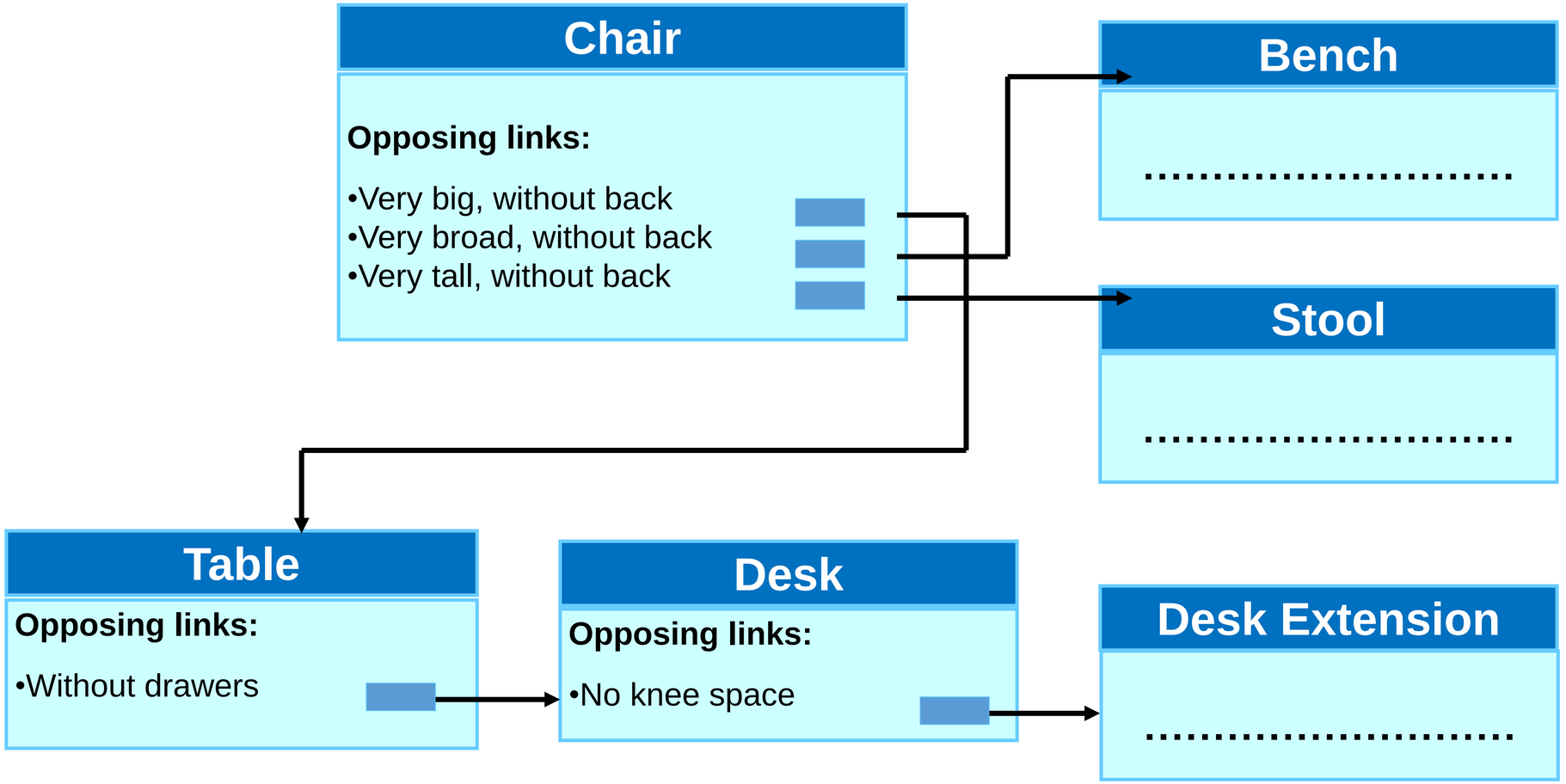
Links between object frames

- **Hierarchical links**
  - relations ‘isa’ and ‘is_part’
- **Opposing links**
  - identify differences between frames with many common characteristics, that as a result of this similarity, the existence of one may be mistaken for the existence of the other
  - The subnet created by the set of opposing links is called the **similarity matrix**
- **Complementary links**
  - they connect one frame to others that represent concepts, complementary to the concept of the first
- **Triggering links**
  - they suggest which frames to activate, by and large initially; these links are usually contained in control frames, and hence they are context-independent, that is, they operate at **global level**

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example Similarity Matrix



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Inheritance

What is the value of characteristic S of object frame instance F?

Inheritance algorithms are used by the frame instantiation process.

Inheritance allows the flow of information from categories to instances. This flow is not monotonic.

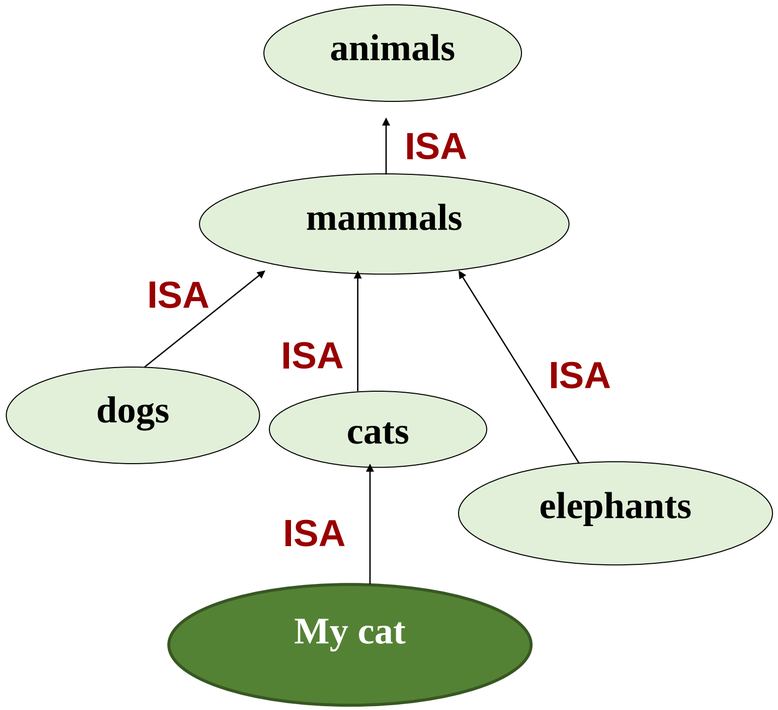
1. **Simple inheritance – strict taxonomies**

2. **Multiple inheritance – a frame can have multiple direct predecessors**

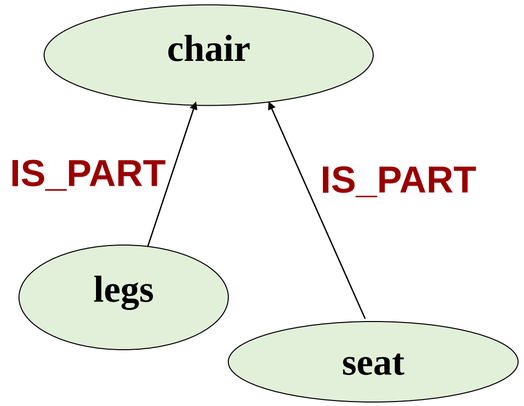
Relevant facets: 'values', 'if-needed', and 'default'

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Taxonomy



Meronomy



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Taxonomies and Meronomies of Concepts

- A **taxonomy** is an organization of categories and subcategories
  - 'ISA' (is-a-kind-of (isa))
- A **meronomy** is an organization of constituent parts/components
  - 'IS_PART' (is_part)
- Relations 'ISA' and 'IS_PART' are transitory and hence the hierarchies formed from these relations are hereditary (support the **inheritance** of properties)
- The top-down flow of information, in some taxonomy, is **not monotonic,** i.e., exceptions should be supported

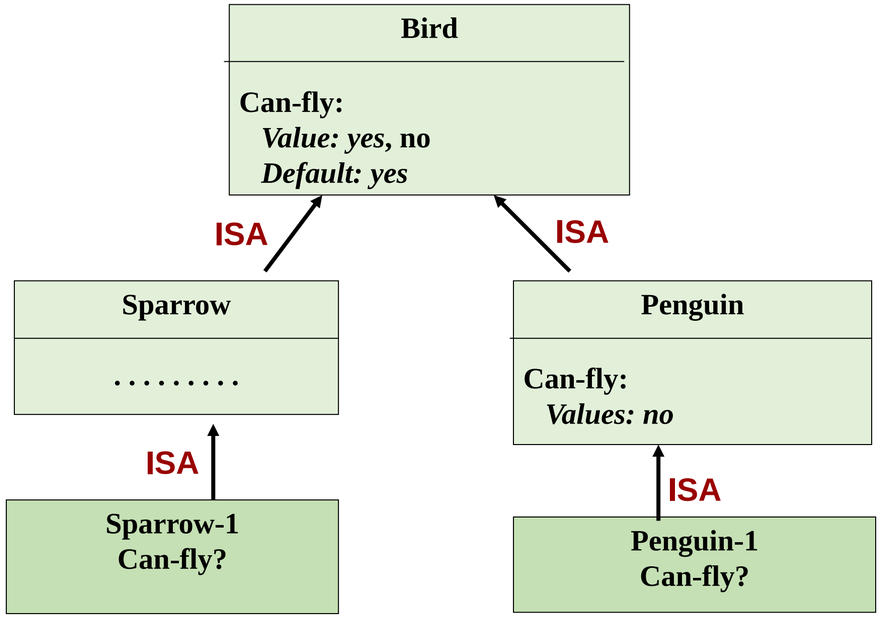
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example

- if all wedges are triangular, it can be concluded that any specific wedge is triangular.
- If we know that the volume of a cube is V and the density of its matter is d, then its weight can be calculated as the product of Vd.
- It can be reasonably assumed that a particular toy cube is made of wood, because most such cubes are made of wood, and that the purpose of a cube is entertainment if seen as a toy, or support if seen as a structure.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

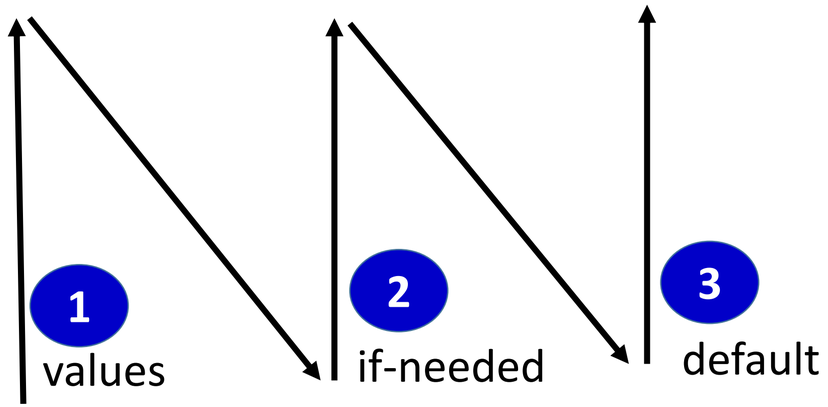
Simple Inheritance



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

N-Inheritance Algorithm

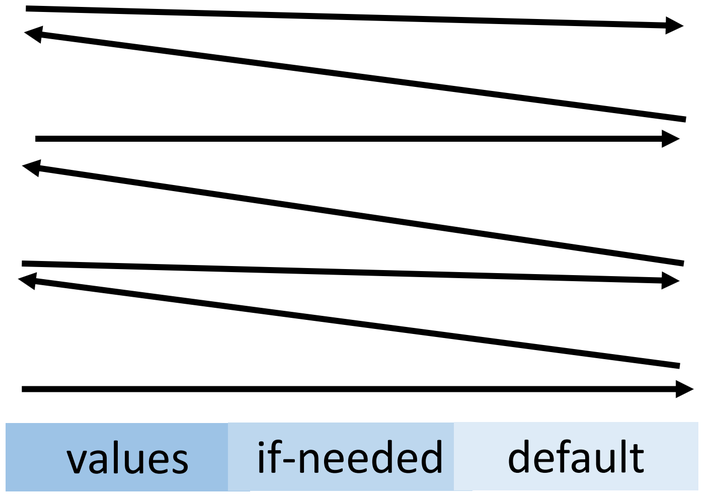
1. Starting from the frame representing object F, move upwards using the ‘ISA’ chain of links. In every frame visited check if there is a **‘values’** facet for characteristic S, that gives a categorical value. If yes, stop.
2. If step 1 failed, repeat the upwards traversing but this time examine the **‘if-needed’** facet.
3. Finally, if step 2 failed, then repeat examining the **‘default’** facet.



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Z-Inheritance Algorithm

Starting from the frame representing object F, move upwards using the 'ISA' chain of links. In each frame visited examine the facets 'values', 'if-needed', and 'default', of slot S, in this order. If a value is obtained for the slot stop, otherwise proceed to the immediately above frame.



Question:Which of the two algorithms is better?

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Remarks

- Z-inheritance is based on the heuristic that the closest answer is the correct one, even if it comes from a default, while N-inheritance gives the highest weight to facet ‘values’ (and the least weight to facet ‘default’), even if the value would come from a frame at a great distance from the starting frame.

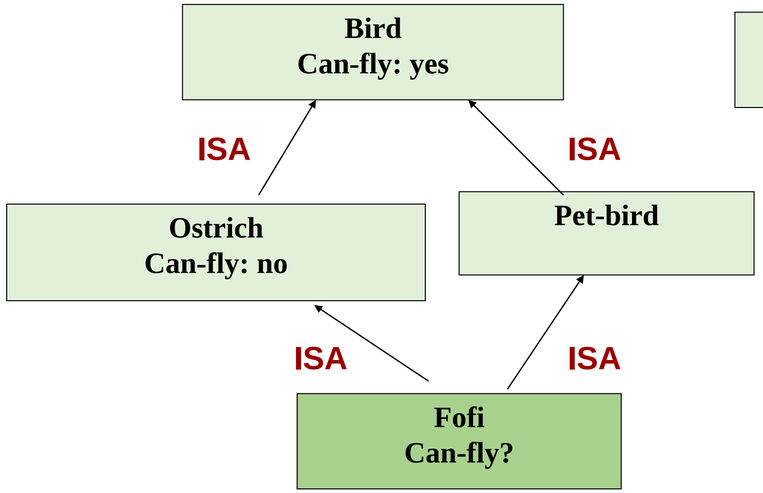
- The examination of facet ‘if-needed’ could lead to a recursive call of the algorithm.

- In the worst case, that is the case where either no value can be derived, or the value comes from the default facet of the highest frame, the two algorithms have the same complexity, provided that the ascent cost is zero.

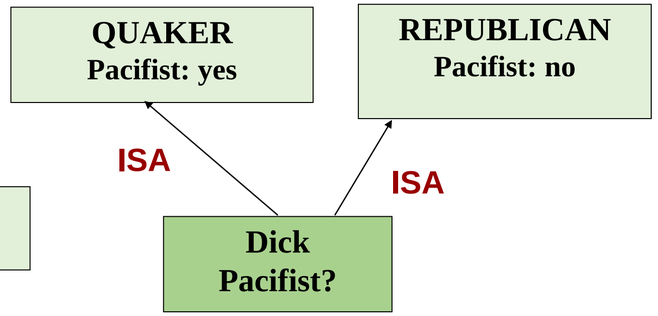
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Multiple Inheritance

In multiple inheritance there could be contradictory suggestions as to the value of characteristic S of frame F.



(a)



(b)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

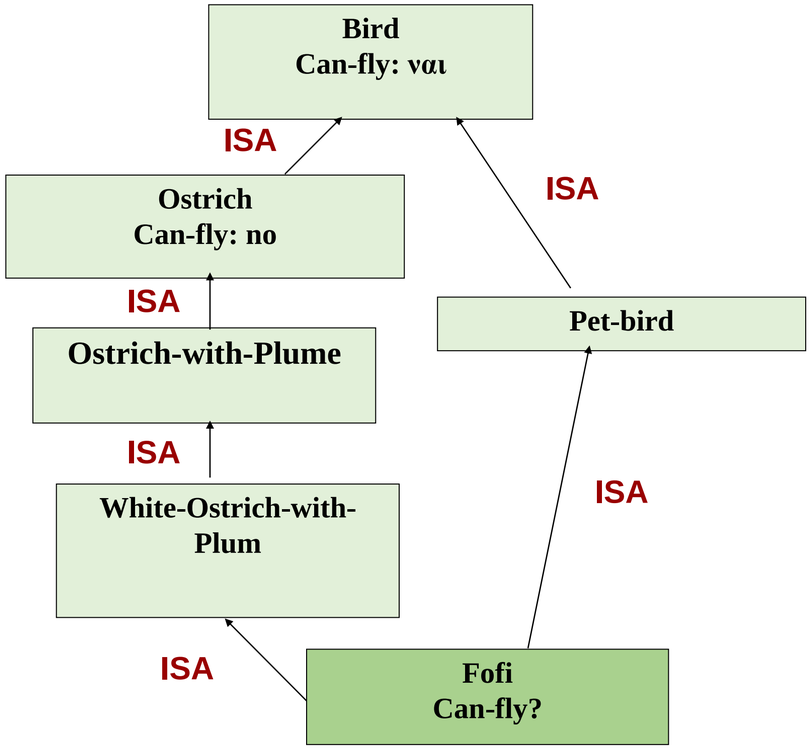
Simple Algorithm

Select the answer that comes from the nearest frame.

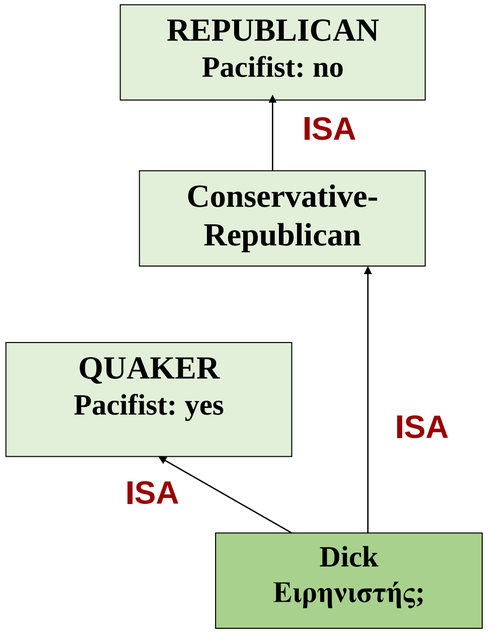
If there is just one value, this is the answer.

If not, the contradiction cannot be resolved.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423



(a)



(b)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

The value is chosen based on the inferential distance and not the path distance.

Inferential Distance

Category C_1 is nearer (reasoning-wise) to category C_2 in relation to category C_3 if and only if C_1 has a reasoning path to C_3 through C_2.

This relation defines a partial order.

Fofi is nearer to ostrich in relation to bird since there is a route from Fofi to bird through ostrich.

However, the relation is not defined for the triplet Dick, Quaker, and Republican.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Set of Conflicting Values for Characteristic S of Object F

Consists of:

- **Values coming from hierarchically higher frames, X.**
- **They are not in contradiction with values coming from some frame, Y, which is at a smaller inferential distance from frame F in relation to frame X.**

The set of conflicting values for characteristic ‘Can-fly’ of object Fofi is {no} while for the characteristic ‘Pacifist’ of object Dick is {yes, no}.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Multiple Inheritance Algorithm based on Inferential Distance

1. CANDIDATES ← [ ]
2. Apply breath-first (or depth-first) search starting from frame F and following upwards all hierarchical links. At every step examine whether there is a value for characteristic S:

   (a) If there is a value, add it in CANDIDATES and terminate the given route

   (b) If there is no value, but there are upwards ‘ISA’ links then follow them

   (c) Otherwise terminate the particular route

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Continuation of Multiple Inheritance Algorithm based on Inferential Distance

3. For every element Y of CANDIDATES:

(a) Examine whether there is another element of CANDIDATES coming from a category which is nearer (reasoning-wise) to F than the category from where Y came

(b) If yes delete Y from CANDIDATES

4. Examine the number of elements of CANDIDATES:

(a) If it is 0, no value has been found
(b) If it is 1, then return the single candidate as the sought value
(c) Otherwise note the contradiction

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Production Rules

The simplest knowledge representation formalism

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on the knowledge representation formalisms of frames and production rules, students will be able:

Regarding Production Rules:

1. Explain what a production rule is and discuss the architecture of a production system in terms of its key components (production memory, working memory and control structure).
2. Present the two basic ways of applying production rules, namely forwards and backwards chaining, and outline the key concepts of the Rete algorithm for forward chaining.
3. Distinguish between chaining and reasoning (goal-driven, or backwards, and event-driven, or forwards, reasoning).
4. Discuss the control structure of a production system, present performance criteria and global strategies, and explain meta-rules for local control.
5. Explain an inference network (or AND/OR tree), the indexing of the production memory, and give algorithms for the procedures, Findout for objects and Monitor for rules that collectively implement backward chaining.
6. Discuss explanations why and how.
7. Present in more detail (than in Unit 4) the blackboard model and its underlining strategy of opportunistic search.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

A central postulate of Artificial Intelligence is that intelligent behavior is governed by rules

If (premise) then (conclusion), or

If (condition) τότε (action)

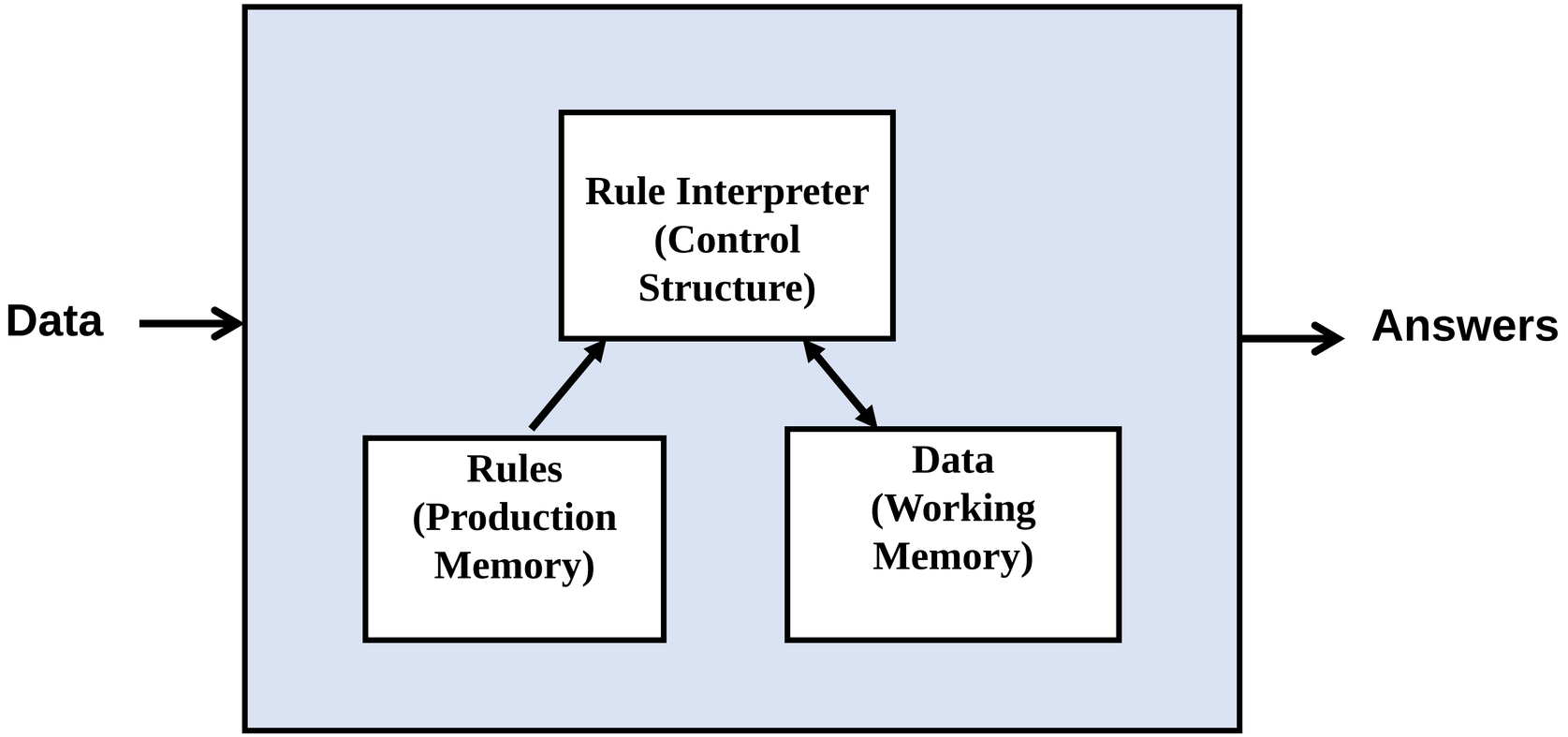
Production System

A production system represents its knowledge as rules.

Its knowledge base is called production memory.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Production System Architecture



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Production Rules

$P_1, \ldots, P_m \rightarrow Q_1, \ldots, Q_n$, where $m, n \geq 1$.

Semantics: If the premises (conditions) $P_1, \ldots, P_m$, are true in the specific contexts, then the conclusions (actions), $Q_1, .., Q_n$, can be derived (executed))

The sentences P_i and Q_j are **symbol structures**, usually triplets of the form:

(⟨object⟩ ⟨attribute⟩ ⟨value⟩)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Dictionary of Rules

1. set **A** of object names
2. set **X** of property names that bestow attributes to objects
3. set **T** of attribute values

Usually sets A and T intersect in order to be able to express relationships between objects, e.g.:

(Knowledge-Representation subunit Artificial-Intelligence)

Where symbols ‘Knowledge-Representation’ and ‘Artificial-Intelligence’ belong to both sets A and T, while symbol ‘subunit’ belongs to set X. ‘

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Several atomic sentences referring to the same object could be merged, e.g.:

$$(\alpha, \chi_1, \tau_1), (\alpha, \chi_2, \tau_2), \ldots, (\alpha, \chi_n, \tau_n)$$

$$\Downarrow$$

$$(\alpha, \chi_1, \tau_1, \chi_2, \tau_2, \ldots, \chi_n, \tau_n)$$

OPS5: **(Production-Rules ^author Keravnou ^subunit AI-Fundamentals)**

CLIPS: **(Production-Rules (author Keravnou) (subunit AI-Fundamentals))**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Rule Syntax

(rule ⟨rule-name⟩
⟨premise-1⟩ . . . . . ⟨premise-n⟩
→ ⟨action-1⟩ . . . . . ⟨action-m⟩)

Example

(rule under-probation
(Student ^name Smith ^semester-average 4.8)
→ (add Place ^name Smith ^under-probation yes))

This formulation does not give the rule, but an instantiation of it. The proper formulation of the rule uses variables.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
(rule under-probation
        (Student  ^name  ?O  ^semester-average  ?A)
        (:less-than  ?A  5.0)
                → (add  Place  ^name  ?O ^under-probation yes))
```

1. The scope of each variable is the given rule. Every appearance of the same variable refers to the same value.
2. Moreover, premises could involve external, computable predicates, such as less-than. These premises are ‘computed’, they are not matched against specific symbolic expressions.
3. The binding of variables with values is done by a pattern matching process.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Let’s assume the following data:

D1: (Student ^name Smith ^semester-average 4.8)
D2: (Student ^name Jones ^semester-average 7.2)
D3: (Student ^name Adams ^semester-average 4.7)

The symbolic expression in the first premise of rule ‘under-probation’ matches against each of the above data.

However, the second, computable, premise is verified only in relation to data D1 and D3..

Hence, in this case there are only two instantiations of the given rule:

1. **In relation to datum D1 where the bindings are**
   ?O = Smith και ?A = 4.8
2. **In relation to datum D3 where the bindings are**
   ?O = Adams και ?A = 4.7

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Working Memory

Data D1-D3 are stored in the working memory, a global data base.

The application of the above rule instantiations would result in the addition of the following data in the working memory:

D4: (Place ^name Smith ^under-probation yes)
D5: (Place ^name Adams ^under-probation yes)

However, the reasoning in a production system, is not entirely monotonic as it is the case with predicate logic.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Basic ways of applying rules

- Forwards chaining
- Backwards chaining

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Forwards Chaining

- The symbolic expressions in the rule premises are matched against the symbolic expressions in the working memory
- The rule instantiations thus derived constitute a **conflict set**, out of which one instantiation is selected and applied, i.e., its action is executed
- The **recognize-act** cycle continues until either there is no rule instantiation (the conflict set is empty) or none of the rule instantiations can lead to progress (modifications or additions to the working memory)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Backwards Chaining

- The rules are applied in the opposite direction, i.e., from actions/conclusions to premises
- Initially some action/goal-conclusion is set
- The rules whose right-hand sides match against the goal constitute a conflict set, from which one rule is then selected whose premises in turn become subgoals to be pursued
- This way an **inference tree** is recursively formed

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Reasoning versus Chaining

- **Goal-driven reasoning**, otherwise known as backwards reasoning
- **Event-driven reasoning**, otherwise known as forwards reasoning
- The form of reasoning is at the level of **design**, while the form of chaining is at the level of **implementation**
  - Reasoning is implemented through chaining, but not the opposite
- The languages OPS5 and CLIPS provide only forwards chaining, but both forms of reasoning (goal-driven or event-driven) can be implemented

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Deductive Reasoning versus Abductive Reasoning

- Deductive reasoning leads to conclusions in a **categorical** way, e.g., the reasoning in predicate logic

- Deductive reasoning is **hypothetical reasoning**; It leads to possible hypotheses for further investigation

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Critical point of both chaining methods

- The **selection** of the next rule (instantiation) to be applied
- The selection is guided by various heuristics, in the form of
  - General control strategies
  - Meta-rules
- **Resolving the conflict set**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Forwards Chaining - ‘Recognize-Act’ cycle

Example (continuation)

Dismissal Rules:

1. **Second failure on a mandatory course**
2. **Failure on more than two mandatory courses in the same semester.**
3. **Under-probation for two consecutive semesters.**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Rule Coding

```
(rule dismissal-1
        (Failure ^name ?O ^course ?C ^semester ?S1)
        (Failure ^name ?O ^course ?C ^semester ?S2)
        (:different ?S1 ?S2)
        (Mandatory ^course ?C ^for ?O)
      → (add Potential-Dismissal ^for ?O
                  ^reason second-failure-on-mandatory))
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
(rule dismissal-2
        (Failure ^name ?O ^course ?C1 ^semester ?S)
        (Failure ^name ?O ^course ?C2 ^semester ?S)
        (Failure ^name ?O ^course ?C3 ^semester ?S)
        (:all-different ?C1 ?C2 ?C3)
        (Mandatory ^course ?C1 ^for ?O)
        (Mandatory ^course ?C2 ^for ?O)
        (Mandatory ^course ?C3 ^for ?O)
        → (add Potential-Dismissal ^for ?O
                    ^reason failure-on-2plus-mandatory))
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

```
(rule dismissal-3
        (Under-probation  ^name  ?O  ^semester  ?S1)
        (Under-probation  ^name  ?O  ^semester  ?S2)
        (:consecutive  ?S1  ?S2)
      → (add Potential-Dismissal  ^for  ?O
                                  ^reason  2-consecutive-under-dismissal))
```

Remark: Each rule is expressed as an independent, self-explanatory entity. Reasoning, however, is achieved based on the indirect correlations of the rules.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Other Rules

```
(rule course-failure
        (Attended ^name ?O ^course ?C ^semester ?S ^mark ?M)
        (:less-than ?M 5.0)
        → (add Failure ^name ?O ^course ?C ^semester ?S)
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Modification of initial ‘under-probation’ rule

```
(rule under-probation
        (Semester-Average  ^name  ?O  ^semester ?S  ^average  ?A)
        (:less-than  ?A 5.0)
        → (add Under-Probation  ^name  ?O  ^semester  ?S))
```

data abstraction

Remark: Each rule needs to be properly ‘integrated’ with the other rules.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Algorithm for Forwards Chaining
(‘Recognize-Act’ Cycle)

Repeat

1. **RINST → all the rule instantiations, the premises of which are verified against the contents of the working memory**
2. **Select, based on the heuristics, R ∈ RINST for execution**
3. **If a rule instantiation R has been selected, execute its action in relation to the working memory**

Until either no rule instantiation R can be selected, or the action of the selected R implies termination of the process

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Rule Application

Initial Working Memory Contents

D1: (Mandatory ^course CS300 ^for Jones)

D2: (Mandatory ^course CS301 ^for Jones)

D3: (Mandatory ^course CS302 ^for Jones)

D4: (Attended ^name Jones ^course CS300 ^semester 1 ^mark 4.5)

D5: (Attended ^name Jones ^course CS300 ^semester 3 ^mark 4.0)

D6: (Attended ^name Jones ^course CS301 ^semester 3 ^mark 3.5)

D7: (Attended ^name Jones ^course CS302 ^semester 3 ^mark 4.0)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

First cycle

The conflict set contains four instantiations of rule ‘course-failure’, say R1-R4, respectively in relation to data D4-D7.

Let’s assume that instantiation R1 is selected, whose execution adds the following new datum (conclusion) to the working memory:

D8: (Failure ^name Jones ^course CS300 ^semester 1)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Second cycle

The conflict set consists of the remaining three rule instantiations, R2-R4, while rules ‘dismissal-1’ and ‘dismissal-2’ have been matched partly due to the new datum, D8.

Let’s assume that instantiation R2 is selected, whose execution adds the following new datum (conclusion) to the working memory:

D9: (Failure ^name Jones ^course CS300 ^semester 3)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Third cycle

A new rule instantiation, say R5, is added in the conflict set, referring to rule ‘dismissal-1’ in connection with the data D1, D8 and D9.

Rule ‘dismissal-2’ continues to be partly matched.

Complete the execution of the forwards chaining algorithm

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Control Structure

- In every cycle, the selection of the rule instantiation to be executed next, is governed by various heuristics that constitute the **control structure** of the rule interpreter
- General performance criteria
  - **Sensitivity**:
    - Ability to respond quicky to changes in the environment as these are depicted in the working memory
  - **Stability**:
    - Ability to demonstrate coherence in the line of reasoning

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Global Control Strategies

- Refractoriness
- Recency
- Specificity

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Refractoriness Control Strategy

- The same rule instantiation can be executed only once
- In monotonic production systems it would not make sense to execute the same rule instantiation more than once
- Weaker version: Only the rule instantiation executed at the immediately preceding cycle to be excluded

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Recency Control Strategy

- Every datum in the working memory is associated with its time of entrance
- Rule instantiations that concern more recent data are preferrable to those that concern older data
- Weakness: Older data can be indefinitely ignored, unless the current line of reasoning fails

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Specificity Control Strategy

- Rule instantiations that have more premises are to be preferred
- The rationale being that such rules take into consideration more of the data
- The use of this strategy facilitates the handling of exceptions, e.g.:
  - General rule: ‘If X is a bird, then X can fly’
  - Rule exceptions:
    - ‘If X is a bird and X is a penguin, then X cannot fly’
    - ‘If X is a bird and X is an ostrich, then X cannot fly’
    - ‘If X is a bird and X has a broken wing, then X cannot fly’
    - Etc.

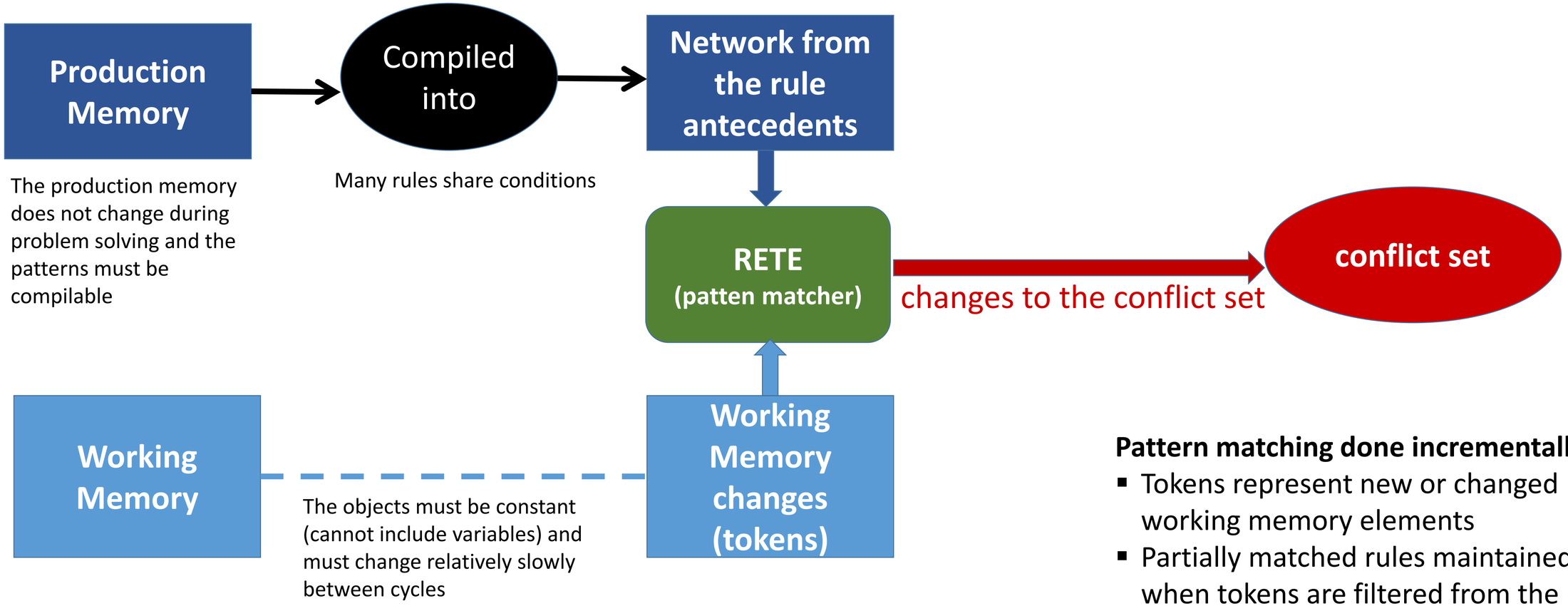
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Rete: a fast algorithm for the many pattern/many object pattern match problem
(http://www.csl.sri.com/users/mwfong/Technical/RETE%20Match%20Algorithm%20-%20Forgy%20OCR.pdf)

- Charles L. Forgy, Artificial Intelligence Journal, 1982
- Incorporated in the OPS5 programming language
- The algorithm and various improvements thereon, are a key component of forward-chaining production systems
  - E.g., the XCON system (originally called R1) contained several thousand rules for designing configurations of computer components for customers of the Digital Equipment Corporations; it was one of the first clear commercial successes in the field of expert systems
  - Many other similar systems have been built and implemented in the general-purpose language OPS5
- **Objective**: How to compute fast, at each cycle, the **conflict set**, i.e., the applicable rule instantiations
  - Even when sophisticated indexing and hashing was used, still 90% of the computation time was used in rule matching

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Rete Algorithm – Key Concepts



Pattern matching done incrementally

- Tokens represent new or changed working memory elements
- Partially matched rules maintained when tokens are filtered from the top of the network downwards

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Overview of Rete Algorithm

From R.J. Brachman and H.J. Levesque, Knowledge Representation and Reasoning, Elsevier, 2004

- “Tokens” are passed incrementally through the network of tests (rule conditions)
- Tokens that make it all the way through the network on any given cycle are considered to satisfy all the conditions of a rule
- At each cycle, a new conflict set can then be calculated from the previous one and any incremental changes made to working memory
- This way, only a very small part of the working memory is rematched against any rule conditions, drastically reducing the time needed to calculate the conflict set

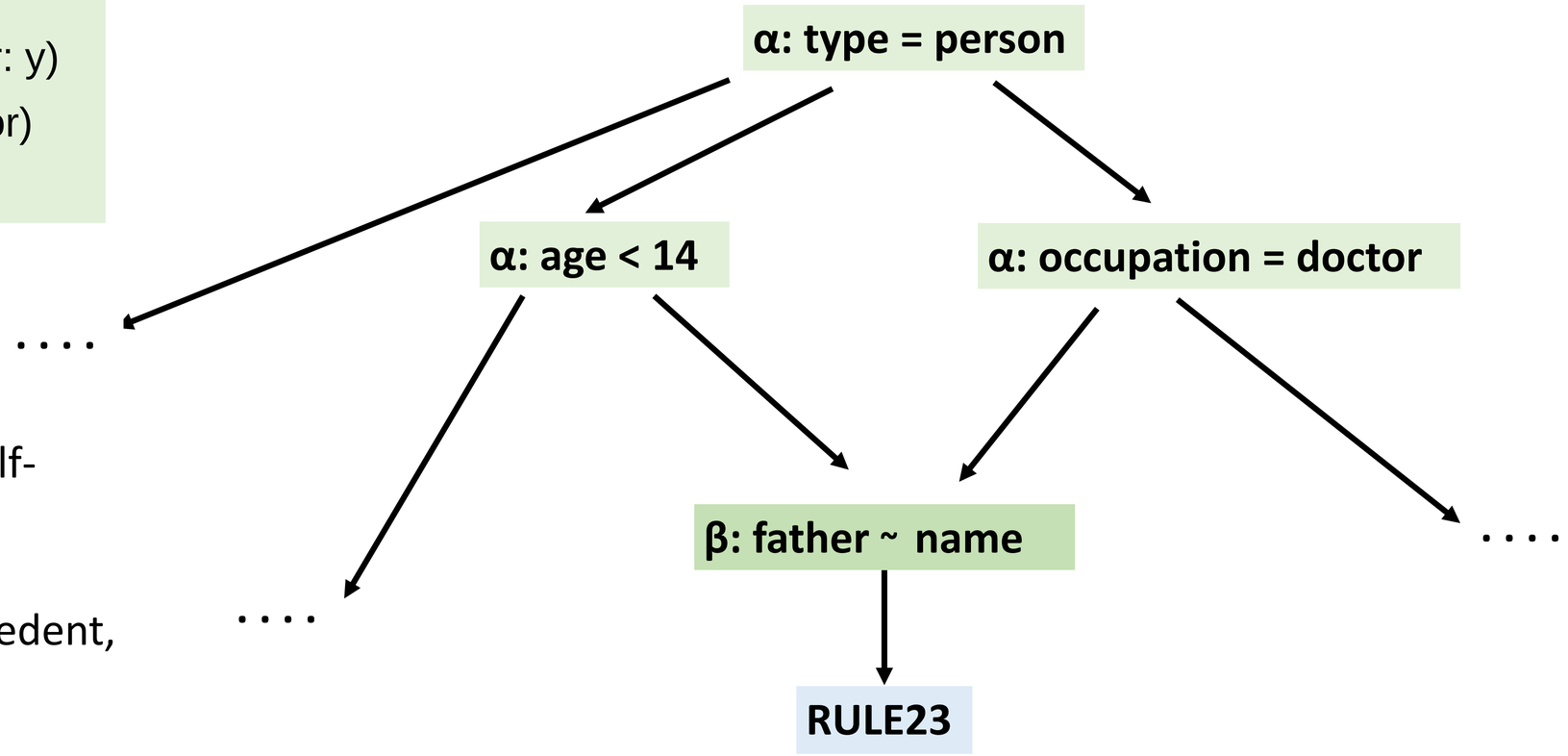
- Thus, at any given point, the state of a Rete network captures all the partial matches of the rules, avoiding a great deal of re-computation

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example: A sample Rete Network

From R.J. Brachman and H.J. Levesque, Knowledge Representation and Reasoning, Elsevier, 2004

RULE23
IF (person name: x age: {< 14} father: y)
(person name: y occupation: doctor)
THEN . . . .



alpha node: represents a simple, self-contained test
beta node: represents a constraint between different parts of an antecedent, emanating from variables

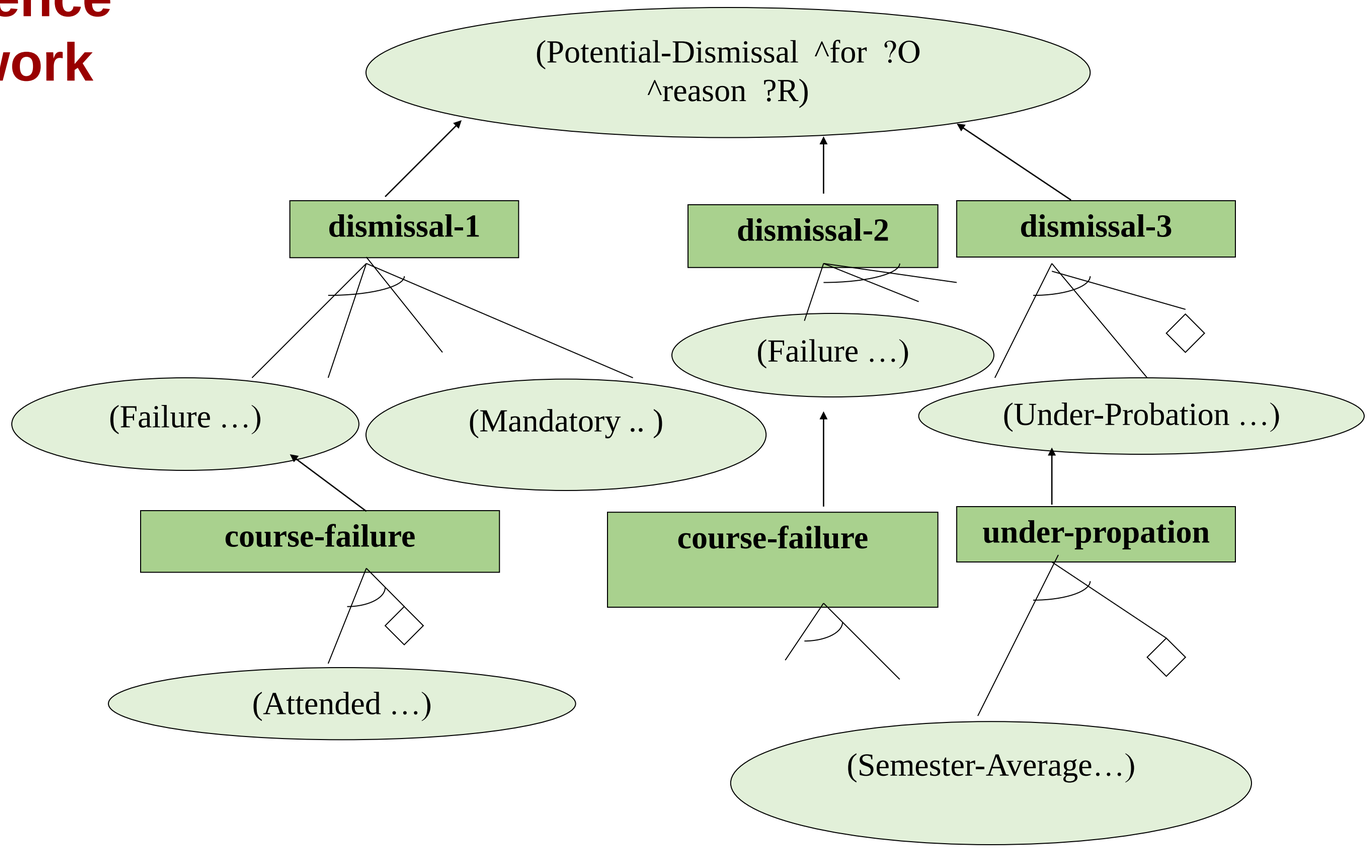
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Backwards Chaining

- The rules discussed above of a system monitoring the performance of students, could have been applied through backwards chaining
- **Inference Network**
  - Or AND/OR tree or goal tree
  - Follows from the indirect connections of the rules
  - The symbolic expressions constitute goals and subgoals

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Inference Network



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example: Production System that advises on the best way to go to a given theater.

Production Memory

```
(rule  r1
    (Distance  ^value  greater-than-5-miles)
    → (add Means  ^value  driving))

(rule  r2
    (Distance  ^value  greater-than-1-mile)
    (Time  ^value  less-than-15-minutes)
    → (add Means  ^value  driving))

(rule  r3
    (Distance  ^value  greater-than-1-mile)
    (Time  ^value more-than-15-minutes)
    → (add Means  ^value walking))
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Production Memory (cont.)

```
(rule  r4
    (Means  ^value  driving)
    (Location  ^value  center)
    → (add Action   ^value  get-taxi))

(rule  r5
     (Means  ^value  driving)
     (Location  ^value  outside-center)
     → (add  Action  ^value drive-your-car))

 (rule  r6
     (Means  ^value  walking)
     (Weather  ^value  bad)
     → (add  Action  ^value take-coat-and-walk))

(rule r7
     (Means  ^value  walking)
     (Weather  ^value  good)
    → (add Action  ^value walk))
```


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Remarks

The rules do not contain any variables and thus they are specific.

