MAI4CAREU
Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI649: PRINCIPLES OF ONTOLOGICAL DATABASES

Adding Recursion - Datalog

Andreas Pieris

Spring 2022-2023

Co-financed by the European Union
Connecting Europe Facility

This Master is run under 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 Outcomes

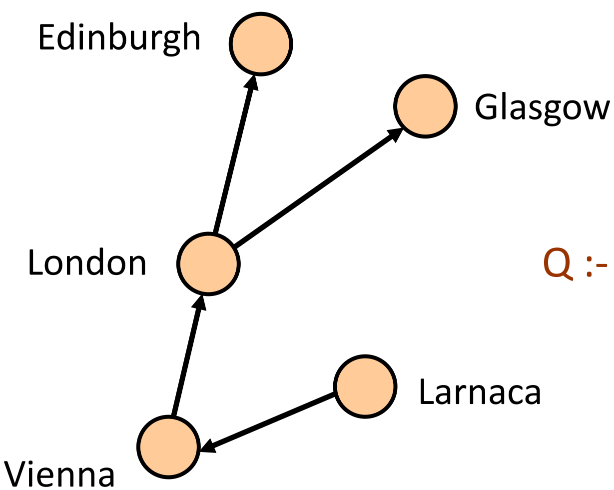
- Syntax and semantics of Datalog (CQs + recursion)
- Analyze the complexity of evaluating Datalog queries
- Static analysis of Datalog queries

Limits of CQs

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline
	VIE	LHR	BA
	LHR	EDI	BA
	LGW	GLA	U2
	LCA	VIE	OS

Airport	code	city
	VIE	Vienna
	LHR	London
	LGW	London
	LCA	Larnaca
	GLA	Glasgow
	EDI	Edinburgh



Q :- Airport(x,Vienna), Airport(y,Glasgow), Flight(x,z,w), Flight(z,y,v)

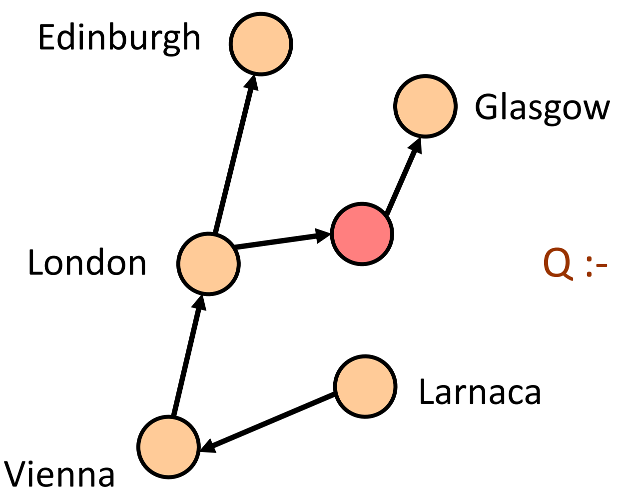
YES

Limits of CQs

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline
	VIE	LHR	BA
	LHR	EDI	BA
	LGW	GLA	U2
	LCA	VIE	OS

Airport	code	city
	VIE	Vienna
	LHR	London
	LGW	London
	LCA	Larnaca
	GLA	Glasgow
	EDI	Edinburgh



Q :- Airport(x,Vienna), Airport(y,Glasgow), Flight(x,z,w), Flight(z,y,v)

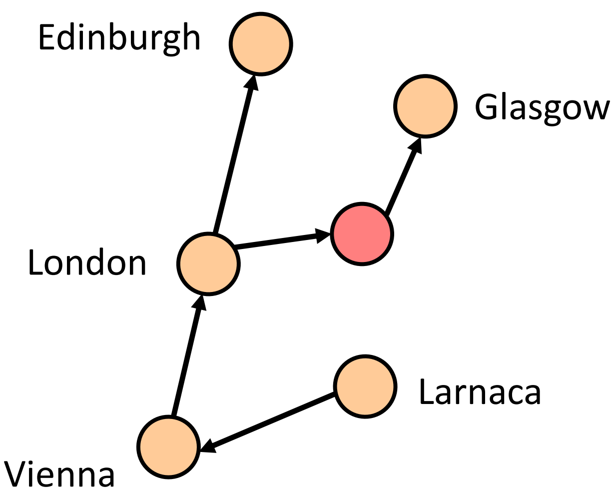
NO

Limits of CQs

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline
	VIE	LHR	BA
	LHR	EDI	BA
	LGW	GLA	U2
	LCA	VIE	OS