Primitive objects: ‘Distance’, ‘Time’, ‘Location’ and ‘Weather’
They are considered *‘askable’* objects

Object which is the ultimate goal: ‘Action’

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Production Memory Indexing

- Based on the appearance of the objects in the rules, the production memory can be indexed in two dimensions:
  - **Look-ahead** dimension
    - Which rules mention the object in their premises
    - It concerns forwards chaining
  - **Updated-by** dimension
    - Which rules mention the object in their conclusions
    - It concerns backwards chaining
- The information on each object, including its look-ahead and updated-by rules could be represented either collectively in a table or as separate frames for each object

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Object Table

Object	Askable?	Question	Values	Look-ahead	Updated-by
Distance	Yes	How far is the theater?	greater-than-1-mile, greater-than-5-miles	r1, r2, r3	
Time	Yes	How much time do you have?	more-than-15-minutes, less-than-15-minutes	r2, r3	
Location	Yes	Where is the theater?	center, outside-center	r4, r5	
Weather	Yes	How is the weather?	good, bad	r6, r7	
Means	No	What means do you have?	walking, driving	r4, r5, r6, r7	r1, r2, r3
Action	No		get-taxi, drive-your-car, take-coat-and-walk, walk		κ4, κ5, κ6, κ7

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Findout and Monitor

- In backward chaining, the rule interpreter has two basic procedures that exhibit chain recursion:
  - **Findout for objects**, and
  - **Monitor for rules**
- These two procedures were invented by Edward Shortliffe in connection to the expert system MYCIN that he implemented, and which is considered one of the founding expert systems.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Findout for Objects

Input: Object A

1. Is A askable?

2. If not, then:

   2.1 U ← rules in the updated-by list for A
   2.2 Call Monitor for each R∈ U
   2.3 If no value is derived for A, then ask the user

3. Otherwise

   3.1 Ask the user
   3.2 If the user does not know the value, then
   U ← rules in the updated-by list for A
   Call Monitor for each R ∈ U

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Monitor for Rules

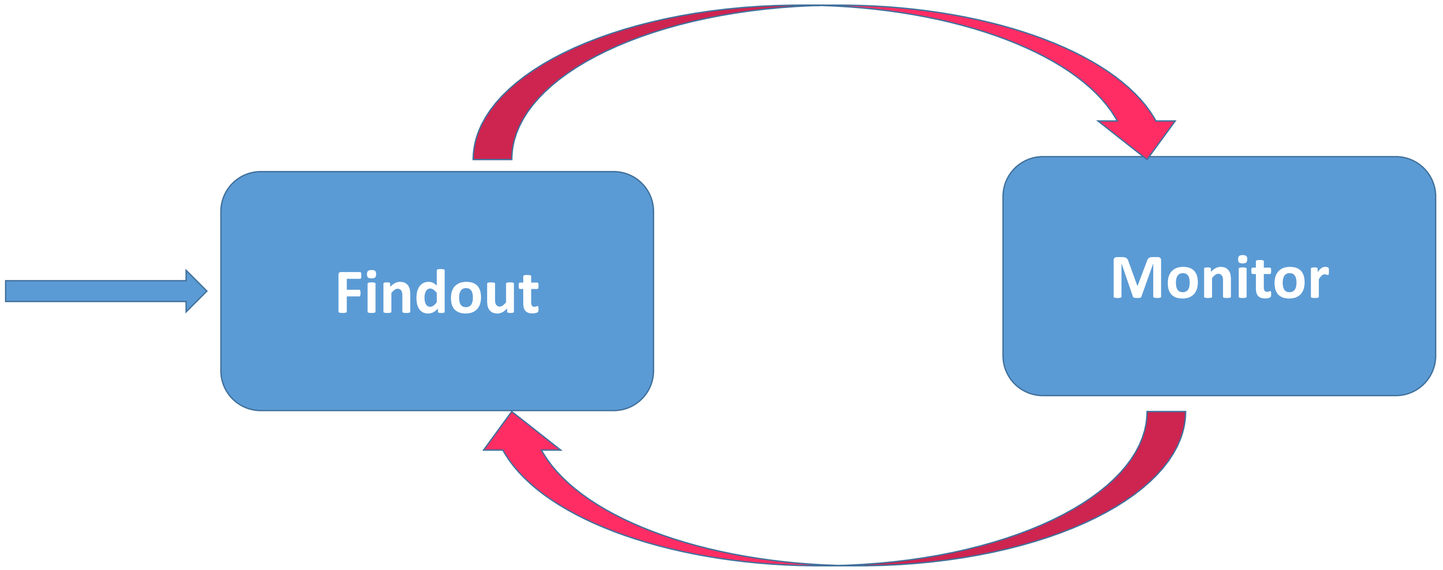
Input: Rule R

1. P ← first condition in the premise of R
2. Does the working memory include all information on P?
3. If yes, and P is verified, then
   - 3.1 If there are other conditions in R's premise, then
     P ← next condition, and
     return to 2
   - 3.2 Otherwise apply R's action to the working memory
4. If yes, and P is either false or unknown, then reject rule R
5. Otherwise solicit the relevant information regarding P's object through the Findout

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Findout and Monitor

- Findout is called initially with input the object of the ultimate goal, e.g., ‘Action’ in the theater example
- In fact, the input to Findout refers to some attribute of the relevant object



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Conversation Example

Findout: Action

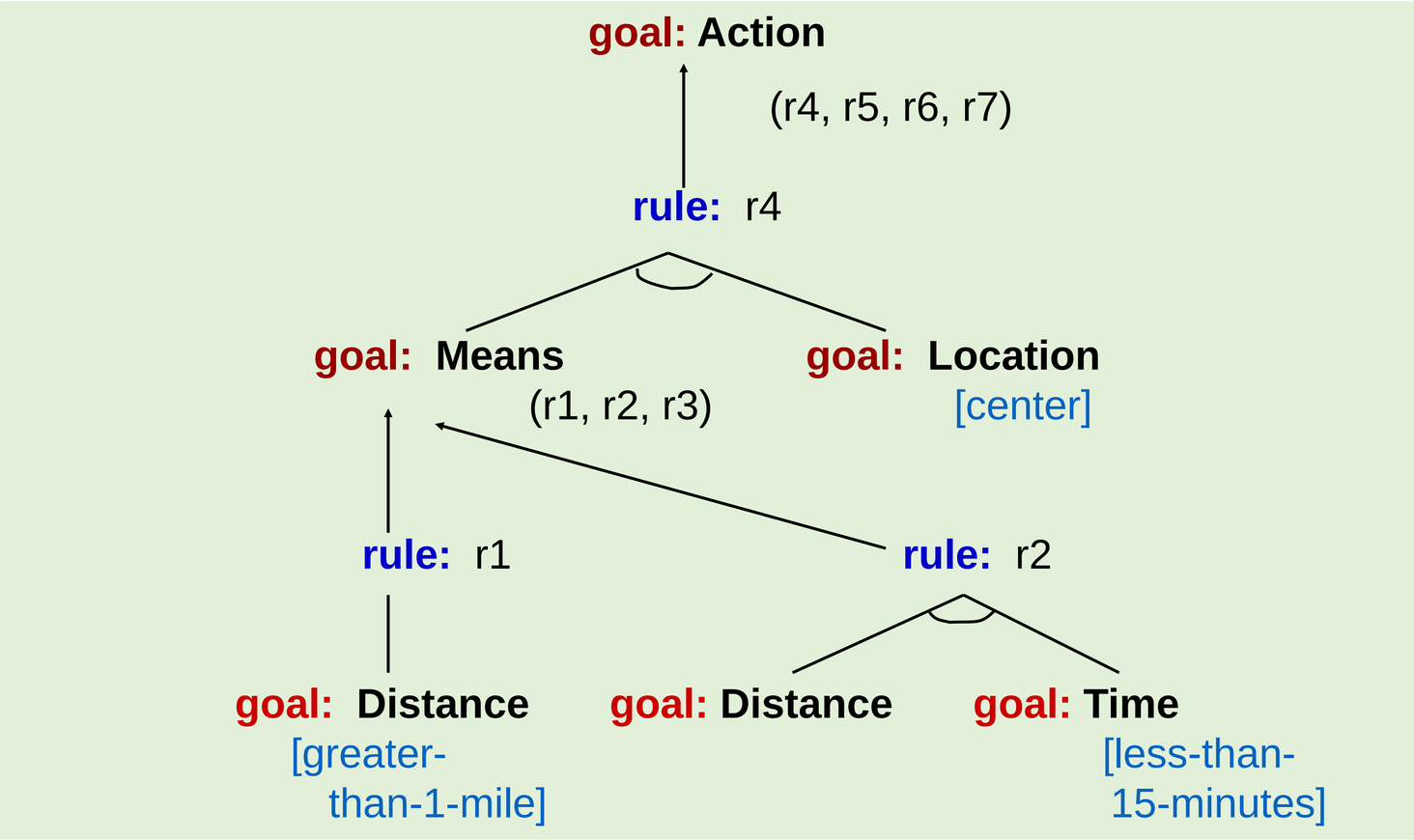
How far is the theater?
1. greater-than-1-mile
2. greater-than-5-miles
→ 1

How much time do you have?
1. more-than-15-minutes
2. less-than-fifteen-minutes
→ 2

Where is the theater?
1. center
2. outside-center
→ 1

Action ^value get-taxi

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423



Inference Tree

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Conversation

- Driven entirely by the system.
- The user simply replies to the questions, which correspond to terminal, askable, subgoals.
- The user answers are recorded in the working memory.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Why and How Explanations

- The inference network records all the reasoning of the system in the given consultative conversation
  - It is a trace of the system's reasoning steps
- **Types of explanation**:
  - '**Why** are you asking me this' – i.e., why are you pursuing this goal?
  - '**How** did you reach this conclusion?' – i.e., how was this goal achieved?
- The explanations are provided by **ascending** ('Why?' questions) or **descending** ('How?' questions) the inference tree

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

‘Why?’ Explanations

Findout: Action

How far is the theater?
1. greater-than-1-mile
2. greater-than-5-miles
→ Why?

I am trying to decide the Distance to derive the Means.

Rule r1:
If the Distance is greater that 5 miles
Then the Means is driving

→ Why?

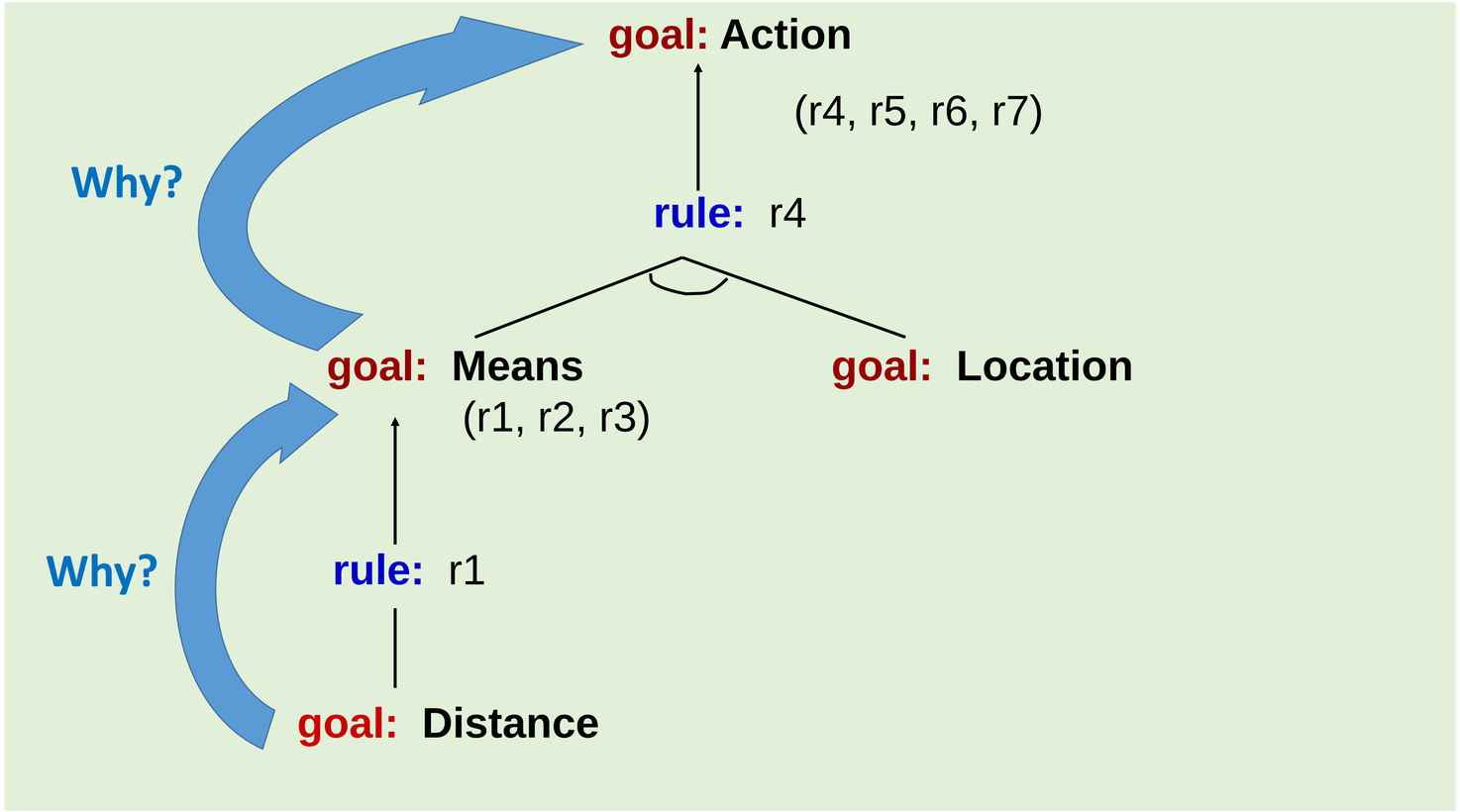
I am trying to decide the Means in order to derive the Action.

Rule r4:
If the Means is driving and
the Location is center
Then the Action is get a taxi

→ Why?

Because you have asked me!

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423



Inference Tree

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

‘How?’ Explanations

Action is get taxi.

→ How?

Based on rule r4:

If 1. Means is driving
2. Location is center
Then Action is get taxi

→ How 1?

Based on rule r2:

If 3. Distance is greater than 5 miles
4. Time is less than 15 minutes
Then Means is driving

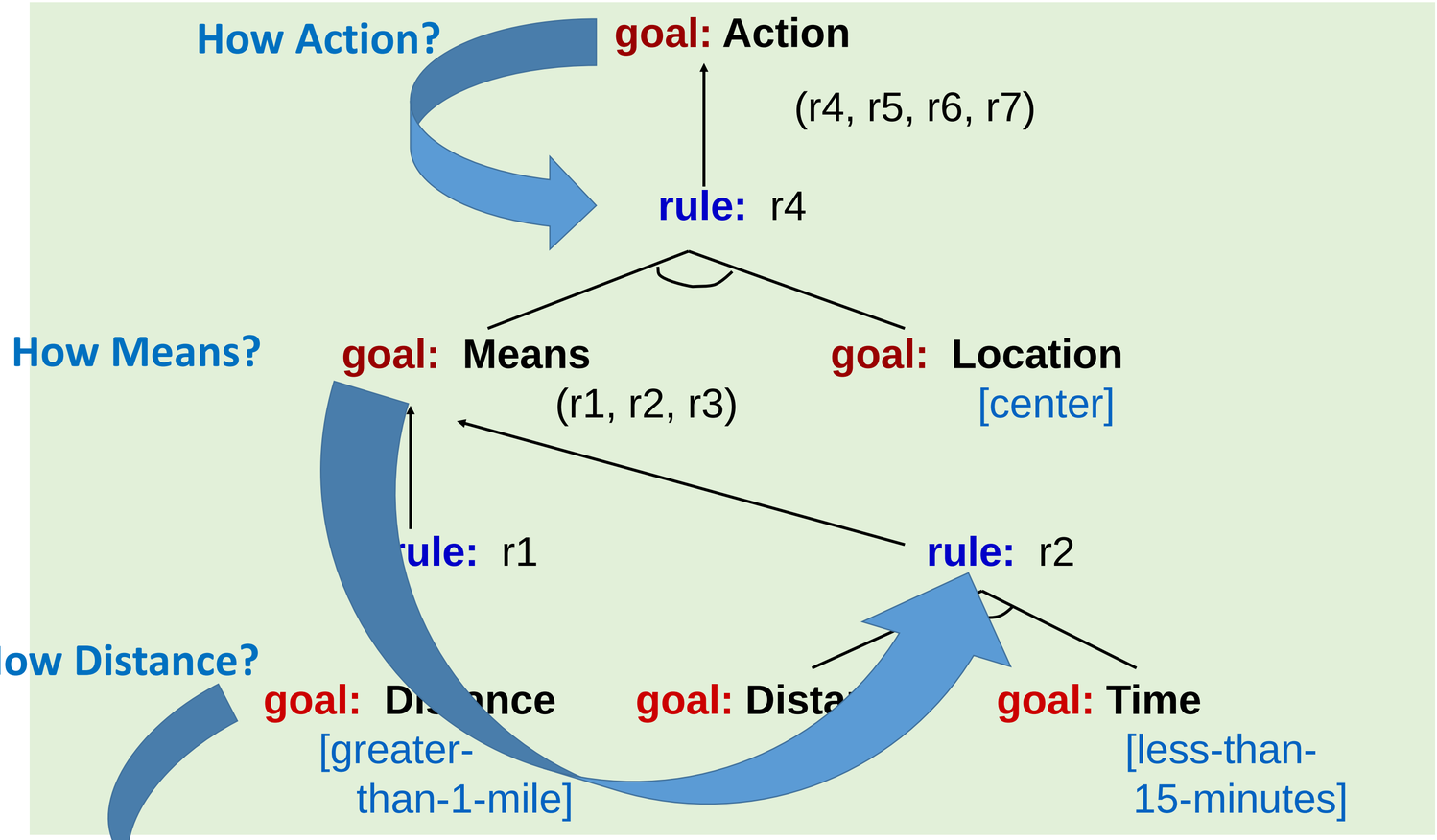
Rule r1 was monitored unsuccessfully.

→ How 3?

You told me that the Distance is greater than 1 mile.

→ OK

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423



Inference Tree

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Why being able to provide explanations is important?

- For the **acceptance** of the (advisory) system by its user
- To **debug** the knowledge of the system
- To use the system as an **educational tool**

Explainable AI - XAI

Now adays we talk about Explainable artificial intelligence (XAI) as **a set of processes and methods that allows human users to comprehend and trust the results and output created by machine learning algorithms**. Explainable AI is used to describe an AI model, its expected impact and potential biases.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Remark

Whether the ‘mechanistic' explanations of the 'Why?' and 'How?' types presented above are satisfactory explanations depends on whether the rules are understood, since explanations are nothing more than the traces of chains of rules.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Local control through meta-rules

- **Meta-rules** are rules about the use of other rules
- They concern a specific knowledge domain
- They specify priorities about groups of rules, e.g., which rules in specific contexts should be ignored or processes before some other rules, etc.
- In a system that applies backward chaining, meta-rules are connected with ojects, or properties of objects.

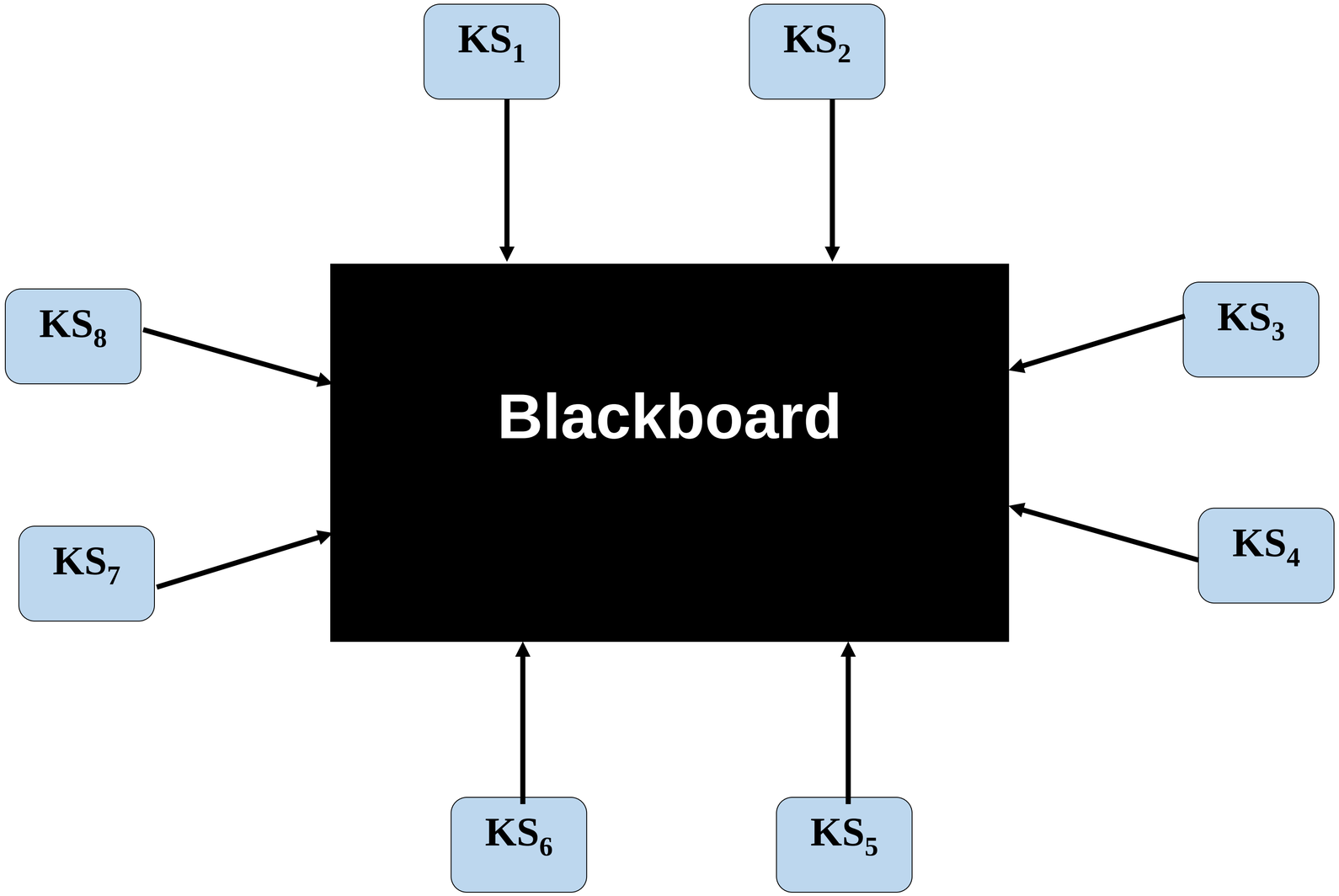
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Blackboard Model

- Revisiting the blackboard model that was previously introduced (unit 4) as a multiagent architecture.
- The **blackboard model** can also be considered as a distributed, heterogeneous, cooperative, production system.
- The central components of a system based on this model are:
  - The **blackboard**
  - The **knowledge sources**; each knowledge source is a separate ‘agent’

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

KS: Knowledge Source

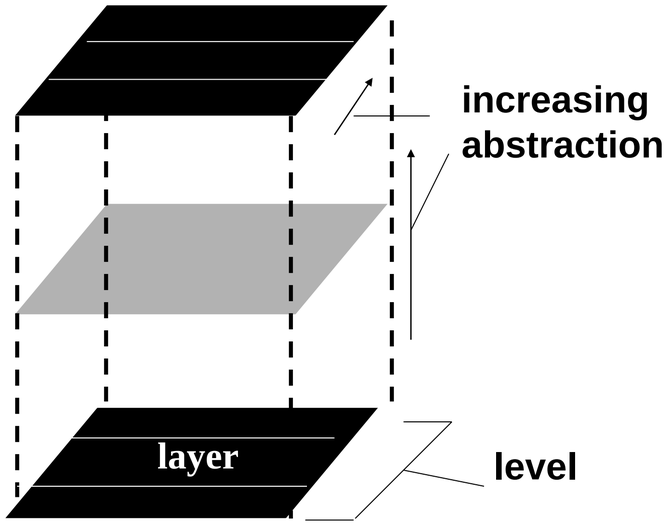


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Blackboard

- It is a **structured working memory**, playing a role analogous to that of a blackboard when used by a group of people with the goal of solving **collaboratively** some problem
- It is a global data structure, consisting (in the general case) of a number of levels (of abstraction), where each level is broken down into several layers (of abstraction)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423



Knowledge Sources

- They correspond to the members of some working team, in other words the various **specialists** for specific parts of the overall problem
- Some of these have the role of **coordinator**, e.g., the teacher, and the rest have the role of **solver**, e.g., the students
- Every knowledge source is a separate production system, with its own production memory, rule interpreter, and working memory, that concerns the solving/coordination of a subproblem of the given problem; in fact, it is not necessary to be a production system, but any computational entity
- Hence every knowledge source is an independent, autonomous system for solving part of the problem:
  - It consists of a public and a private part
  - The public part defines its interface with the blackboard
  - The private part implements its task
- The functioning of a set of knowledge sources is underlined by the so call **opportunistic search**

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Blackboard use

- The information (symbolic expressions) that can be entered on each layer of every level are defined
- There are two basic relations between the layers of the same level, and between the levels:
  - **reduction**, from bottom to top
  - **expectation**, from top to bottom

- Initially, information is entered at the lowest layer of the lowest level
- Eventually the solution of the problem will be derived at the highest layer of the highest level
- The information entered on the blackboard is classified as facts, goals or expectations

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Cognitive Architectures in terms of Production Systems

- Cognitive architectures are models of human reasoning
- Production systems are popular in cognitive architectures, e.g., ACT and SOAR
- In such systems, the “working memory” models human short-term memory, and the productions are part of long-term memory

Production Systems and Databases

- In comparison to databases, production systems often have many rules and relatively few facts
- With suitably optimized matching technology, production systems can operate in real time with tens of millions of rules

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Summary

- ❑ Frames – slots and facets
- ❑ Frame System – links between object frames
- ❑ Inheritance – simple and multiple
- ❑ Production Systems – architecture
- ❑ Forwards chaining – the Rete algorithm
- ❑ Backwards chaining – inference tree, Monitor and Findout
- ❑ Why and How explanations
- ❑ Blackboard Model

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September – December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Expert Systems Technology

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 7

Expert Systems Technology

CONTENTS

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on the expert systems technology, students will be able:

1. To explain what an expert system is and what is the purpose of the expert systems technology.
2. To outline the architecture of an expert system and illustrate it by reference to actual expert systems.
3. To point out the key principles of the expert systems technology and to list the distinctive characteristics of expert systems in general, and particularly of the first generation of expert systems, including weaknesses.
4. To present the application areas of several successful, early expert systems.
5. To analyze three potential roles for an expert system, namely advisor, critic, tutor.
6. To discuss the three basic forms of reasoning, deduction, abduction and induction, and explain the ‘collusion’ between abduction and deduction, forming the hypothetico-deductive scheme of reasoning, and how this is implemented in INTERNIST-1 and PROSPECTOR.
7. To present the uncertainty models in MYCIN, INTERNIST-1 and PROSPECTOR.
8. To explain the hybrid knowledge representation used in PROSPECTOR (production rules and semantic networks), the frame system in INTERNIST-1, and how MYCIN applies deductive reasoning through backward chaining.
9. To explain the notion of ‘compiled knowledge’ and its interpretation in the MYCIN system.
10. To discuss knowledge acquisition systems and empty systems (expert system shells).

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Expert Systems brought a strong new perspective to AI:
the mechanization of powerful, domain-specific knowledge as a viable alternative to weak methods, for handling large, difficult problems

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Weak methods led to the birth of expert systems

- Weak methods were the focus during the first decade of AI research, aiming to find complete solutions to problems by using general-purpose search mechanisms stringing together elementary reasoning steps
- They were called **weak methods** because they could not scale up to large or difficult problem instances
- The alternative was to use more powerful, **domain-specific knowledge**, allowing larger reasoning steps that could handle typical cases in narrow areas of expertise
- The technology of **expert systems** was thus born and was a main area of AI research for almost two decades (1969-1986)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

DENDRAL, MYCIN and R1: successful early examples

DENDRAL

- Developed at Stanford by Ed Feigenbaum, Bruce Buchanan and Joshua Lederberg (a Nobel laureate geneticist)
- It inferred the molecular structure from the information provided by a mass spectrometer, given as input the elementary formula of the molecule and the mass spectrum
- The naïve version generated all possible structures, predicted their mass spectrum and compared these with the actual spectrum to home onto a solution – intractable approach even for moderate-sized molecules
- The powerful, revised version embodied the **knowledge of analytical chemists**, not in the form of first principles but in efficient "cookbook recipes"
- DENDRAL was the first successful knowledge-intensive system, whose expertise derived from large numbers of **special-purpose rules**
- The **Heuristic Programming Project** (HPP) that followed at Stanford by Feigenbaum and others aimed to investigate the extent to which the new methodology of expert systems could be applied to other areas

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

DENDRAL, MYCIN and R1: successful early examples

MYCIN

- MYCIN (developed by Edward Shortliffe) to be discussed in detail later in this unit, was the next major system
- Its domain was the diagnosis of blood infections and its knowledge base consisted of about 600 rules
- It performed as well as some experts, and considerably better than junior doctors
- Unlike the DENDRAL rules, no general theoretical model existed from which the MYCIN rules could be deduced, but they had to be acquired from extensive interviewing with the experts
- In addition, the rules had to reflect the uncertainty associated with medical knowledge; MYCIN incorporated a calculus of uncertainty called **certainty factors** that aimed to model how experts assessed the impact of evidence on the diagnosis

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

DENDRAL, MYCIN and R1: successful early examples

- **R1**
  - R1 (later called XCON) was the first successful commercial expert system
  - It operated at the Digital Equipment Corporation and helped configure orders for new computer systems
  - By 1986, it was saving the company an estimated $40 million a year
  - By 1988, DEC's AI group had 40 expert systems deployed
  - Nearly every major U.S. corporation had its own AI group and was either using or investigating expert systems

Building and maintaining expert systems for complex tasks could be a laborious task, especially if the systems could not learn from experience

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

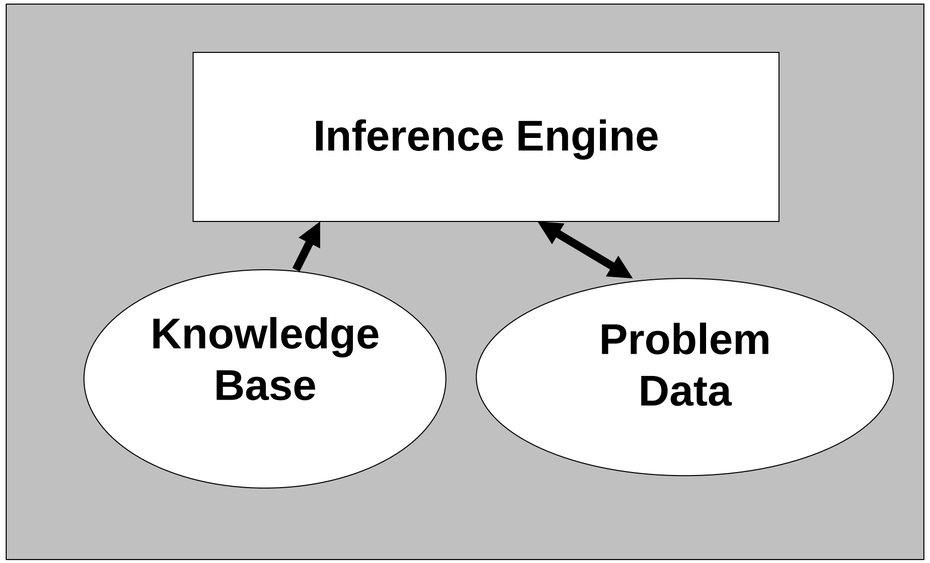
Expert Systems

- Expert systems belong to the applied part of Artificial Intelligence.
- An expert system possesses knowledge which it uses based on some reasoning mechanism.
- The body of knowledge, or knowledge base, of an expert system represents domain expertise.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

What is an Expert System?

Expert systems belong to the category of knowledge-based systems



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

General Definition

An expert system is a computing system that can efficiently and effectively solve realistic problems, the solution of which on the part of a human implies the existence of some form of expertise.

The general definition makes no commitment as to how to solve the problems.

Acquiring expertise is a painstaking process achieved through extensive experience.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

(General) Purpose of Expert Systems Technology

The widespread dissemination of the ability to solve problems, involving expertise, for social, economic or other reasons.

Expert systems are **decision support systems** that need to interact with their user and provide **explanation/justification** of their suggestions.

Thus, it's not just the result that matters, but the reasoning and knowledge that led to it.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Refined Definition:

An expert system is a knowledge base system that extensively 'embodies' the **expertise** of one or more experts in the relevant (specialized) field.

The performance of the system in solving these realistic problems should be **comparable** to that of experts.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Key Principles of the Expert Systems Technology

- The (wide) spread of a given expertise for social, economic or other reasons.
- The typical user of an expert system is expected to belong to the relevant knowledge field, but not to be an expert himself.
- The expert system helps that user perform at a level of competence comparable to that of the expert.
- Even the experts themselves can use the system with significant benefits.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Breadth of Expertise Spreading

- The breadth of the desired spread of some expertise depends on the domain.
- The scope of the usefulness of an expert system can be the entire universe or be limited to the context of a company/organization, which can of course be multinational.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Indicative application areas of first expert systems

- **Scientific Analysis**
  - The very successful DENDRAL system, mentioned above, is considered the first expert system.
- **Geology – Mineral Exploration**
  - PROSPECTOR, to be discussed in the sequel, aided geologists in evaluating the favorability of an exploration site or region for occurrences of ore deposits of particular types. Once a site had been identified, PROSPECTOR could also be used for drilling-site selection. It is rumored that PROSPECTOR discovered a very rich ore deposit.
- **Engineering**
  - The XCON system (initially named R1), mentioned above, is a prime example of a very successful expert system in the engineering field.
- **Medicine**
  - A plethora of famous expert systems (MYCIN already mentioned, INTERNIST-1, CASNET and many others) that contributed significantly to the development of the technology

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Each of the primary expert systems was developed as a stand-alone system, often running on special hardware (e.g., a LISP machine), but with the advent of the Internet, the cloud, etc. this no longer needs to be so, and the spreading is hugely facilitated.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Roles of Expert Systems

- **Consultant/advisor**
  - This is the principal role
- **Critic**
- **Tutor**
- All these roles involve **dialogue** between the user and the system
- Hence expert systems are **highly interactive systems**
  - The **quality of the interface** between the system and the user is a critical factor

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Expert System Interface Requirements with User

- **The system interacts with the user to:**
  - 'understand' the problem better
  - elicit more information about the problem

- **The user interacts with the system to:**
  - offer more information about the problem
  - better understand the reasoning of the system
  - be convinced of the validity of the proposed solution
  - identify gaps or errors in the knowledge base of the system for improvement purposes (knowledge debugging) – here it is understood that the user belongs to the category of experts

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Distinctive Characteristics of Expert Systems

- The dialogue element and its specific requirements
- The capacity for self-development and self-improvement
  - Expertise is never static, but continuously improves and evolves through experience and information/knowledge exchanges/sharing between the community of experts
  - Knowledge base updates (automatically through learning, or through external guidance) are a must
    - Content, structure or both?

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Developmental Requirements

An expert system should:

- have the ability to self-improve (i.e., learn) based on its own experiences in solving problems
- facilitate the external updating of its knowledge base

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Characteristics of Expert Systems

- They simulate human reasoning, knowledge and cognition
- They solve problems using heuristic or approximate methods
- They deal with problems of realistic complexity, the efficient and effective resolution of which requires human expertise
- They demonstrate high levels of performance in both speed and correctness of solutions
- They interact with the user
- They explain and justify their recommendations
- They self-evolve
- In short, they are competent problem solvers for the specific, specialized fields

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Basic Forms of Reasoning

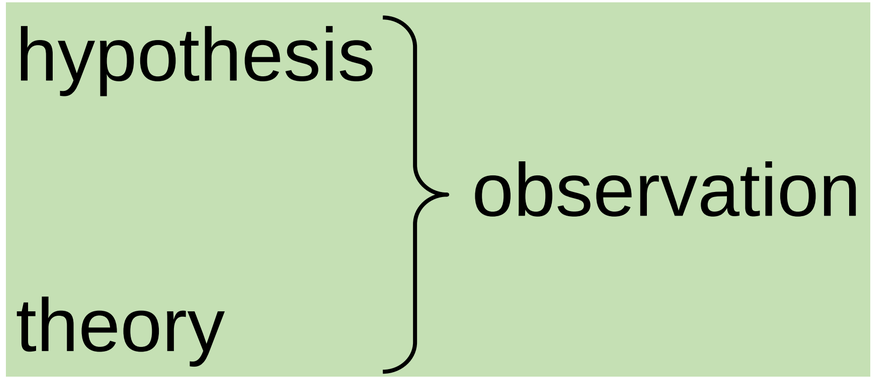
The philosopher C.S. Peirce distinguished three basic forms of reasoning:

- **Deduction** – prediction
- **Abduction** – explanation
- **Induction** – learning

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Deduction:

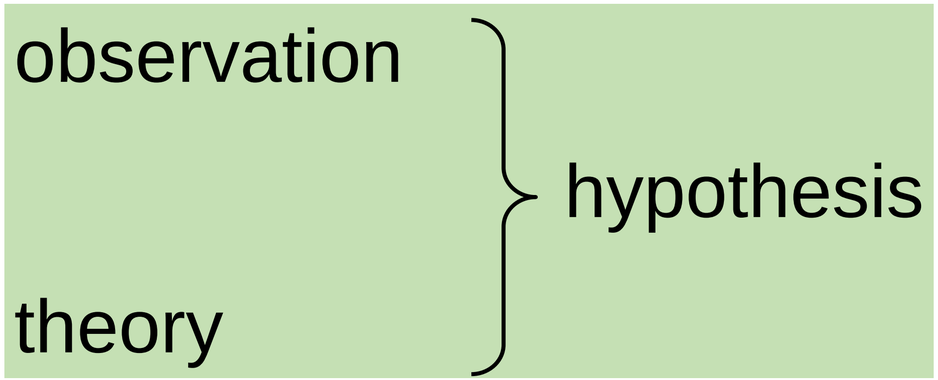
Based on a hypothesis (or a fact) and a theory, observations are predicted



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Abduction:

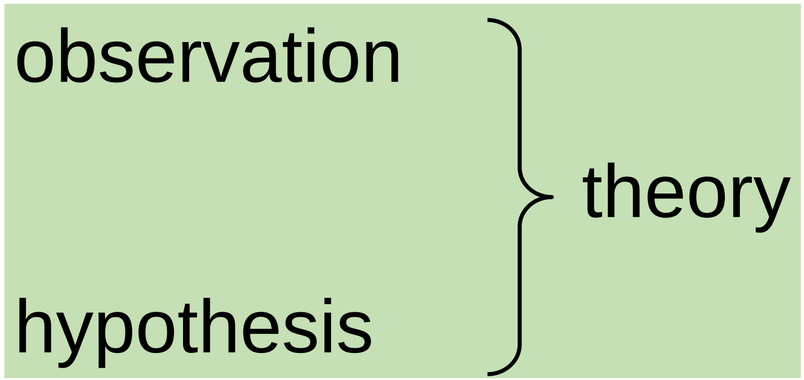
Based on an observation of unknown cause and a theory, a hypothesis is generated for explaining the observation



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Induction:

Based on several observations and hypotheses regarding possible associations, a theory is derived, i.e., new knowledge is learned



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Let’s consider the following theory (knowledge):

$\forall x$ Has_flu(x) $\Rightarrow$ Manifests(x, fever)

$\forall x$ Has_flu(x) $\Rightarrow$ Manifests(x, headache)

$\forall x$ Has_measles(x) $\Rightarrow$ Manifests(x, fever)

$\forall x$ Has_measles(x) $\Rightarrow$ Manifests(x, red_rushes)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Deduction

Based on the theory and the hypothesis (or the fact) that

“Kostas has flue”

Has_flu(Kostas)

the following conclusions may be derived

Manifests(Kostas, fever)

Manifests(Kostas, headache)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Deduction

- The consequences of the given hypothesis are **predicted** based on the given theory
- Deductive reasoning is used when it is necessary to prove whether a given proposition is true

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Abduction

- Abduction is hypothetical reasoning
- Based on the given theory and the observation that Kostas has a fever, **plausible hypotheses** regarding the cause of the fever are the flu or measles.
- **Crucial Question**: Which is the best of the plausible hypotheses, in short which is the best explanation of Kostas' fever

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Abduction

- Competing plausible hypotheses need to be evaluated in order to select the one that is the best explanation
- This evaluation/investigation involves deductive reasoning:
  - If the cause is flu, this implies the occurrence of a headache
  - But if the cause is measles, this implies the appearance of red rashes

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Abduction

- In general, opposing implications from opposing hypotheses are good points of separation:

$$\forall x \;\; H_1(x) \Rightarrow P(x)$$

$$\forall x \;\; H_2(x) \Rightarrow \sim P(x)$$

the question P(A) is a good point of separation between the rival hypotheses

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Deductive reasoning:

to prove whether something holds

Abductive reasoning:

to answer why something holds

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Abduction and Deduction

- In order to answer a **why** it is important to decide **whether**
- Hence, deduction may be considered a sub-process of abduction
- This ‘collusion’ is the **hypothetico-deductive** scheme of reasoning that appears in many expert systems, mainly systems involving diagnosis and debugging

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Induction

- Inductive reasoning aims to develop theories, to discover knowledge
- Hence it concerns **learning**
- Let's suppose that the following observations were made:

Has_flu(Kostas)	Manifests(Kostas, fever)
Has_flu(Maria))	Manifests(Maria, fever)
Has_flu(Eleni)	Manifests(Eleni, fever)
. . . . .	
Has_flu(Yiannis)	Manifests(Yiannis, fever)

- From the above observations, the following general association may be derived:

$$\forall x \; \text{Has\_flu}(x) \Rightarrow \text{Manifests}(x, \text{fever})$$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Induction

- Induction concerns the developmental aspect of the expert system
- An expert system, as already mentioned, should have the ability of incremental self-improvement based on its problem-solving experiences
- Abduction and deduction concern the expert system's problem-solving, i.e., the system's reasoning regarding the derivation of solutions to problems

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Exemplar First Generation Expert Systems: MYCIN, PROSPECTOR, INTERNIST-1

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

First Generation of Expert Systems

- Chronologically set from the late 60s to the late 70s
- Weaknesses
  - Stiffness in reasoning
  - Inadequate user interface
  - Limited scalability
- The initial approach focused on the **representation level** emphasizing uniformity in knowledge representation, and consequently simplicity of reasoning
- The first systems were not necessarily accurate simulators of the relevant areas of expertise
- Important elements of expertise were absent or appeared in an implicit way

.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Characteristics of First-Generation Expert Systems

By and large first-generation expert systems:

- ❑ Had a uniform representation and relatively simple reasoning mechanisms
- ❑ Were shallow systems
- ❑ Used mainly classification problem solving
- ❑ Had a centralized control structure

New approach

The new approach, which is a significant change of orientation, focuses on the **knowledge level** with the aim of creating a **conceptual model of expertise** which is then translated at the representation level.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MYCIN

- The MYCIN system was developed at Stanford University with Edward Shortliffe as the principal investigator
- The aim of the system was to assist non-experienced doctors in the diagnosis and treatment of microbiological infections of the blood, such as meningitis

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Motivation for Creating MYCIN

- The need to spread expertise because errors were often observed with tragic consequences for patients
- The advisory system to be an efficient basis for the development of an intelligent educational tool for use by medical students

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Advisory System

Diagnostic Part

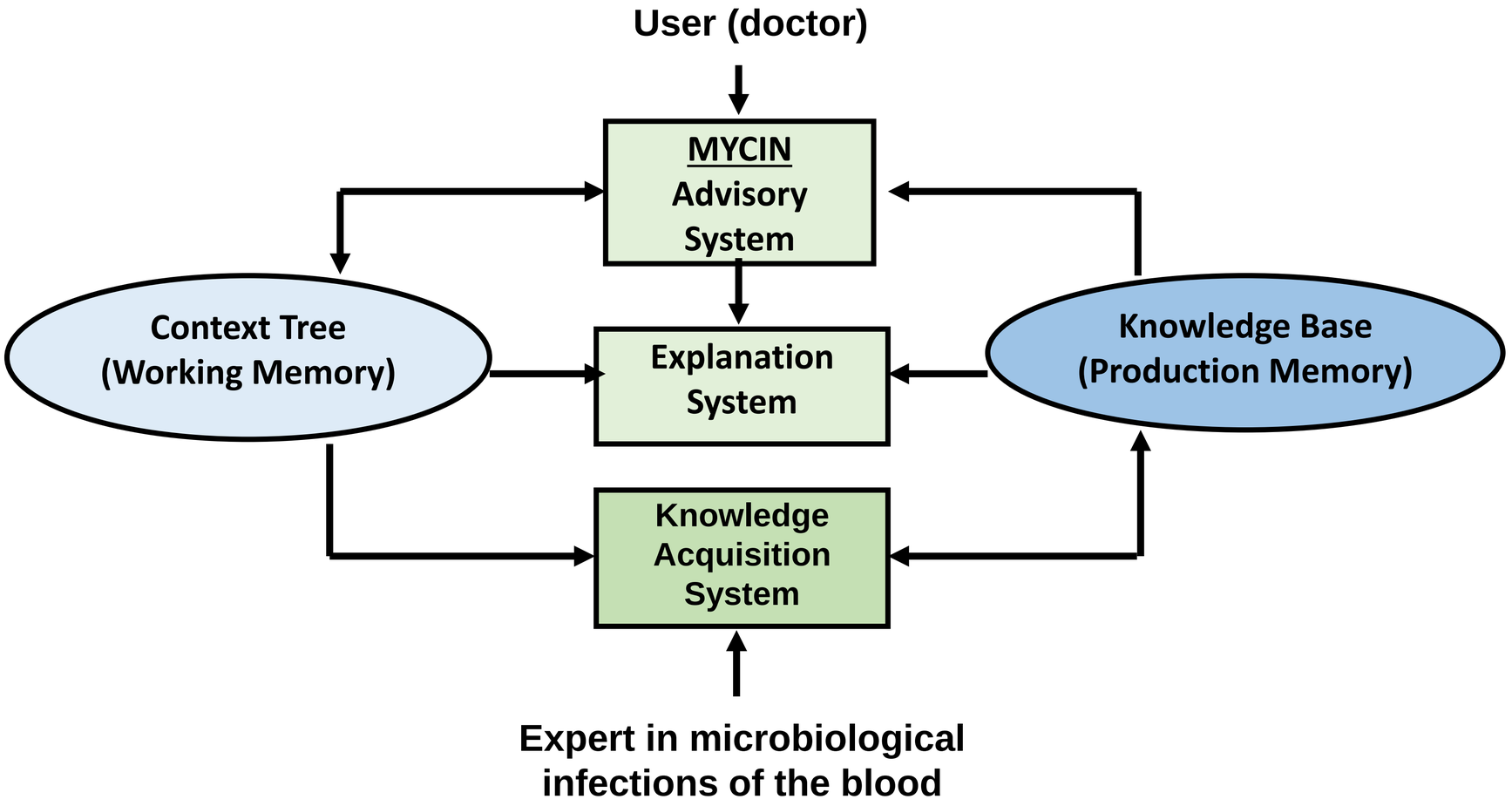
What are the most likely identities for the microorganisms that cause the infection?

Recommendation for Treatment

Based on the findings of the diagnostic part as well as the patient's general condition and history, what is the best combination and corresponding doses of antibiotics?