Airport	code	city
	VIE	Vienna
	LHR	London
	LGW	London
	LCA	Larnaca
	GLA	Glasgow
	EDI	Edinburgh



Q :- Airport(x,Vienna), Airport(y,Glasgow), Flight(x,z,w),

Flight(z,z_1,w_1), Flight(z,y,v)

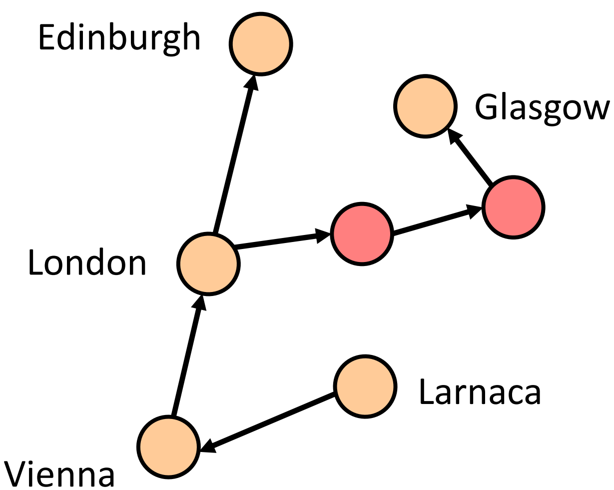
YES

Limits of CQs

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline
	VIE	LHR	BA
	LHR	EDI	BA
	LGW	GLA	U2
	LCA	VIE	OS

Airport	code	city
	VIE	Vienna
	LHR	London
	LGW	London
	LCA	Larnaca
	GLA	Glasgow
	EDI	Edinburgh



Q :- Airport(x,Vienna), Airport(y,Glasgow), Flight(x,z,w), Flight(z,z_1,w_1), Flight(z,y,v)

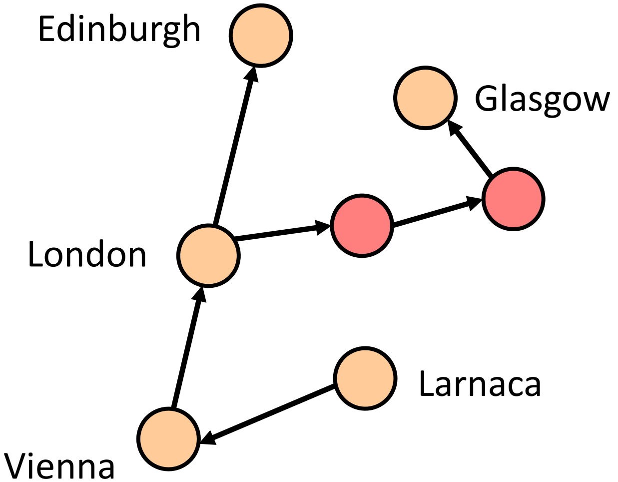
NO

Limits of CQs

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline
	VIE	LHR	BA
	LHR	EDI	BA
	LGW	GLA	U2
	LCA	VIE	OS

Airport	code	city
	VIE	Vienna
	LHR	London
	LGW	London
	LCA	Larnaca
	GLA	Glasgow
	EDI	Edinburgh