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MYCIN and Auxiliary Subsystems



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Base

IF:

1. **the strain of the organism is gram-pos**
2. **the morphology of the organism is coccoid**
3. **the development-pattern of the organism is chain-like**

THEN

There is suggestive evidence (0.7) that the identity of the organism is streptococcus

The knowledge base (or production memory) contains around 600 rules.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Rules are described as

- Heuristic
- Empirical associations
- ‘Compiled knowledge’
- Judgmental

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Rule structure

- Rule premises have the usual (object, attribute, value) structure
  - organism is an object (or **context** in MYCIN's vocabulary)
  - stain, morphology and development-pattern are attributes (or **clinical parameters** in MYCIN's vocabulary)
  - gram-pos, coccoid and chain-like are values
- Object categories are few and are hierarchically related
  - persons, current-cultures, current-organisms, past-cultures, past-organisms, etc.
  - Every object has several attributes/characteristics

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Rule organization

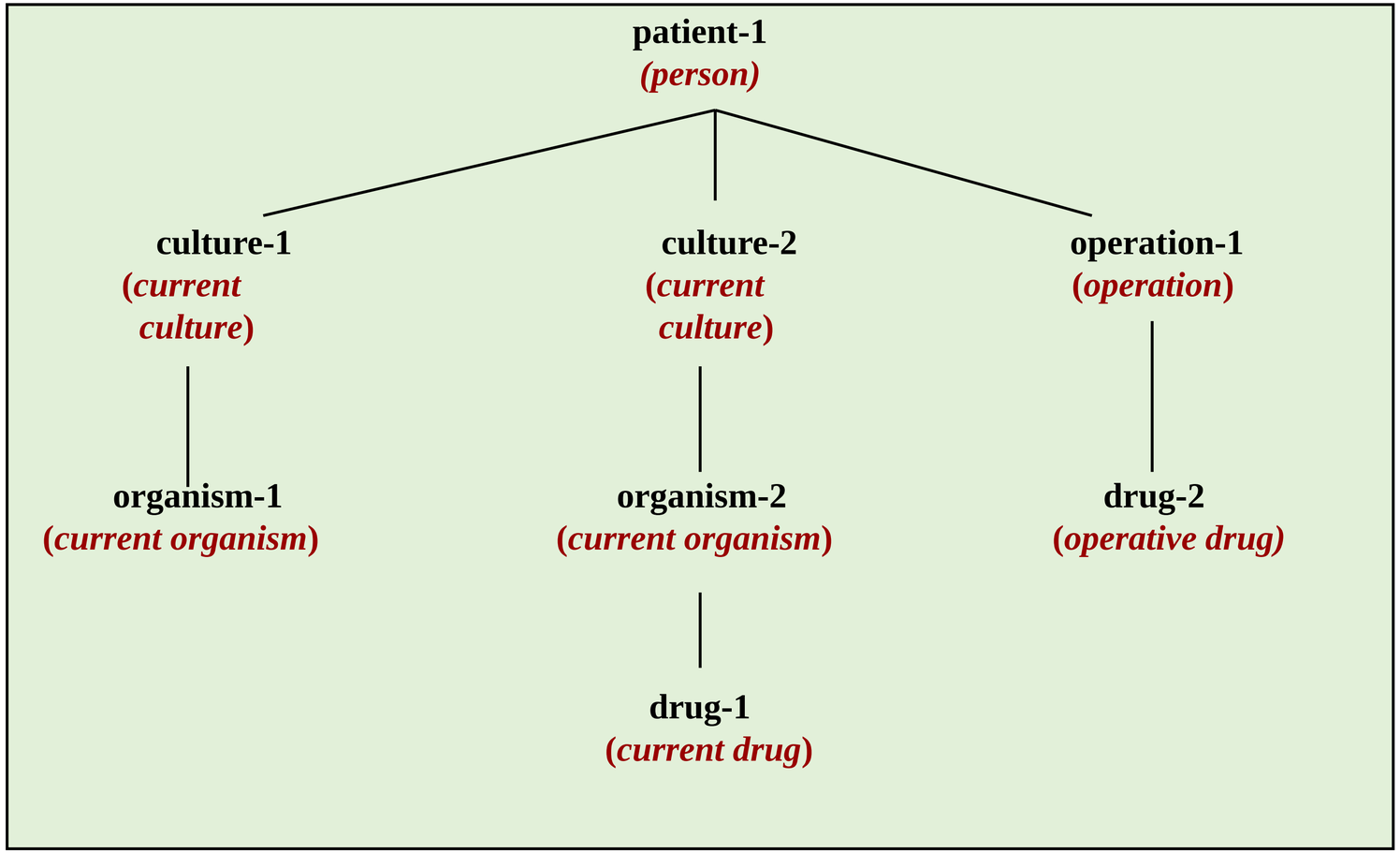
- Rules are partitioned based on the objects that make up their contexts
- The set of rules for the same object is indexed in the two dimensions, prediction and update, based on the characteristics of the object

Working memory

- The working memory or **object tree** (context tree) is hierarchical
- Nodes represent objects, where the root represents the patient
- The object tree is built incrementally through the application of production rules

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Reasoning

- MYCIN applies backwards reasoning, implemented through backwards chaining
- The reasoning is **deductive** – top to bottom

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Goal-Rule

IF:

1. there is at least one organism that requires treatment, and
2. the possibility of other organisms, being present, that require treatment has been explored

THEN:

1. compile a list of possible treatments
2. select the best treatment for the patient

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Flow of Reasoning

The flow of reasoning, for the diagnostic part of the system, results in the following three stages (that do not appear explicitly):

1. Get more clinical information about the patient
2. Find out which microorganisms, if any, could have caused the infection
3. Name the most likely microorganisms

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Rule chaining

Subgoal: What is the locus of infection?

IF:

1. culture site known
2. specimen collected this week
3. **organism causes a therapeutically significant disease**

THEN:
A locus of infection is definitely at culture site

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Rule chaining
Subgoal: What is the locus of infection?

IF:

1. culture site known
2. specimen collected this week
3. **organism causes a therapeutically significant disease**

THEN:

A locus of infection is definitely at culture site

IF:

1. culture site not normally sterile
2. culture collected by sterile method
3. there is sufficient numbers of the organism

THEN:

It **suggests strongly** that the **organism causes a therapeutically significant disease**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Control Structure

- It is mainly based on the use of **meta-rules**
- The general strategy is to give preference to the rules with the **highest certainty factors**
- Meta-rules are associated with object characteristics

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Meta-rule for the Identity of Organisms

IF:

1. the source of the culture is not sterile, and
2. there are rules that mention in their premise some past organism that could be the same as the current organism

THEN:

it is definite (1.0) that none of these rules will be useful

Pruning meta-rule

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

General Meta-rule

IF:

1. there are rules that do not mention the current goal in their premise, and
2. there are rules that mention the current goal if their premise

THEN:

It is definite (1.0) that the first should be tried before the second

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Reasoning with uncertainty

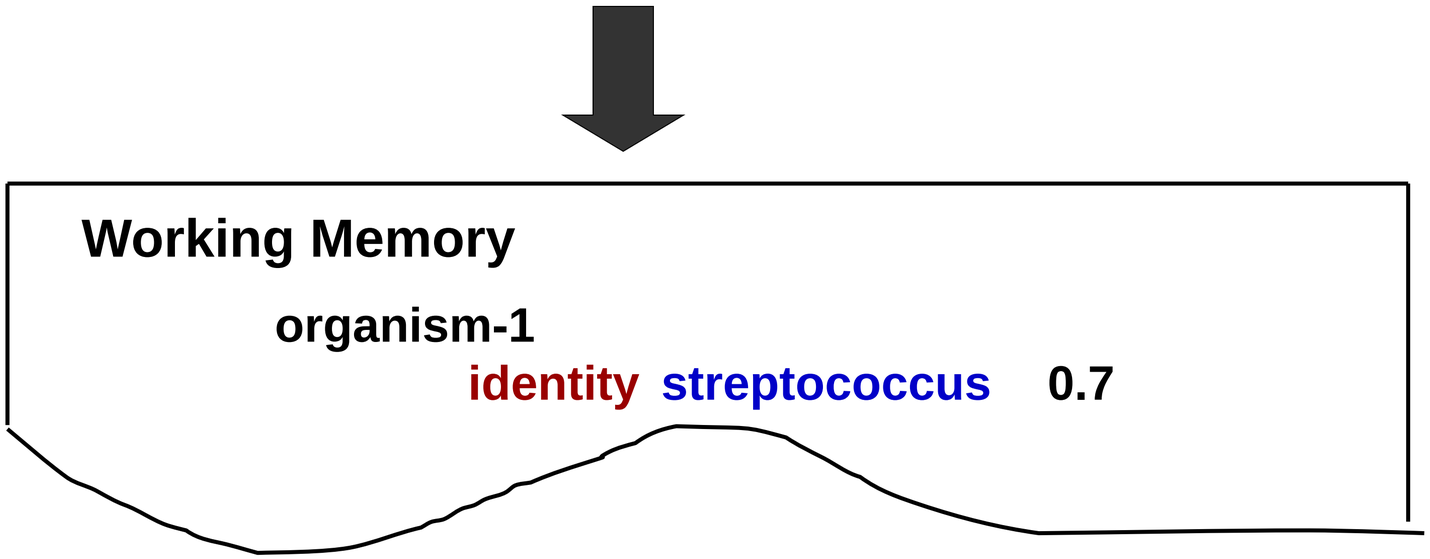
IF:

1. the stain of the organism is +ve
2. - - - - - - -
3. - - - - - - -

THEN:

There is **suggestive evidence** (0.7) that

the identity of the organism is streptococcus



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

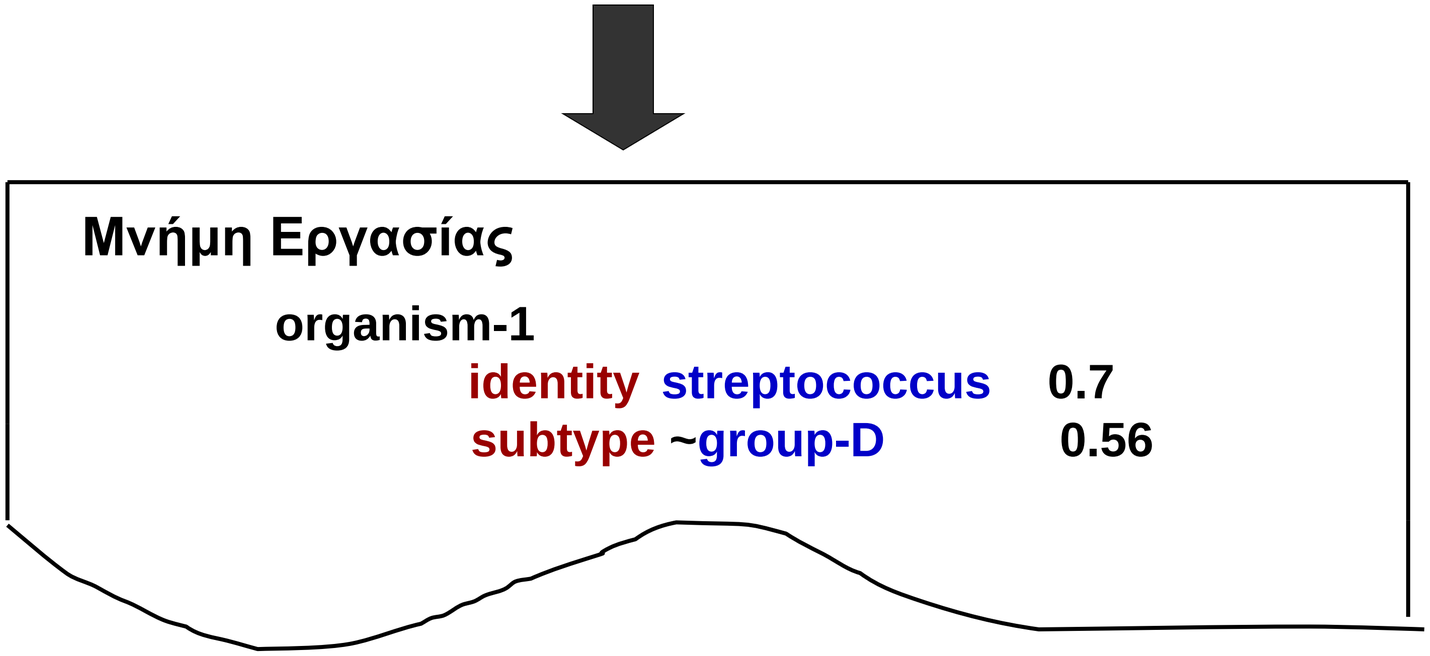
IF:

1.- - - - - - - -

2.the identity of the organism is streptococcus

THEN: 0.8

There is **suggestive evidence** that the subtype of the organism is not group-D



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Uncertainty Model

- Uncertainty occurs at two levels:
  - in relation to system knowledge and
  - in relation to the data of the problem being solved

- Therefore, regarding MYCIN we talk about rule uncertainty and uncertainty in the values of the characteristics of the objects involved.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Rule Uncertainty

$-1 \leq CF[H,E] \leq 1$

CF: Confidence Factor

Each rule has a Certainty Factor (CF), a numeric value from the domain [–1,1].

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Certainty Factors

- If **CF[H,E] > 0**, the verification of hypothesis H is supported to the degree CF[H,E], by the categorical verification of premise E, in some case
  - Premise E represents **evidence in favor** of H
- If **CF[H,E] < 0**, then the negation of hypothesis H is supported to the degree |CF[H,E]|, i.e., the absolute value of CF[H,E], by the categorical verification of premise E, in some case
  - Premise E represents **evidence against** H
- There are no rules where CF[H,E] = 0, since that would mean that evidence E is independent of hypothesis H

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Certainty Factors

- The rule uncertainties satisfy the relation

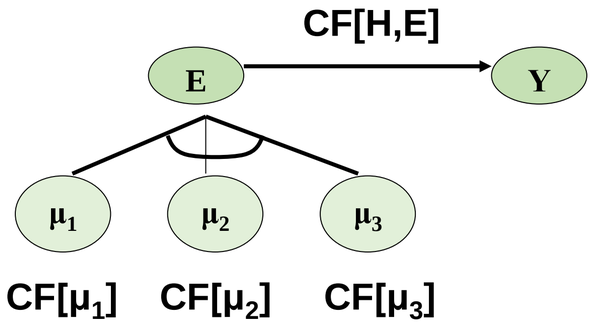
$$CF[H,E] + CF[\sim H,E] = 0$$

- Hence the same evidence E cannot be at the same time both in favor and against some hypothesis H

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Evidence Uncertainty

Based on the weakest link principle



$$CF[E] = \min_{i} \{ CF[\mu_i] \}$$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Evidence Uncertainty

- The CF of a condition that constitutes direct evidence (askable characteristic) is given by the user
- The premise CF are also drawn from the domain [–1,1]
- A rule can be applied only when the CF of its premise, in the given problem case, is at least 0.2, i.e., $CF[E'] \geq 0.2$, where E' are the observations in relation to the premise (evidence) E

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Combining Evidence towards the same Hypothesis, H

Initially CF[H]=0. The relevant pieces of evidence are processed sequentially. The processing terminates as soon CF[H] becomes 1 or –1. Evidence E contributes to the updating of CF[H], only if $CF[E'] \geq 0.2$, where E′ are the observations relating to E, as follows:

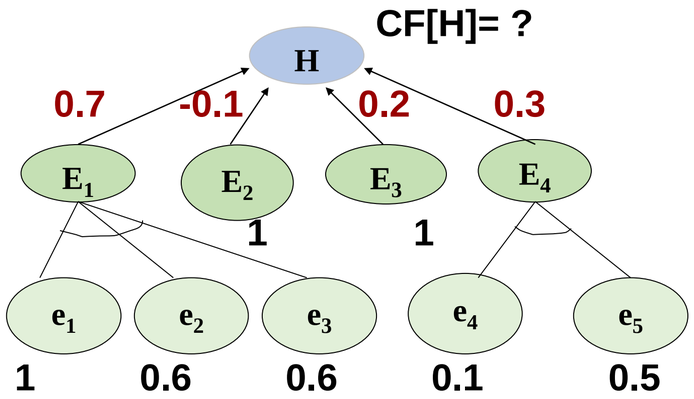
$$CF[H] := \begin{cases} CF[H] + CF[H,E'] - CF[H] \times CF[H,E'], & \text{if } CF[H] > 0 \text{ and } CF[H,E] > 0 \\ CF[H] + CF[H,E'] + CF[H] \times CF[H,E'], & \text{if } CF[H] < 0 \text{ and } CF[H,E] < 0 \\ (CF[H] + CF[H,E']) \,/\, (1 - \min(|CF[H]|, |CF[H,E']|)), & \text{otherwise} \end{cases}$$

where $CF[H,E'] = CF[E'] \times CF[H,E]$

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example



Initially CF[H] = 0

CF[E_1] = 0.6, ∴ CF[H,E_1′] = 0.6 × 0.7 = 0.42,
∴ CF[H] := 0.42

CF[E_2] = 1, ∴ CF[H,E_2] = –0.1,
∴ CF[H] := (0.42 – 0.1) / (1 – 0.1) := 0.35

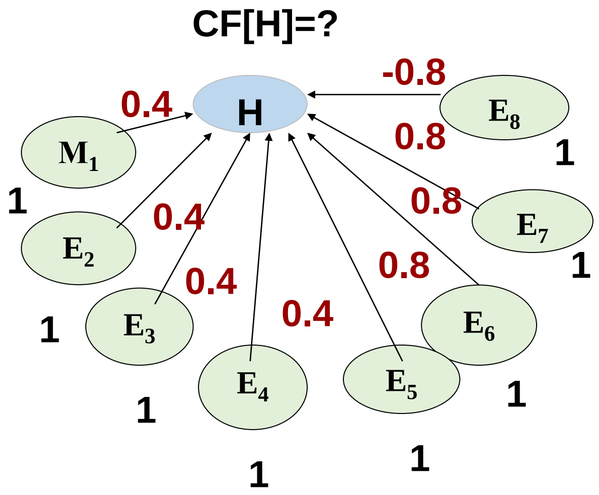
CF[E_3] = 1, ∴ CF[H,E_3] = 0.2,
∴ CF[Y] := 0.35 + 0.2 – 0.35 × 0.2 := 0.48

CF[E_4] = 0.1, ∴ the given rule cannot be applied

∴ the final CF for hypothesis H comes to 0.48

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example



Initially CF[H] = 0.

The pieces of evidence, E_1 – E_7 are all in favor of hypothesis H.

The sequence of values for CF[H] is 0.4, 0.64, 0.784, 0.8704, 0.974, 0.9948, και 0.999.

The processing of evidence E_8, which is the only one against hypothesis H, results in reducing CF[H] to 0.199 < 0.2.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Explanation System

- ‘Why?’, ‘How?’ (and ‘Why not?’) Explanations
- Inform about the CFs of the various hypotheses (e.g., to what degree of certainty it is believed that the identity of organism-1 is streptococcus?)
- ‘Justification’ of rules - ‘canned’ text
- Rule presentation in ‘pseudo’ natural language form

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

User-tailored explanations

- **User Model**: Expressed as two natural numbers from the domain {1,2, ..., 10}, which represented:
  - the level of **expertise** (E) of the user, and
  - the degree of **detail** (D) that the user wanted with respect to explanations

- In fact, only E was required, since D could be computed as a function of E (some levels above E).

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Complexity Level and Rule and Concept Importance

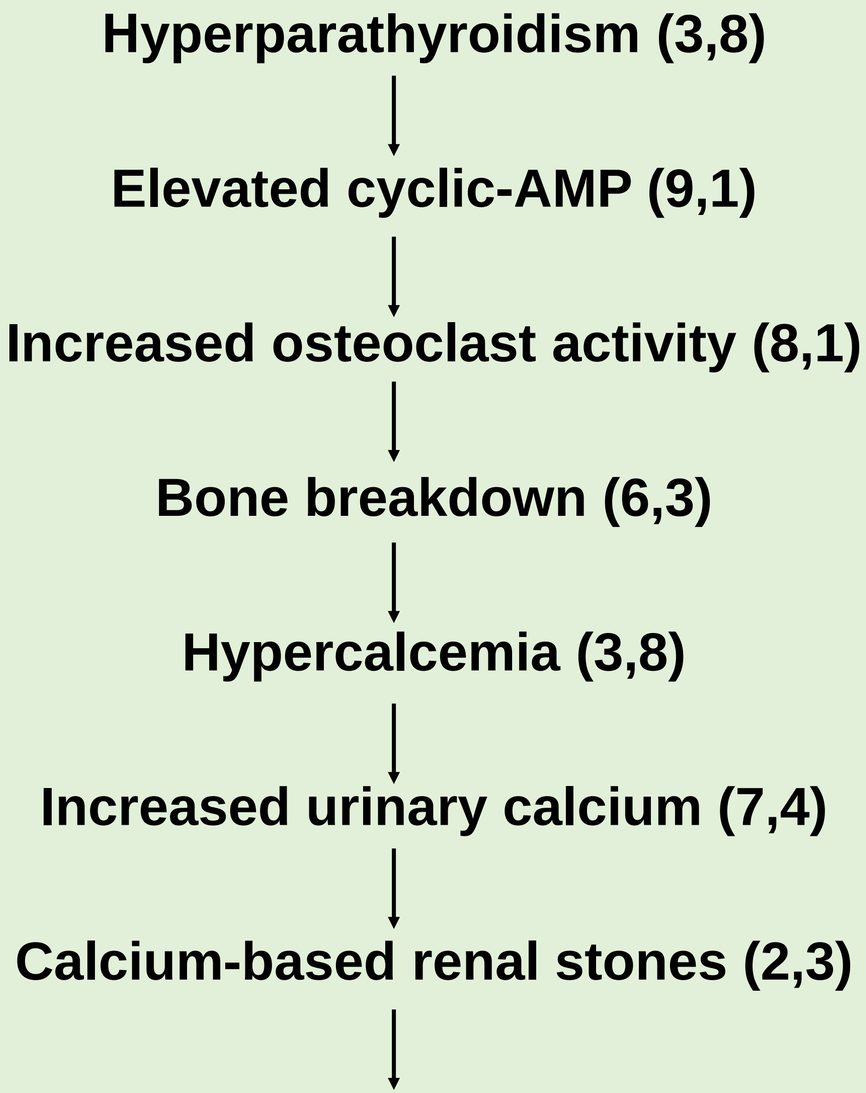
Rules and the concepts involved in them (values of object characteristics) were associated with:

- Level of **Complexity** (C)
- Level of **Importance** (I)

again, from the domain of values {1,2,…,10}

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

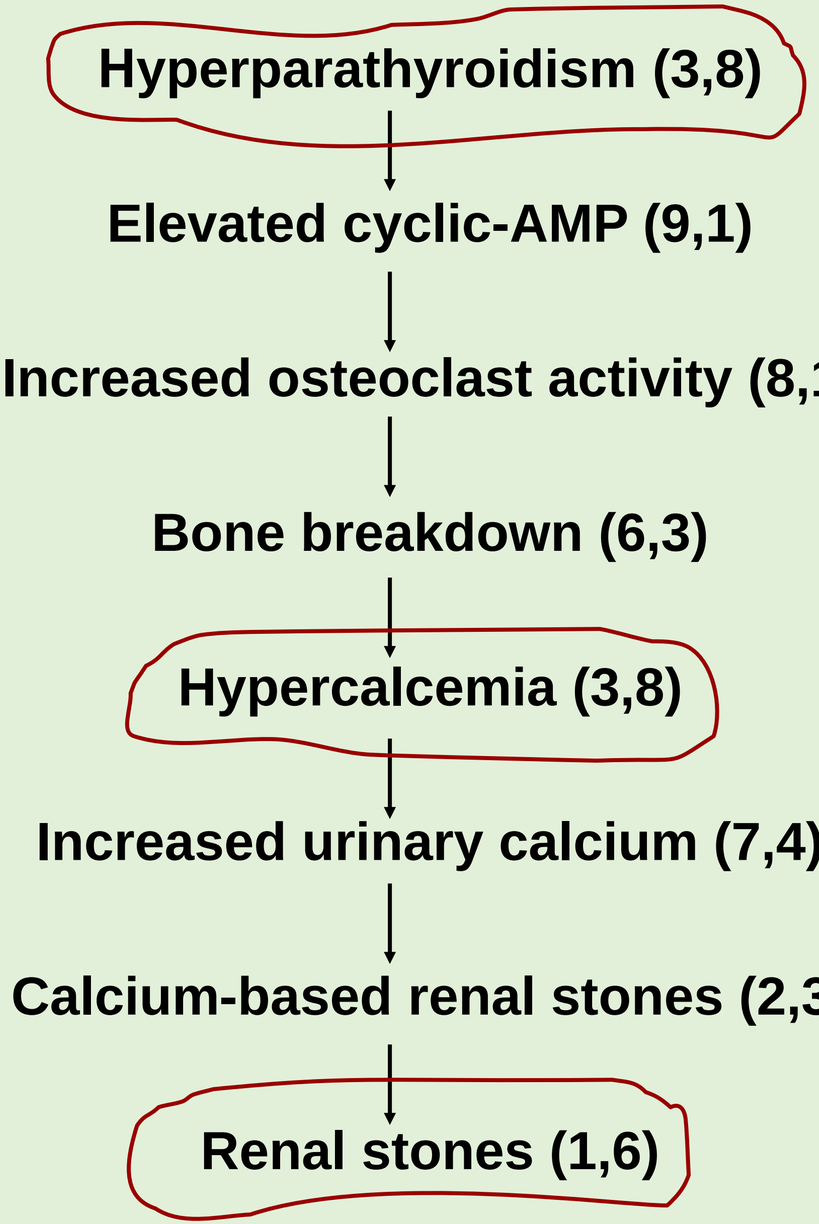
Example: Causal Reasoning Chain



Concept (C,I)

Co-financed by
Connecting Euro

Tailored for a User with E = 3



Co-financed by
Connecting Euro

Tailored for a User with E = 3

Hyperparathyroidism (3,8)

Hypercalcemia (3,8)

Renal stones (1,6)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Tailored for a user with E = 3

More detail!

Hyperparathyroidism (3,8)

↓

Bone breakdown (6,3)

↓

Hypercalcemia (3,8)

↓

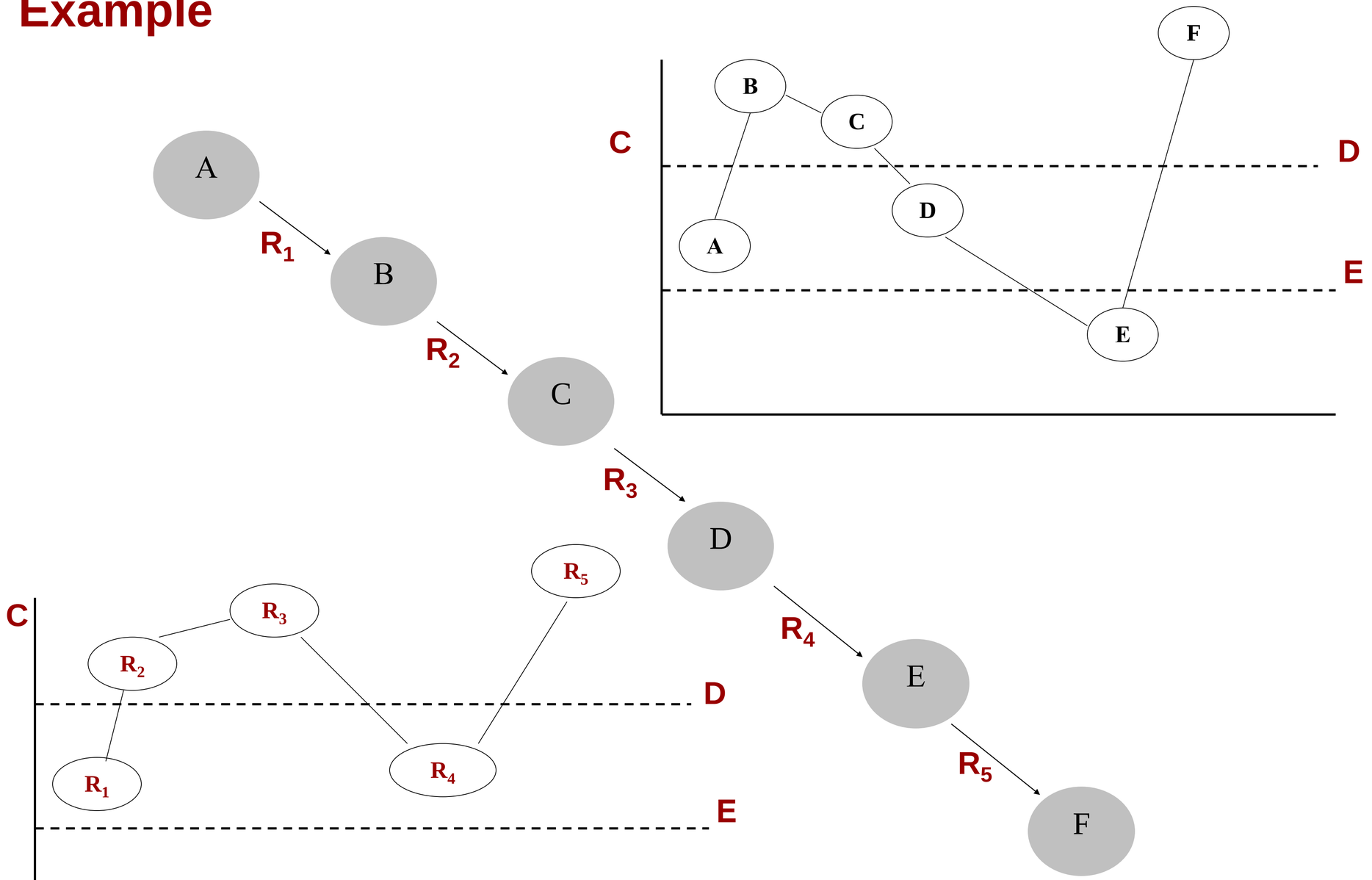
Increased urinary calcium (7,4)

↓

Renal stones (1,6)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example



Knowledge Acquisition System

- It aims at debugging and expanding the knowledge of the expert system
- The most famous primary knowledge acquisition system is the TEIRESIAS system, named after the blind seer Teiresias
- The TEIRESIAS system is also a knowledge base system
  - Its knowledge is entirely meta-knowledge since it is knowledge on MYCIN's knowledge

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

TEIRESIAS Knowledge Base

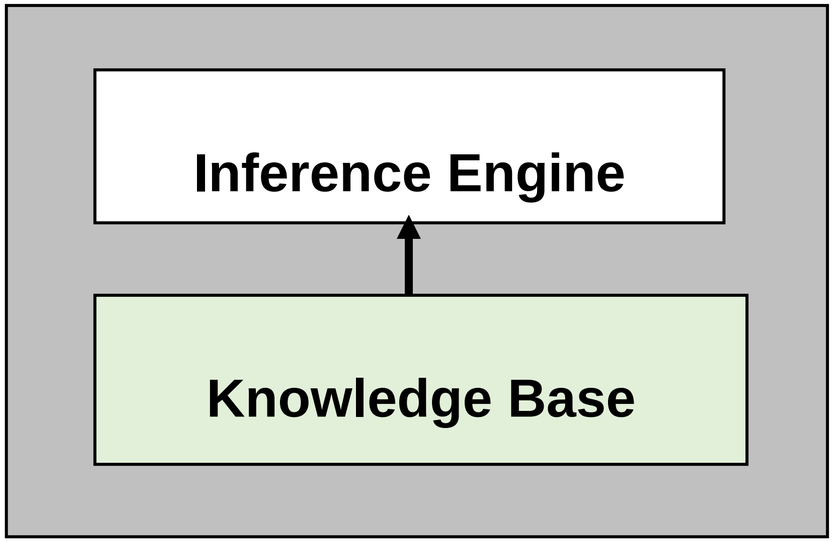
Includes:

- Knowledge on the syntax and the organization of MYCIN's rules (**syntactic knowledge**)
- Knowledge on the MYCIN rules per se that relates to the domain of microbiological infections (**semantic knowledge**)

TEIRESIAS Operation

- The TEIRESIAS system helps improve the MYCIN knowledge base
- Knowledge bugs means incomplete or incorrect rules, which are externalized as irrelevant or incorrect questions, or incorrect conclusions, on the part of MYCIN
- The improvement of the knowledge base is usually done in the context of consulting sessions to solve specific problems - **knowledge acquisition in situ**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



Modifying the Content, not the Structure, of the Knowledge Base

A knowledge acquisition system operates under the assumption that only the content of the knowledge base needs to be modified to improve the performance of the (advisory) system, not the way in which that base is organized and consequently not the way in which the knowledge is applied.

In other words, the general structure of the knowledge base and the reasoning mechanism, which is based on this structure, do not need to be modified.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Functional Requirements of a Knowledge Acquisition System

- After each modification to the knowledge base, it is important to verify that an improvement has indeed occurred
- A significant part of the 'intelligence' of a knowledge acquisition system depends on its ability to propagate the consequences of a modification to all implicated components in the knowledge base so that the knowledge base remains in a consistent state
- As such a knowledge acquisition system needs to know the structural elements of the knowledge base and their interdependencies, i.e., to be an **intelligent structured editor**

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

TEIRESIAS Functionality

- TEIRESIAS can conduct, for a given problem, a detailed comparison of the new result and the way it was derived, with the previous result, and start a discussion with the expert on their differences
- It is also able to identify contradictions at the level of individual rules:

  $A \rightarrow B$

  $A \rightarrow \sim B$

- But it cannot detect contradictions involving multiple rules,

  $A \rightarrow B$, $B \rightarrow C$, and $C \rightarrow D$,

  $A \rightarrow \sim D$

- However, it can detect if a rule is covered by another rule:

  the rule $A \,\&\, B \rightarrow D$ is covered by the rule

  $A \,\&\, B \,\&\, C \rightarrow D$

  Solution: Change the first rule to

  $A \,\&\, B \,\&\, \sim C \rightarrow D$

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

TEIRESIAS meta-knowledge

❑ Syntactic knowledge:

- It concerns the **organization** of the MYCIN knowledge and is expressed through a high-level descriptive language.
- It is independent of the knowledge domain of MYCIN.
- It consists of a **hierarchy of schemas** where each schema describes a category of the representation, e.g. rule, context, clinical parameter, etc. It is a frame with the relevant slots which describe the realization of the instances of the category, the associations of the category with other categories, etc.

❑ Semantic knowledge:

- It concerns the specific knowledge domain and is expressed through **rule models**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Examples of Rule Models

If the rule type is orgrule

Then 1. the premise of the rule should mention culture
2. the premise of the rule should mention the kind of infection
3. the conclusion of the rule must mention identity

If the premise of the rule mentions the source of the culture and
the premise of the rule mentions the kind of infection

Then the premise of the rule must mention the port of entry of the organism

TEIRESIAS could in fact deduce the port of entry from the source of the culture and the kind of infection.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

EMYCIN: Empty/Essential MYCIN

- The knowledge acquisition system concept led to the **shell or skeletal system concept**, which dominated the initial approach to creating expert systems.
- Skeletal systems aim to semi-automate the creation of new expert systems.
- A skeletal system, through its built-in knowledge acquisition subsystem, guides **the creation from scratch of a new knowledge base** for some new knowledge domain.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Functionality for monitoring the performance of a rule base

- **EXPLAIN** a conclusion
- **TEST** the result of some advisory session with the stored correct result
- **REVIEW** the system's inferential conclusions against a library of cases (e.g. success rate), thus monitoring the consequences of rule base modifications

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Systems that have been built through EMYCIN

- HEADMED clinical psychopharmacology
- PUFF disorders of the pulmonary system
- SACON engineering
- ONCOCIN cancer
- CLOT blood disorders
- DART computer networks

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Shell System Methodology

- The shell system 'methodology' was a low-level, or **representation-level**, interpretation of the concept of reuse.
- It became clear that the use of shell systems was a way of understanding expertise because it did lead to the quick construction of a prototype for the expert system (rapid prototyping) which could have been a very distant and imprecise approximation of the intended final system, but through its testing, experts were facilitated to externalize their expertise simply by criticizing the prototype's performance.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Compiled Knowledge

IF:

1. **the infection is meningitis**
2. **the type of meningitis is bacterial**
3. **there is only circumstantial evidence**
4. **the patient is at least 17 years**
5. **the patient is alcoholic**

THEN:

There is considerable evidence that one of the organisms causing the meningitis is pneumonic diplococcus

Behind the specific rule and in particular its premise, there is hidden important knowledge, descriptive and strategic, which cannot be expressed explicitly due to the uniform representation imposed by the (simple) formalism of production rules.

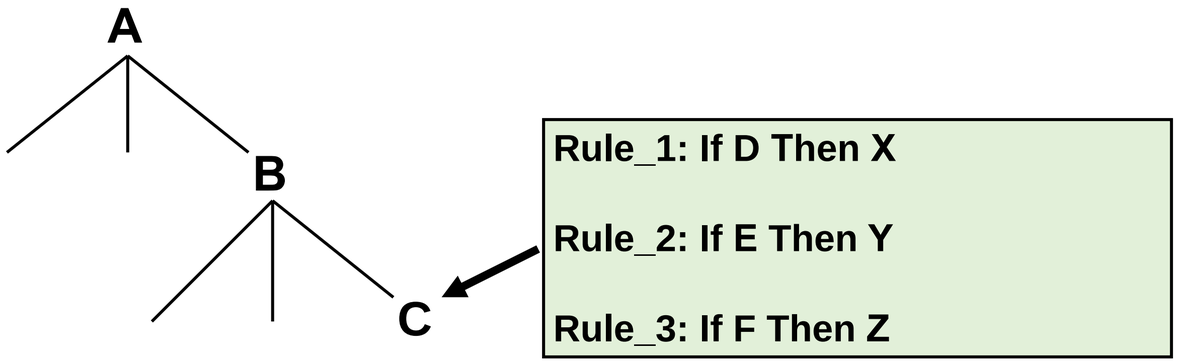
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Implicit Knowledge

Taxonomies of concepts are implicit

Rule_1: If A and B and C and D Then X
Rule_2: If A and B and C and E Then Y
Rule_3: If A and B and C and F Then Z



Explicit Representation

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Implicit Knowledge

World facts or common knowledge is implicit.

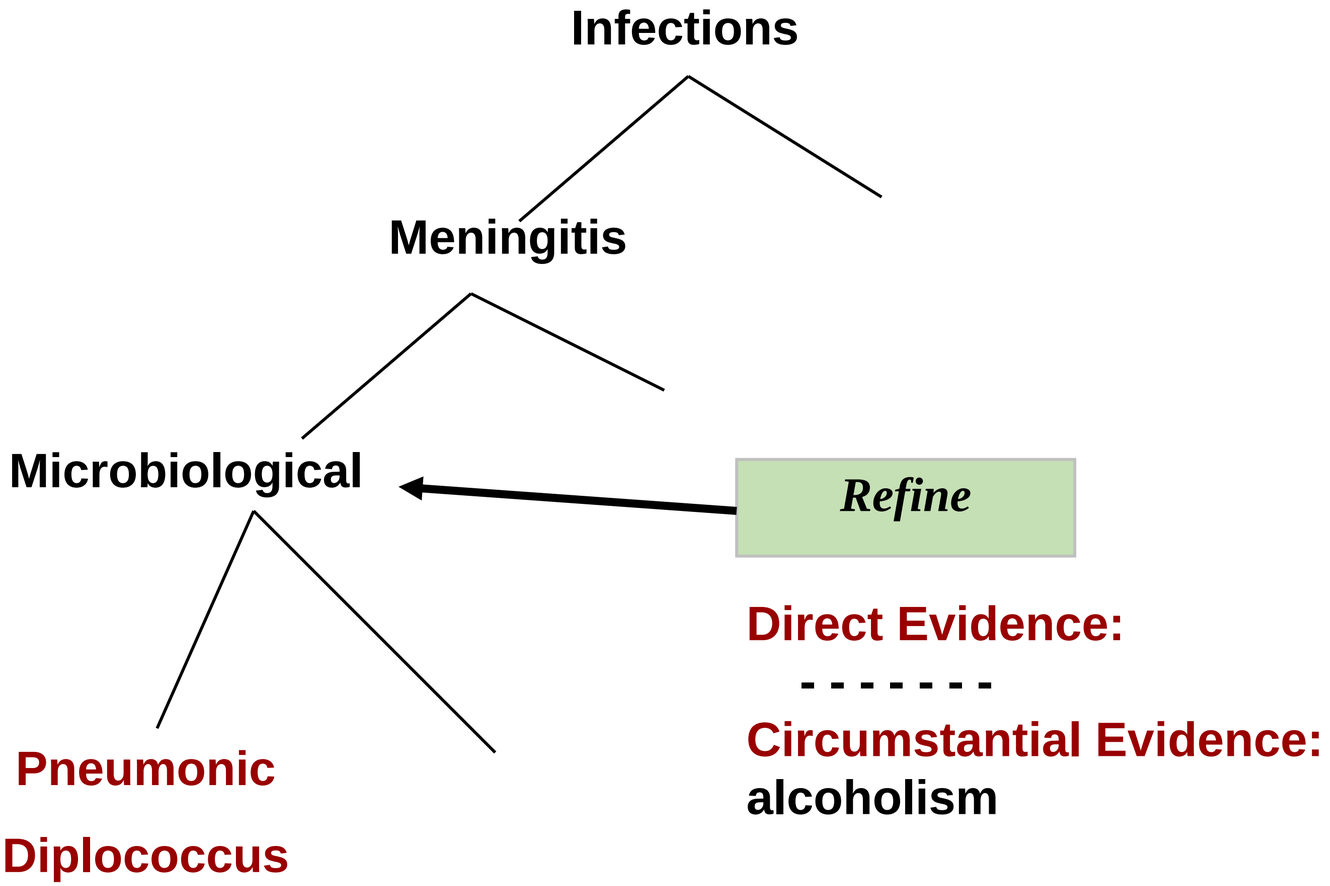
For example, MYCIN does not know that “a child is not usually an alcoholic”.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Implicit Reasoning Strategies

1. Hypothesize and Refine
2. Circumstantial evidence is processed differently in the presence or absence of direct evidence
3. Before asking a question, it should be confirmed that its answer cannot already be deduced from what has been said so far

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Rule Justification

- Another notable omission is the **justification of the rules**, not in canned text form which is obviously sufficient for explanation purposes, but in a form that can be used by the inference engine when solving the problem:
- E.g., why is there a connection between the observation of alcoholism and the hypothesis that the meningitis was caused by pneumonic diplococcus? The causal chain underlying this link is absent from the knowledge base.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example

Consider the following rule:

‘If the age of the patient is under 8, do not administer tetracyclines’.

The canned text justification of the given rule states:

‘tetracyclines in children are absorbed by developing bones, which causes tooth discoloration which is an undesirable physical change and therefore these substances should not be administered to children'.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Rule Justification

- Any causal chain that constitutes the justification of a link can be expressed at infinitely many levels of detail.
- The choice of intermediate concepts is random
  - E.g., in the above justification it is not mentioned how the absorption takes place and under what conditions
- From the point of view of an advisory system what is needed is for the system to be able to 'understand':
  - when violating a rule is justified, and
  - when the application of a rule whose premise does not hold is justified
  - E.g. to 'know' that the above rule may be violated if treatment involving tetracyclines is the only one

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The choice of the term ‘compiled’ knowledge is not accidental. The analogy with the concept of a compiled program is obvious.

Compiled Knowledge – MYCIN Interpretation

It is the transformation of (source) knowledge into a uniform, standardized form of rules where various elements of knowledge appear implicitly or are completely omitted. The compilation is done purely for functionality purposes, with the aim of efficiently solving not necessarily all instances of the problem, but at least most of them.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

PROSPECTOR

- The PROSPECTOR system, developed by SRI, aided geologists in evaluating the favorability of an exploration site or region for occurrences of ore deposits of particular types.
- It discovered a very rich ore deposit.
- Combines production rules and semantic networks in its knowledge representation.
- Bayes Theorem underlies its uncertainty model where pseudo-probabilities are used.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

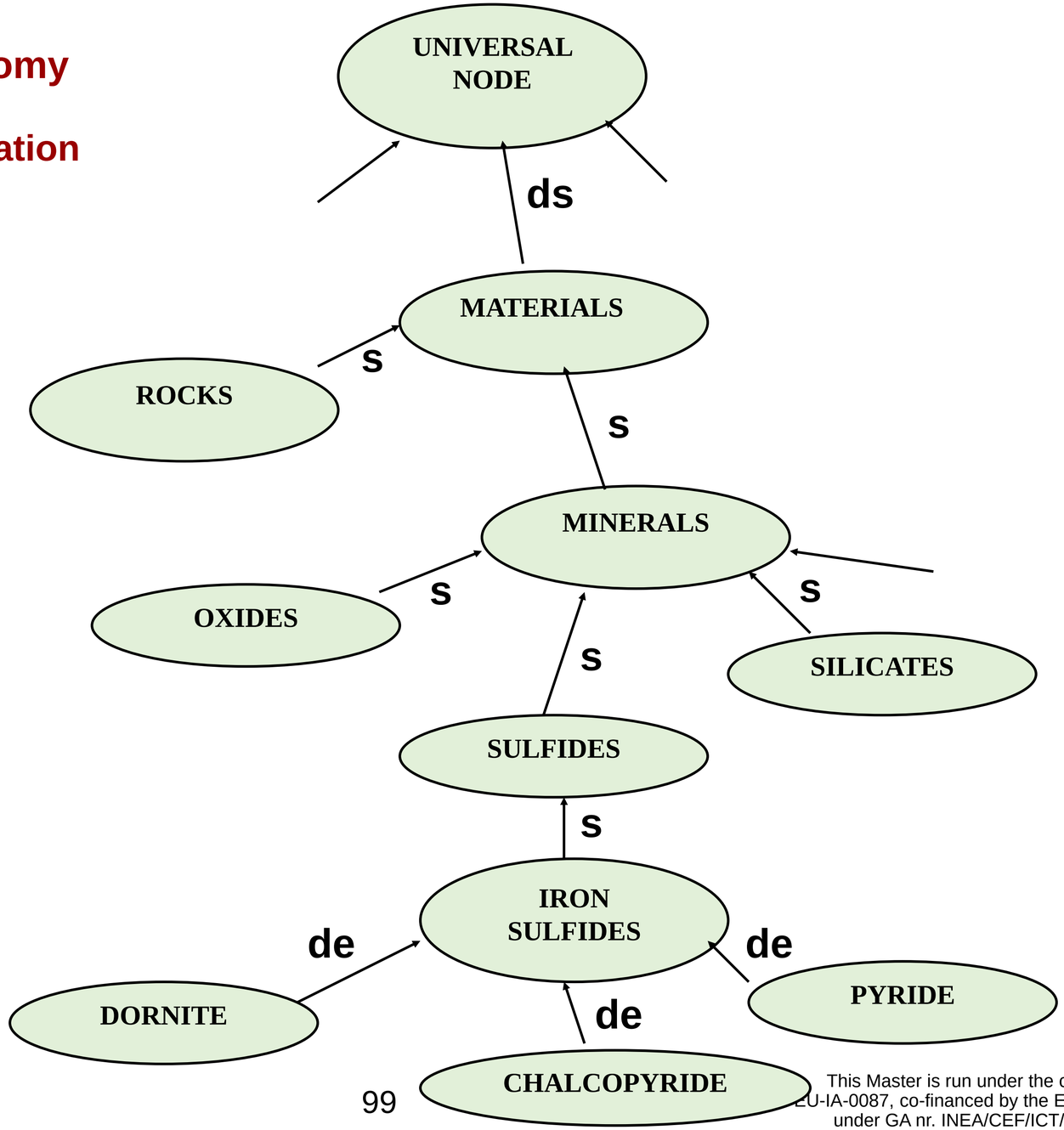
PROSPECTOR's hybrid knowledge representation

- The representation of the system's knowledge base is hybrid, combining:
  - production rules and
  - partitioned semantic networks
- Concept taxonomies
  - rock types, minerals, natural forms, geological ages

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Part of the Materials Taxonomy

Generalization and concretization relations

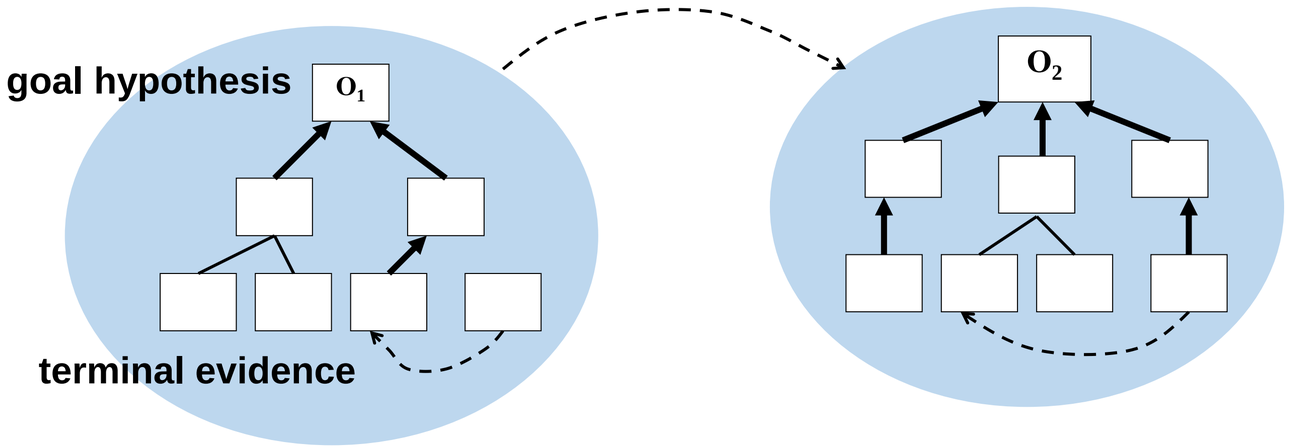


ds: disjoint subset
s: subset
de: distinct element

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Base Organization: Models for Ore Categories



Model (inference network) for ore category O_1.
Each node is a partition.

Model for ore category O_2.

rule **logical link** **contextual link**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Contextual Links

- They specify the order in which hypotheses/evidence should be explored
- They indicate whether a hypothesis/evidence has geological significance only if some other hypothesis/evidence has already been verified
- They aim at strengthening the **dialogue** of the system with its user

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Assertions (Hypotheses/Evidence)

- They are represented as partitioned semantic networks:
  - Nodes represent physical entities, processes, locations, relationships
  - Arcs represent arguments

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example Relations

Binary

AGE-OF
COMP-OF
FORM-OF
GRAIN-SIZE-OF
LOC-OF
SIZE-OF
TEXTURE-OF

Non-Binary

ALTERED-TO
COMPONENTS-OF
CONTAINED-IN
RELATIVE-AGE-OF

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example

A ≡ ‘A RHYOLITE PLUG IS PRESENT’

Analysis of A

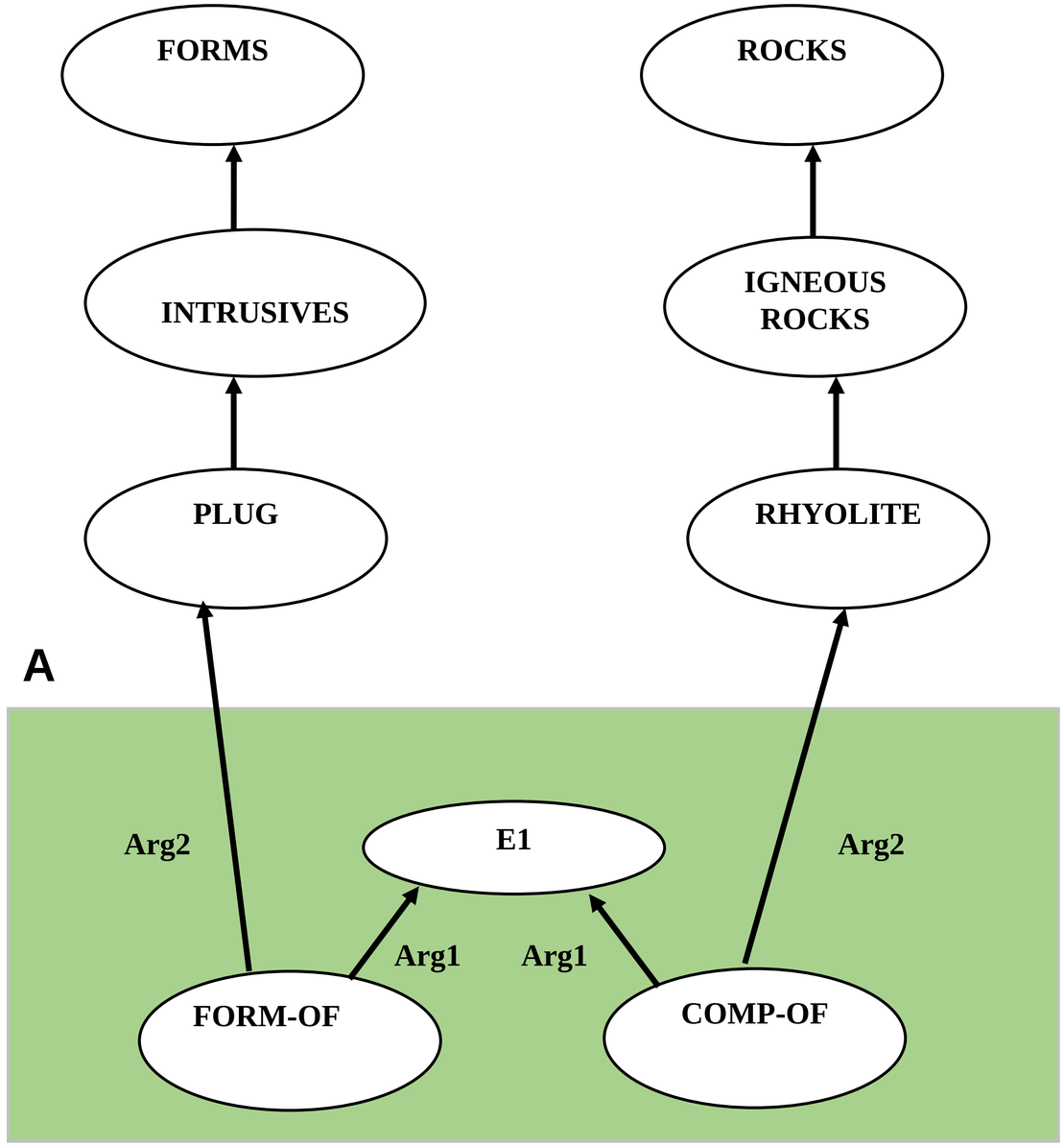
A1: There is a physical entity E1

A2: The composition of E1 είναι rhyolite

A3: The form of E1 is plug

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

A's representation



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Mixed Chaining

- The system combines
  - Goal-driven reasoning (**backwards reasoning**) with
  - Event-driven reasoning (**forwards reasoning**)
  - Implemented through backwards and forwards chaining respectively

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Central Reasoning Processes

- Each reasoning process can be activated
  - Externally, on the initiative of the user
  - Internally, based on the results of some other process

Matching of Observations

- Observations are matched against assertions that represent direct evidence (and are terminal nodes) in the various mineral models
- These observations are mainly entered voluntarily by the user

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Mixed Reasoning/Chaining

Forwards Reasoning (through forwards chaining)

- From evidences that have been activated (that is, their probabilities have been modified) to all involved goal-hypotheses
- The result of this process is to emerge a new ultimate goal-hypothesis, which will be explored within the context of backwards reasoning
- The ultimate goal-hypothesis has the currently highest posterior probability

Backwards Reasoning (through backwards chaining)

- From the currently most likely goal-hypothesis to the (direct) evidence presently believed to have the greatest impact on the given goal-hypothesis

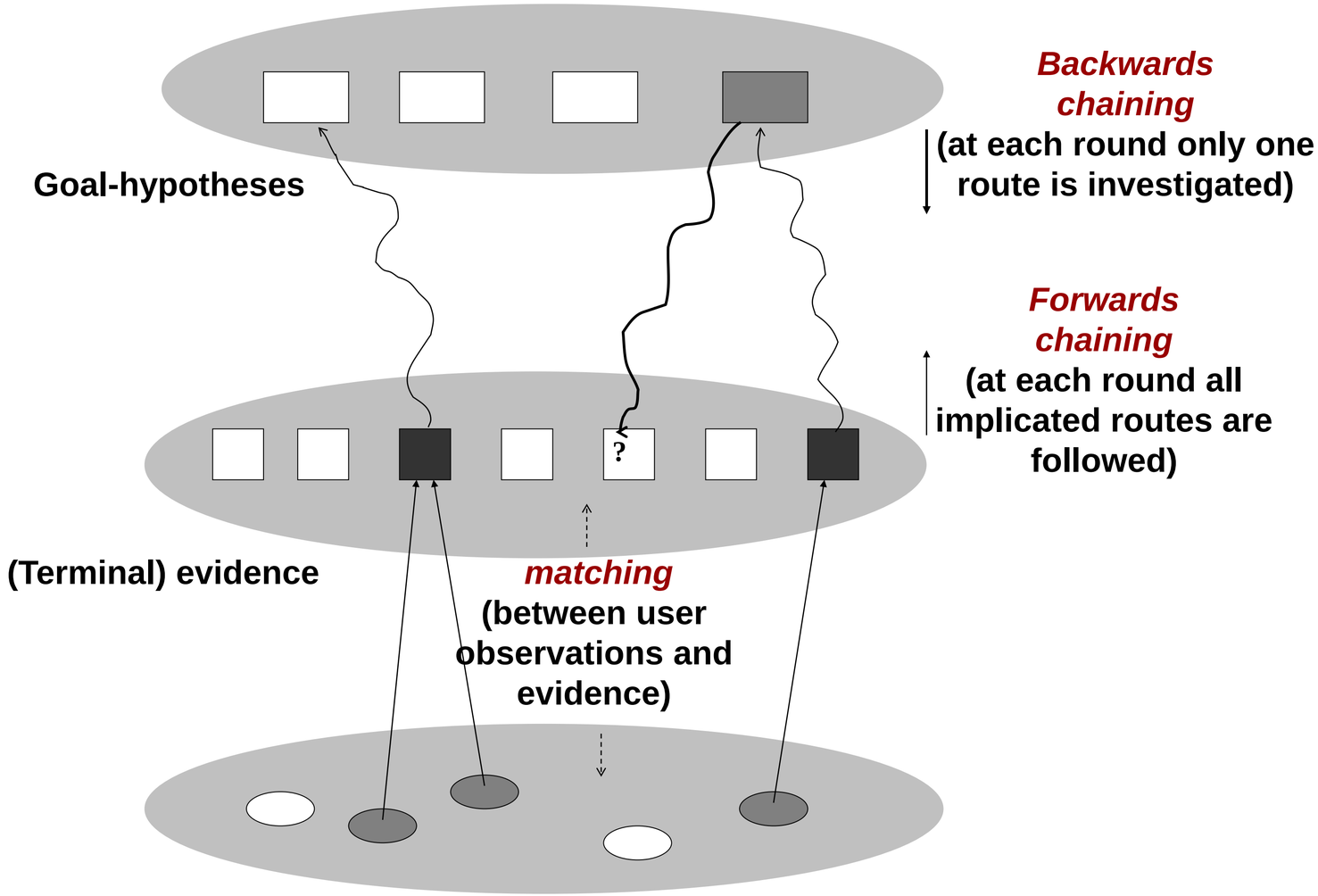
Conclusions

- For those goal-hypotheses whose posterior probabilities are high enough PROSPECTOR can conclude that there is sufficient evidence for the existence of deposits of the ores in question

Co-financed by the European Union
Connecting Europe Facility

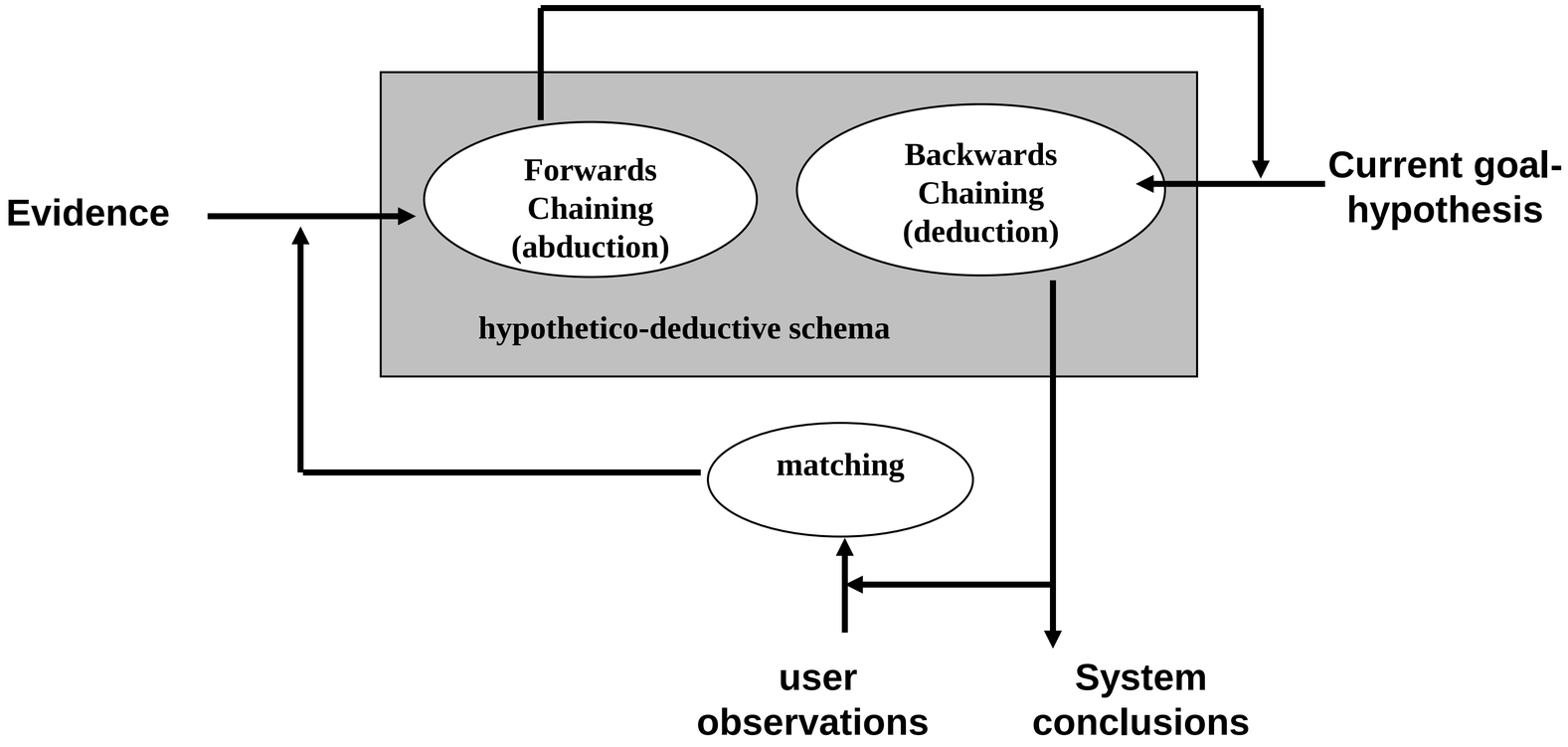
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Reasoning Processes – Matching, Forwards and και Backwards Chaining



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Hypothetico-Deductive Schema

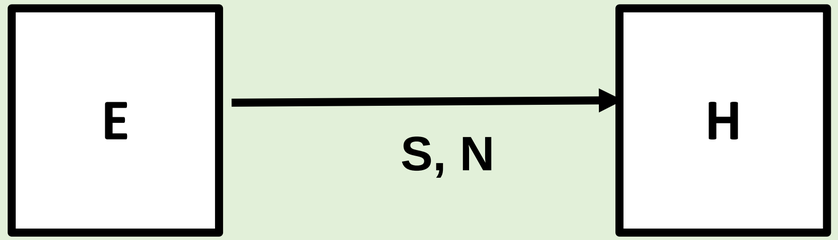


This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Uncertainty Model

- **PROSPECTOR's uncertainty model is based on probabilities, in reality pseudo-probabilities, since these are subjective values given by the experts and not real probabilities**

- **Rule uncertainty**



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Sufficiency Factor, S

$$S = P(E/H) / P(E/\sim H)$$

- The sufficiency factor, S, represents how sufficient evidence E is, to verify hypothesis H
- If E is only associated with occurrences of H (not necessarily all occurrences), E alone is sufficient to verify H
- In this case the probability P(E/~H) approaches 0 and thus S approaches ∞
- S is used when E has been verified

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Necessity Factor, N

$$N = P(\sim E/H) / P(\sim E/\sim H)$$

- The necessity factor, N, represents how necessary evidence E is for hypothesis H
- If E is associated with all occurrences of H, E is necessary for the verification of H; in other words, the negation of E is sufficient for the negation of H
- In this case the probability P(~E/H) approaches 0 and hence N approaches 0
- N is used when E has been refuted

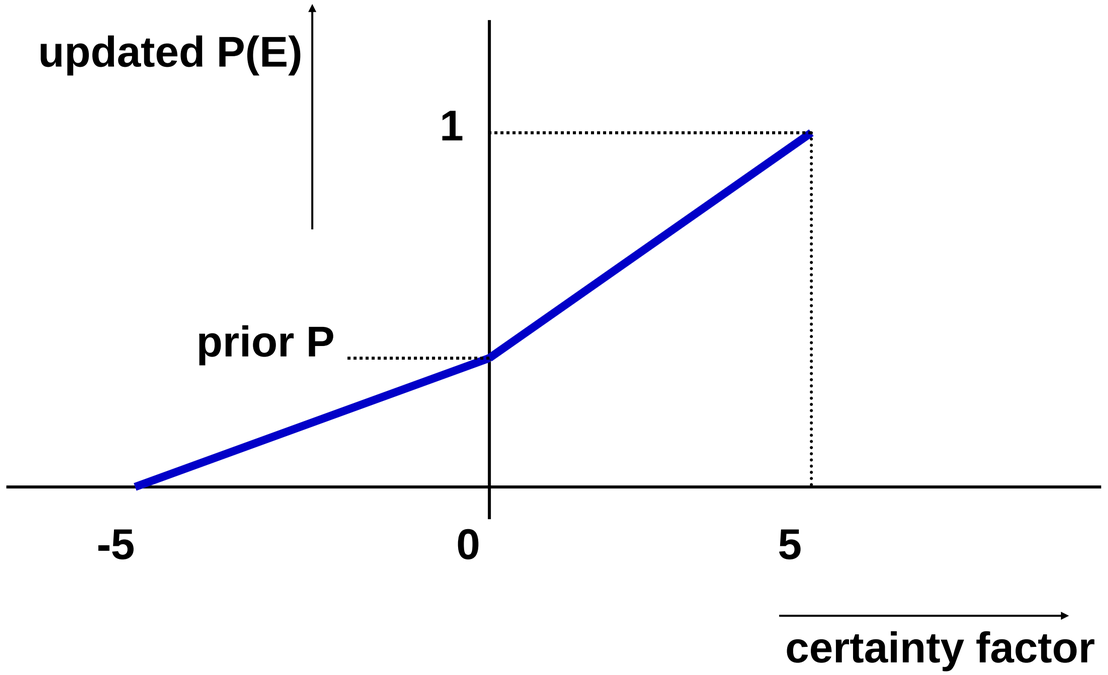
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Evidence Uncertainty

- The various nodes (evidences and hypotheses) in the ore models are associated with subjective, a priori probabilities of existence
- Like S and N, prior probabilities are formulated by experts
- When solving a problem, the probabilities are updated (via forwards chaining) based on the user's observations
- Usually, the user can only specify some degree of certainty, from the range [–5, 5], regarding the presence of the pursued evidence

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Relation between certainty factor and posterior probability



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Combining Rule and Evidence Uncertainty

- **Using Bayes Theorem:**

$$P(H/E) = (P(E/H)\ P(H)) / P(E)$$

- **The collusion of abduction and deduction, that is, the hypothetico-deductive schema, is evident in this theorem:**
  - **The left side of the equation represents abduction while the probability P(E/H) shown on the right side represents deductive reasoning**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Posterior odds

- Using Bayes Theorem, posterior odds can be expressed as:

$$O(H/E) = P(H/E) / P(\sim H/E)$$
$$= (P(E/H) / P(E/\sim H)) \times (P(H) / P(\sim H))$$
$$= S \times O(Y) \quad (1)$$

$$O(H/\sim E) = P(H/\sim E) / P(\sim H/\sim E)$$
$$= (P(\sim E/H) / P(\sim E/\sim H)) \times (P(H) / P(\sim H))$$
$$= N \times O(Y) \quad (2)$$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Relation between probabilities and odds

- **P = O / (O + 1)**

  or O = P / (1 – P)

- **The term odds represents the ratio of the probability that a given hypothesis is true, compared to the probability that it is not true**

  O(Y)=P(Y)/P(~Y)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Using Equations (1) and (2)

- Equation (1) is used to update the probability of hypothesis H when evidence E has been categorically verified
  - The updating is based on the sufficiency factor
- Equation (2) is used to update the probability of H when the evidence E has been categorically refuted
  - The updating is based on the necessity factor
- These two equations cover the extreme cases

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Uncertain Evidence

- What happens when there is uncertainty in the evidence? Let E′ be the observations relating to evidence E. The general update equation is the following:

$$O(H/E') = \lambda \times O(H)$$

Where $\lambda = S$ when $P(E/E') = 1$

$\lambda = N$ when $P(E/E') = 0$, that is $P(\sim E/E') = 1$

$\lambda = 1$ when $P(E/E') = P(E)$,

that is nothing is known about E

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Probability Theory

- Based on the general theory of probabilities,

$$P(H/E') = P(H,E/E') + P(H,\sim E/E')$$

$$= P(H/E,E')\,P(E/E') + P(H/\sim E,E')\,P(\sim E/E')$$

- If it is known that E exists or does not exist, the observations E′ in relation to E do not provide any further information about the hypothesis H. That is, it can be considered that:

$$P(H/E,E') = P(H/E) \text{ και}$$

$$P(H/\sim E,E') = P(H/\sim E)$$

- This leads to the equation

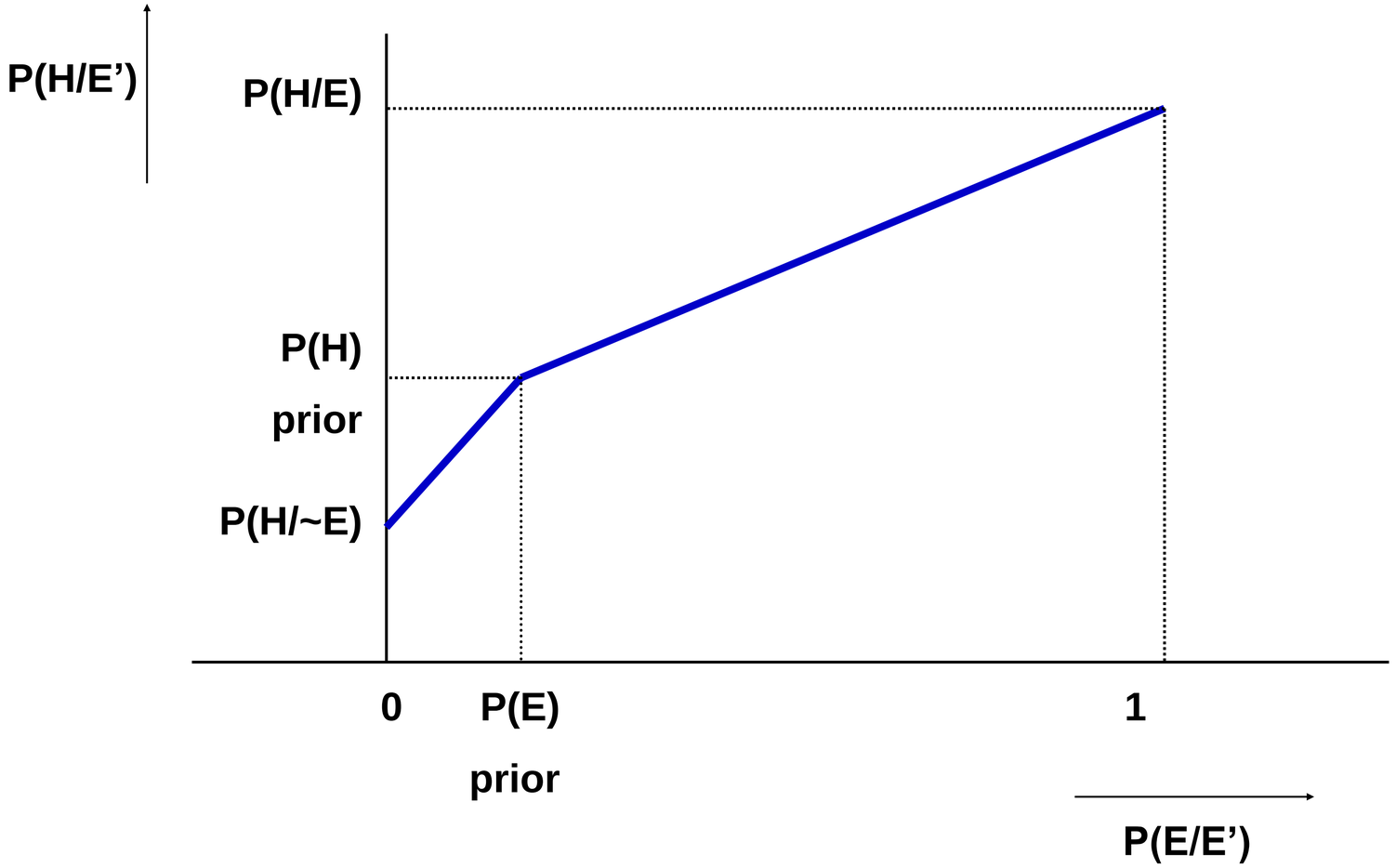
$$P(H/E') = P(H/E)\,P(E/E')$$

$$= P(H/\sim E)\,(1 - P(E/E'))$$

- Finally, if nothing is known about evidence E, that is $P(E/E') = P(E)$, then $P(H/E') = P(H)$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Probability of Conclusion as a Function of the Probability of Premise



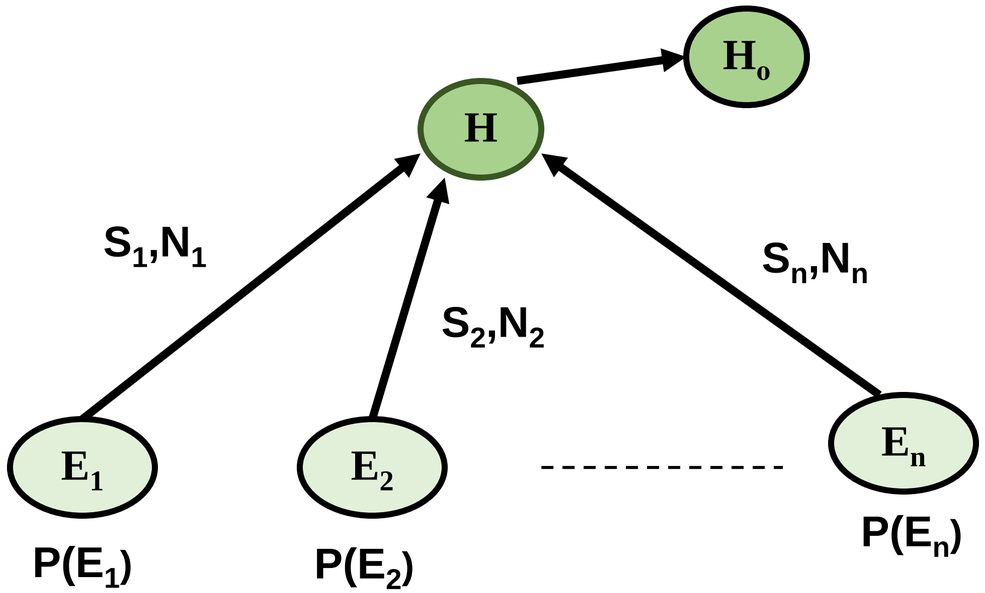
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

If $P(E/E') > P(E)$,
then $\lambda = S \times ((P(E/E') - P(E)) / (1 - P(E))$,
that is the proportion of S.

If $P(E/E') < P(E)$,
then $\lambda = N \times (P(E/E') / P(E))$,
that is the proportion of N.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Accumulating evidence for the same hypothesis



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Multiple Evidence

When there is multiple evidence, say E_i, i = 1, .., n, for the same hypothesis H, which are conditionally independent under H and ~H, the update equation is generalized as

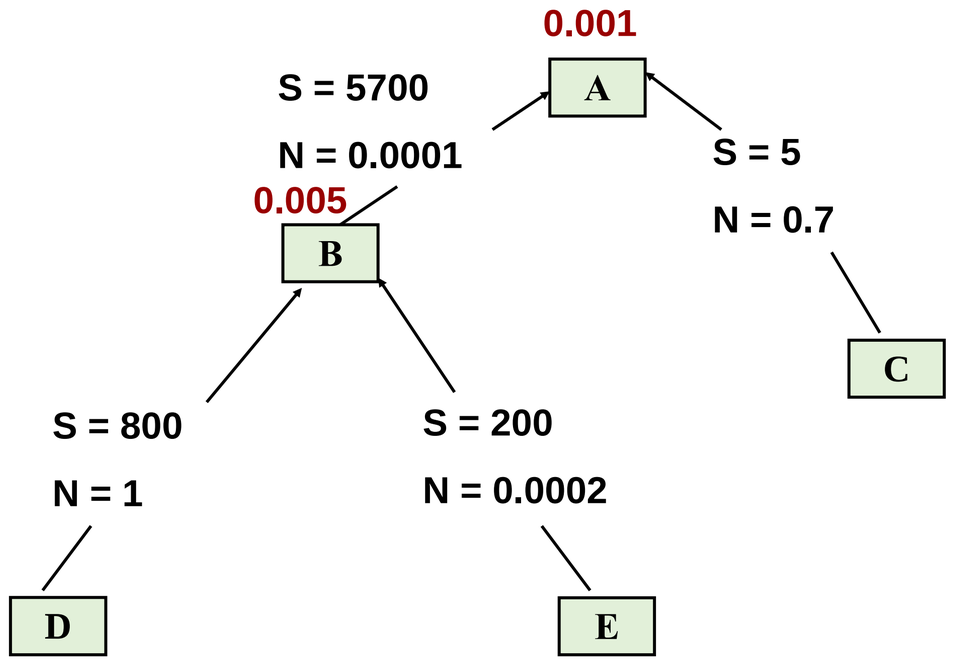
$$O(H/E_1, E_2, .., E_n) = \left\{ \prod_{i=1}^{n} \lambda_i \right\} O(H)$$

where λ_i is the multiplier for evidence E_i.

Each new terminal evidence is propagated across all relevant chains, thus updating the (posterior) probabilities of all implicated goal-hypotheses.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example



Inference Tree

Assume the following posterior probabilities:

P(D) = 1
P(E) = 1
P(~C) = 1

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

What is the posterior probability of the goal-hypothesis A?

$$O(B/D,E) = 800 \times 200 \times (0.005 / (1 - 0.005)) = 804.016$$

$$\therefore P(B/D,E) = 804.016 / (804.016 + 1) = 0.9988$$

$$\therefore \lambda_{B'} = 5700 \times ((0.9988 - 0.005) / (1 - 0.005)) = 5693$$

$$O(A/B',\sim C) = 5693 \times 0.7 \times (0.001/ (1 - 0.001)) = 3.989$$

$$\therefore P(A/B',\sim C) = 3.989 / (3.989 + 1) = 0.7995$$

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristic for guiding the backward chaining

- Give priority to rules with small N, when the pursued hypothesis has a small probability, and give priority to rules with high S, when its probability grows.
- In the first case, the possibility of rejecting the hypothesis investigated and in the second case the possibility of its verification.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

INTERNIST-1

- The INTERNIST-1 system was developed at the University of Pittsburgh (Dr Jack Myers) with the aim of assisting physicians in performing differential diagnosis in the field of general internal medicine.
- INTERNIST-1 has the most extensive knowledge base among medical expert systems, covering 80% of general internal medicine and taking 15 person-years to build.
- More specifically its knowledge base include 600 diseases or categories of diseases with 2600 links between them and 3550 manifestations with 6500 links between them.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

INTERNIST-1

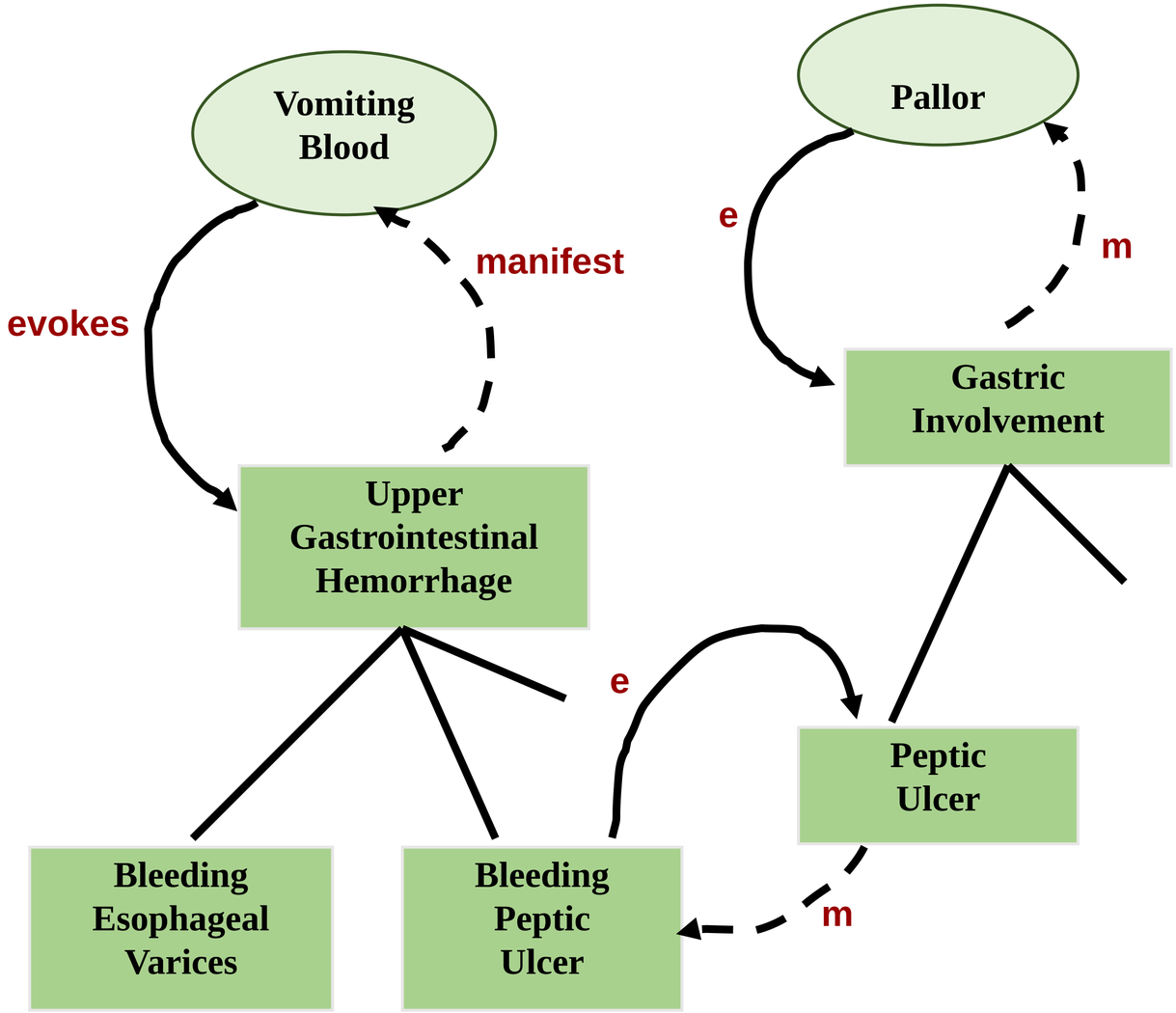
Dr Myers was noted to state “The method used by physicians to arrive at diagnoses requires complex information processing which bears little resemblance to the statistical manipulations of most computer-based systems”

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTERNIST-1 Knowledge Base

Hierarchically organized, around the concept of a disease profile.

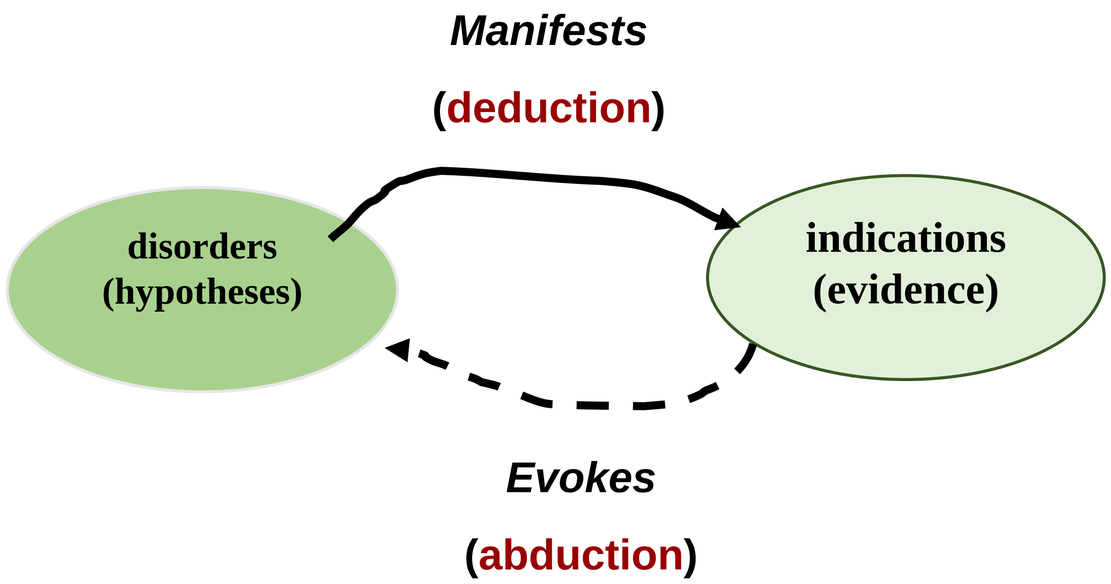


This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Base

- The central entities are **diseases** and **manifestations** with the following relations:
  - Relation **Manifest** from diseases to symptoms: A disease, D, manifests a symptom, M, to a given degree, from the set {1, 2, ..., 5}, which indicates the frequency of occurrence of M in cases of A.
  - Relation **Evokes** from manifestations to diseases: A manifestation, M, evokes the hypothesis of a disease, A, to a given degree from the set {0, 1, ..., 5}. The degree indicates how indicative M is of A.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



Degree of Importance of Indications

Each indicator, E, is assigned a degree of importance (**import**) from the set {1, 2, ..., 5} which indicates the universal importance of the indicator, regardless of its various causes, and thus the degree to which its presence to a patient needs to be explained, i.e., be ascribed to a cause.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Frame Structure for Diseases/Disease Categories

- **Subcategories: . . . . . .**

- **Indication:**
  indicator-1 evocative-strength manifest-degree
  . . . . . .

- **Complementary relations with other diseases:**

  - **predisposition-for:**
    disease evocative-strength manifest-degree
    . . . . . .
  - **causes: . . . . . .**
  - **coincides-with: . . . . . .**
  - **precedes: . . . . . .**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

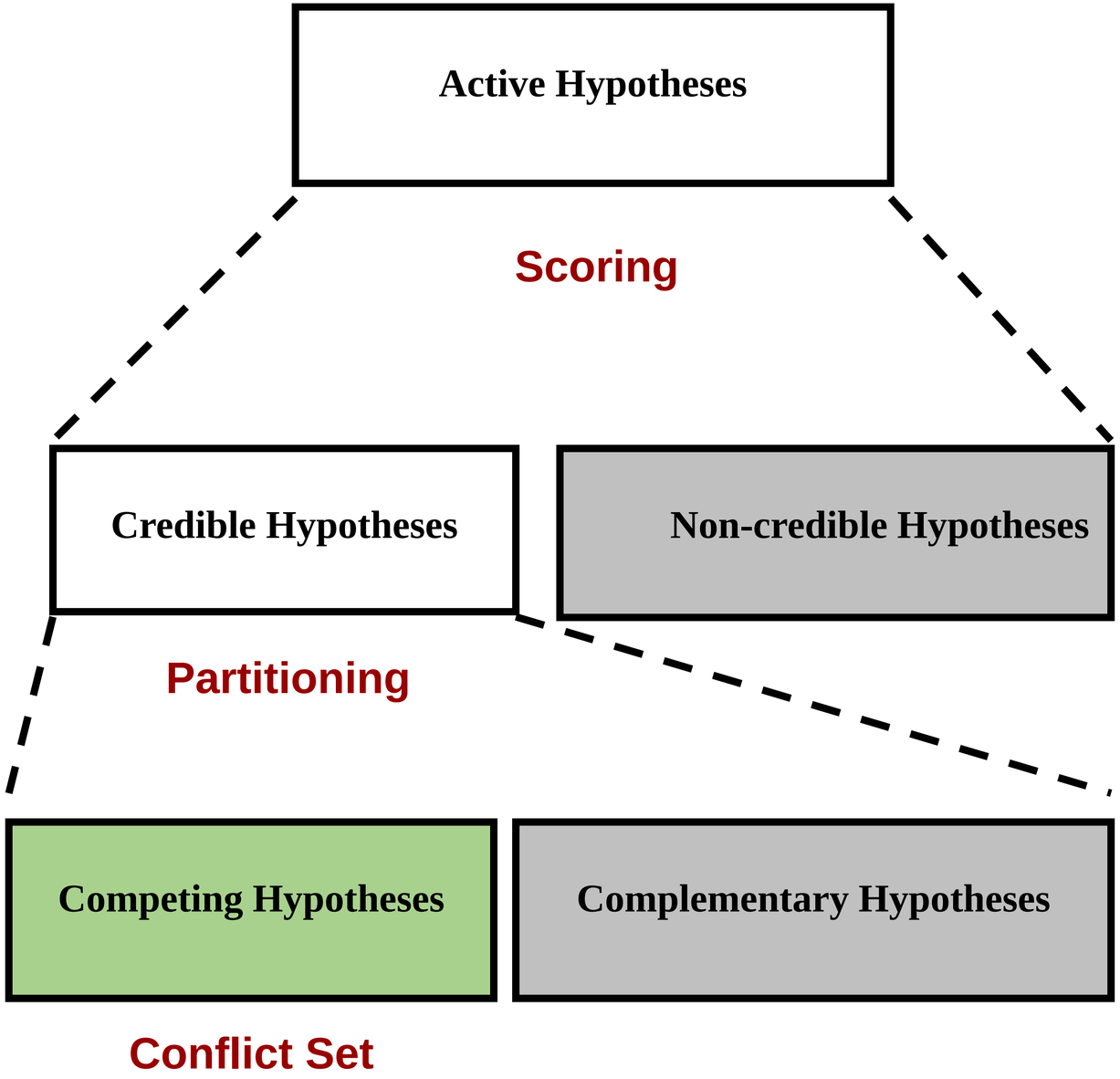
Reasoning Processes

- Reasoning is based on the **hypothetico-deductive** scheme
- Evidence refers to **positive** and **negative** indications
- Positive indications whose **import** is at least 3 activate disease hypotheses – relation Evokes
- Active hypotheses are evaluated – **scoring function**
- Hypotheses, selected as **credible**, are those whose score exceeds a certain threshold
- Credible hypotheses are divided into **complementary** and **competing** to the most credible hypothesis
- The most credible hypothesis together with its competitors constitute the current **conflict set** which is investigated based on its composition

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Partitioning heuristic:
Two hypotheses can be regarded as competing if the positive evidence unexplained by one is a subset of the positive evidence unexplained by the other.



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Conflict Resolution

- The reasoning that is carried out is mainly **deductive** (based on the Manifest relation) with the aim of extracting new information from the user
- According to the composition of the conflict set various **strategies** are applied to resolve the conflict

Alternative Strategies (heuristics)

- **Ruleout**: It is applied when in the conflict set there are at least 5 hypotheses which, in terms of credibility, are close enough to the most credible hypothesis. Looking for 'cheap' indications which are necessary for some of the opponents.
- **Discriminate**: It is applied when in the conflict set there are 2 to 4 close rivals of the most credible hypothesis. Not-so-cheap indications are sought provided that they support one opponent at the expense of another.
- **Pursue**: It is applied when the second-best competing hypothesis is relatively far from the most credible hypothesis. Indications are sought, regardless of cost, with the goal of verifying the most credible hypothesis.

Completion of reasoning process - When a hypothesis is verified, the positive indications explained by it, i.e., related to it under the Manifest relation, are considered 'covered'. The process continues if high import positives remain uncovered (unexplained).

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Hypothesis Evaluation (Scoring) – Uncertainty Model

- Uncertainty at the level of evidence is not allowed.
- Each positive indication can have only one cause (disease) which is its explanation, although the simultaneous existence of multiple diseases in the same person is allowed.
- For every active hypothesis four lists are formed:
  - L1, the **unexplained positive indications**, the existence of which can be **explained** by the hypothesis in question.
  - L2, the hypothesis manifestations which have been **refuted**.
  - L3, the unexplained positives that are **not related** to the hypothesis.
  - L4, the hypothesis manifestations whose **truth state is unknown.**

- **Scoring function:**

$$Score(H_i) = Positives(H_i) - Negatives(H_i) + Bonus(H_i)$$

  - **$Positives(H_i)$** – summarizes evidence in favor of the hypothesis from the elements of L1 (**degrees of sufficiency**).
  - **$Negatives(H_i)$** – summarizes evidence against the hypothesis from the elements of L2 (**degrees of necessity**) and L3 (**global importance**).
  - **$Bonus(H_i)$** – summarizes evidence in favour from already confirmed hypotheses that are related to hypothesis H_i through some complementary relation.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTERNIST 1: Reasoning Critique

- The purely sequential way of building complex solutions is at a disadvantage
  - The system does not have alternative, "complete" pictures of what is probably happening to the patient
  - There is no way to undo previous conclusions
- Uncertainty is not allowed at the level of evidence
- A positive indication cannot have multiple causes at the same time

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Summary

- The Expert Systems Technology
- A strong alternative to weak methods
- Deduction, Abduction, Induction
- First Generation Expert Systems: MYCIN, PROSPECTOR, INTERNIST-1
- User Interface Requirements - Explanation
- Models of Uncertainty
- Knowledge Acquisition – Expert System Shells
- Compiled Knowledge

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September – December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Deep Knowledge-Based Systems:
the second generation of expert systems

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 8

Deep Knowledge-Based Systems: the second generation of expert systems

CONTENTS

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on deep knowledge-based systems: the second generation of expert systems, students will be able:

1. To explain the notion of deepness with respect to knowledge-based systems, particularly the second generation of expert system.
2. To discuss the distinguishing features of the second generation of expert systems.
3. To discuss the use of multiple models (structural knowledge and support knowledge) in NEOMYCIN, as well as the representation and interpretation of strategic knowledge in this system, and the production and semantics of strategic explanations.
4. To explain the generic problem-solving method of heuristic classification.
5. To present an alternative meaning of the notion of ‘compiled knowledge’ and to compare it with MYCIN’s proposition.
6. To explain the collaborative model of specialists and the distributive architecture of MDX, and its auxiliary subsystems.
7. To present the generic tasks architecture and to analyze the notion of reusability with respect to deep knowledge-based systems.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

The notion of deepness with respect to knowledge-based systems

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Deep vs Shallow

- Early knowledge-based (expert) systems were characterized as **shallow**, aiming for the second-generation to be characterized as **deep** systems
- Deepness/shallowness are relative (not absolute), multi-dimensional notions encompassing:
  - Human-Computer Interaction
  - Problem-Solving Flexibility
  - Extensibility
- All dimensions must be analyzed wholistically to reveal the root causes of the limitations

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

‘Deepness’ of human specialized knowledge/expertise

- Human specialists can deal with exceptional cases, not just routinely occurring cases – **flexibility**
- They refer clients to other specialists when a client’s case is outside their own expertise – **self-reflection** on limits of their knowledge/expertise
- They continuously upgrade (revise) their knowledge/expertise, both through their own experience as well as from the literature or reports of other experts - **learning**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Limitations regarding Problem-solving flexibility

- Monolithic, rigidly-applied reasoning
  - Inability to dynamically plan its reasoning strategy for a specific case, based on the characteristics of the case
  - Alternative, orthogonal strategies not supported
- Performance degrades dramatically when dealing with difficult (rare) cases
- Inability to recognize that a problem case is at the periphery or outside of its area of expertise

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Limitations regarding Human-computer interaction

- Inadequate user interface
  - Information required to be entered in very specific terminologies and formats
  - Historic information on a case not maintained
- Inadequate dialogue structure
  - System raises incoherent or redundant questions
  - User not allowed to volunteer information or focusing guidance
  - User not allowed to revoke an answer, or to pursue the effects of an alternative answer (to see "what if . . . .")
- Inadequate explanation structure
  - "Explanations" are just rule playbacks, and not meaningful
  - Explanations by and large are not user-tailored and do not cover all the explanation needs of the user

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Limitations regarding Extensibility (maintainability)

- Difficult to modify the system knowledge, both manually and automatically; consistency checks not facilitated
- Inability of the system to evolve based on its experiences in problem solving

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Cause of Limitations

- The cause of the serious limitations of the first generation of knowledge-based (expert) systems was attributed to the fact that important knowledge was completely absent or appeared in an implicit way.
- Specifically, three types of knowledge did not appear explicitly:
  - **'How'** Knowledge
    - control or strategic knowledge
  - **'What'** Knowledge
    - descriptive functional knowledge
  - **'Why'** Knowledge
    - justification knowledge or causal knowledge

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Main Distinguishing Features of Second Generation

- The use of multiple models and corresponding reasoning mechanisms. There are usually two models:
  - The **heuristic model**
  - Some **'deeper' model**, mainly a causal model
- Knowledge Level Design
  - demonstrating higher abstraction in how to solve relevant problems
  - more rational way of development
- Greater emphasis on the ability of the system to learn and self-improve, where for example the heuristic model can be gradually generated from the 'deeper' model, in a (semi)automatic way

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Reusability

- In the **first generation**, it was purely at the **representation or implementation level**:
  - It was manifested through the concept of the **system shell** which involved the reuse of the symbolic way of representing knowledge as well as the code of the corresponding reasoning mechanism
- The **second generation** provides new interpretations for reusability, mainly at the **level of knowledge:**
  - reuse of conceptual knowledge models
  - reuse of knowledge bases or symbolic units, i.e., code
  - Reusable modules (conceptual or symbolic) are not necessarily complete views of a system, as is the case with system shells, but are simply **building blocks** that need to be composed to produce the complete view of the system

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

NEOMYCIN: multi-modelling, strategic explanations

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

NEOMYCIN

- The NEOMYCIN system is the 'reconstruction' of MYCIN
- It was also developed at Stanford University, with William J. Clancey as principal investigator

Key motivation behind the creation of NEOMYCIN

- The various shortcomings and inherent problems of GUIDON, MYCIN's teaching system
- GUIDON's failure lay in the fact that MYCIN's reasoning, in other words the backwards chaining of rules, which GUIDON was trying to teach, did not match human reasoning

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Central system goal

- The explicit **modelling of strategic knowledge** as a basic precondition for the effective teaching of diagnostic reasoning and the accurate interpretation of a student's behaviour during problem solving.