Recursive query - not expressible in **CQ**

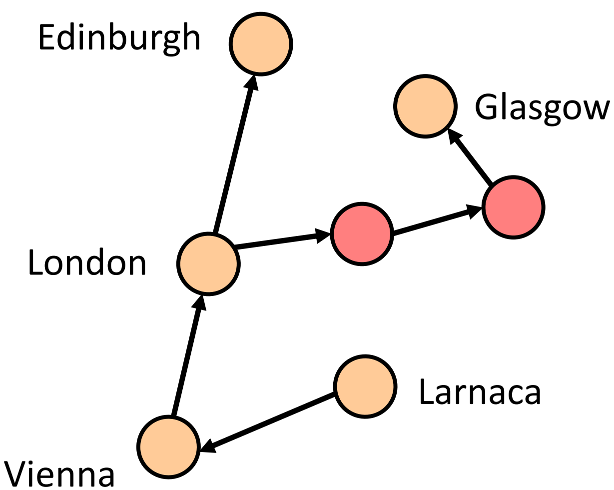
(or even in **RA** and **RC**)

A Possible Strategy

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline
	VIE	LHR	BA
	LHR	EDI	BA
	LGW	GLA	U2
	LCA	VIE	OS

Airport	code	city
	VIE	Vienna
	LHR	London
	LGW	London
	LCA	Larnaca
	GLA	Glasgow
	EDI	Edinburgh



- List all the pairs (a,b) such that b is reachable from a
- Check if there exists a pair (a,b) such that a is in Vienna and b is in Glasgow

A Possible Strategy

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline

Airport	code	city

- List all the pairs (a,b) such that b is reachable from a

Reachable(x,y) :- Flight(x,y,z)

Reachable(x,w) :- Flight(x,y,z), Reachable(y,w)

- Check if there exists a pair (a,b) such that a is in Vienna and b is in Glasgow

Answer() :- Airport(x,Vienna), Airport(y,Glasgow), Reachable(x,y)

A Possible Strategy

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline

Airport	code	city

- List all the pairs (a,b) such that b is reachable from a

Reachable(x,y) :- Flight(x,y,z)

Reachable(x,w) :- Flight(x,y,z), Reachable(y,w) - recursion

- Check if there exists a pair (a,b) such that a is in Vienna and b is in Glasgow

Answer() :- Airport(x,Vienna), Airport(y,Glasgow), Reachable(x,y)

A Possible Strategy

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline

Airport	code	city

- List all the pairs (a,b) such that b is reachable from a

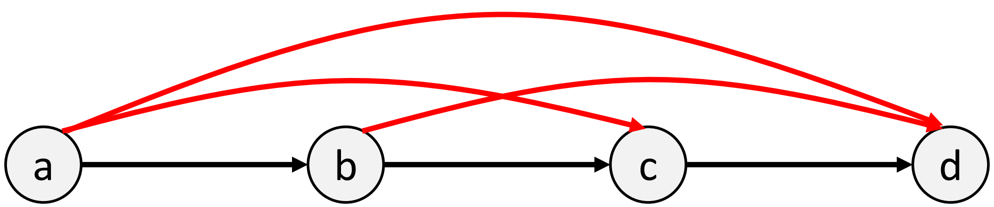
Reachable(x,y) :- Flight(x,y,z)

Reachable(x,w) :- Flight(x,y,z), Reachable(y,w) - recursion

DATALOG

Datalog at First Glance

Transitive closure of a graph



Datalog at First Glance

Edge	start	end
	a	b
	b	c
	c	d

TrClosure(x,y) :- Edge(x,y)

TrClosure(x,y) :- Edge(x,z), TrClosure(z,y)

Answer(x,y) :- TrClosure(x,y)

Answer	start	end
	a	b
	a	c
	a	d
	b	c
	b	d
	c	d

Datalog at First Glance

- **Semantics:** a mapping from databases of the **extensional** schema to databases of the **intensional** schema, and the answer is determined by the output relation

Edge	start	end
	a	b
	b	c
	c	d

Answer	start	end
	a	b
	a	c
	a	d
	b	c
	b	d
	c	d

- Equivalent ways for defining the semantics
  - **Model-theoretic:** logical sentences asserting a property of the result
  - **Fixpoint:** solution of a fixpoint procedure

Syntax of Datalog

A **Datalog rule** is an expression of the form

$$\underbrace{S(\mathbf{x})}_{\text{head}} \;\text{:-}\; \underbrace{R_1(\mathbf{x_1}),\ldots,R_n(\mathbf{x_n})}_{\text{body}}$$