- In addition, the scope of the **knowledge base was expanded** beyond microbiological infections of the blood by the addition of other categories of infections.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

NEOMYCIN's Knowledge Base

- Knowledge about infections is described in two different ways:
  - **Taxonomic Model** – structural knowledge
  - **Causal Model** – support knowledge

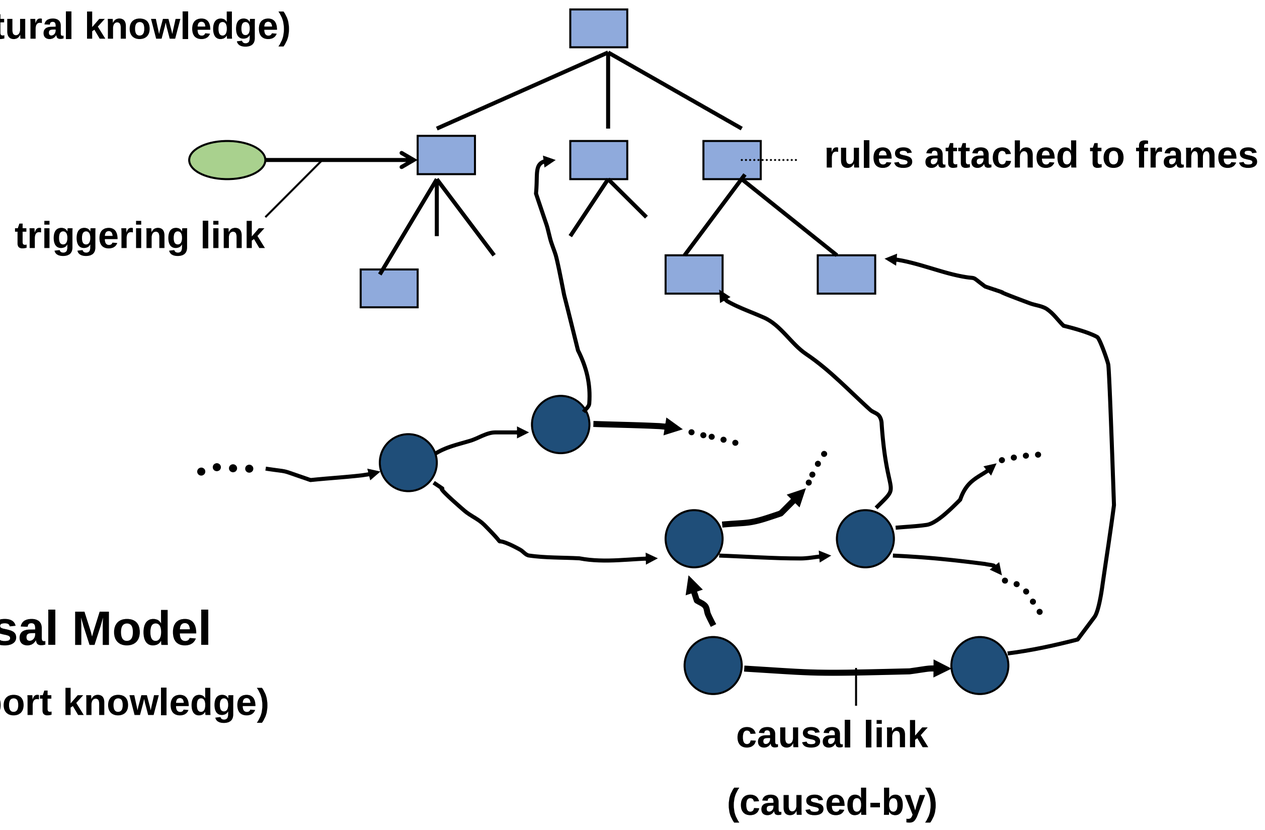
- The taxonomic model is considered the heuristic model and the causal model the 'deep' model.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

NEOMYCIN's Descriptive Knowledge Models

Taxonomic Model

(Structural knowledge)



Causal Model

(support knowledge)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Taxonomic Model Representation

Each node of the taxonomy is a **frame**, which among others contains all the **rules** related to the given disease or class of diseases.

Taxonomic Model Reasoning

- The **'hypothesize and refine'** strategy is applied
- By using **triggers**, the initial goal is to activate an intermediate node of the taxonomy and from there gradually select successor nodes (subcategories) until a terminal node (individual disease) is selected.
  - "trigger" example: "headache along with neck stiffness triggers the meningitis hypothesis"

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Causal Model

- It depicts the given knowledge at a deeper level
  - The relationship that connects the nodes in the various causal chains is **'may-be-caused-by'**
- Therefore, chains are **abductive** in form (from indications to causes, backwards in time), in contrast to taxonomic paths which are **deductive** in form
- Some nodes in the causal chains also belong to the taxonomic model, not necessarily as terminal nodes
  - These nodes are the connecting links between the two models

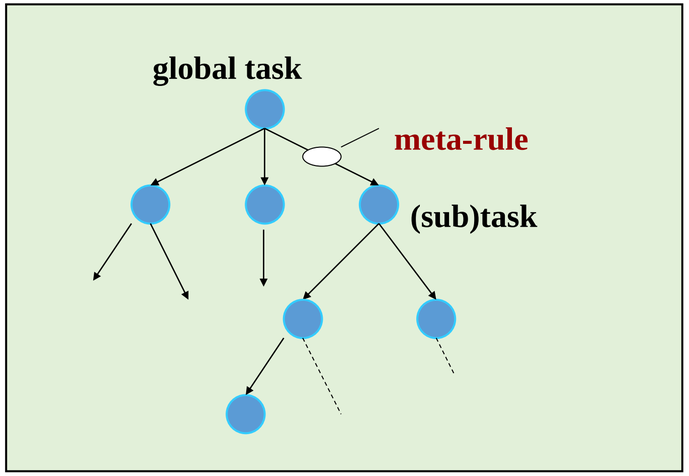
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Alternative Problem-Solving Strategies

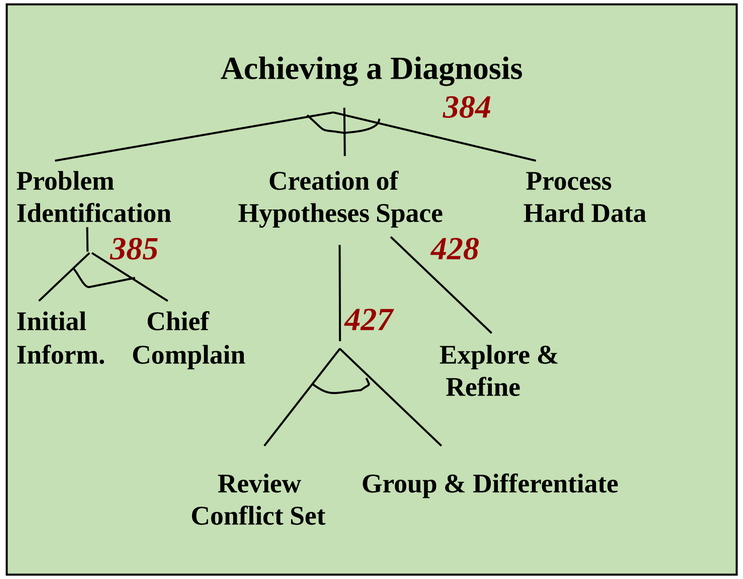
- The taxonomic model provides a more efficient way to solve problems.
- The causal model leads to a more thorough and detailed search for the solution. This strategy is expected to be useful in relation to difficult problems that are not adequately covered by the abstract rules of taxonomy nodes.
- The main use of the causal model is to justify, at a deeper level, the solutions produced through the taxonomic model, where the two models are applied in a purely collaborative manner.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Modelling Strategic Knowledge



Linking tasks to subtasks via meta-rules



Part of NEOMYCIN's Task Taxonomy

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Meta-rules

- The various strategies regarding the execution of tasks, which result in the involvement of other (sub)tasks, are expressed in a purely declarative way in the form of meta-rules.
- The premise of a meta-rule states the conditions governing the use of that strategy, while its action states the subtask or sequence of subtasks involved.
- Reasoning knowledge is expressed in an **abstract way**, that is, a way independent of the knowledge domain.
- Meta-rules refer to abstract concepts such as 'indication' and ‘cause' rather than specific concepts such as 'headache' and 'meningitis'.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example Tasks and Meta-rules

Task: Group & Differentiate

Meta-rule:

If there are two elements in the conflict set, which differ on some characteristic

Then pose a question to the user in relation to that characteristic

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Task: Test Hypothesis

Meta-rule:

If the element under question is closely related to the hypothesis

Then apply the relevant rules with the goal of answering the question

Meta-rule:

If the element under question will result in increasing the credibility of the hypothesis

Then apply the relevant rules with the goal of answering the question

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Task: Explore & Refine

Meta-rule:

If the explored hypothesis has some refinement that has not been explored yet

Then the refinement should be **explored**

Task: Characterize Data

Meta-rule:

If there is some unknown information in relation with the most recent datum

Then then this information should be **explored**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Task: Exploring

Meta-rule:

If the desired information is a subcategory of some category that has been excluded

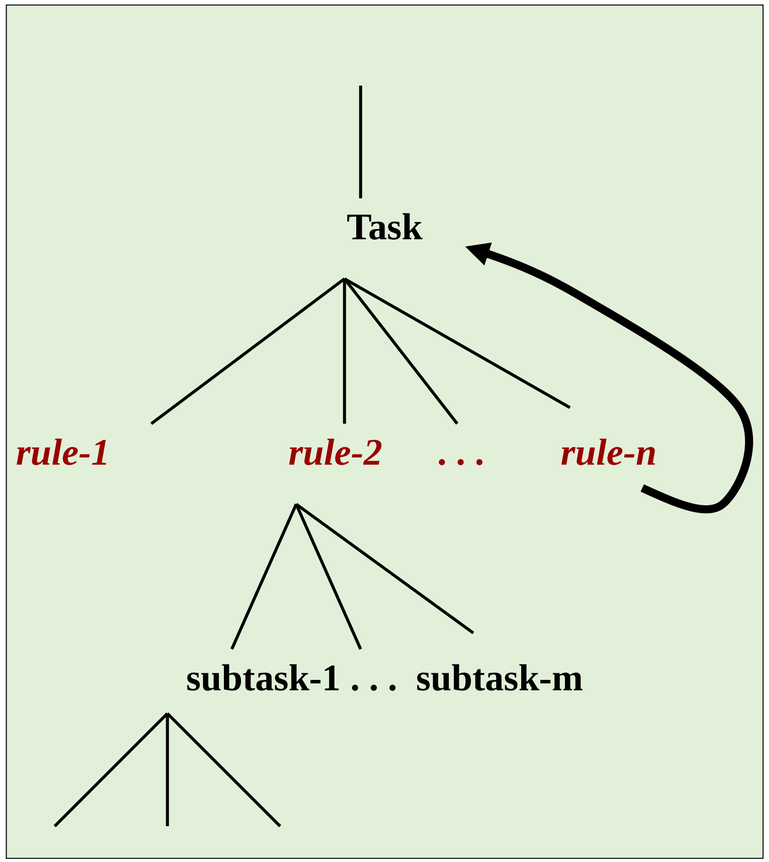
Then **draw the conclusion** that the information is not valid

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

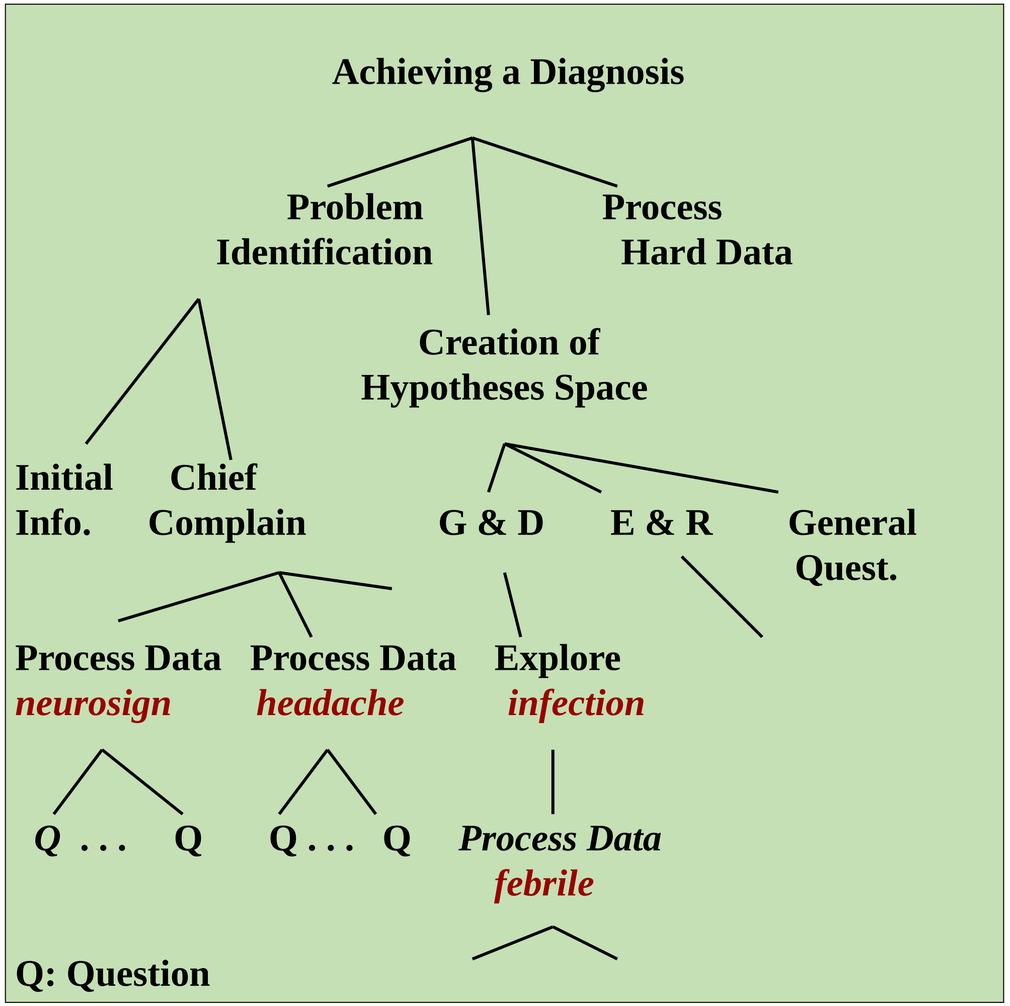
Task Interpreter

- The meta-rules of the task under execution are interpreted based on forward chaining.
- The rules are arranged in a given order. Usually, the action of the last rule is that the execution of the task is complete, that is, its premise states the termination condition.
- The premises of the rules are successively examined, each time starting from the beginning of the sequence of rules, and the first rule in the sequence whose premise is verified is applied, until either the last rule is verified, or some rule that expressly states the termination of the execution of the task is verified.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423



Meta-rule interpretation



Inference Tree

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Strategic Explanations

- The inference tree records all the reasoning of the system in the context of a given consultative conversation
- The way of representing and interpreting strategic knowledge allows the system to provide the user with **strategic explanations**:
  - at a **specific level** (applying strategies in specific situations) as well
  - at an **abstract level** (general strategies)
- As in MYCIN, there are two types of explanations, 'Why?' explanations and 'How?' explanations.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Strategic Explanations

Strategic ‘WHY’ Explanations

- A 'Why?' type question from the user refers to a specific task and means 'Why was it decided to perform this task?’
- The decision rationale is provided by the meta-rule which represents the reasoning strategy linking the two tasks in the specific contexts and more precisely the premise of the meta-rule.

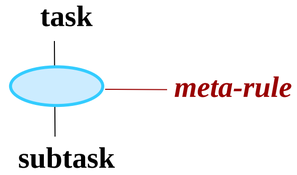
Strategic ‘HOW’ explanations

- A 'How?' type question also refers to a specific task and means 'How was this task accomplished?’
- Such queries concern non-terminal tasks (in the inference tree).
- The explanation consists of the (instantiations of) sub-tasks that had to be involved based on the interpretation of the meta-rules for that task, in other words the actions of the meta-rules that have been applied.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Strategic "Why?" Explanations

Why was the decision to execute a specific subtask taken?



The subtask justification is given by the premise of the meta-rule.

Example

Explore and Refine
(*meningitis*)

Explore Hypothesis
(*bacterial-meningitis*) **Why?**

If **the explored hypothesis has some refinement that has not been explored yet**
Then **the refinement should be explored**

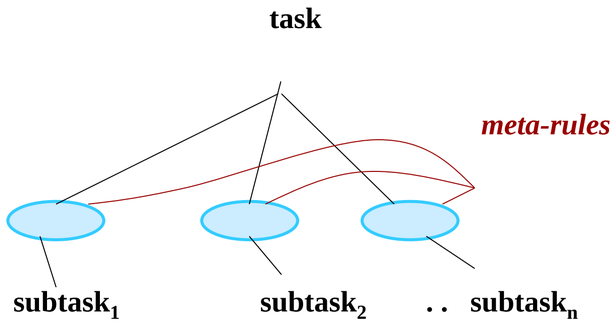
Meningitis is the explored hypothesis.

Bacterial meningitis is one of its refinements that has not been explored yet.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

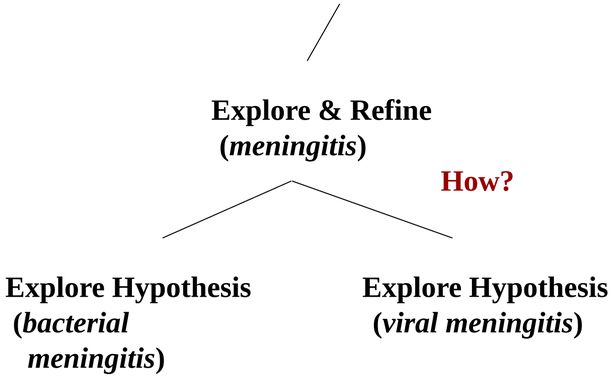
Strategic "How?" Explanations

How was a specific task achieved?



The justification comes from the actions of the meta-rules.

Example



The hypothesis of meningitis was explored and refined as follows:

The hypothesis of bacterial meningitis was explored

The hypothesis of viral meningitis was explored

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Heuristic Classification: the generalization of NEOMYCIN's reasoning

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

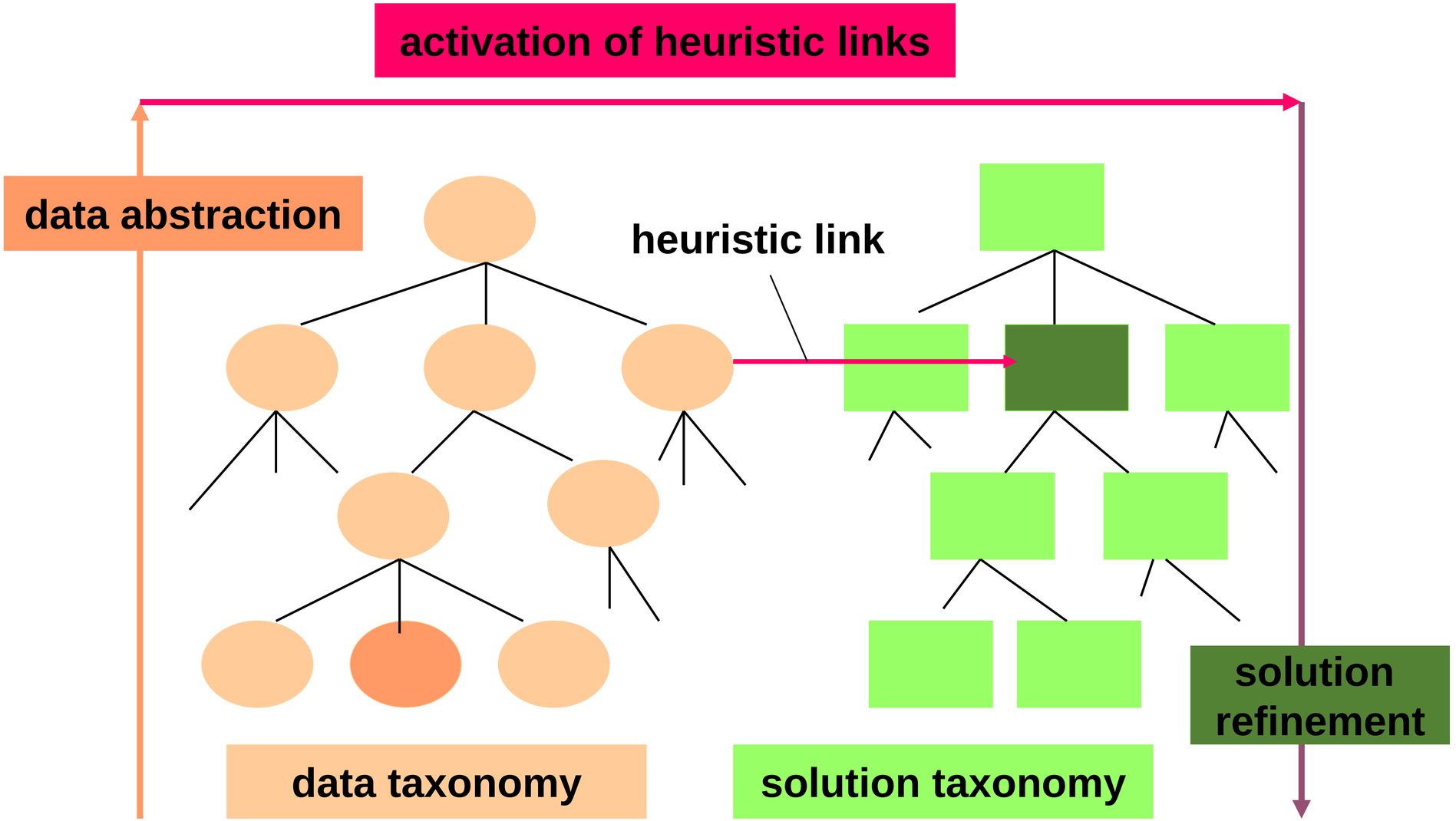
Heuristic Classification

- For its application, three basic knowledge structures are needed:
  - the **data taxonomy**,
  - the **taxonomy of solutions** (solution taxonomy) and
  - the **heuristic links** which connect nodes of the data taxonomy to nodes of the solution taxonomy and therefore constitute the connecting links between the two taxonomies
- The heuristic classification method has been incorporated into the generalized system HERACLES

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristic Classification



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Reasoning Processes: Data Abstraction

- It is applied to the data taxonomy, aiming to produce more abstract data from the original, very specific data of a problem.
- The goal is to gradually activate nodes at higher levels of the data taxonomy.

Types of Data Abstraction

- **Qualitative abstraction,** where quantitative expressions, e.g., 'temperature 41° C', are translated into qualitative expressions, e.g., 'High fever'
- **Generalization abstraction,** where a specific expression, e.g., 'Augmentin administration', translates to category level, e.g., 'administration of antibiotic'
- **Definitional abstraction,** where a given datum that belongs to a conceptual category is translated into its 'equal' in another, non-hierarchically related, conceptual category, e.g., 'generalized platyspondyly' can be translated as 'short trunk'

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Temporal Data Abstraction

- The above types of data abstraction do not use the time dimension.
- Recently, **temporal data abstraction** has attracted great research interest, where the raw data are time series of various forms, and the goal is the production of temporal abstractions such as trends and periodic events.

Types of Temporal Data Abstraction

- **Merge abstraction,** where a series of concatenable (time-point based) data are abstracted to a single time-interval based datum
- **Persistence abstraction,** where a maximal interval spanning the extent of some property is derived (backwards and/or forwards in time)
- **Trend abstraction,** where the aim is to derive the significant changes and the rates of change in the progression of some parameter
- **Periodic abstraction,** where repetitive occurrences with some regularity in the pattern of repetition, are derived, e.g., headache every morning, for a week, with increasing severity

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Heuristic Classification: Further Reasoning Processes

Activation of Heuristic Links

- This is a relatively simple, but critical process.
- Its purpose is to direct attention to nodes of the solution taxonomy since the ultimate goal is to select terminal nodes in this taxonomy.

Solution Refinement

- Operates on the solution taxonomy
- In contrast to the data abstraction process which starts from low-level nodes and moves up the relative taxonomy, the process of solution refinement starts from high-level nodes and aims to move downwards
- That is, starting from the nodes that the heuristic links led to, it gradually moves towards terminal nodes

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

MDX: Community of collaborating specialists, new meaning of ‘compiled knowledge’

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

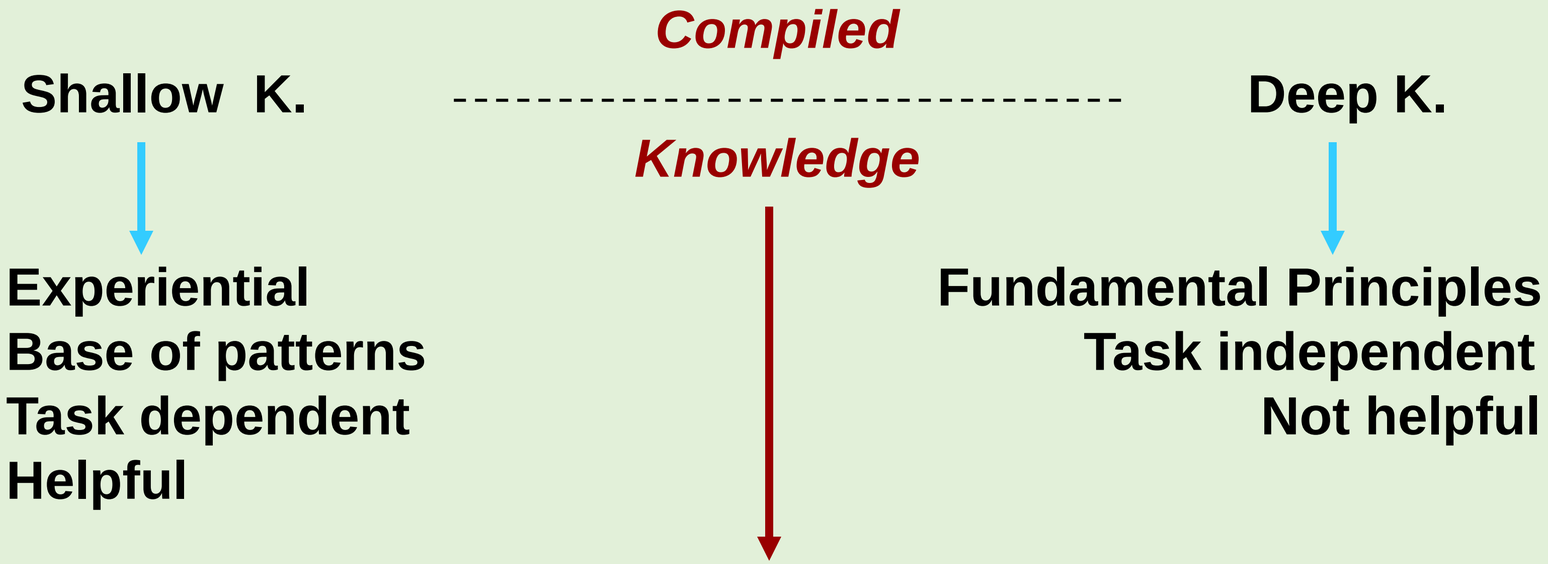
MDX System

- The MDX system was developed at The Ohio State University with B. Chandrasekaran and S. Mittal as the principal investigators.

- The aim of the system is to diagnose the syndrome of the liver, known as cholestasis.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Compiled Knowledge
Alternative Definition



It depends on the task
Very helpful
It can handle any problem that can be solved through deep knowledge,
But the solution is done in a much more efficient way

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Compiled Knowledge

- Deep knowledge is **comprehensive** but **not operational**, while shallow knowledge is **operational** in terms of the needs of the given task but usually has deficiencies, mainly in terms of the more difficult problems that concern it.
- Compiled knowledge has the positive characteristics of the other two, the comprehensiveness of deep knowledge and the functionality of shallow knowledge.

Analogy with Compiled Programs

This interpretation of compilation is compatible with the concept of program compilation where the object code is (logically) identical to the source code in terms of the set of problems that can be solved, but the object code is more efficient in terms of execution time.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Compiled Knowledge – MDX Interpretation

It is the transformation of deep knowledge, into operational form, from the perspective of a given task, without reducing the comprehensiveness of the knowledge.

This means that any problem that falls under the scope of the given task and can be solved through the deep knowledge, it can also be solved through the compiled knowledge, but the solution is achieved much more efficiently.

The same body of deep knowledge can have multiple compilations from the perspective of different tasks.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

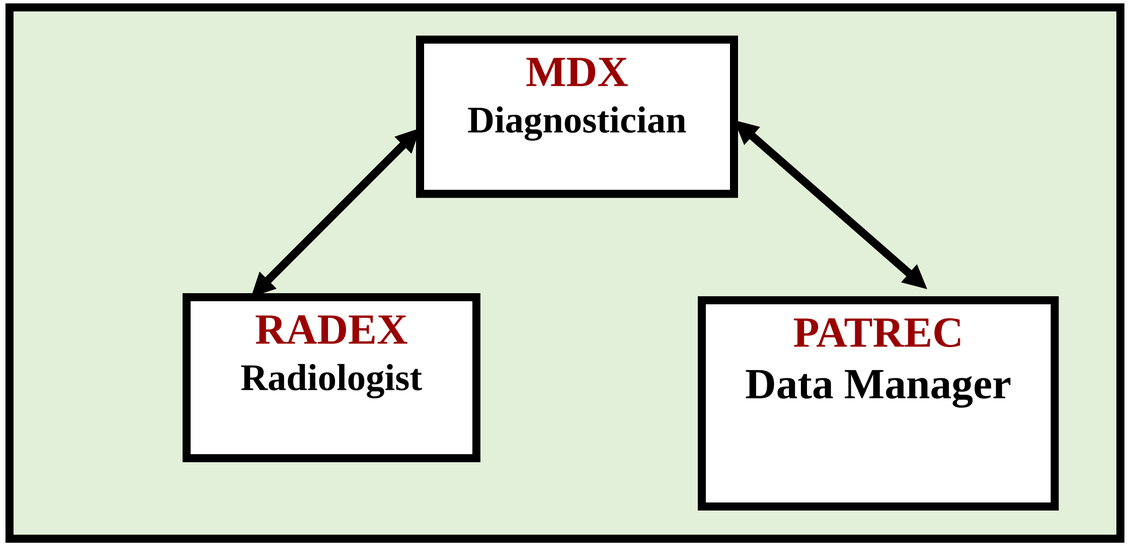
Collaboration of Specialists

The MDX system consists of three separate units:

- The main unit, the diagnostician, also called MDX
- The data manager, PATREC
- The radiologist, RADEX

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Collaboration of Specialists



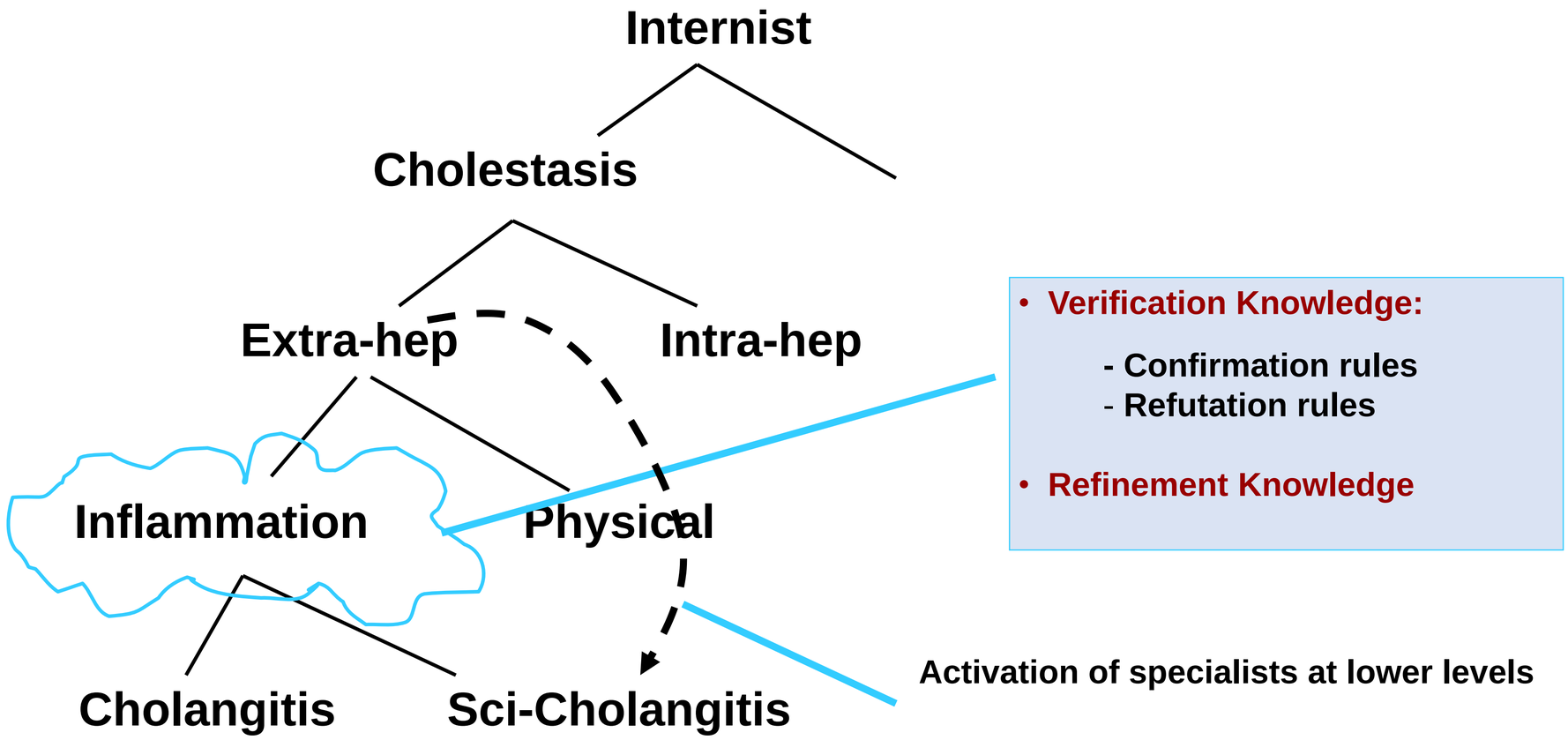
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Each system is a community of specialists

- Internally, each of these systems consists of a **community of specialists**, who are organized in a hierarchical manner and cooperate based on a specific protocol.
- Each specialist is considered an autonomous entity with its own knowledge and control/reasoning mechanism, which can perform a specific task in the entire solution plan.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Part of the Hierarchy of Specialists of the Diagnostician



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Collaboration Protocol

- First, the problem is referred to one of the specialists, usually the one who is the most senior in the hierarchy.
- Each specialist should be able to decide whether the problem referred to it really belongs to its specialty. If so:
- In the case of a non-terminal specialist, it will then have to decide to which subordinate, not necessarily his immediate subordinate, to refer the problem to.
- In the case of a terminal specialist, it should try to solve the problem and then inform its superior.
- If a specialist decides that the problem is outside its specialty, it should refer it back to the referring specialist along with relevant guidelines.
- The collaboration protocol is a satisfactory simulation of reality, where the patient initially visits his physician and from there, he can be referred to various specialists and sub-specialists for a more accurate diagnosis of the causes of his problem.
- The distributed way of organizing knowledge and reasoning is the compilation of the corresponding deep knowledge, aiming for its efficient use.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

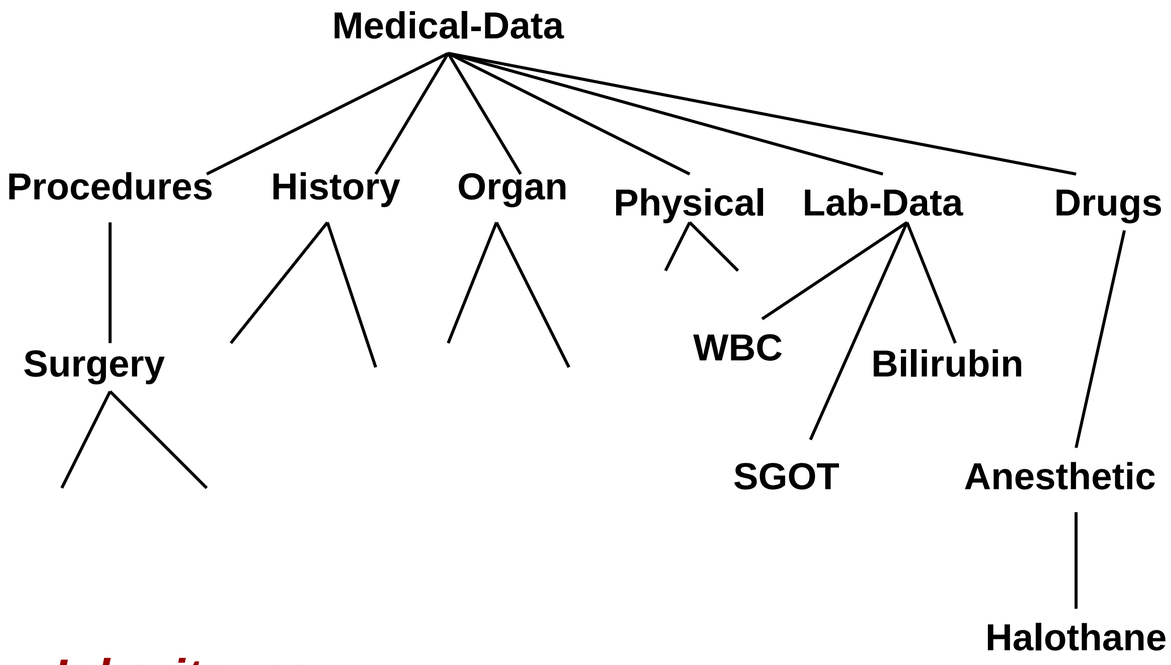
Multi-modeling in MDX

- In MDX there are no alternative models for the same body of knowledge as in NEOMYCIN.
- Here the whole body of knowledge is divided into three separate models, one for diagnostic concepts, one for medical data concepts, and one for radiology concepts. These concepts constitute the individual specialists for each of these models.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

PATREC

Taxonomy of Specialists in Medical Data Concepts



Inheritance

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Frame for Specialists

Classification links:

Type: super-concept
Instances: subconcept$_1$ ... subconcept$_n$

Characteristics:

C_1 values-domain default constraints
. . . .
C_m values-domain default constraints

Deductive links:

Positive links: . . . .
Negative links: . . .

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example

Let’s assume the following patient data:

- **The substance ‘halothane’ was administered during the ‘cholecystectomy’.**
- **The patient’s ‘bilirubin’ was 12.2 three days after the surgery.**
- **The patient had ‘pruritus’ one week after.**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Suppose the cholestasis specialist needs to evaluate the following refinement rule:

If the appearance of ‘jaundice’ was within a week after surgery, and ‘pruritus’ developed after ‘jaundice’

Then consider post-surgical cholestasis, which has been caused by anesthetics

The diagnostician (MDX) calls the patient manager (PATREC) to evaluate the rule premise.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Data Abstraction

1. ‘Cholecystectomy’ is a type of surgery.

Generalization

2. ‘Bilirubin 12.2’ means ‘bilirubin high’.

Qualitative Abstraction

3. ‘Bilirubin high’ indicates the presence of ‘jaundice’.

Data Dependency

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Temporal Reasoning

4. Three days is a smaller time period than one week and ‘jaundice’ appeared at the same time as ‘bilirubin’.

5. ‘Pruritus’ developed after ‘jaundice’ (one week after the time the ‘bilirubin’ was high).

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

PATREC Operation

- The various procedures that are distributed in the taxonomy of PATREC specialists, have the function of deciding whether a specific datum is true or not for the given patient.
- In brief:
  - Inference rules are applied (with backwards chaining) to infer the presence or absence of the given datum from the patient information, while hierarchical relationships allow the application of principles such as:

    If a concept is present,

    then its super-concept is also present.
  - Also, many inferences are based on the characteristics of the concepts, their value domains, and any defaults.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Generic Tasks Architecture

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Generic Tasks Architecture

- The Generic Tasks Architecture was proposed by B. Chandrasekaran.

- It is the generalization of the MDX system architecture and provides a more conceptual interpretation of the notion of **reusability** than the interpretation given in the context of the first generation.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Generic Tasks

- A **generic task** is linked to a specific goal and is characterized by:
  - The knowledge structures it uses, in other words its knowledge model.
  - The types of information that make up its input and output.
  - The control mechanism.

Generic Tasks as the 2nd Generation Building Blocks

- Each generic task is a **conceptual unit** for a given purpose.
- It is proposed that such units form the **building blocks** of second-generation systems, so that the system's task-level control structure is explicitly displayed.
- Primitive generic tasks can be composed to build **complex generic tasks**.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Generic Tasks

Complex Generic Tasks

- A complex task is characterized by: (a) the generic tasks that make it up, and (b) the routes and conditions governing the transfer of information between them.
- A complex generic task can be implemented as a **community of collaborating specialists** communicating via messages.

Library of Generic Tasks

- The research team has developed a library of basic tasks that can be reused in the context of different complex tasks.
- However, one weakness of this architecture, which became apparent in practice, is that it supports the splitting of tasks into subtasks, etc., but not the existence of alternative models to achieve the goal of a given task:
- In other words, each task can be carried out based on one and only one way.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Summary

- Notion of deepness
- Approach and distinguishing features of 2nd generation systems
- New interpretation of reusability
- NEOMYCIN: multiple models, strategic explanations
- Heuristic classification
- MDX: alternative meaning of ‘compiled knowledge’, community of collaborating specialists
- Generic Tasks Architecture

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September - December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Knowledge Engineering

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 9

Knowledge Engineering

CONTENTS

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on knowledge engineering, students will be able:

1. To point out the key differences between knowledge engineering and software engineering.
2. To explain how expertise is developed and what the modelling of expertise entails.
3. To present example total task investigation methods of low, medium and high fidelity.
4. To discuss the Knowledge Engineering processes, particularly the development of a model of expertise.
5. To present and compare various interview techniques.
6. To discuss the principles, the various models and the layers of knowledge advocated by the CommonKADS Knowledge Engineering methodology.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

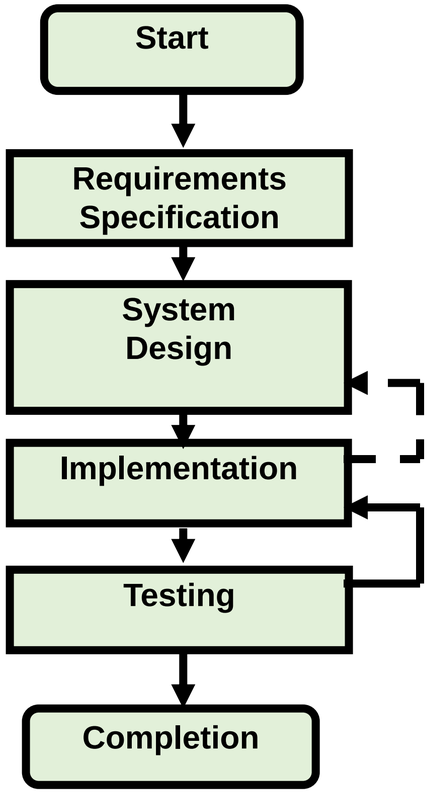
Master programmes in Artificial Intelligence 4 Careers in Europe

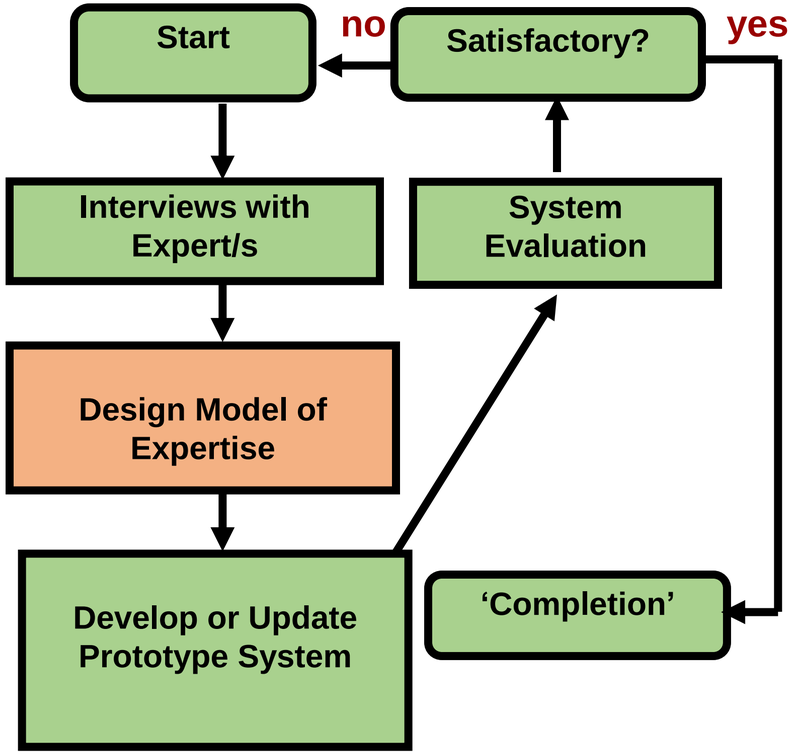
Knowledge Engineering versus Software Engineering

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Software Engineering versus Knowledge Engineering





Software Engineering **Knowledge Engineering**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Developing and Modelling Expertise

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Model of Expertise

- An expert system is a **model of expertise.**
- Like all modelling tasks, the creation of an expert system involves the analysis of mainly qualitative data and the design of an abstract structure, the model.
- The model externalizes the implied structure of the data and can therefore be considered the interpretation or explanation of the data:
  - Model elicitation
  - Model representation
- The conceptual distance between the qualitative data, expressed in natural language, and the symbolic implementation constructs provided by AI, is large.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Conceptual Models of Expertise

- The second-generation approach is the intermediate design of **conceptual models of expertise**, as the bridge between data and symbolic implementation constructs
- Models of expertise are expressed in terms of some intermediate representation
- At the same time, as an additional means of bridging the gap between the data and the intended system, the technique of rapid prototyping is applied

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Developing Expertise

Many studies have been conducted regarding the differences in solving related problems between experienced and inexperienced people in some domain, or between categories of people with different amounts of experience.

Results of Studies

- The initially unstructured knowledge base of the inexperienced person, through experience in various ways of solving problems, gradually acquires shape and structure so that the pieces of knowledge are organized for immediate and efficient use.
- Structures learned through experience are 'orthogonal' to the traditional structures contained in books.
- An inexperienced person's descriptive knowledge is 'raw' because:
  - focuses on **classical descriptions**,
  - is **scattered** because there are not many connections between its elements and
  - the internal structure of its elements is **imprecise**.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge ‘Compilation’ Process

Recall, that according to Chandrasekaran's proposal, the process of acquiring expertise is the process of 'compiling' knowledge for efficient and effective use in carrying out specific problem-solving tasks.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

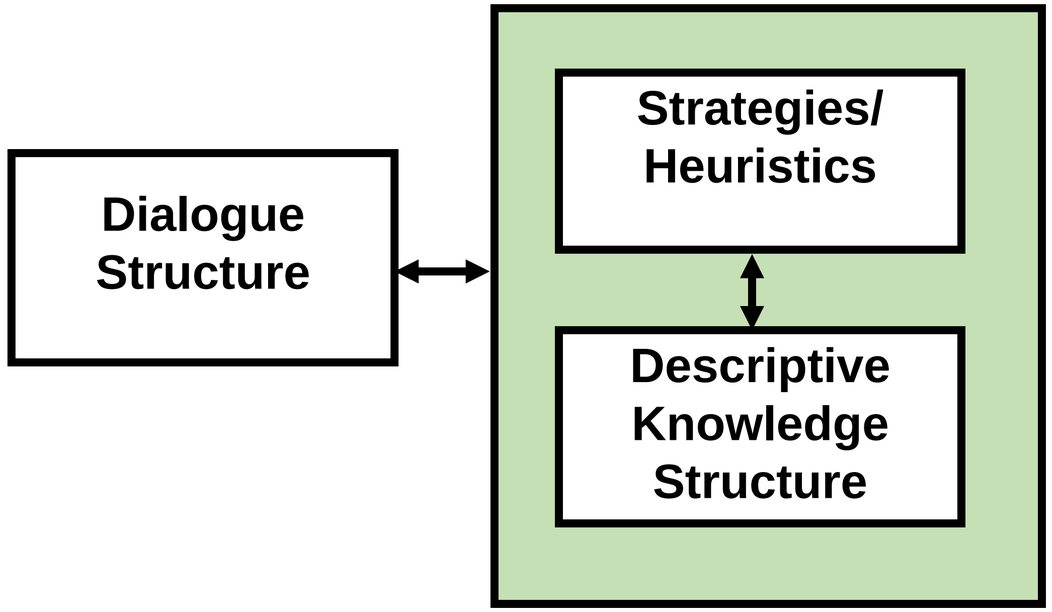
Master programmes in Artificial Intelligence 4 Careers in Europe

Total Task Investigation Methods

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Total Task Investigations



Interplay of Strategies, Descriptive Knowledge Structure and Dialogue Structure

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Total Task Investigations

Total task investigations aim to investigate the sequential nature of the search for information that leads to conclusions and decisions.

- Such methods were developed and applied by Elstein, Shulman, and Sprafka with the goal of understanding the skills, strategies, or other qualities that characterize the performance of skilled clinicians.
- The study focused on disentangling the two components of the expertise model, search strategies and memory structures.
- The motivation behind this study was to improve or accelerate how medical students can learn these skills.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Total Task Investigations: Research Findings

- Hypotheses (regarding the solution to the problem) are generated early.
- The number of active hypotheses is very small, rarely exceeding 5 and almost never exceeding 7.
- The most common error is over-interpretation (giving more weight than it should to evidence consistent with the intended hypothesis, while important evidence against the hypothesis may be ignored).
- Ability can be situational, and knowledge and experience are essential to ability.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Total Task Investigation Methods

The application of a total task investigation method implies the simulation of the environment of the task in question. Methods can be categorized based on how the task environment is simulated:

- **high fidelity** simulation where the participation of actors may be required,
- **moderate fidelity** simulation, otherwise ‘basket’ methods, and
- **low fidelity** simulation.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example of a method of low fidelity simulation

- The analyst creates a list of all possible actions and their corresponding results.
- The list of actions, without the results, is presented to the expert. The expert successively chooses actions, until either all actions are exhausted or a solution to the problem is reached.
- During the process the expert does not need to reveal his thought.
- The analyst records the sequence of the expert's choices to analyse it to discover the expert's thinking.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Actions

A_1

A_2

.

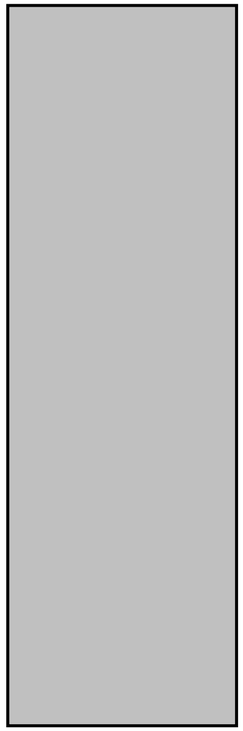
.

.

.

A_n

Results



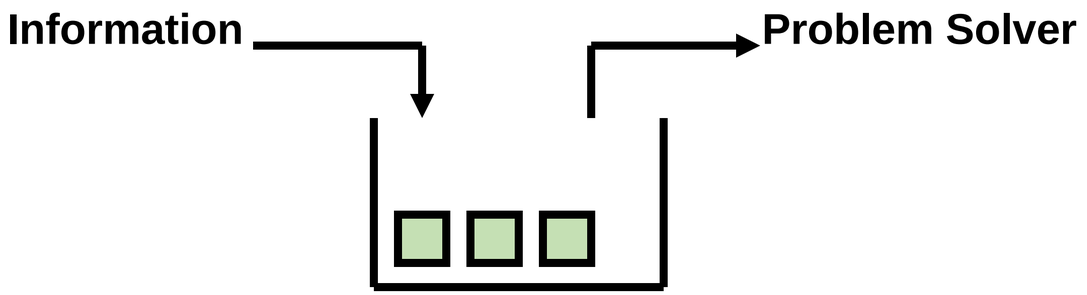
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Example of a method of medium fidelity simulation

- The analyst sequentially reveals information about the problem being solved by placing relevant notes in a basket.
- The expert, who does not raise questions, sees this information, withdraws from the basket those he wants to use, and at the same time reveals his thinking to the analyst.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

“Basket” Method



Argument against this method: The process of thinking loudly modifies the content and the process of thinking.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Another example of a method of medium fidelity simulation

- Some initial information about the problem is presented to the expert, and then the expert is allowed to ask up to twenty questions to find the solution to the problem.
- The sequence of questions is recorded and analysed with the aim of discovering the thinking of the expert.
- This method, which is a version of the well-known game of twenty questions, leads to binary decision trees.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example of a method of high-fidelity simulation

- According to one such method, for application to clinical problems, actors are trained to play the roles of patients during physical examination and history taking.
- Additional information, e.g., laboratory test results, is provided by a nurse acting as an information bank.
- At certain stages of the process, the expert needs to express his thinking so far. **Retrospective evaluations** are considered an important feature of the method because no precise conclusions can be drawn regarding the practitioner's strategy, based only on the practitioner's interaction with the patient.
- The expert's interaction with the patient is videotaped, which is then presented to the expert as a stimulus for further memory recall. This is an additional way of getting behind the visible behaviour, into associations, strategies, etc.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Additional data solicited from the high-fidelity simulation method

Overall, this method provides three additional sources of data to complement the analysis of the task:

- material from any simultaneous thinking aloud, which the expert voluntarily provides, e.g., short sentences about what he had learned, or what he is going to do and why
- longer episodic assessments conducted during natural breaks of the task
- material obtained during the presentation of the video film

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

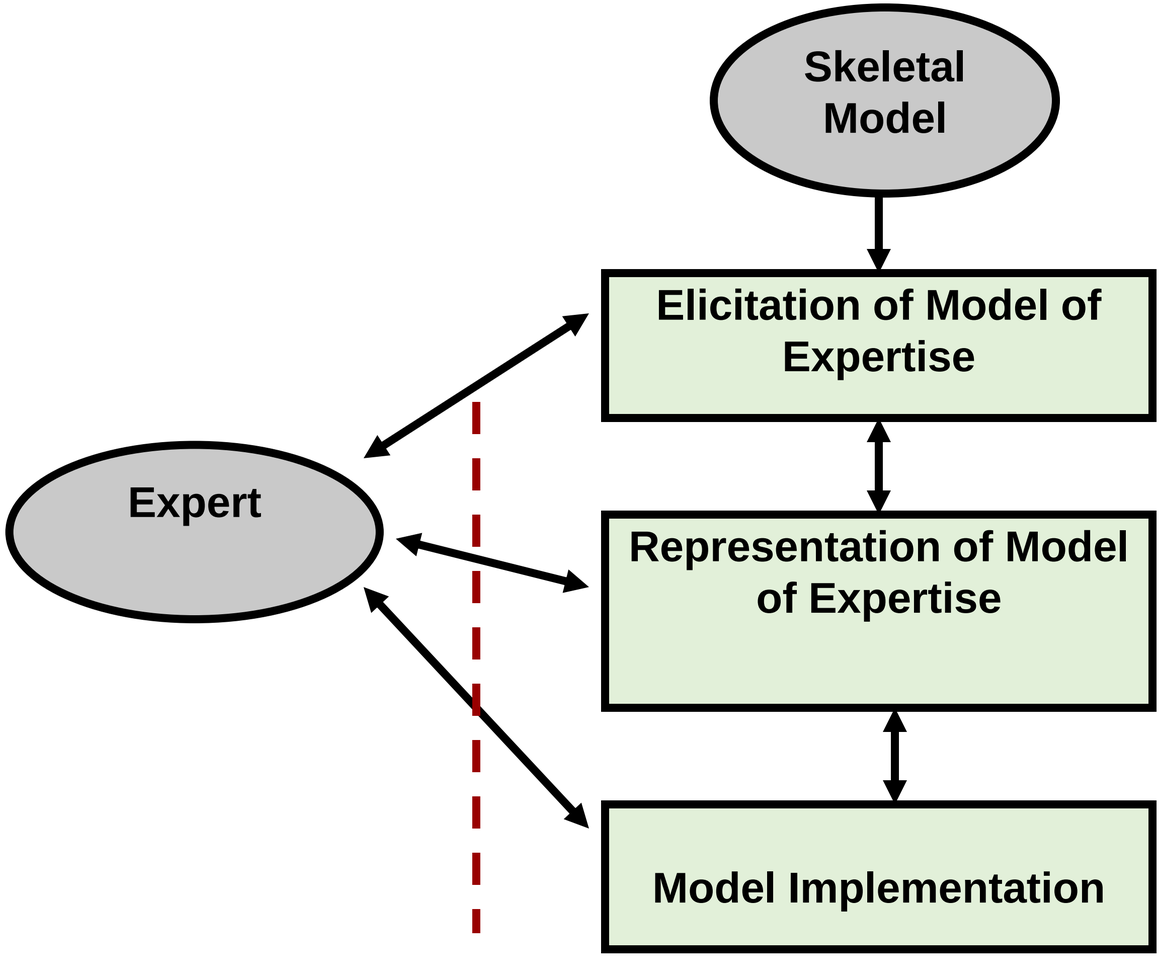
Master programmes in Artificial Intelligence 4 Careers in Europe

Knowledge Engineering Processes – Interview Techniques

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Engineering Processes



intermediate representation

symbolic representation

AI techniques

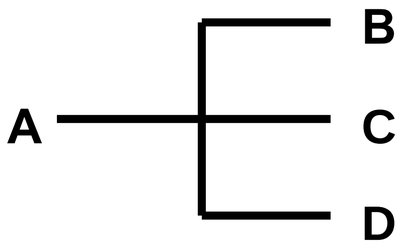
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Elicitation of Model of Expertise

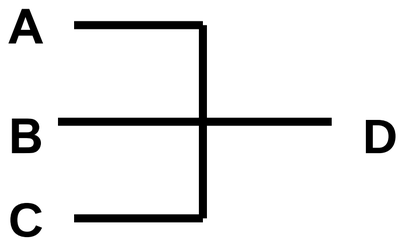
- During the elicitation process, the model is expressed at a high level of abstraction through some intermediate or mediating representation.
- An example of an intermediate representation is **systemic grammar networks**, which have been used to analyse qualitative data, as well as to define and classify concepts.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

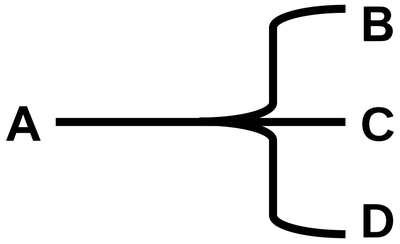
Notation of Systemic Grammar Networks



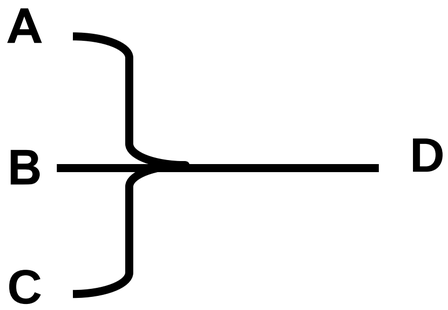
If A, then select one out of B, C, or D



Select D, only if one or more from A, B, or C

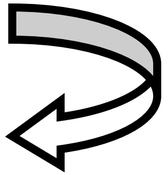


If A, then select all of B, C and D



Select D, only if all of A, B and C

recursion



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Engineering Processes

- The processes of elicitation, representation, and implementation of the model of expertise can be considered parallel processes with mutual effects.
- The knowledge engineer must be willing to face even situations that involve a complete re-engineering of the system. Experimentation through rapid prototyping, e.g., using some shell system is indicated, as a means of better understanding the given expertise.
- Also, the use of a knowledge acquisition tool is indicated, especially when the symbolic knowledge structures have reached a satisfactory level of reliability.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Interview Techniques: “Teach Back” Technique

- It is based on Pask's conversational theory.
- It focuses on a program of semi-structured interviews.
- In each meeting, the subject of the discussion and the means of communication are decided (systemic grammar networks have been used as a means of communication in the context of this technique).
- The dialogue between the expert and the analyst takes place at **two levels**:
- Level '0' is about explanations of how an algorithm is executed
- Level '1' is about explanations as to why the algorithms work, i.e., explanations of the explanations (meta-explanations)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

"Teach Back" Technique: Dialogue Levels

Level '0' Dialogue

- The expert describes a process to the analyst.
- The analyst then teaches it back to the expert using the expert's terminology and, in a manner, satisfactory to the expert.
- At the end of this stage **the analyst and the expert can be said to share the same concept.**

Level '1' Dialogue

- The analyst asks the expert to give him an explanation as to how the given concept can be reconstructed.
- The analyst then teaches this back to the expert, until the expert is satisfied with the expert's version, at which point it can be said **the analyst has understood the expert**.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Therefore, at level '0' concepts are taught back (to the expert by the analyst) with the aim of achieving common concepts, while at level '1' memories are taught back with the aim of achieving understanding.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

“Teach Back” Technique: Strengths and Weaknesses

Strengths

- Objectivity
- Success in gaining and maintaining (by the analyst) the interest of the expert
- There is no doubt as to the authenticity of the extracted data

Weaknesses

- The technique is quite tedious for the analyst, who should be trained in interviewing
- It also leads to too much material to analyse, and interviews should be short

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Other Interview Techniques

Tutorial interview:

The expert gives a tutorial on the main topics and concepts of the field.

Focused interview:

The analyst prepares the topics for discussion in advance.

The goal is to acquire descriptive knowledge, the types of problems that the expert solves and in general the functions he performs as an expert.

Distinction of goals:

The analyst presents the expert with a specific goal and asks what evidence is necessary and sufficient for distinguishing that goal from other goals.

This technique is suitable for diagnostic fields.

Reclassification:

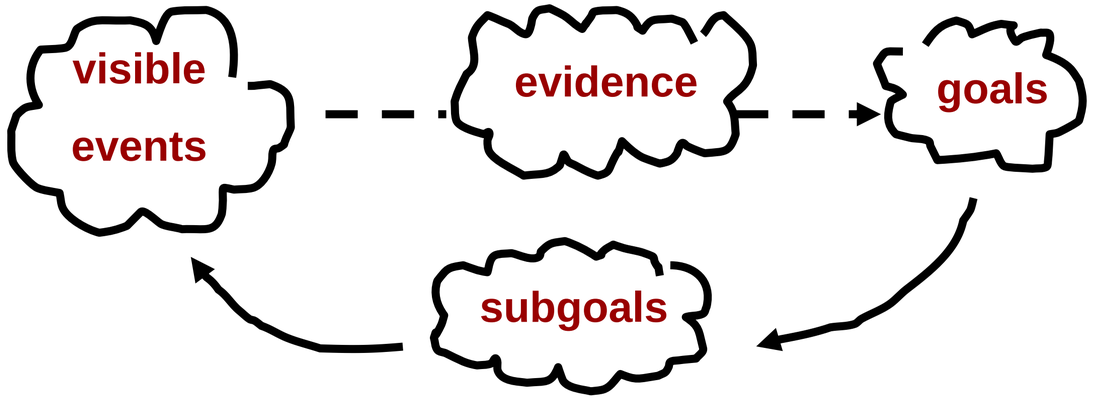
The expert is asked to conduct reasoning in two directions

From visible events, through evidence, to goals (abductive reasoning) and in

Reverse from goals, through subgoals, to visible events (deductive reasoning)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

abductive reasoning



deductive reasoning

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

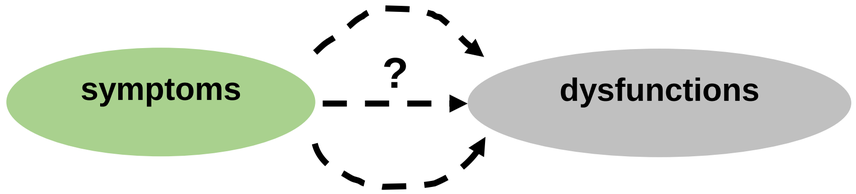
Other Interview Techniques

Systematic symptom-to-fault links:

The expert is presented with a set of symptoms and a set of dysfunctions and asked to associate symptoms with dysfunctions.

This technique, which again requires the expert to conduct abductive reasoning, is applicable only in very simple and limited areas.

The order in which symptoms occur in a real situation is completely ignored.



Intermediate reasoning steps:

It aims at obtaining information to complement the links extracted by the previous technique.

Structured interview:

The goal here is to acquire all the knowledge that pertains to a particular concept.

The technique involves the detailed and in-depth analysis of a sequence of topics related to the concept.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Other Interview Techniques

Twenty questions:

The same technique mentioned earlier with the extension that the expert can be asked by the analyst why he is asking the specific questions.

Laddered grid:

The analyst presents the expert with a set of concepts and asks him to classify them in several alternative ways and each time to explain the classification.

This technique is suitable for areas where the analyst suspects the existence of hierarchical structures.

Introspection:

The analyst asks the expert to imagine how he would solve (or has solved) a problem or class of problems.

This technique, like the basket method, requires the expert to reveal his thinking.

At the same time, the analyst can ask the expert questions of the type 'How' and 'What', but not of the type 'Why', because such questions may result in the attention being diverted somewhere else.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Other Interview Techniques

Retrospective case description:

The expert is asked to describe how one or more typical incidents, preferably from the recent past, were processed.

The analyst needs to assess how representative each case is.

It is a fact that rarer and therefore remarkable incidents are better recorded in one's memory.

Critical incident:

In contrast to the previous technique, here the expert is asked to describe his experiences regarding remarkable or difficult incidents.

The use of this technique can create a stimulating beginning in the knowledge acquisition process, since such critical incidents are of greater interest to the expert than typical incidents, while at the same time it aims to reveal any 'bottlenecks' in the expert's work.

Forward scenario simulation:

The expert describes in detail how he would process a hypothetical incident, chosen either by himself or by the analyst.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The use of interview techniques aims to achieve understanding between the expert and the analyst and to express this understanding in a form that allows public scrutiny.

This is considered the central function of the knowledge acquisition process.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

CommonKADS Methodology: Basic Principles and Models

- The CommonKADS methodology is the most widespread knowledge engineering methodology, at least in the European area
- The original name of the methodology was simply KADS (Knowledge Acquisition and Document Structuring)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CommonKADS Library

- It facilitates the application of the methodology
- The library is organized based on a categorization of problems and inferences – more abstract elements
  - Categories or types of problems, e.g., diagnosis, prognosis, planning, design, etc.
  - Reusable, canonical inferences – primary elements

Other elements of the library:

- **Modelling Units** (Building Blocks)
- **Generalized models** (skeleton plans with constraints on how they can be 'filled')
- **Modelling procedures** (the steps that should be followed to create the models)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Need for a Methodological Approach

- As a means of overcoming the so-called 'knowledge acquisition bottleneck'
- Also, because the reasoning methods used in knowledge-based systems were not always fully understood

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The CommonKADS methodology is based on treating the knowledge acquisition process as a modelling activity.

A knowledge-based system is considered as a functional model that manifests some desired behaviour, which is visible or prescribed through realistic phenomena.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Principles of CommonKADS Methodology

Multiple Models

Modelling expertise

Reusability

Separation of knowledge

Design that preserves the structure of knowledge

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Multiple Models

- Building a knowledge-based system is a complex process, which can be seen as a search process in a large space consisting of knowledge engineering methods, techniques, and tools.
- This complexity can be controlled to some extent by the use of **multiple models**, each of which presents the system from a different point of view, thus emphasizing certain features of the system while abstracting away the other features.

Multiple models (views) of a knowledge-based system

- Organizational Model
- Application Model
- Task Model
- Cooperation Model
- Expertise Model
- Conceptual Model
- Design Model

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Organizational Model

Entails the following:

- Definition of the problem that the system aims to solve within the Organization.
- Analysis of the socio-organizational structure of the environment in which the system will operate.
- Description of the functions, tasks, and 'bottlenecks' of the Organization.
- Anticipation of how the introduction of the system will affect the Organization and the people working within it.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Application Model

- It defines the operation of the system in relation to its future users, who are either humans or other systems.
- It identifies external constraints related to the development of the application, e.g., constraints regarding speed, efficiency, use of special software or computer, etc.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Task Model

- It defines the specific tasks that the system will perform in relation to its defined function, i.e., to achieve its goals.
- Achieving a goal can be done in alternative ways.
- Therefore, the appropriate method needs to be chosen for the specific application, considering the characteristics of the application, existing knowledge and data, user requirements, as well as external factors.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Task Model

Views of Task Model:

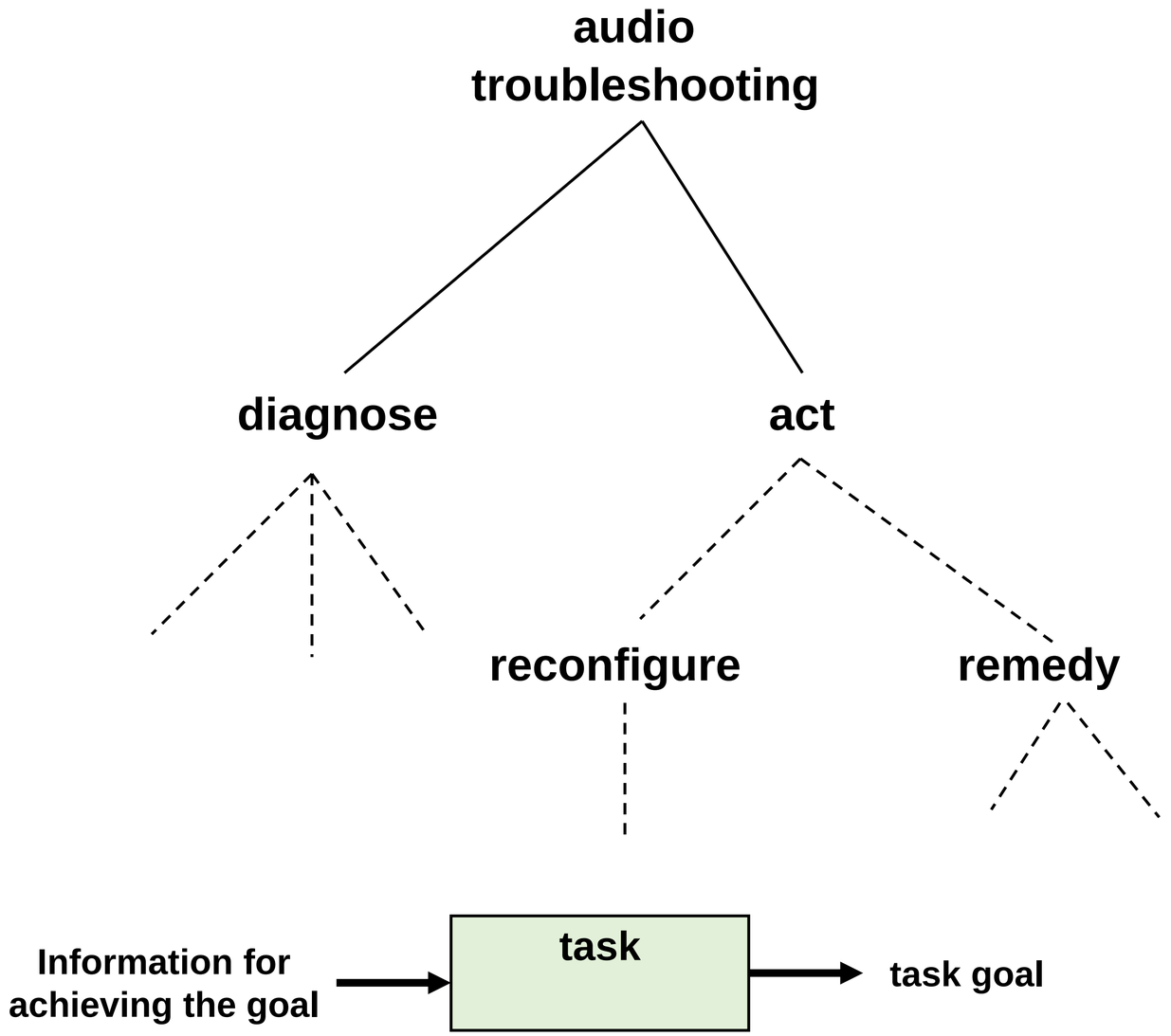
- Task Decomposition
- Task Allocation
- Task Environment

Task Decomposition

- The task is broken down into subtasks, etc.
- For each subtask, its interface is defined, i.e., what its goal is and what information it needs to achieve this goal.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Task Model

Task Allocation

- Subtasks are allocated to **agents**.
- At this level, agents are the under-construction system itself, the user, or some other system.
- The user or other systems are external agents.

Task Environment

- The constraints imposed by the task environment.
- These constraints influence the scope and overall nature of the expertise and cooperation models.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Cooperation Model

- It specifies the functionality of the sub-tasks, in the task model, which need collaborative effort, such as for example sub-tasks concerning the acquisition of data, or the provision of explanations, etc.
- These subtasks are called **transfer tasks** because their processing involves the transfer of some information from the system to an external agent or vice versa.
- Therefore, what is specified is a **cooperative problem-solving model**, in which the system together with the user accomplishes some goal in a way that satisfies the constraints of the task environment.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Expertise Model

- It is the crux of the matter.
- It is this element that distinguishes knowledge engineering from traditional software engineering.
- The goal is to specify, at a knowledge level rather than a symbol level, the expertise required to perform the (problem solving) tasks assigned to the system.
- The creation of the expertise model focuses on the **behaviour** that the system needs to exhibit and the kinds of knowledge that lead to that behaviour, without considering how that reasoning is implemented.
- The creation of the model is guided by the analysis of the expert's behaviour, but also by the bias as to what the intended system should and can do.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Conceptual Model

- It is the integration of the expertise and cooperation models and therefore provides a comprehensive picture of what the functionality of the system is expected to be and what its interfaces with external agents are.
- The conceptual model, as well as the models that make it up, is implementation independent.

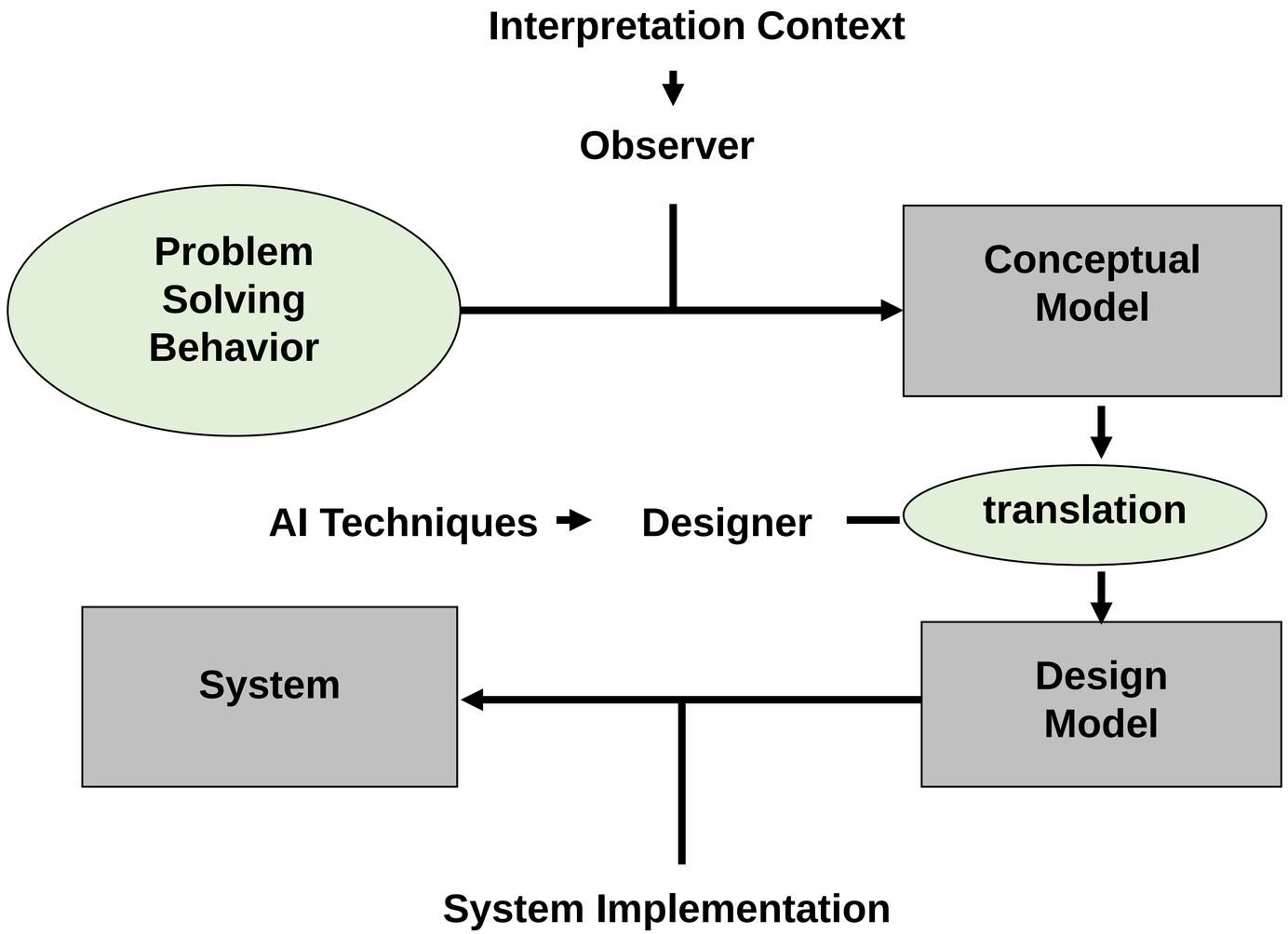
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Design Model

- The design model is the translation of the conceptual model into symbols, based on knowledge representation formalisms, reasoning mechanisms and other computational techniques, which should be used to implement the system.
- External requirements such as speed, hardware, software, etc. must be considered in the design model.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Roles of Conceptual Model and Design Model



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

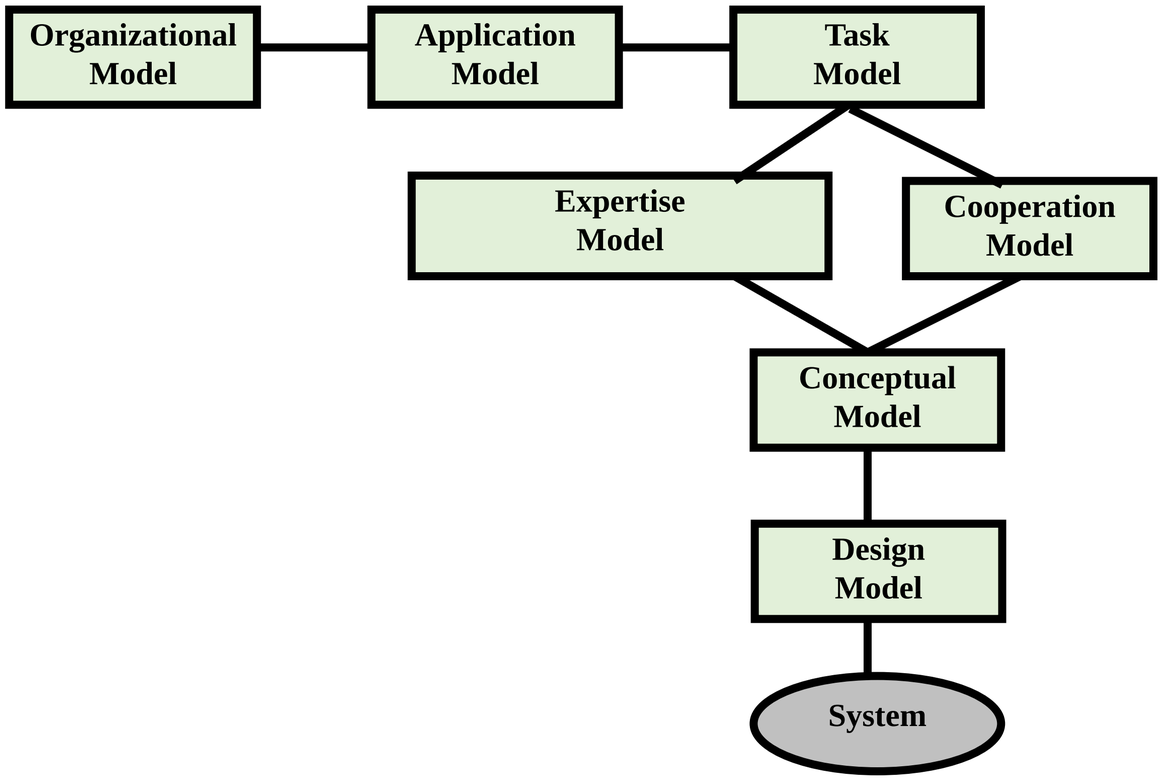
Separation between conceptual modelling and design

- It is considered the strong and at the same time the weak point of the methodology.
- Strong point because when modelling the knowledge engineer is not biased based on the limitations of the computational framework.
- Weak point because after modelling the problem of implementation remains and possibly elements of the model may not be implementable based on the existing technology.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Model Dependencies



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Modelling Expertise

The biggest challenge of knowledge engineering is answering satisfactorily the question

"**How is expertise modelled?"**

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

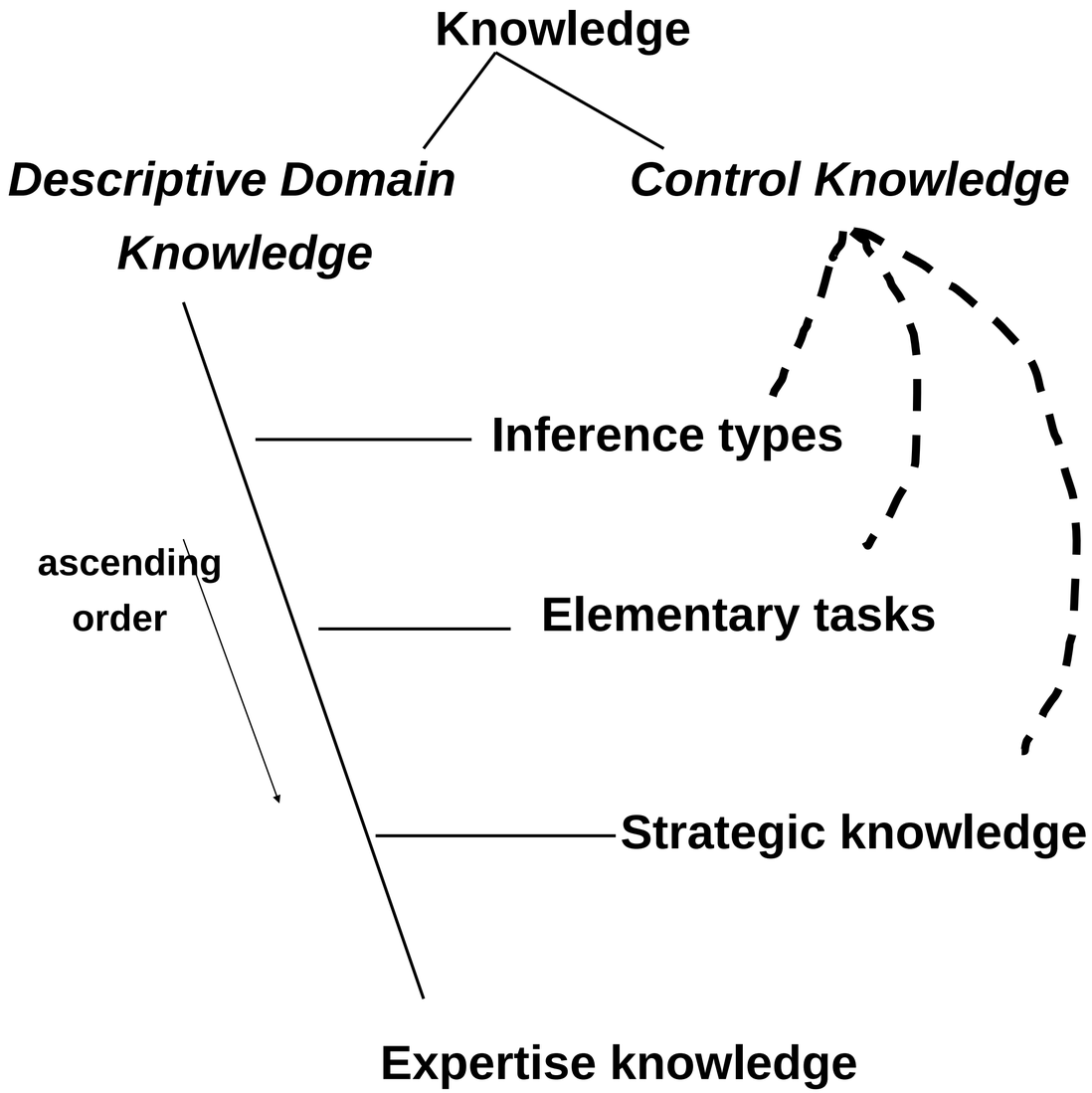
Modelling Expertise: CommonKADS Methodology approach

The CommonKADS methodology proposes the following basic conditions regarding the modelling of expertise:

- It is possible and useful to separate different generic types of knowledge according to the different roles that knowledge plays in reasoning processes.
- These types of knowledge can be organized into several layers which have limited interconnections.

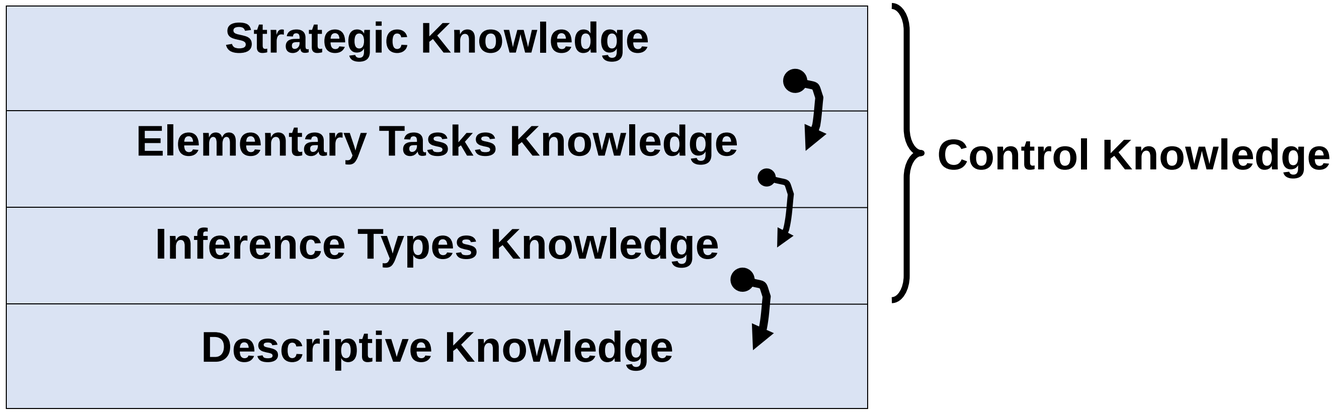
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Basic Decomposition of Knowledge



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Layers



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Descriptive Domain Knowledge

- Static knowledge, which constitutes the **theory** of the field of application.
- It is expressed in a declarative manner and consists of the following primary elements:
  - **Concepts**, which represent the relevant objects. Each concept has its name.
  - **Properties** and **values**. Each concept has a set of characteristics, its properties. Each property has a name and a set of values.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Other Primitive Elements

- **Relationships between concepts**: The most common relationships are hierarchical relationships, 'isa' and 'is_part'
- **Relationships between property expressions**: For example,

  Amplifier: power-button = pressed CAUSES

  Amplifier: volume = up
- **Structure**, for representing complex objects

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Domain Schema

- For a given model of expertise, the subset of primitives is selected which provides the knowledge engineer with the required expressivity.
- Primary elements are used to define the **domain schema**.
- The schema provides the reference points regarding the processing of descriptive knowledge (theory of the domain) from the layer of knowledge above it, i.e., the types of inference.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Domain Schema Example

Primitive Element	Name
Concept	unit
Relationship amongst concepts	unit ISA unit
Relationship amongst concepts	unit SUB_UNIT unit
Property	unit: state
Relationship amongst property expressions	unit: state CAUSES unit: state
Concept	test
Property	test: value
Relationship amongst property expressions	test: value INDICATES unit: state

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Descriptive knowledge independent of task

- Descriptive knowledge is considered relatively **independent of task.**
- The names given to the (categories of) concepts, their properties, and relationships, etc., should not imply any specific use of this knowledge in the domain schema.
- The same body of descriptive knowledge can be used in the context of different problem-solving tasks.
- The separation of descriptive knowledge from the specific way of using it in the context of a given task is considered the first step towards the flexible use and reuse of descriptive knowledge.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Inference Types Knowledge

- **Inferences** are considered basic in the sense that they can be fully described through their name, their interface (what information is their input and output), and the elements of descriptive knowledge (domain schema) they process.
- For modelling purposes, how some inference is implemented does not matter. Only the functionality it provides matters.
- Descriptive knowledge is expressed independently of its uses. Reasoning knowledge assigns **roles** to descriptive knowledge according to the various uses of the latter in the context of basic inferences.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Elements of Inference Knowledge

Meta-classes:

- A meta-class describes some **role** that can be assigned to objects (concepts) when solving problems and gives the classes of objects (from the domain schema) to which this role can be assigned.
- For example, the concept unit can have the role of **hypothesis** in the context of some inference, and the role of **solution** in the context of another inference.

Knowledge sources, where the term has different semantics than the homonymous term in the blackboard model:

- Knowledge sources represent the processing performed by the basic inferences.
- At this level, the knowledge sources are simple names since the 'how's' of the inferences are considered as 'black boxes'.

Domain views, from the point of view of the various knowledge sources:

- A domain view therefore identifies the descriptive knowledge elements that constitute the 'body of knowledge' processed by the given knowledge source.

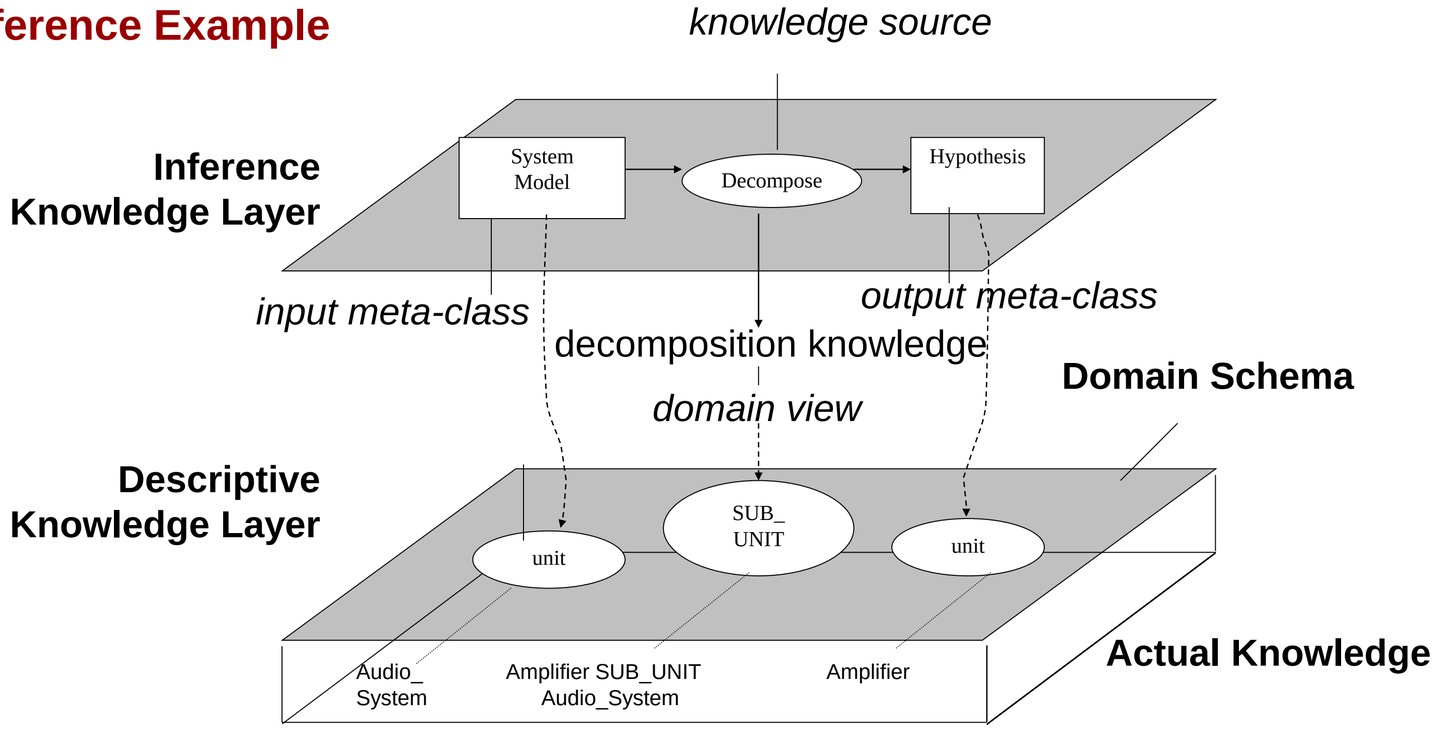
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Basic Inference Composition

- The meta-class that constitutes its input
- The knowledge source and its corresponding view (at the descriptive knowledge layer), that constitutes its processing
- The meta-class that constitutes its output

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Basic Inference Example



Knowledge-Source Decompose

Input-Meta-Class: System_Model → unit

Output-Meta-Class: Hypothesis → unit

Domain-View: Decompose(System_Model, Hypothesis)

→ SUB_UNIT(unit, unit)

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Advantages of separating descriptive knowledge from how it is viewed in the context of inferences

- Multiple uses of the same knowledge are allowed.
- Redundancy is avoided.
- A dual way of naming knowledge is provided, independent and dependent on its use.
- The scope of knowledge is usually wider than what is required by the given problem-solving task, i.e., by the requirements of the involved set of basic inferences.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Basic Inferences

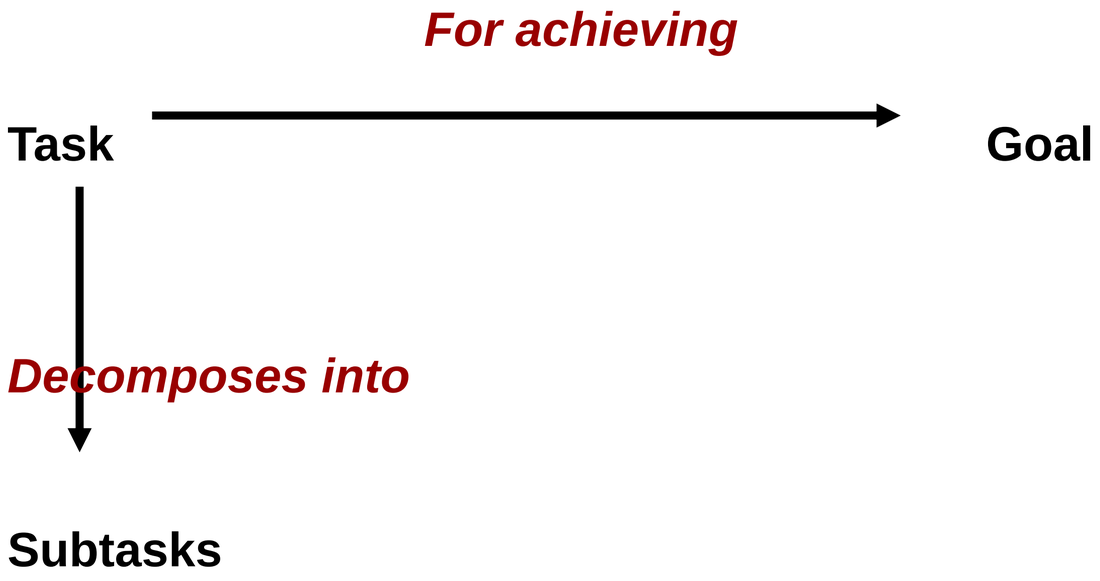
- Basic inferences are the basic building blocks of the knowledge-based system.
- Basic inferences are interconnected to form **inference networks**. At this knowledge layer, no control is defined in relation to such an inference network.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Elementary Tasks Knowledge

- This knowledge layer provides the knowledge of how basic inferences can be put together to achieve a goal.
- The same reasoning task can be associated with multiple goals and the same goal with multiple tasks.
- Therefore, assigning goals to tasks is not necessarily something simple since the relationship between tasks and goals is many-to-many (multi-faceted). However, the function of assigning goals to tasks belongs to the highest layer of knowledge, that of strategic knowledge.
- Elementary tasks represent (lower level) strategies for achieving problem-solving goals.
- The basic inferences are individual, non-decomposable, tasks.
- Therefore, this layer concerns tasks that involve a set of basic inferences whose application is governed by a specific control structure.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Difference with NEOMYCIN

A task in the CommonKADS methodology has one important difference with a task in the NEOMYCIN system:

- In NEOMYCIN the meta-rules that constitute the implementation strategies of the various non-terminal tasks can refer directly to the descriptive knowledge.
- Here tasks refer to basic inferences (terminal tasks) and not directly to descriptive knowledge.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Task Categories

- **Basic tasks** (primitive tasks), i.e., basic inferences.
- **Composite tasks**, where a composite task can be the recursive call of the same task.
- **Transfer tasks** which require an interface with an external agent, usually the user. At this level they are simply seen as black boxes.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Types of Transfer Tasks

- **Obtain**: The system requests some information from the external agent, having the initiative itself.
- **Present**: The system presents some information to the external agent, again having the initiative.
- **Receive**: The system accepts some information from the external agent, based on the latter's initiative.
- **Provide**: The system provides some information to the external agent, again based on the latter's initiative.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Composite Task

- It consists of the relevant goal or goals, the subtasks that make it up, and the control that governs the application of the subtasks.
- For the specification of the control structure, the so-called **control terms** are defined:
  - These are names for relevant sets of elements from the meta-classes involved, e.g., the terms **focus** and **conflict-set** which, in the context of a diagnostic system, name sets of elements from the meta-class hypothesis.
- Also, the specification of the control structure makes use of well-known procedural programming algorithmic structures, e.g., repeat-until, for-do, etc.
- Any subtask that is also complex is analysed in the same way.
- Therefore, a complex task is broken down into sub-tasks, sub-sub-tasks, etc. where the terminal tasks are either knowledge sources (basic inferences), or transfer tasks.
- The execution of transfer tasks is specified in the cooperation model.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Strategic Knowledge

- The highest knowledge layer is the one least explored by the developers of the methodology.
- The knowledge of this layer provides **higher-level strategies** than those concerning ways of accomplishing tasks.
- Strategic knowledge is related to problem analysis and the formulation of relevant goals, as well as the assignment of these goals to tasks. Once a given (sub)goal is assigned to a given task, how to achieve it is decided by that task's knowledge.
- Therefore, strategies control tasks that apply inferences, which use descriptive domain knowledge.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge Chain

Knowledge Category	Organization	Knowledge Types
strategy	strategies	plans meta-classes
↓ *controls*		
task	tasks	goals control terms control structures
↓ *applies*		
inference	inference structure	knowledge source meta-class domain view
↓ uses		
descriptive knowledge (domain schema)	domain theory	concept property relations

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Summary

- ❑ Knowledge Engineering versus Software Engineering
- ❑ Developing and Modelling Expertise
- ❑ Total Task Investigation Methods
- ❑ Knowledge Engineering Processes - Interview Techniques
- ❑ CommonKADS Methodology – Multiple Models

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September - December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Case-Based Reasoning and Intelligent Data Analysis

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 10

Case-Based Reasoning and Intelligent Data Analysis

CONTENTS

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on cased-based reasoning and intelligent data analysis, students will be able:

Regarding case-based reasoning (CBR):

1. To explain case-based reasoning and to outline its perceived advantages over model-based reasoning.
2. To point out the historical roots of CBR and the essence of CBR techniques.
3. To present the CBR cycle, and explain the four REs comprising it
4. To discuss case representation, the different indexing methods, and the two case memory models (dynamic storage model, category-exemplar model).
5. To present various retrieval methods and adaptation techniques.
6. To give some example academic demonstrators and commercial applications.
7. To outline the key challenges of CBR.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on cased-based reasoning and intelligent data analysis, students will be able:

Regarding intelligent data analysis (IDA):

1. To distinguish between IDA, Knowledge Discovery in Databases (KDD) and Data Mining (DM).
2. To explain the role of IDA systems in a clinical setting and draw a comparison between the classical knowledge-based systems and the knowledge-based systems augmented with IDA capabilities.
3. To present the categorization of IDA methods into data abstraction and data mining methods.
4. To discuss the importance of time and give key temporal abstraction methods.
5. To distinguish between a directed, goal-driven, and a nondirected, event-driven, mode of deployment of data abstraction, and explain the two basic ways of integrating a data abstraction process into a problem-solving system.
6. To give a cursory listing of some data mining methods for symbolic classification, CBR being an exemplar of these.
7. To conjecture on the potential integration of temporal data abstraction and data mining for the purpose of knowledge discovery.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Case-Based Reasoning

Adapted from Ian Watson and Farhi Marir, “Case-Based Reasoning: A Review”

https://www.cs.auckland.ac.nz/research/groups/ai-cbr/classroom/cbr-review.html

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Case-Based Reasoning

Case-based reasoning (CBR) is **a paradigm of artificial intelligence and cognitive science that models the reasoning process as primarily memory based**. Case-based reasoners solve new problems by retrieving stored 'cases' describing similar prior problem-solving episodes and adapting their solutions to fit new needs.