- $n \geq 0$ (the body might be empty)
- $S, R_1, \ldots, R_n$ are relation names
- $\mathbf{x}, \mathbf{x_1}, \ldots, \mathbf{x_n}$ are tuples of variables
- each variable in the head occurs also in the body (**safety condition**)

Syntax of Datalog

- **Datalog program P:** a finite set of Datalog rules

- **Extensional relation:** does not occur in the head of a rule of P

- **Intensional relation:** occurs in the head of some rule of P

- EDB(P) is the set of extensional relations of P

- IDB(P) is the set of intensional relations of P

the **schema** of P

SCH(P) = EDB(P) ∪ IDB(P)

- **Datalog query Q:** a pair of the form (P, Answer), where P is a Datalog program, and Answer a distinguished intensional relation, the **output relation**

Example of Datalog

Is Glasgow reachable from Vienna?

Flight	origin	destination	airline

Airport	code	city

$$P = \left\{ \begin{array}{l} \text{Reachable(x,y) :- Flight(x,y,z)} \\ \text{Reachable(x,w) :- Flight(x,y,z), Reachable(y,w)} \\ \text{Answer() :- Airport(x,Vienna), Airport(y,Glasgow), Reachable(x,y)} \end{array} \right\}$$

EDB(P) = {Flight, Airport}   IDB(P) = {Reachable, Answer}

Q = (P, Answer)

Semantics of Datalog

…it relies on the notion of immediate consequence operator

- Given a database D and a Datalog program P, an atom $R(a_1,\ldots,a_n)$ is an **immediate consequence** for D and P if:
  - $R(a_1,\ldots,a_n)$ belongs to D, or
  - There exists a rule $R(x_1,\ldots,x_n)$:- body in P, and a homomorphism h from body to D such that $R(h(x_1),\ldots,h(x_n)) = R(a_1,\ldots,a_n)$

- $T_P(D) = \{R(a_1,\ldots,a_n) \mid R(a_1,\ldots,a_n)$ is an immediate consequence for D and P$\}$

- The immediate consequence operator T_P should be understood as a function from databases of SCH(P) to databases of SCH(P)

Semantics of Datalog

…it relies on the notion of immediate consequence operator

Theorem: For every Datalog program P and database D of EDB(P), the immediate consequence operator T_P has a minimum **fixpoint** containing D

a database D' is a fixpoint of T_P if $T_P(D') = D'$

the semantics of P on D, denoted P(D), is the minimum fixpoint of P containing D

for a Datalog query Q = (P, Answer), Q(D) = {t | Answer(t) ∈ P(D)}

…how do we compute P(D)?

Semantics of Datalog

...it relies on the notion of immediate consequence operator

$T_{P,0}(D) = D$ and $T_{P,i+1}(D) = T_P(T_{P,i}(D))$

$$\mathbf{T_{P,\infty}(D) = T_{P,0}(D) \cup T_{P,1}(D) \cup T_{P,2}(D) \cup T_{P,3}(D) \cup \cdots}$$

Semantics of Datalog: Example

…it relies on the notion of immediate consequence operator

$$D = \{Edge(a,b), Edge(b,c), Edge(c,d)\} \qquad P = \left\{ \begin{array}{l} TrClosure(x,y) \text{ :- } Edge(x,y) \\ TrClosure(x,y) \text{ :- } Edge(x,z), TrClosure(z,y) \\ Answer(x,y) \text{ :- } TrClosure(x,y) \end{array} \right\}$$

$$T_{P,0}(D) = D$$

$$T_{P,1}(D) = T_P(T_{P,0}(D)) = D \cup \{TrClosure(a,b), TrClosure(b,c), TrClosure(c,d)\}$$

$$T_{P,2}(D) = T_P(T_{P,1}(D)) = T_{P,1}(D) \cup \{TrClosure(a,c), TrClosure(b,d), Answer(a,b), Answer(b,c), Answer(c,d)\}$$