Hence CBR is **an experience-based approach to solving new problems by adapting previously successful solutions to similar problems**.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Modelling Expertise is Challenging . . . . CBR is an alternative

Expert or knowledge-based systems (KBS) undoubtedly are one of the success stories of AI. However, modeling expertise, i.e., having an explicit model of the knowledge required to solve a problem, both in the shallow first-generation knowledge-based systems, and the deeper second-generation systems having deep causal models to enable them to reason from first principles, is certainly challenging. The CommonKADS methodology proposes a multi-modeling approach for tackling this challenge.

The CBR paradigm is a practical, theoretically founded alternative, to model-based approaches grounded on the premise that expertise/advanced problem-solving skills are acquired through experience.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Problems facing model-based approaches

- Knowledge elicitation bottleneck
- Special skills and many person years required to implement a KBS given the inherent complexity of realistic domains
- KBS are frequently slow
- Inability of KBS to access and manage large volumes of information
- Once developed KBS are difficult to maintain or evolve through their own learning

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

What CBR advocates as an alternative problem-solving paradigm

- **No explicit domain model** is required and hence the elicitation becomes a task of gathering case histories
- Implementation is reduced to **identifying significant features** that describe a case, which is easier than creating an explicit model
- Volumes of information can be managed by applying **database techniques**
- CBR systems **can learn** by acquiring new knowledge as cases thus making maintenance easier

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CBR's Historical Roots (Roger Schank's group at Yale University)

- **Scripts**, a variant of frames, recording general knowledge about situations, allowing the setting up of expectations and the of performing inferences
  - Proposed as a structure of conceptual memory describing information about stereotypical events (going to a restaurant, visiting a doctor)
  - Not a complete theory of memory representation, as they do not address concept formation, problem solving and experiential learning
- **Memory Organisation Packets (MOPs)** – the role that the memory of previous situations (i.e., cases) and situation patterns or MOPs play both in problem solving and learning
- **The dynamic memory model**, implemented using two kind of MOPs, instances and abstractions

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

CBR Techniques: A case-based reasoner solves new problems by adapting solutions that were used to solve old problems

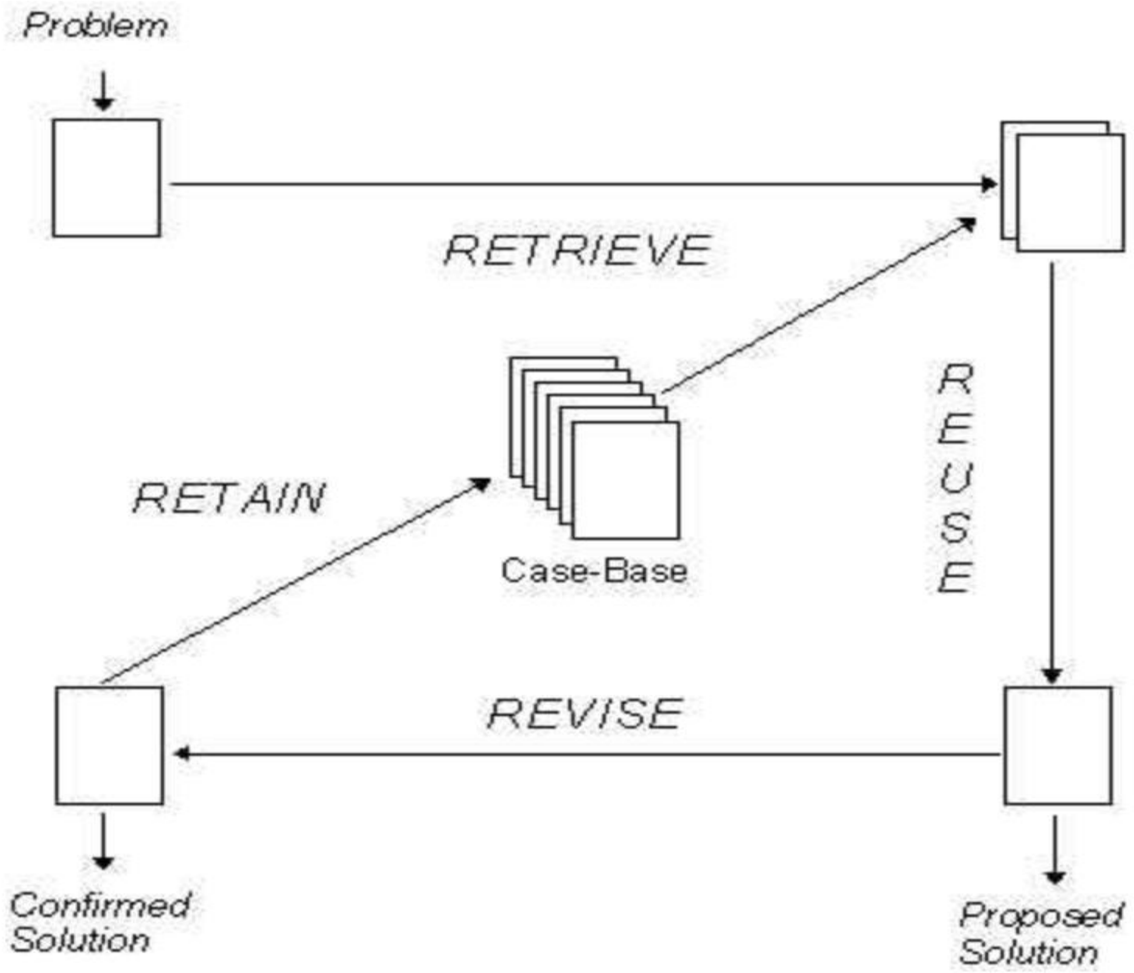
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The CBR Cycle: the four REs

- **RETRIEVE** the most similar case(s);
- **REUSE** the case(s) to attempt to solve the problem;
- **REVISE** the proposed solution if necessary, and
- **RETAIN** the new solution as a part of a new case.

- As many CBR tools act primarily as case retrieval and reuse systems, case revision (i.e., adaptation) could be undertaken by the case base managers; this is not necessarily a weakness as it encourages human collaboration in decision support.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Case Representation

A case is a contextualized piece of knowledge representing an experience that typically comprises:

- the **problem** that describes the state of the world when the case occurred,
- the **solution** which states the derived solution to the problem, and/or
- the **outcome** which describes the state of the world after the case occurred.

- Cases can be represented using the full range of AI representational formalisms (frames, objects, predicates, semantic networks and rules) – the frame/object representation being used by the majority of CBR software.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Case Representation

problems, solutions } derive solutions to new problems

problems, outcomes } evaluate new situations

problems, solutions, outcomes } Evaluate the outcome of proposed solutions and prevent potential problems

There is no consensus as to exactly what information should be in a case. However, two pragmatic measures can be considered when designing a case representation: (a) functionality, and (b) ease of acquisition.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Case Indexing

Assigning **indices** to cases to facilitate their retrieval. Indices should:

- ❑ be predictive
- ❑ address the purposes the case will be used for
- ❑ be abstract enough to allow for widening the future use of the case-base
- ❑ be concrete enough to be recognized in future

Indexing can be done manually or automatically.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example automated indexing methods

- **Checklist-based indexing**: the domain is analyzed, computing the dimensions which are responsible for solving or influencing the outcome of cases; these are put in a checklist and all cases are indexed by their values along these dimensions.
- **Difference-based indexing**: select indices that differentiate a case from other cases, i.e., in the process choose those features that differentiate cases best.
- **Similarity and explanation-based generalization methods**: produce an appropriate set of indices for abstract cases created from cases that share some common set of features, whilst the unshared features are used as indices of the original cases.
- **Inductive learning methods**: identify predictive features that are then used as indices (commonly use variants of the ID3 algorithm for rule induction).
- **Explanation-based techniques**: analyze each case to find which of their features are predictive.

However, despite the success of many automated methods, people tend to do better at choosing indices than algorithms.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Storage: Case memory models

- **Dynamic memory model**
- **Category-exemplar model**

The aim is to strike a balance between the preservation of the semantic richness of cases and their indices on the one hand, and the simplicity of access and retrieval of relevant cases on the other hand, so that **the case-base is organized into a manageable structure that supports efficient search and retrieval methods.**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Dynamic Memory Model

- Memory Organization Packets (MOPs), a form of frames, are the basic unit in dynamic memory.
- Two kinds of MOPs are used:
  - **instances** representing cases, events or objects
  - **abstractions** representing generalized versions of instances or of other abstractions
- The case memory is a **hierarchical structure** of episodic memory organization packets (E-MOPs), also referred to as generalized episodes (GEs), where specific cases which share similar properties are organized under a more general structure (a generalized episode)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Generalized Episodes (GEs)

- A GE contains three different types of objects:
  - **cases**, i.e., the cases indexed under it
  - **norms**, which are the features common to its cases
  - **indices**, which are the features that discriminate between its cases; an index may point to a more specific GE or to a case, and is composed of an index name and an index value
- The case-memory is a **discrimination network** where nodes are either a GE, an index name, index value or a case; index name-value pairs point from a GE to another GE or case
- The primary role of a GE is as an **indexing structure** for storing, matching and retrieval of cases

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Storing a new case

- When a feature (i.e., an index name-value pair) of a new case matches a feature of an existing case, a new GE is created
- If the new case is not identical with the existing case, the two cases are discriminated by indexing them under different indices below the new GE
- Hence the case memory is **dynamic** in that similar parts of two cases are dynamically generalized into a new GE, the cases being indexed under the GE by their differences
- This process can lead to an explosive growth in the number of indices as case numbers increase; a practical solution is to limit the number of permissible indices to a limited vocabulary

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Category-Exemplar Model

- Based on the view that the real world should be defined extensionally with cases being referred to as **exemplars**
- The case memory is a **network structure** of categories, semantic relations, cases and index pointers
- Each case is associated with a category; different case features are assigned different importance in describing a case's membership to a category
- There are three types of indices, which may point to a case or a category:
  - **feature links,** that point from problem descriptors (features) to a case or category
  - **case links** that point from categories to its associated cases
  - **difference links** pointing from categories to the neighboring cases that only different in a small number of features

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Storing a new case

- A category's exemplars are stored according to their degree of prototypicality to the category, and the categories are interlinked within a semantic network containing the features (name-value pairs)
- This network represents a background of general domain knowledge that enables explanatory support to some CBR tasks
- A new case is stored by searching for a matching case and by establishing the relevant feature indices; if a case is found with only minor differences to the new case, the new case may not be retained, or the two cases may be merged

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Retrieval

- CBR would be viable for large scale problems only when retrieval algorithms are efficient at handling thousands of cases
- Bearing in mind that in general no existing case matches exactly a new case, a retrieval algorithm using the indices in the case-memory should retrieve the most similar cases to the current problem or situation
- Well known methods for case retrieval are:
  - Nearest neighbor
  - Induction
  - Knowledge guided induction
  - Template retrieval

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Retrieval Methods

Nearest Neighbor

$$\frac{\sum_{i=1}^{n} w_i \times sim\left(f_i^I, f_i^R\right)}{\sum_{i=1}^{n} w_i}$$

Where, w is the importance weighting of a feature, sim is the similarity function, and $fI\ and\ fR$ are the values for feature i in the input and retrieved cases, respectively.

The biggest problem is to determine the weights of the features. In general, the retrieval time increases linearly with the number of cases; hence the method is more effective with small case bases.

Induction

Induction algorithms (e.g., ID3) determine which features do the best job in discriminating cases and generate a decision tree type structure to organize the cases in memory. This approach is useful when a single case is required as a solution, and where that case feature is dependent upon others.

Knowledge guided induction

Applies knowledge to the induction process by manually identifying the case features that are known or thought to affect the primary case feature.

Template retrieval

Similar to SQL-like queries, template retrieval returns all cases that fit within certain parameters. Often used before other methods (e.g., nearest neighbor) to limit search space.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Adaptation

- Once a matching case is retrieved, its solution should be **adapted** to the needs of the current case.
- Adaptation looks for **prominent differences** between the retrieved case and the current case and then applies formulae or rules that take those differences into account when suggesting a solution.
- Adaptation kinds:

- **Structural adaptation** – adaptation rules applied directly to the solution stored in cases
- **Derivational adaptation or reinstantiation** – reuses the algorithms, methods or rules that generated the original solution to produce a new solution to the current problem; hence the planning sequence that constructed the original solution must be stored in memory along with the solution. This kind of adaptation can only be used for cases that are well understood.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Structural and Derivational Adaptation

- Ideally the adaptation rules must be strong enough to generate complete solutions from scratch.
- Structural adaptation rules adapt **poorly understood solutions**.
- Derivational mechanisms adapt solutions of **cases that are well understood**.
- An efficient CBR system may need both – yet again flexibility calls for multiplicity of mechanisms.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Adaptation Techniques

Null adaptation: Simply use the solution of the retrieved case to the current problem without adapting it. Appropriate for problems involving complex reasoning but with a simple solution.

Parameter adjustment: This is a structural adaptation technique that compares specified parameters of the retrieved and current case to modify the solution in an appropriate direction.

Abstraction and respecialisation: A general structural adaptation technique that is used in a basic way to achieve simple adaptations and in a complex way to generate, novel, creative solutions.

Critic-based adaptation: A critic looks for combinations of features that can cause a problem in a solution and repairs these problems.

Reinstantiation: Instantiate features of an old solution with new features.

Derivational replay: Use the method of deriving an old solution or solution piece to derive a solution in the new situation.

Model-guided repair: Uses a causal model to guide adaptation.

Case-based substitution: Uses cases to suggest solution adaptation.

.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Categories of CBR research

- **true faith** CBR – relating to the cognitive science and AI theoretical issues of CBR
- **hard-core** CBR – testing and refining true faith theories in challenging practical applications
- **CBR-lite** – using selected CBR techniques, such as case-retrieval, to solve particular problems (e.g., help-desks)

Example CBR Applications

- CBR could be applied to solve a problem where **no explicit model exists**
- CBR systems **can learn by acquiring new cases** and so improve their performance with time

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Academic Demonstrators

Legal reasoning

- JUDGE – represents a case-based model of criminal sentencing (murder, man-slaughter and assault cases) and uses its case-base to maintain a consistent sentencing pattern.
- HYPO – case-based reasoning in the area of patent law, centered on cases claiming violation of a patent; the system uses its case-base of precedents to generate plausible arguments for the prosecution and the defense.
- GREBE – uses knowledge in form of generalizations and category exemplars to determine the classification of new cases, its domain being injuries to workers travelling outside of the workplace.
- KICS – accumulates case histories of interpretation of building regulations used to establish precedents; these precedents can be used when revising statutory regulations and enrich the resulting new versions of regulations, as well as equip experts with relevant information to interpret and make decisions about cases coming to appeal.

Knowledge acquisition

- REFINER - helps an expert refine his knowledge in a more natural way than extracting rules from the expert; the system can recognize inconsistent classifications and suggest ways of resolving the inconsistency, using both inferred rules as well as individual cases.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Academic Demonstrators

Explanation of anomalies

- SWALE – a case-based creative explainer, utilizing a library of cases explaining how animals and people die; when given an anomalous event such as the death of a healthy, young horse, it searches for explanations of death in another context such as rock stars dying from drug overdose, or a spouse being murdered for life insurance; such explanations are adapted to fit the current situation.

Diagnosis

- PROTOS – was developed for the domain of clinical audiology using cases from a speech and hearing clinic and reaching an absolute accuracy of 100%
- CASEY – diagnoses heart failure using as input a patient's symptoms and produces a causal network of possible internal states that could lead to those symptoms; retrieves similar cases and adapts the retrieved diagnosis by considering differences in symptoms.
- CASCADE – diagnoses the causes of crashes to the VMS operating system to suggest a solution; it has no adaptation but assists effective recovery from a system crash.
- PAKAR – identifies possible causes for building defects and suggests remedial actions.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Academic Demonstrators

Arbitration

- MEDIATOR – works in the domain of dispute resolution; given a conflict between several parties, it proposes possible compromises, and if one proposal fails to satisfy all parties involved, it generates new proposals and records the failure thus avoiding a similar failure in the future.
- PERSUADER – proposes resolution of disputes between labor and management; a unique feature of this system is that it has a backup planner that can generate new contracts when no existing ones can be found or adapted.

Tutoring

DECIDER – helps students understand or resolve a pedagogical problem by selecting and presenting appropriate cases.

Design

- CYCLOPS – solves design problems in the domain of landscaping.
- JULIA – works in the domain of meal planning, using cases to propose solutions, decomposing the problem as necessary and posting constraints to guide synthesis.
- CADET – functions as designer's assistant for mechanical design.
- ARCHIE – helps architects in the high-level task of conceptual design, by giving them access to office building designs created by other architects and points to factors that should be considered in solving a given case.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Academic Demonstrators

Planning

- BATTLE – projects the results for plan evaluation in the domain of land warfare planning; the user describes a battle situation and chosen battle plan, and the system retrieves cases that are composed of pieces of battles and experts' evaluation to project the outcome; the system was built from an existing database of 600 cases.
- BOLERO – builds a diagnostic plan according to information known about a patient; used collaboratively with a rule-based system that has knowledge about specific diseases while BOLERO has planning knowledge about these diseases.
- TOTLEC – solves complex manufacturing planning problems.

Repair and adaptation

- CHEF – creates new recipes from old ones; one of its important aspects is explanation of failures through a causal description of why they occurred, where recipes are linked to a failure's explanation to predict the failure in similar circumstances; stores new recipes indexed by the goals that they satisfy and the problems that they avoid.
- PLEXUS – is a planner that adapts old plans to new situations.
- COACH – generates new football plays by improving old plays by debugging and repairing stored plans; it has several strategies for adaptation to form new plans.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Example Commercial Applications

- Lockheed – CLAVIER
- British Airways – CaseLine
- Legal & General - SWIFT

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Lockheed – CLAVIER

https://www.aaai.org/Papers/IAAI/1994/IAAI94-007.pdf

- The first commercially fielded CBR application
- Applying CBR to composite part fabrication:
  - reuse previously successful loadings
  - reduce the pressure of work on one or two experts
  - secure the expertise of the experts as a corporate asset
  - help to train new personnel
- CLAVIER acts as a collective memory for Lockheed and as such provides a uniquely useful way of transferring expertise between autoclave operatives.
- It also demonstrates the ability of CBR systems to learn, its case-base growing from 20 to over 150 successful layouts

Clavier is a case-based reasoning (CBR) system that assists in determining efficient loads of composite material parts to be cured in an autoclave. Clavier's central purpose is to find the most appropriate groupings and configurations of parts (or *loads*) in order to maximize autoclave throughput while assuring that parts are properly cured. Clavier uses case-based reasoning to match a list of parts that need to be cured against a library of previously successful loads and suggest the most appropriate next load. Clavier also uses a heuristic scheduler to generate a sequence of loads that best meets production goals while satisfying operational constraints. The system is being used daily on the shop floor and has virtually eliminated the production of low-quality parts that must be scrapped, saving thousands of dollars each month. As one of the first fielded case-based reasoning systems, Clavier demonstrates CBR to be a practical technology that can be used successfully in domains where more traditional approaches are difficult to apply.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

British Airways - CaseLine

- Assists Boeing 747-400 technical support engineers with aircraft fault diagnosis and repair between aircraft arrival and departure.
- It advises on past defects and known successful recovery and repair procedures; users can input diagnostic information and control the search for available repair and recovery information.
- BA identified the following benefits of CBR:
  - CBR is intuitive to both developers and users
  - CBR complements human reasoning and problem solving
  - CBR retains a rich context of a problem situation – they discard nothing but simply index on different features of what they store; very important from the **legal perspective** since a simple defense accrues by demonstrating that engineers had followed procedures that had proved successful in case X, while a more complex defense accrues from the use of a rule-based system where an expert witness might need to prove that a rule (representing distilled knowledge) was theoretically correct

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Legal & General - SWIFT

- L&G are a major UK provider of financial services
- As part of a business process reengineering project, they wanted to provide a streamlined service to employees purchasing PC's, peripherals, software or upgrades.
- All L&G's employees would have access to a single point of contact for ordering PC products and upgrades.
- After approximately one month's prototyping, a demonstration system, called SWIFT, was shown at several seminars to stakeholders from all the business units within L&G; the presentations were carefully organized as part of a **comprehensive communications plan**.
- Following this an intensive system development stage took place that involved establishing the required databases and obtaining cases from different business units.
- SWIFT demonstrates that:
  - CBR systems can be developed quickly, can be integrated within a wider information system, and once established can be maintained by people who are not programmers.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Expert and Experience derive from the same root …

- Experts solve problems by applying their experience
- Novices attempt to solve problems by applying rules they have recently acquired
- The application of experience to problem solving is the hallmark of CBR
- Claims made by CBR researchers:
  - CBR is proposed as a psychological theory of human cognition
  - CBR provides a cognitive model of how people solve problems
  - CBR offers a paradigm close to the way people solve problems and that overcomes the brittleness of model-based reasoning systems

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Potential advantages of CBR

- Avoids the knowledge elicitation bottleneck … acquiring past cases is easier
- A model does not need to exist
- Implementation is an easier task as it focuses on identifying relevant case features, and in fact a CBR system can be rolled out with only a partial case-base; indeed, such a system need never be complete since it will be continually growing with the addition of new cases – by analogy how to tell when a knowledge-base is complete
- No need to infer an answer from first principles
- Individual or generalized cases can be used to provide explanations that are perhaps more satisfactory than explanations generated by chains of rules
- Learning is done through the acquisition of new cases making maintenance easier
- CBR systems can grow to reflect their organization's experience

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

But there are also challenges for CBR

- How cases should be represented
- How indices should be chosen for organizing memory efficiently
- How to structure relationships between cases and parts of different cases
- How to handle cases containing multimedia
- How to handle massive case-bases
- How to develop general adaptation heuristics for modifying previous cases or their solutions to fit new cases – this is considered the **biggest challenge**
- Moreover, CBR systems should be able to:
  - forget unused cases in order to maintain case-base efficiency
  - learn about the emergence of any indices that had not been previously been thought significant
  - Continually learn from interactions with new students

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Finally, the issue of validation

- Relying on previous experience without validation may result in inefficient or incorrect solutions being recommended causing an increase in problem-solving time or errors that may have a negative effect on the process of learning.
- The validation issue and the detection of biases very much haunts the application of data-driven approaches in general.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Intelligent Data Analysis

with a focus on medicine

- ❑ Data Abstraction
- ❑ Data Mining

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on cased-based reasoning and intelligent data analysis, students will be able:

Regarding intelligent data analysis (IDA):

1. To distinguish between IDA, Knowledge Discovery in Databases (KDD) and Data Mining (DM).
2. To explain the role of IDA systems in a clinical setting and draw a comparison between the classical knowledge-based systems and the knowledge-based systems augmented with IDA capabilities.
3. To present the categorization of IDA methods into data abstraction and data mining methods.
4. To discuss the importance of time and give key temporal abstraction methods.
5. To distinguish between a directed, goal-driven, and a nondirected, event-driven, mode of deployment of data abstraction, and explain the two basic ways of integrating a data abstraction process into a problem-solving system.
6. To give a cursory listing of some data mining methods for symbolic classification, CBR being an exemplar of these.
7. To conjecture on the potential integration of temporal data abstraction and data mining for the purpose of knowledge discovery.

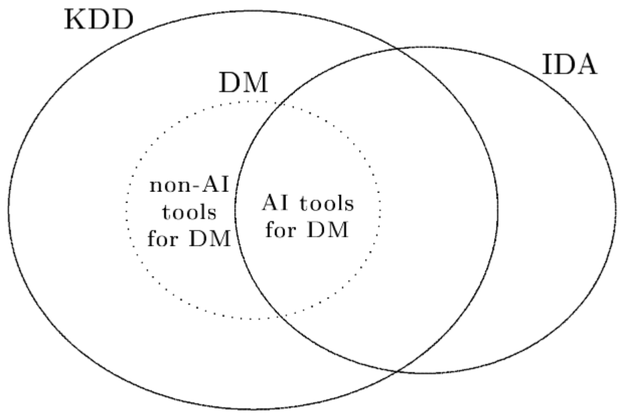
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

IDA, KDD and DM

Intelligent Data Analysis (IDA) encompasses statistical, pattern recognition, machine learning, data abstraction and visualization tools to support the analysis of data and discovery of principles that are encoded within the data.

Knowledge Discovery in Databases (KDD) is a process consisting of the following steps: understanding the domain, forming the dataset and cleaning the data, extracting the regularities hidden in the data thus formulating knowledge in the form of patterns, rules, etc. [this step is usually referred to as **Data Mining** (DM)], postprocessing of discovered Knowledge and exploitation of results.

IDA and KDD have in common the topic of investigation, which is data analysis, and they share many common methods; however, IDA uses AI methods and tools while KDD employs both AI and non-AI methods. Moreover, KDD is typically concerned with the extraction of knowledge from very large datasets, whereas in IDA the datasets are either large or moderately sized.



CBR can be categorized under DM

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The role of IDA systems in a clinical setting

- Their role is that of an intelligent assistant that tries to **bridge the gap between data gathering and data comprehension,** in order to enable the physician to perform his task more efficiently and effectively; the physician must have at his disposal the right information at the right time.
- Nowadays is possible to store large volumes of data from diverse sources on electronic media; data could be on a single case (e.g., one patient) or multiple cases.
- Raw data are of little direct use; their sheer volume and/or very specific level makes impossible their operationalization in problem solving.
- But such data can be converted to a mine of information wealth if the real gems of information are extracted from the data by computationally intelligent means.
- Useful, operational information/knowledge is expressed at the right level of abstraction and made readily available to support the decision making.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The globality of data and information calls for …

- The provision of standards in terminology, vocabularies, and formats to support multilingualism and sharing of data
- Standards for the abstraction and visualization of data
- Standards for interfaces between different sources of data
- Integration of heterogeneous types of data, including images and signals
- Standards for electronic patient records
- Reusability of data, knowledge, and tools

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

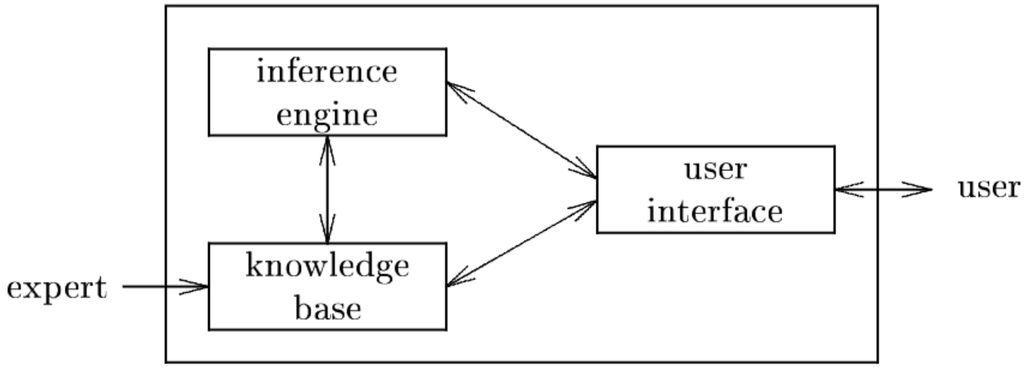
Data Abstraction and Data Mining

- IDA methods applied to supporting decision making in medicine can be classified into two main categories: data abstraction and data mining.
- **Data abstraction** is concerned with the intelligent interpretation of patient data in a context-sensitive manner and the presentation of such interpretations in a visual or symbolic form, where the temporal dimension in the representation and intelligent interpretation of patient data is of primary importance.
- **Data mining** is concerned with the analysis and extraction (discovery) of medical knowledge from data, aimed at supporting diagnostic, screening, prognostic, monitoring, therapy support, or overall patient management tasks.
- Most DM methods belong to **machine learning** and most data abstraction methods perform **temporal abstraction**.

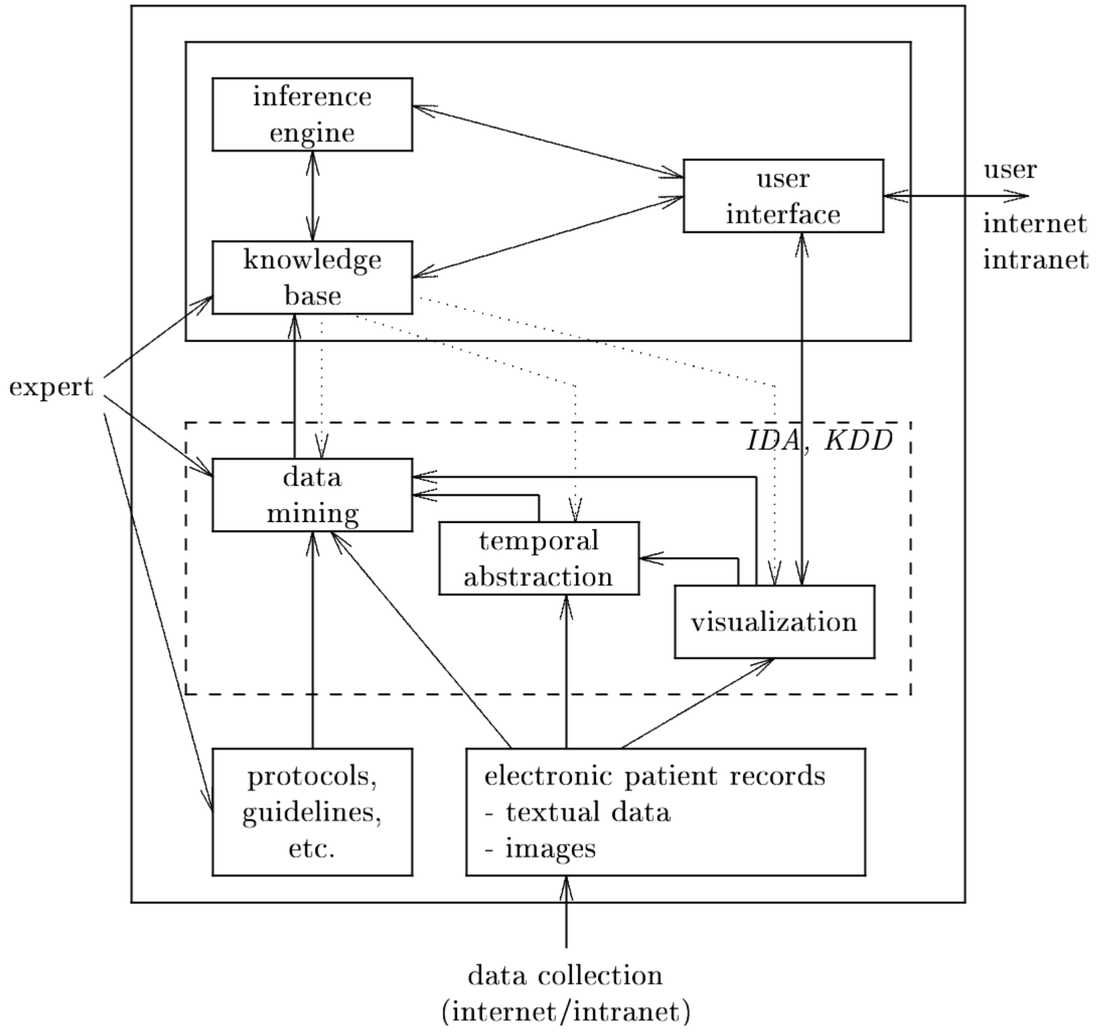
Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Knowledge versus Data



The classical architecture of an expert system



A decision support schema in the age of data: arrows denote normal information flow, and dotted arrows represent information flow in processes involving iteration and loops between the different steps of the IDA process

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Data Abstraction Methods

- **Support specific knowledge-based problem-solving activities** (data interpretation, diagnosis, prognosis, monitoring, etc.) by extracting useful abstractions from the raw, mostly numeric data.
- Temporal data abstraction methods represent an important subgroup where the processed data are temporal.
- The derivation of abstractions is often done in a context-sensitive and/or distributed manner and it applies to discrete and continuous supplies of data.
- The abstraction can be performed over a single case (e.g., a single patient) or over a collection of cases.
- The data abstraction methods are **knowledge-driven** (both general and specialist knowledge)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Data Mining Methods

- Extract knowledge, preferably in a **meaningful and understandable symbolic** form.
- Most frequently applied methods are supervised symbolic machine learning methods, e.g., effective tools for inductive learning exist that can be used to generate understandable diagnostic and prognostic rules.
- Other methods include symbolic clustering, discovery of concept hierarchies, qualitative model discovery and learning of probabilistic causal networks.
- Sub-symbolic learning (e.g., nearest-neighbor method, Bayesian classifier, and (non-symbolic) clustering) and case-based reasoning methods can also be classified in the DM category.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Simple atemporal data abstractions – also presented in Unit 8

- **Qualitative abstractions**, where a numeric expression is mapped to a qualitative expression; such abstractions are based on simple associational knowledge, e.g., (“a temperature of at least 39 degrees C”, “fever”).
- **Generalization abstraction**, where an instance is mapped to (one of) its classes; such abstractions are based on strict or tangled concept taxonomies.
- **Definitional abstraction**, where a datum from one conceptual category is mapped to a datum in another conceptual category that happens to be its definitional counterpart in the other context; the resulting concept must be more abstract than the originating concept, e.g., it refers to something more easily observable. The knowledge driving such abstractions consists of simple associations between concepts across different categories.
- In an atemporal situation everything is assumed to refer to “now”: $\boldsymbol{holds(P,D) \rightarrow holds(P,abs(D))}$

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Why time is important … particularly in medicine

- Time is intrinsic to many problem domains where **dynamic situations** arise – temporal reasoning is largely **commonsense reasoning**
- In medicine:
  - Disease processes evolve in time
  - Patient records give the history of patients
  - Therapeutic actions, like all actions, are indescribable without considering time
- The modelling of time enables a more accurate formation of potential solutions:
  - The presence of an abnormality may not be diagnostically significant as such, but its specific pattern of appearance is
  - The expected picture of a disease is different, depending on the state of its evolution
- These call for **temporal abstractions**

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Temporal data abstraction

- **Temporal data abstraction** is a fundamental intermediate reasoning process for the intelligent interpretation of temporal data in support of tasks such as diagnosis, monitoring, etc.; unlike atemporal abstraction where a single datum is abstracted, here sets of data, or more accurately time series of data, can be abstracted to single data.
- **Background domain knowledge** can be effectively utilized in the context of temporal data abstraction, e.g., persistence semantics of concepts
- **Commonsense reasoning** involves the intuitive handling of multiple time granularities and temporal relations such as before, overlaps, and disjoint.
- Patient data can be considered as **temporal objects**, where a temporal object is an integral association between an item of information and a time.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Types of temporal data abstractions – also presented in Unit 8

- **Merge abstraction**, where a collection of data, sharing the same **concatenable property** and whose temporal aspects collectively form a (possibly overlapping) chain (at some time granularity) are abstracted to a single datum with the given property whose temporal aspect is the maximal time interval spanning the original data; also known as **state abstraction**.
- **Persistence abstraction**, where again the aim is to derive maximal intervals spanning the extent of some property; here, though, there could be just one datum on that property, and hence the difficulty is in filling the gaps by ‘seeing’ both backward and forward in time from the specific, discrete recording of the given property.
  - **Default persistence rule**: some property is assumed to persist indefinitely until some event (e.g., a therapy) is known to have taken place and this terminates the persistence of the property.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Types of temporal data abstractions – also presented in Unit 8

- **Trend abstraction**, where the aim is to derive the significant changes and the rates of change in the progression of some parameter; it entails merge and persistence abstraction in order to derive the extents where there is no change in the value of the given parameter.
- **Periodic abstraction**, where repetitive occurrences with some regularity in the pattern of repetition are derived, e.g., headache every morning for a week of increasing severity:
  - repetition element, e.g., headache – it can be of any order of complexity, e.g. it could itself be a periodic abstraction, or a trend abstraction
  - repetition pattern, e.g., every morning for a week
  - progression pattern, e.g., increasing severity
- The data abstraction types can be combined in a multitude of ways, yielding **complex abstractions.**

Co-financed by the European Union
Connecting Europe Facility

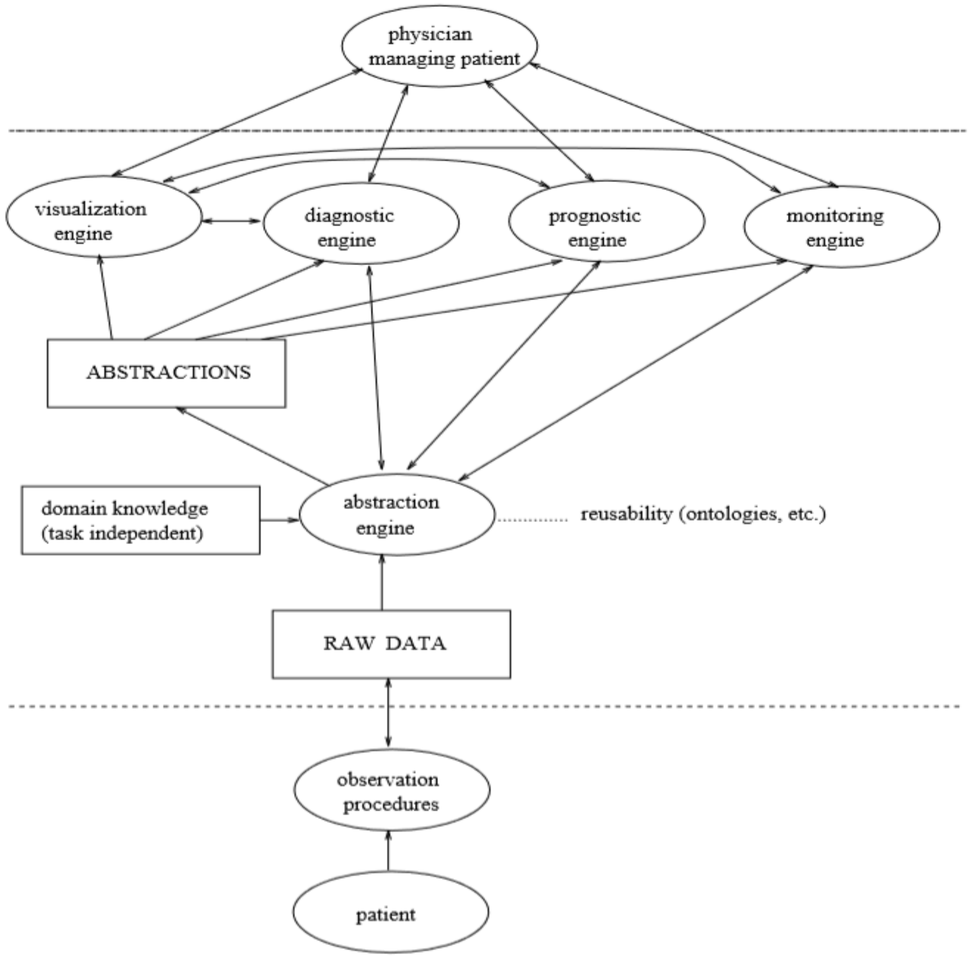
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Modes of deployment of data abstraction

- **Directed or goal-driven,** i.e., the problem-solving system (in exploring its hypothesis space) predicts various abstractions that the data abstraction process is required to corroborate against the raw patient data.
- **Nondirected or event-driven**, e.g., in a monitoring system; the aim is to comprehensively interpret all the data covered by a (moving) time window.
- Nondirected data abstraction can be used in a **stand-alone fashion** where the derived abstractions should be presented to the user in a visual form; **visualization** is also of relevance when a data abstraction process is not used in a stand-alone fashion as it is a good way of justifying the reasoning.
- **Truth maintenance** is of relevance to any inference system: as raw data may be received out of temporal sequence, abstractions referring to the present may need to be modified, or abstractions referring to the past are revoked by new data.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

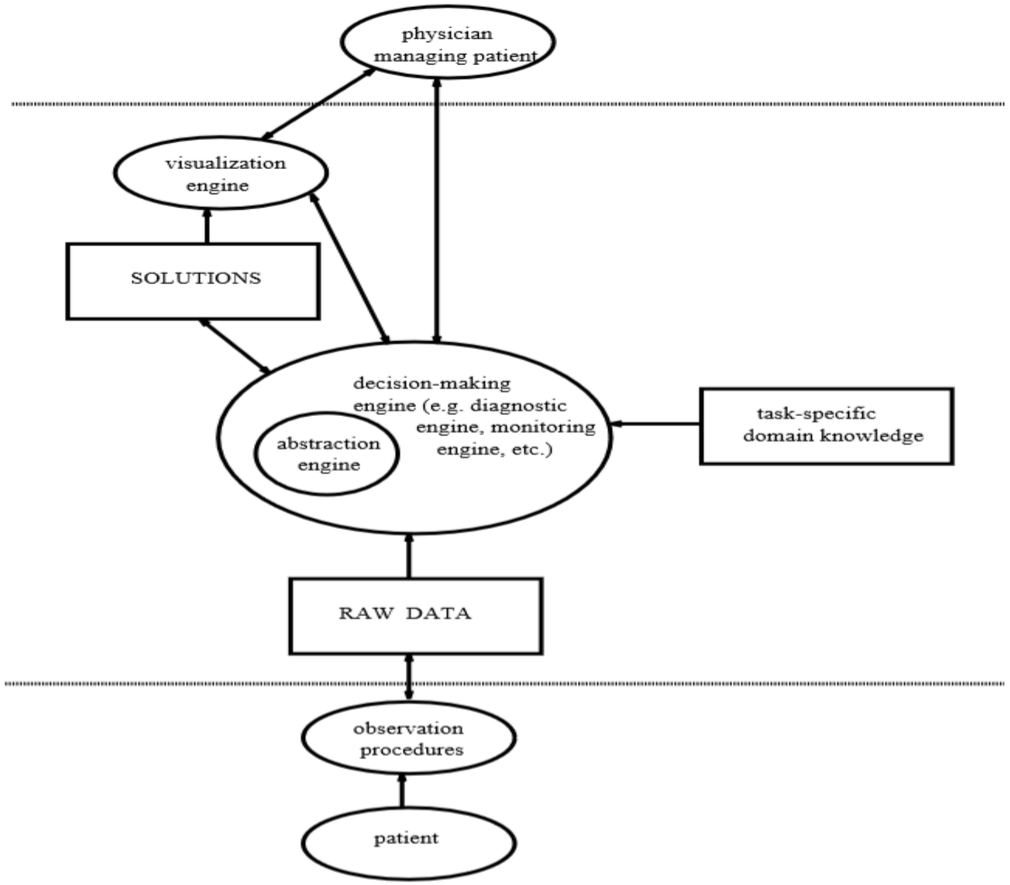
Integration of data abstraction into a problem-solving system



Data abstraction as a loosely coupled process

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Integration of data abstraction into a problem-solving system



Data abstraction as a task-dependent process

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Data Mining through Symbolic classification methods

- **Rule Induction** – given a set of classified examples, a rule induction system constructs a set of rules: IF Conditions THEN Conclusion

Example rule induced by CN2 in the domain of early diagnosis of rheumatic diseases

IF Sex = male

AND Age > 46

AND Number_of_painful-joints > 3

AND Skin_manifestations = psoriasis

THEN Diagnosis = Crystal_induced_synovitis

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Data Mining through Symbolic classification methods

- **Rough Sets** – If-then rules can also be induced by using the theory of rough sets which are concerned with the analysis of classificatory properties of data aimed at approximations of concepts.
- The basic concept is an **indiscernibility** relation: two objects x and y are indiscernible based on the available attribute subset B if they have the same values of attributes B; the set of objects indiscernible from x using attributes B forms an equivalence class.
- A main task is to find minimal subsets of attributes that preserve the indiscernibility relation; this is called **reduct** computation.
- Decision rules are generated from reducts by reading off the values of the attributes in each reduct.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Data Mining through Symbolic classification methods

- **Association Rules** – given a set of transactions, where each transaction is a set of items (i.e., literals of the form *Attribute* = *value*), an association rule is an expression of the form $X \rightarrow Y$, where X and Y are sets of items; the intuitive meaning of such a rule is that transactions in a database that contain X tend to contain Y.
- Example" "80%" of patients with pneumonia also have high fever. 10% of all transactions contain both these items; 80% is the confidence of the rule and 10% its support.
- **Confidence** of the rule is the ratio of the number of records having true values for all items in X and Y to the number of records having true values for all items in X.
- **Support** of the rule is the ratio of the number of records having true values for all items in X and Y to the number of all records in the database.
- Association rule learners use minimum support and minimum confidence constraints.

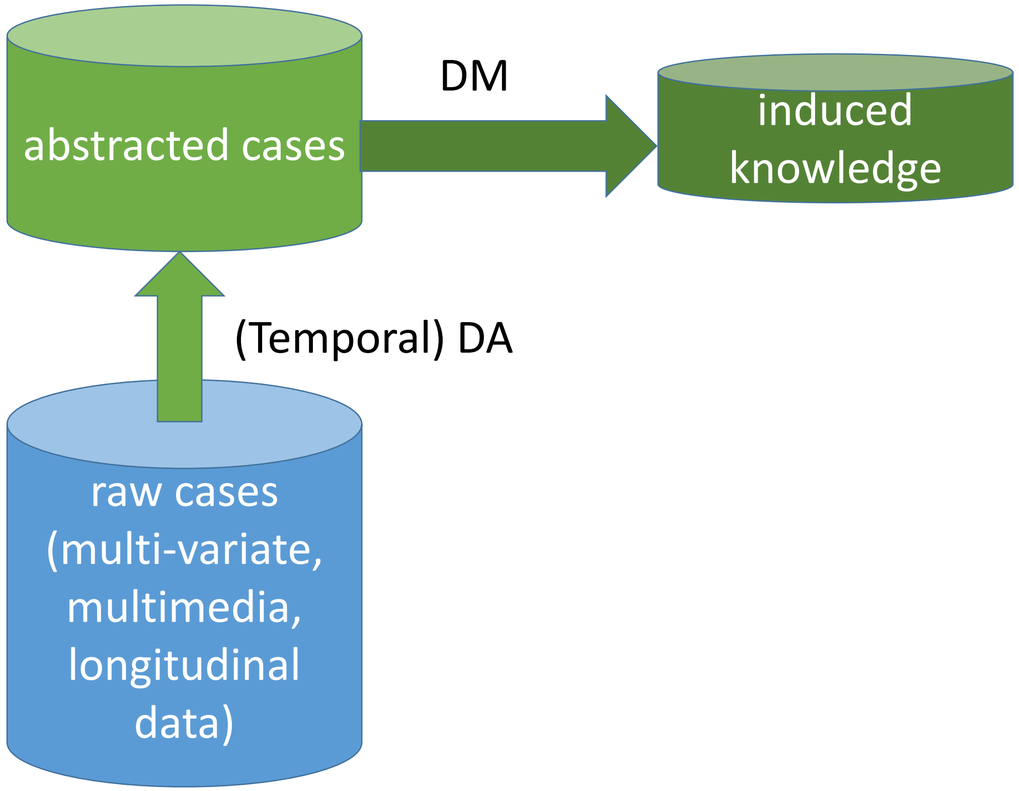
This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Other symbolic classification methods

- **Learning of classification and regression trees** – Systems for top-down induction of decision trees generate a decision tree from a given set of attribute-value tuples; each of the interior nodes of the tree is labeled by an attribute, and branches that lead from the node are labeled by the values of the attribute. The tree construction process is heuristically guided by choosing the most informative attribute at each step, aimed at minimizing the expected number of tests needed for classification.
- **Inductive Logic Programming (ILP)** – ILP systems learn relational concept descriptions from relational data, in the form of Prolog clauses.
- **Discovery of concept hierarchies** – Decompose a classification dataset to equivalent but smaller, more manageable, and potentially easier to comprehend datasets. Function decomposition is such a method, which besides the discovery of appropriate datasets it arranges them into a concept hierarchy.
- **Constructive induction** – An ability of the system to derive and use new attributes in the process of learning.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Data abstraction for knowledge discovery



If the same complex abstraction, such as a nested periodic occurrence, is associated with a significant number of patients from a representative sample, it makes a strong candidate for being a significant piece of knowledge; sharing a complex abstraction is a strong similarity, whereas sharing a concrete datum is a weak similarity, if at all.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Summary

- CBR as an alternative to Model-Based Reasoning
- The CBR Cycle – the four REs
- CBR example academic demonstrators and commercial applications
- IDA, KDD and DM
- KBS augmented with IDA
- Data abstraction – temporal data abstraction – modes of deployment and problem-solving system integration
- Integration of data abstraction with data mining for knowledge discovery

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI611 Fundamentals of Artificial Intelligence

Elpida Keravnou-Papailiou

September – December 2022

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Artificial Intelligence and Explanation

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

UNIT 2

Artificial Intelligence and Explanation

CONTENTS

1. The significance of explanation in AI
2. Some theories of explanation that have influenced AI
3. Tracing the history of explanations in symbolic AI
4. The resurgence of interest in explanation in connectionist AI – opening the ‘black box’

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

INTENDED LEARNING OUTCOMES

Upon completion of this unit on Artificial Intelligence and explanation, students will be able:

1. To enhance further their understanding (as distilled from the previous course units) of the significance of explanation in relation to AI systems.
2. To discuss C.S. Peirce's and P. Thagard's general theories of explanation that have influenced the AI field, and the role of causality in the production of explanations.
3. To trace the history of explanations in symbolic AI, pointing out key milestones (rule-based explanations, strategic explanations, user-tailored explanations, case-based explanations).
4. To outline the recent resurgence of interest in explanation, in relation to connectionist AI, and the establishment of the research field referred to as XAI (eXplainable AI) aiming to 'open' the black box.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

The significance of explanation in AI

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Why do AI systems need explanations?

There are many (critically) important reasons ……

- In general, decision support systems must be **interpretable** and not black-boxes – recall the roles of expert knowledge-based systems as consultants, critics or tutors.
- By and large AI systems are **interactive**, i.e., they do not just get an input, process it and give an output, but they engage in a dialogue with a human user, who needs to take a 'final' decision that could impact on another human (e.g., a patient) or an organization, the society, etc.
- Particularly **critical domains** are the medical/health care, legal, and defense domains.
- Explanations have at least a dual purpose: (i) **understanding the logic/model of the system**, also facilitating 'debugging' (e.g., revealing biases in logic/data and erasing them); (ii) **understanding the rationale of the recommended outcome of a specific consultation** and be convinced of its validity; recall that AI systems are complex software systems deploying algorithms and knowledge/data.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Why do AI systems need explanations?

- Traditionally, the central role of the explanation model is to reveal the system's reasoning; however, it has a subsidiary role in relation to **information-acquisition interactions**, that concerns individual items of information rather than the system reasoning processes:
  - The user needs to be able to ask, not only why the system is asking a particular question (i.e., how does it relate to the reasoning process), but also what the given question means.
- Nowadays the strive for **responsible, trustworthy and ethical AI**, emphasizes even more the need for AI systems to be bestowed with appropriate, user-tailored and hence **fit for purpose**, explanation models; different categories of users have different explanation needs.
- EU's General Data Protection Regulation (**GDPR**) and **ACM's Statement on Algorithmic Transparency and Accountability** make direct references to the need for explanation while the European Research Consortium for Informatics and Mathematics (ERCIM) devoted one of its special issues on **transparency in algorithmic decision making**.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

GDPR

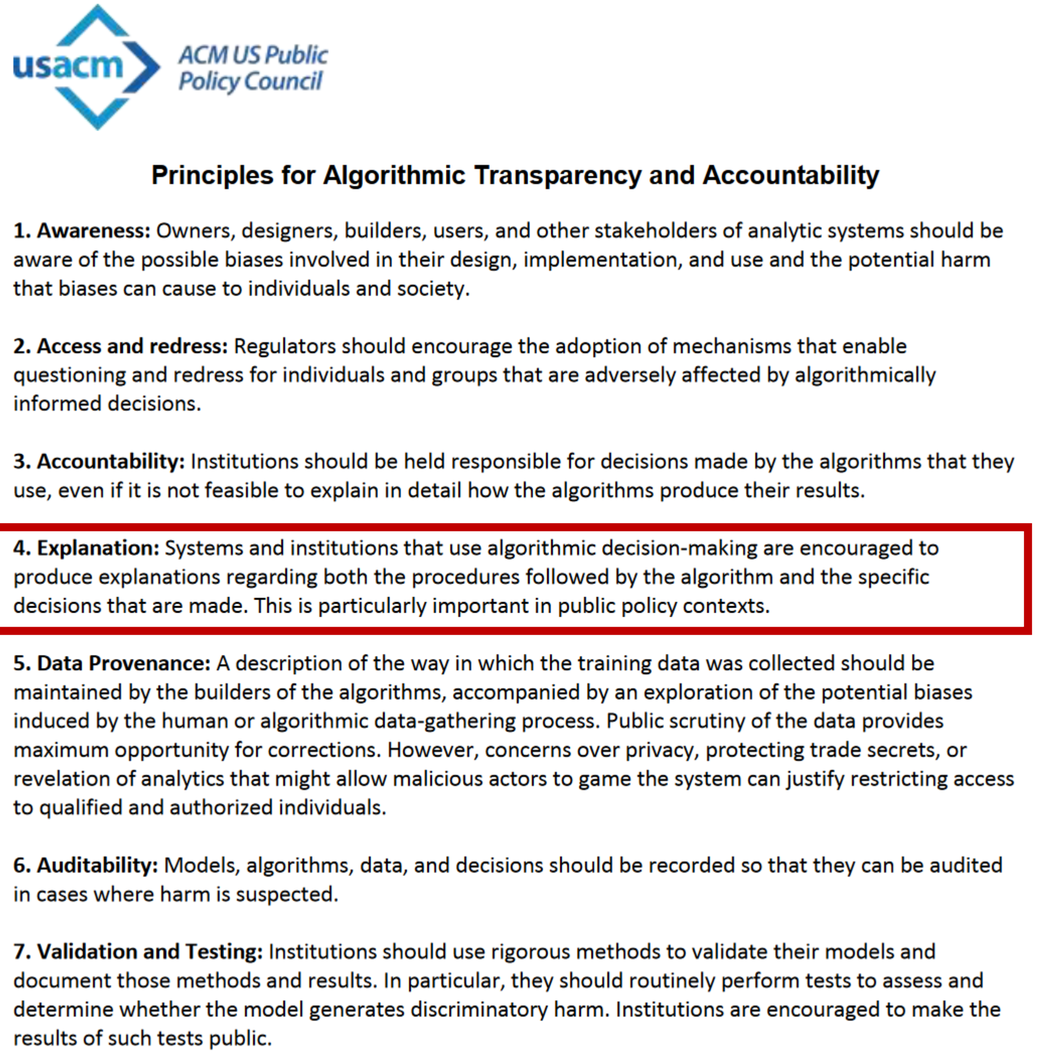
According to R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi and F. Giannotti, “A survey of methods for explaining black box models”, ACM Computing Surveys 51(5):93, 2018, DOI 10.1145/3236009:

An innovative aspect of the GDPR, which has been debated, are the clauses on automated (algorithmic) individual decision-making, including profiling, which for the first time introduce, to some extent, a right of explanation for all individuals to obtain **“meaningful explanations of the logic involved”** when automated decision making takes place. Despite divergent opinions among legal scholars regarding the real scope of these clauses, everybody agrees that the need for the implementation of such a principle is urgent and that it represents today a huge open scientific challenge. **Without an enabling technology capable of explaining the logic of black boxes, the right to an explanation will remain a “dead letter”.**

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

ACM Policy Council: Statement on Algorithmic Transparency and Accountability, 2017.
https://www.acm.org/binaries/content/assets/public-policy/2017_usacm_statement_algorithms.pdf



usacm ACM US Public Policy Council

Principles for Algorithmic Transparency and Accountability

1. Awareness: Owners, designers, builders, users, and other stakeholders of analytic systems should be aware of the possible biases involved in their design, implementation, and use and the potential harm that biases can cause to individuals and society.

2. Access and redress: Regulators should encourage the adoption of mechanisms that enable questioning and redress for individuals and groups that are adversely affected by algorithmically informed decisions.

3. Accountability: Institutions should be held responsible for decisions made by the algorithms that they use, even if it is not feasible to explain in detail how the algorithms produce their results.

4. Explanation: Systems and institutions that use algorithmic decision-making are encouraged to produce explanations regarding both the procedures followed by the algorithm and the specific decisions that are made. This is particularly important in public policy contexts.

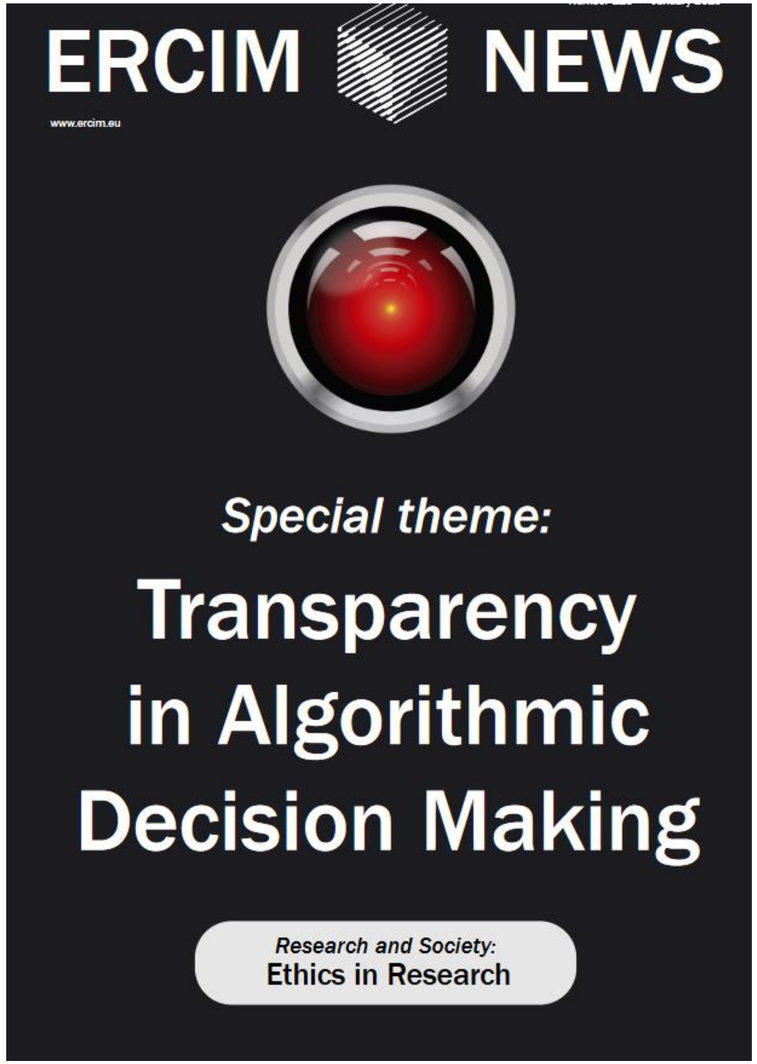
5. Data Provenance: A description of the way in which the training data was collected should be maintained by the builders of the algorithms, accompanied by an exploration of the potential biases induced by the human or algorithmic data-gathering process. Public scrutiny of the data provides maximum opportunity for corrections. However, concerns over privacy, protecting trade secrets, or revelation of analytics that might allow malicious actors to game the system can justify restricting access to qualified and authorized individuals.

6. Auditability: Models, algorithms, data, and decisions should be recorded so that they can be audited in cases where harm is suspected.

7. Validation and Testing: Institutions should use rigorous methods to validate their models and document those methods and results. In particular, they should routinely perform tests to assess and determine whether the model generates discriminatory harm. Institutions are encouraged to make the results of such tests public.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

A. Rauber, R. Trasarti and F. Giannotti (eds.), Transparency in algorithmic decision making, Special theme, ERCIM News, Number 116, January 2019.
https://ercim-news.ercim.eu/images/stories/EN116/EN116-web.pdf



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

High-Level Expert Group on Artificial Intelligence

Sabine Theresia Köszegi, Professor of Labor Science and Organization Institute of Management Science, TU Wien, Chair of the Austrian Council on Robotics and Artificial Intelligence, BMVIT, Member of the High-Level Expert Group on Artificial Intelligence of the European Commission.

On 25 April 2018, the Europaen Commission published a Communication in which it announced an ambitious European Strategy for Artificial Intelligence (AI). The major advances in AI over the last decade revealed its capacity as a general-purpose technology and pushed inventions in areas of mobility, healthcare, home & service robotics, education and cyber security, to name just a few. These AI-enabled developments have the capability to generate tremendous benefits not only for individuals but also for the society as a whole. AI has also promising capabilities when it comes to address and resolve the grand challenges, such as climate change or global health and wellbeing, as expressed in the United Nations Sustainable Development goals. In competition with other key players, like the United States and China, Europe needs to leverage its current strengths, foster the enablers for innovation and technology uptake and find its unique selling proposition in AI to ensure a competitive advantage and a prosperous economic development in its Member States. At the same time, AI comes with risks and challenges associated to fundamental human rights and ethics. Europe therefore must ensure to craft a strategy that maximizes the benefits of AI while minimizing its risks.

The Commission has set out an interwoven strategy process between the development of a European AI Strategy and the development of a Coordinated Action Plan of Member States (hosted under the Digitising European Industry framework). The publication of the European policy and investment strategy on AI is envisaged for Summer 2019. To support this strategy development process and its implementation, the Commission has called for experts to establish a High-Level Expert Group on Artificial Intelligence (AI HLEG). Following an open selection process by DG Connect in spring 2018, the Commission has appointed 52 experts encompassing representatives from different disciplines of academia, including science and engineering disciplines and humanities alike, as well as representatives from industry and civil society. As an expert in labor science and with a research background in decision support systems, I was selected to join the exciting endeavor to lay the foundations for a human-centric, trustworthy AI in Europe that strengthens European competitiveness and addresses a citizen perspective to build an inclusive society.

Our mandate includes the elaboration of recommendations on the policy and investment strategy on ethical, legal and societal issues related to AI, including socio-economic challenges. Additionally, we serve as a steering group for the European AI Alliance to facilitate the Commission's outreach to the European society by engaging with multiple stakeholders, sharing information and gathering valuable stakeholder input to be reflected in our recommendations and work.

On 18 December 2018, we proposed a first draft on "Ethics Guidelines towards Trustworthy AI" to the Commission, setting out the fundamental rights, principles and values that AI has to comply with in order to ensure its ethical purpose. Additionally, we have listed and operationalized requirements for trustworthy AI as well as provided possible technical and non-technical implementation methods that should provide guidance on the realization of trustworthy AI. This draft on ethics guidelines is currently in a public consultation process in the European AI Alliance platform. Through this engagement with a broad and open multi-stakeholder & citizen forum across Europe and beyond, we aim to secure the open and inclusive discussion of all aspects of AI development and its impact on society. The finalised draft will be formally presented in the First Annual Assembly of the European AI Alliance in Spring 2019.

To advise the Commission with regards to the European policy and investment strategy, we are currently preparing a set of recommendations on how to create a valuable ecosystem for AI in Europe in order to strengthen Europe's competitiveness. The draft document of recommendations should be published in April 2019 and will undergo a public consultation process as well. The recommendations will primarily address European policy makers and regulators but also relevant stakeholders in Member States encompassing investors, researchers, public services and institutions. I would like to use the opportunity, to invite the readers of ERCIM News to engage in the European AI Alliance (see the link below) and to contribute your expertise and input to our policy and investment recommendations.

The complexity of AI-related challenges requires to set up a problem-solving process with highest information processing capacities that allows to consider different perspectives and to resolve conflicts of interest between different stakeholders. It can easily be imagined that our discussions as an inter-disciplinary expert and multi-stakeholder group are intense, difficult and at times emotional. In difficult situations, I remind myself of our commitment to the following statement in our ethics guidelines: "Trustworthy AI will be our north star, since human beings will only be able to confidently and fully reap the benefits of AI if they can trust the technology."

AI Alliance:
https://ec.europa.eu/digital-single-market/en/european-ai-alliance

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Some theories of explanation that have influenced AI

- C.S. Peirce's hypothesis of abduction – finding the most likely explanation of a set of observations
- P. Thagard's theory of explanatory coherence

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

C.S. Peirce's hypothesis of abduction

Has its origins on Peirce's **architecture of theories** (https://arisbe.sitehost.iu.edu/menu/library/bycsp/arch/arch.htm)

The Architecture of Theories

By Charles S. Peirce

The Monist, v. I, n. 2, 1891 January, pp. 161–176. At Google Books. At Internet Archive.
Reprinted: *Writings* v. 8 (2010), 199–211; *The Essential Peirce* v. 1 (1992), 285–297; *Collected Papers* v. 6 (1931), paragraphs 7–34.
Also: *Logic of Interdisciplinarity* (2009), 58–69; *Values in a Universe of Chance* (1958), 142–159; *Philosophical Writings* (1940), 315–323; *Chance, Love and Logic* (1923), 157–178.

OF the fifty or hundred systems of philosophy that have been advanced at different times of the world's history, perhaps the larger number have been, not so much results of historical evolution, as happy thoughts which have accidently occurred to their authors. An idea which has been found interesting and fruitful has been adopted, developed, and forced to yield explanations of all sorts of phenomena. The English have been particularly given to this way of philosophising; witness, Hobbes, Hartley, Berkeley, James Mill. Nor has it been by any means useless labor; it shows us what the true nature and value of the ideas developed are, and in that way affords serviceable materials for philosophy. Just as if a man, being seized with the conviction that paper was a good material to make things of, were to go to work to build a *papier mâché* house, with roof of roofing-paper, foundations of pasteboard, windows of paraffined paper, chimneys, bath tubs, locks, etc., all of different forms of paper, his experiment would probably afford valuable lessons to builders, while it would certainly make a detestable house, so those one-idea'd philosophies are exceedingly interesting and instructive, and yet are quite unsound.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

From the Stanford Encyclopedia of Philosophy

https://plato.stanford.edu/entries/abduction/index.html#DedIndAbd

Abduction

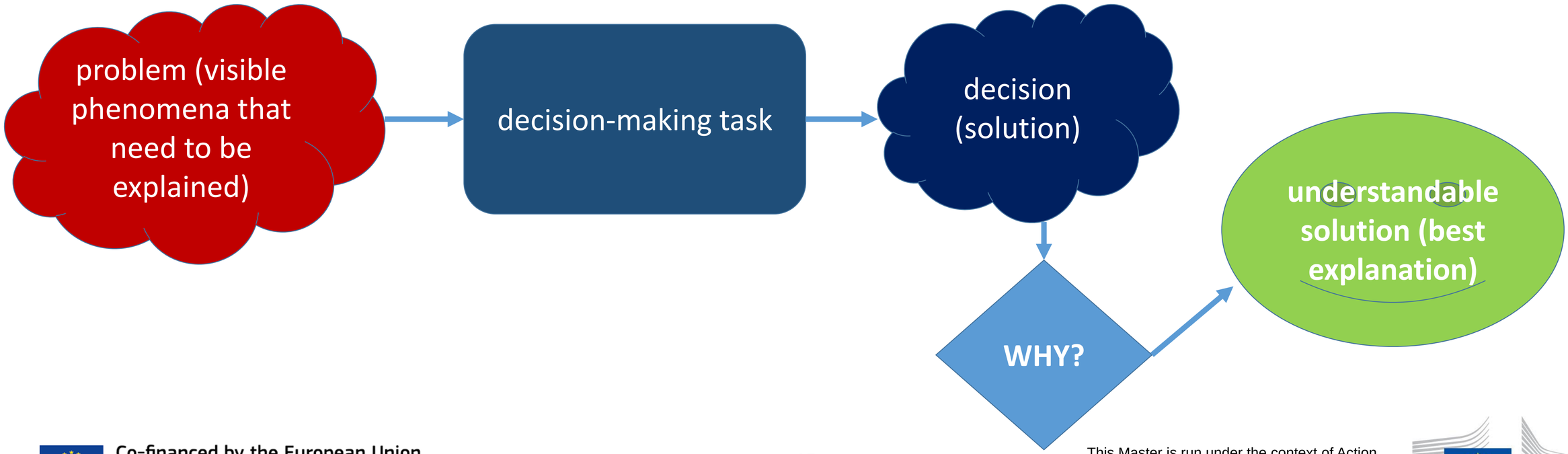
First published Wed Mar 9, 2011; substantive revision Tue May 18, 2021

In the philosophical literature, the term "abduction" is used in two related but different senses. In both senses, the term refers to some form of explanatory reasoning. However, in the historically first sense, it refers to the place of explanatory reasoning in *generating* hypotheses, while in the sense in which it is used most frequently in the modern literature it refers to the place of explanatory reasoning in *justifying* hypotheses. In the latter sense, abduction is also often called "Inference to the Best Explanation."

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Relevance to AI decision-making tasks

- Any decision-making task strives to reach a decision that constitutes the best solution and hence **best explanation** of the problem at hand, whether it refers to a classification, prediction, plan of action, etc.
- Explanations are essential when decisions are critical, unclear or not easily understandable.



This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The basic reasoning methods

- **Abduction**: Formulates hypotheses, making a combined space to create new ideas
- **Deduction**: Tests hypotheses to narrow down existing choices
- Recall **hypothetico-deductive model** of reasoning leading to best explanation, where deduction is a sub-process of abduction: Contextualized versus unconstrained deductions
- **Induction**: Reaches conclusions and generalizes existing ideas
- **Explanations arise as rational connections between hypotheses and observations**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

H.E. People's Mechanization of Abductive Logic

https://www.ijcai.org/Proceedings/73/Papers/017.pdf

- In a **deduction**, the objective is to determine **whether** some statement is true
- In an **abduction**, the objective is to determine **why** something is true (i.e., why the observed abnormalities hold)
- In answering the why question, it is obviously important to be able to determine whether, thus deduction may be considered a process subordinate to deduction

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Abduction and Deduction

- Abduction is far more complicated than deduction.
- A queried statement may be deduced (derived) in a multitude of ways, and any of these suffices; effective deductive systems are able to follow the simplest derivation paths, but this is an implementation rather than a conceptual issue.
- In abduction, it is not sufficient just to generate one **plausible explanation** of the observed situation; instead, all plausible explanations need to be compared and contrasted.
- **An explanation is usually not deducible**, and so once an explanation is hypothesized, it is not possible to deduce it.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

(i) What are the plausible explanations and (ii) How is the best explanation selected?

- Peirce has not specified any criteria …
- A trend in abductive diagnosis has been to explore how much can be achieved with somewhat restrictive and thus nonpragmatic criteria
- **Explanation plausibility**: complete accounting (coverage) of all observations of abnormality irrespective of their relative importance, say, for therapy
- Two celebrated theories of **abductive diagnosis** are based on this restricted notion of explanation plausibility:
  - Peng and Reggia's parsimonious covering theory
  - Poole's logic-based theory
  - The principle used to select the best explanation from the plausible ones is that of **simplicity**

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Peng and Reggia's parsimonious covering theory

Parsimonious criteria based on:

- **Relevancy** – every disorder hypothesis included in an explanation is **causally related** to some observation of abnormality
- **Irredundancy** – none of the proper subsets of an explanation is itself an explanation
- **Minimality** – prefer the explanation with the minimum cardinality

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Pool's logic-based theory

Criteria based on:

- **Minimality** – prefer the explanation that makes the fewest, in terms of set inclusion, assumptions
- **Least presumption** – prefer the explanation that makes the fewest, in terms of what can be implied, assumptions
- **Minimal abnormality** – prefer the explanation that makes the fewest failure assumptions or makes the same abnormality assumptions but fewer normality assumptions

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

P. Thagard's theory of explanatory coherence

http://cogsci.uwaterloo.ca/Articles/1989.explanatory.pdf

BEHAVIORAL AND BRAIN SCIENCES (1989) 12, 435–502
Printed in the United States of America

Explanatory coherence

Paul Thagard
Cognitive Science Laboratory, Princeton University, 221 Nassau St., Princeton, NJ 08540
Electronic mail: *pault@confidence.princeton.edu*

Abstract: This target article presents a new computational theory of explanatory coherence that applies to the acceptance and rejection of scientific hypotheses as well as to reasoning in everyday life. The theory consists of seven principles that establish relations of local coherence between a hypothesis and other propositions. A hypothesis coheres with propositions that it explains, or that explain it, or that participate with it in explaining other propositions, or that offer analogous explanations. Propositions are incoherent with each other if they are contradictory. Propositions that describe the results of observation have a degree of acceptability on their own. An explanatory hypothesis is accepted if it coheres better overall than its competitors. The power of the seven principles is shown by their implementation in a connectionist program called ECHO, which treats hypothesis evaluation as a constraint satisfaction problem. Inputs about the explanatory relations are used to create a network of units representing propositions, while coherence and incoherence relations are encoded by excitatory and inhibitory links. ECHO provides an algorithm for smoothly integrating theory evaluation based on considerations of explanatory breadth, simplicity, and analogy. It has been applied to such important scientific cases as Lavoisier's argument for oxygen against the phlogiston theory and Darwin's argument for evolution against creationism, and also to cases of legal reasoning. The theory of explanatory coherence has implications for artificial intelligence, psychology, and philosophy.

Keywords: artificial intelligence; attribution theory; coherence, connectionism; epistemology; explanation; legal reasoning; scientific reasoning; theory evaluation

Co-financed by the European Union
Connecting Europe Facility

P. Thagard's theory of explanatory coherence

Thagard is quite emphatic about the need for a **tight coupling between the formation and evaluation of hypotheses** in computational, abductive systems.

More specifically, he says that there are three possible models:

1. the two processes are completely independent, and hypotheses are formed in a random fashion, a nonviable option under limited resources;
2. the processes are weakly related, and only hypotheses that explain at least something are formed, or
3. they are strongly related, and only hypotheses that constitute likely possibilities are formed.

He also points out the inability (of some AI systems) to recognize those **observations in need of explanation** (this is a limitation because not every observation demands explanation) and subsequently the need to identify **how evaluation constraints can be used more effectively** to help limit the range of hypotheses that can be generated in order to lead to ones more likely to be accepted.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Thagard's general criteria for measuring the quality of explanatory hypotheses

- **Consilience** which is concerned not only with how much a hypothesis explains but also the variety of things it explains; a hypothesis is **dynamically consilient** if it becomes more credible over time
- **Simplicity** which is concerned with the number of supporting assumptions, the well-known Occam's razor: What can be done with fewer assumptions is done in vain with more
- **Analogy** which advocates the reusability of successful explanation models in analogous situations
- The above notions have been incorporated in the theory of explanatory coherence.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU | Master programmes in Artificial Intelligence 4 Careers in Europe

Abductive Diagnosis using Time-Objects: criteria for the evaluation of solutions

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1069.4339&rep=rep1&type=pdf

Computational Intelligence, Volume 17, Number 1, 2001

ABDUCTIVE DIAGNOSIS USING TIME-OBJECTS: CRITERIA FOR THE EVALUATION OF SOLUTIONS

ELPIDA T. KERAVNOU
Department of Computer Science, University of Cyprus

JOHN WASHBROOK
Department of Computer Science, University College London

Diagnostic problem solving aims to account for, or explain, a malfunction of a system (human or other). Any plausible potential diagnostic solution must satisfy some minimum criteria relevant to the application. Often there will be several plausible solutions, and further criteria will be required to select the "best" explanation. Expert diagnosticians may employ different, complex criteria at different stages of their reasoning. These criteria may be combinations of some more primitive criteria, which therefore should be represented separately and explicitly to permit their flexible and transparent combined usage.

In diagnostic reasoning there is a tight coupling between the formation of potential solutions and their evaluation. This is the essence of abductive reasoning. This article presents an abductive framework for diagnostic problem solving. *Time-objects*, an association of a property and an existence, are used as the representation formalism and a number of primitive, general evaluation criteria into which time has been integrated are defined. Each criterion provides an intuitive yardstick for evaluating the space of potential solutions. The criteria can be combined as appropriate for particular applications to define plausible and best explanations.

The central principle is that when time is diagnostically significant, it should be modeled explicitly to enable a more accurate formulation and evaluation of diagnostic solutions. The integration of time and primitive evaluation criteria is illustrated through the Skeletal Dysplasias Diagnostician (SDD) system, a diagnostic expert system for a real-life medical domain. SDD's notions of plausible and best explanation are reviewed so as to show the difficulties in formalizing such notions. Although we illustrate our work by medical problems, it has been motivated by consideration of problems in a number of other domains (fermentation monitoring, air and ground traffic control, power distribution) and is intended to be of wide applicability.

Key words: diagnostic problem solving, temporal abductive diagnosis, diagnostic solution, time-object, evaluation criteria.

Primitive Evaluation Criteria

- **Coverage**: focus-coverage, hard-coverage, current-coverage
- **Consistency**: case-consistent
- **Strength of integration**: strongly-integrated (or coherent), loosely-integrated (or incoherent); single or multiple point of failure
- **Satisfiability**: N/T/C-satisfiable (necessary, typical, common expectations)
- **Ambiguity**: alternative explanations for focus-abnormalities
- **Redundancy**: a strict subset has the same coverage
- **Minimality**: not redundant
- **Optimality**: has focus-coverage, and it is case-consistent, satisfiable, strongly-integrated and minimal

Causality

The notion of causality is strongly coupled to the ‘quality’ of explanations:

- Theories of explanation implicitly or explicitly entail causality, e.g., Peng and Reggia’s relevancy criterion states that every disorder hypothesis included in an explanation is **causally related** to some observation of abnormality
- **Causal models** are deeper than associational (rule-based) models and can provide justifications to associationally derived solutions; recall the case of NEOMYCIN and many other second-generation knowledge-based systems
- Association can arise between variables having causation or those not having causation; hence causality implies association but not the opposite

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Causal Explanation

- Is the strictest form of explanation
- It arises from the construction of causal models, which require that explanations for arising predictions are, in fact, “recipes” for reconstructing that prediction
- A causal model captures directed causal relationships, usually in a graphical representation:
  - Either forwards in time (A causes B) or backwards in time (B caused-by A)
  - Richness of temporal and other semantics varies in different models

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Judea Pearl, the father of Bayesian networks and probabilistic reasoning states …

“To build truly intelligent machines, teach them cause and effect”

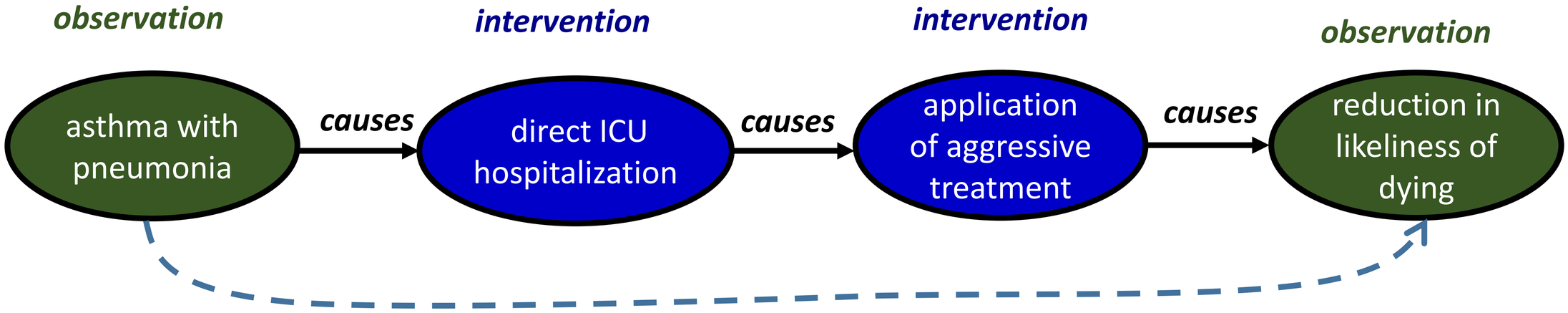
This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Paradoxical association due to lack of causal knowledge …

Asthmatics are less likely to die from pneumonia!

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

AI needs more WHY



associated-with – hence do not recommend asthmatics to the ICU!

Pearl's three levels of causality:

1. **Association:** invokes purely statistical relationships defined by the data – *What does a symptom tell me about a disease?*
2. **Intervention:** not just observing what is, but changing what one sees, e.g., reliably estimating the effect if one performs an action – *If I take a baby Aspirin, will my risk of heart failure reduce?*
3. **Counterfactuals:** reasoning about hypothetical situations which enables us to estimate the *unobserved* outcomes (this is *abduction*) – *Was the Aspirin that saved me from a heart attack?*

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

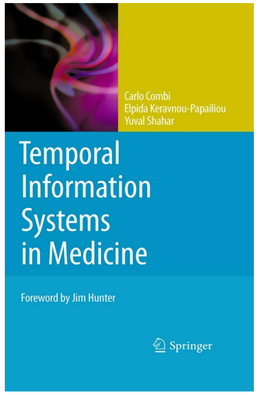
A. Lavin, "AI needs more why", Forbes, 2019.

https://www.forbes.com/sites/alexanderlavin/2019/05/06/ai-needs-more-why/#70ea2a5f156d

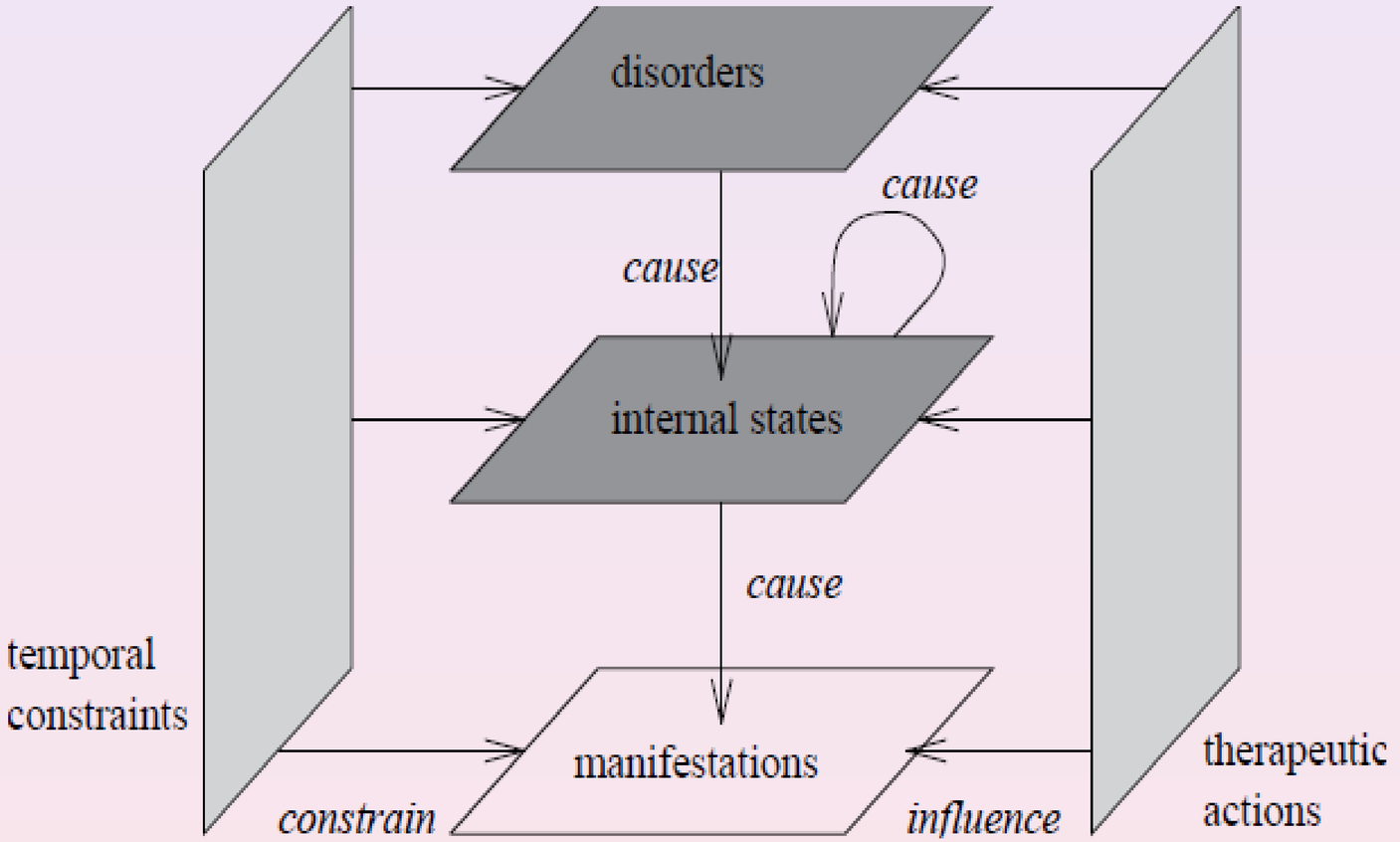
- The pneumonia example shows that without considering clinical **contexts**, counterintuitive predictions and models with unintended consequences can be derived.
- By taking into consideration **domain expertise of the hospital's policy**, level 2 causal structure (clinical context, i.e., interventions) can be added.
- The incorporated knowledge in the form of causal graph depicts which associations in the observed data are assumed to be **valid cause-effect relationships**.
- However, this is not enough since relationships caused by action policies, won't necessarily generalize when the policy changes.
- Reliable decision support models need to learn counterfactual objectives; for levels 2 and 3 Pearl proposes the use of do-calculus, a formalism for causal logic.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The Causal-Temporal-Action (C-T-A) Model



Causal inference can be leveraged to reason explicitly about actions-and-effects underlying observational data.



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Challenge: infer causality from purely observational data

Pearl's stance: "Causal reasoning is an indispensable component of human thought that should be formalized and algorithimitized toward achieving human-level machine intelligence."

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Tracing the history of explanations in symbolic AI

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Kim et. al., A multi-component framework for the analysis and design of explainable AI (https://www.mdpi.com/2504-4990/3/4/45)

"Explanations have always been an indispensable component of decision making, learning, understanding, and communication in the **human-in-the-loop** environments. After the emergence and rapid growth of artificial intelligence as a science in the 1950s, an interest in interpreting underlying decisions of intelligent systems also proliferated."

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Rule-based expert systems championed explanations in symbolic AI

- The MYCIN system pioneered symbolic explanations for different purposes:
  - **Justification** of the system's recommendations (MYCIN – 'intelligent' consultant)
  - Knowledge-base **debugging** (TEIRESIAS – 'intelligent' debugger)
  - **Tutoring** medical students (GUIDON – 'intelligent' tutor)

- Revealing **chains of rules** in the derived inference trees, also giving unsuccessful rules; pseudo natural language presentation
- Presenting the **current confidence** in (context-attribute-value) derivations stored in the context tree (working memory)
- Presentation and **comparative analysis of full inference trees** and other explanatory features of TEIRESIAS
- **Canned text** for individual rules; presentation of rules independently of specific consultations

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Despite their pioneering significance, problems soon surfaced with rule-based explanations, deeming them largely inadequate ….

- ❑ “Explanations” were **just rule playbacks** and not meaningful
  - Missing/implicit knowledge
  - No support (causality) or strategic knowledge
- ❑ User-tailored explanations subsequently added through a **rudimentary user model**
  - Complexity, importance of concepts and rule associations
  - User level of knowledge/detail of explanations

- ❑ Adequate explanations to be an **inborne feature of the design** of a knowledge-based system from the start and not a subsequent add-on or reengineered into the system
  - Differentiating, explicating and implementing relevant knowledge types (e.g., causality)
  - Modelling human expertise (factual and reasoning knowledge) – **bottleneck**!

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Second-generation, deep knowledge-based systems, offered new, promising avenues towards more adequate symbolic explanations …

- NEOMYCIN explicated important knowledge types utilized in explanations
  - Support knowledge in the form of a **causal model**, having a dual purpose
    - As an alternative means to solving problems
    - For augmenting rule-based explanations with a more detailed/deep justification
  - Strategic knowledge, enabling the provision of **strategic explanations**

- GUIDON2 was more successful than GUIDON as an ‘intelligent’ tutoring system

- Still **many challenges remained**, e.g.,
  - Explaining the rational basis of strategies
  - Revoking choices (including strategic choices) and/or derivations and explaining these
    - Inadequacies with reasoning, truth maintenance, non-monotonicity
  - Handling and justifying exceptions
  - User tailoring

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Case-based explanations

- **CBR** offered yet another paradigm to symbolic explanations
- **Contextualized**, evidence-based explanations
- The similarity between the current case and the retrieved/selected past case needs to be explained
- Where solution adaptation is made, this would also need to be explained
- In many domains the case-based element is the domineering element in decision making, e.g., legal system in Cyprus
  - A past case sets **precedence** for future similar cases – fair/consistent handling
  - A decision is justified based on past cases – transparency, trust
- Repeating a successful past solution for a new similar case is sufficient explanation on its own without requiring further justification – **It worked for a similar case in the past!**
- Listing unsuccessful past cases, similar to the new case, provides further explanation/ justification for not adopting their (erroneous) solution and opting for something different
  - Avoiding past mistakes and reinforcing successes – **learning** from them

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

The resurgence of interest in explanation in connectionist AI

– opening the ‘black box’ and the creation of the acronym XAI (eXplainable AI)

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The knowledge acquisition bottleneck of symbolic AI systems, coupled with the performance success of Machine Learning and more recently Deep Neural Networks triggered interest in data-driven approaches.

But a new challenge emerged … making the resulting, highly-performing "black boxes", interpretable and explainable!

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Not all ML approaches result in ‘black boxes’; e.g., decision trees are not, and symbolic rules can result from each branch from root to leaf of such trees; hence a decision tree can be flattened into a set of if-then rules.

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

Some survey papers on the topic of eXplainable AI (XAI) ….

- M-Y Kim et. al., A multi-component framework for the analysis and design of explainable AI (https://www.mdpi.com/2504-4990/3/4/45)
- G. Vilone and L. Longo, Classification of explainable AI methods through their output formats (https://www.mdpi.com/2504-4990/3/3/32)
- A.B. Arrieta et. al., Explainable AI (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI (https://www.sciencedirect.com/science/article/abs/pii/S1566253519308103)
- R. Guidotti et. al., A survey of methods for explaining black box models (https://www.researchgate.net/publication/322976218_A_Survey_of_Methods_for_Explaining_Black_Box_Models)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The opening remarks of these surveys ….

- M-Y Kim et. al.: “The rapid growth of research in explainable artificial intelligence (XAI) follows on two substantial developments. First, the enormous application success of modern machine learning methods, especially deep and reinforcement learning, having created high expectations for industrial, commercial, and social value. Second, the emerging and growing concern for creating ethical and trusted AI systems, including compliance with regulatory principles to ensure transparency and trust.”

- G. Vilone and L. Longo: “Machine and deep learning have proven their utility to generate data-driven models with high accuracy and precision. However, their non-linear, complex structures are often difficult to interpret. Consequently, many scholars have developed a plethora of methods to explain their functioning and the logic of their inferences.”

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

The opening remarks of these surveys ….

- A.B. Arrieta et. al.: “In the last few years, AI has achieved a notable momentum that, if harnessed appropriately, may deliver the best of expectations over many application sectors across the field. For this to occur shortly in Machine Learning, the entire community stands in front of the barrier of explainability, an inherent problem of the latest techniques brought by sub-symbolism (e.g., ensembles or Deep Neural Networks) that were not present in the last hype of AI (namely, expert systems and rule-based models).”

- R. Guidotti et. al.: “In the last years many accurate decision support systems have been constructed as black boxes, that is as systems that hide their internal logic to the user. This lack of explanation constitutes both a practical and an ethical issue.”

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Opening the Black Box

- Explaining the black box model
- Explaining the outcome
- Inspecting the black box internally
- Providing a transparent solution

Source: R. Guidotti et. al., A survey of methods for explaining black box models (https://www.researchgate.net/publication/322976218_A_Survey_of_Methods_for_Explaining_Black_Box_Models)

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Black box and comprehensible predictors

- A **black box** predictor ***b*** belongs to the set of **uninterpretable** data mining and machine learning models
  - The reasoning behind the function is not understandable by humans and the outcome returned does not provide any clue for its choice
  - In real-world applications, ***b*** is an opaque classifier
- A **comprehensible** predictor is one for which a global or a local explanation is available; its performance is generally evaluated by two measures:
  - **Accuracy**: comparing the real target values against the respective predicted target values of the black box and comprehensible predictors
  - **Fidelity**: how good is the comprehensible predictor in mimicking the black box predictor

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Explaining the black box model

Given a black box predictor ***b*** and a dataset ***D = {X, Y},*** the **black box explanation problem** consists in finding a function ***f*** which takes as input a black box ***b*** and a dataset ***D***, and returns a comprehensible global predictor c_g, i.e., $f(b, D) = c_g$, such that c_g is able to mimic the behavior of ***b*** and exists a global explanator function that can derive from c_g a set of explanations modeling in a human understandable way the logic behind c_g .

For example, the set of explanations can be modelled by a decision tree or by a set of rules.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Explaining the outcome

Given a black box predictor ***b*** and a dataset ***D = {X, Y},*** the **black box outcome explanation problem** consists in finding a function ***f*** which takes as input a black box ***b*** and a dataset ***D***, and returns a comprehensible local predictor c_l, i.e., $f(b, D) = c_l$, such that c_l is able to mimic the behavior of ***b*** and exists a local explanator function that takes as input the black box ***b***, the comprehensible local predictor c_l and a data record ***x***, and returns a human understandable explanation for the record ***x***.

The various approaches proposed to implement function ***f***, aim to overcome the limitations of explaining the whole model. The returned explanation may be either a path of a decision tree or an association rule.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Inspecting the black box internally

Given a black box predictor ***b*** and a dataset ***D = {X, Y},*** the **black box inspection problem** consists in finding a function ***f*** which takes as input a black box ***b*** and a dataset ***D*** and returns a visual (or textual) representation of the behavior of the black box, i.e., ***f(b, D) = v.***

For example, the visualization returned highlights the feature importance for the predictions. Overall, the aim is either to understand how the black box model works or why the black box returns certain predictions more likely than others.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Providing a transparent solution

Given a dataset ***D = {X, Y},*** the **transparent box design problem** consists in finding a learning function L_c which takes as input the dataset ***D*** and returns a (locally or globally) comprehensible predictor ***c***, i.e., $L_c(D) = c$.

This implies that there exists a local or a global explanator function that takes as input the comprehensible predictor ***c*** and returns a human understandable explanation or explanations.

For example, L_c and ***c*** may be the decision tree learner and predictor respectively, while the global explanator may return the choices taken along the various branches of the tree and the local explanator may return the textual representation of the path followed according to the (particular) decision suggested by the predictor.

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Agnostic Explanator

- Is a comprehensible predictor, not tied to a particular type of black box, explanation or data type.
- In theory it can explain indifferently a neural network or a tree ensemble using a single tree or a set of rules.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Reverse Engineering

- A common approach to solve the black box model and outcome explanation problems and the black box inspection problem
- The black box predictor ***b*** is queried with a certain test dataset in order to create an oracle dataset that in turn is used to train the comprehensible predictor
- The name **reverse engineering** comes from the fact that only the input and output of the black box can be observed.

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

Summary

- The significance of explanation in AI
- Abduction and Explanatory Coherence
- Causality
- Revisiting explanations in symbolic AI
- Opening the ‘black box’ in connectionist AI

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423