$$T_{P,3}(D) = T_P(T_{P,2}(D)) = T_{P,2}(D) \cup \{TrClosure(a,d), Answer(a,c), Answer(b,d)\}$$

$$T_{P,4}(D) = T_P(T_{P,3}(D)) = T_{P,3}(D) \cup \{Answer(a,d)\}$$

$$T_{P,5}(D) = T_P(T_{P,4}(D)) = T_{P,4}(D)$$

$$\boxed{T_{P,\infty}(D) = T_{P,4}(D)}$$

Semantics of Datalog

...it relies on the notion of immediate consequence operator

$T_{P,0}(D) = D$ and $T_{P,i+1}(D) = T_P(T_{P,i}(D))$

$$\mathbf{T_{P,\infty}(D) = T_{P,0}(D) \cup T_{P,1}(D) \cup T_{P,2}(D) \cup T_{P,3}(D) \cup \cdots}$$

Theorem: For every Datalog program P and database D of EDB(P), $P(D) = T_{P,\infty}(D)$

Complexity of **DATALOG**

QOT(**DATALOG**)

Input: a database D, a Datalog query Q/k, a tuple of constants **t** ∈ **adom**(D)^k

Question: **t** ∈ Q(D)? (i.e., whether Answer(**t**) ∈ P(D))

Theorem: It holds that:

- QOT(**DATALOG**) is EXPTIME-complete (combined complexity)
- QOT[Q](**DATALOG**) is PTIME-complete, for a fixed Datalog query Q (data complexity)

Complexity of **DATALOG**

- Recall that $P(D) = T_{P,\infty}(D)$
- Computing $T_{P,i}(D)$ takes time

$$O(|P| \cdot |\mathbf{adom}(D)|^{\text{maxvar}} \cdot \text{maxbody} \cdot |T_{P,i-1}(D)|)$$

where <u>maxvar</u> is the maximum number of variables in a rule-body, and <u>maxbody</u> is the maximum number of atoms in a rule-body

- It is clear that $|T_{P,i-1}(D)| \leq |T_{P,\infty}(D)|$, and thus, computing $T_{P,i}(D)$ takes time

$$O(|P| \cdot |\mathbf{adom}(D)|^{\text{maxvar}} \cdot \text{maxbody} \cdot |T_{P,\infty}(D)|)$$

- Consequently, computing $T_{P,\infty}(D)$ takes time

$$O(|P| \cdot |\mathbf{adom}(D)|^{\text{maxvar}} \cdot \text{maxbody} \cdot |T_{P,\infty}(D)|^2)$$

- It is not difficult to verify that

$$|T_{P,\infty}(D)| \leq |\text{SCH}(P)| \cdot |\mathbf{adom}(D)|^{\text{maxarity}}$$

where <u>maxarity</u> is the maximum arity over all relations of $\text{SCH}(P)$

- Consequently, $T_{P,\infty}(D)$ can be computed in time

$$O(|P| \cdot |\mathbf{adom}(D)|^{\text{maxvar}} \cdot \text{maxbody} \cdot |\text{SCH}(P)|^2 \cdot |\mathbf{adom}(D)|^{2\text{maxarity}})$$

Complexity of **DATALOG**

QOT(**DATALOG**)

Input: a database D, a Datalog query Q/k, a tuple of constants $\mathbf{t} \in \mathbf{adom}(D)^k$

Question: $\mathbf{t} \in Q(D)$? (i.e., whether $\text{Answer}(\mathbf{t}) \in P(D)$)

Theorem: It holds that:

- QOT(**DATALOG**) is EXPTIME-complete (combined complexity)
- QOT[Q](**DATALOG**) is PTIME-complete, for a fixed Datalog query Q (data complexity)

P(D) can be computed in time

$$O(|P| \cdot |\mathbf{adom}(D)|^{\text{maxvar}} \cdot \text{maxbody} \cdot |\text{SCH}(P)|^2 \cdot |\mathbf{adom}(D)|^{2\text{maxarity}})$$

What About Optimization of Datalog?

SAT(**DATALOG**)

Input: a query $Q \in$ **DATALOG**

Question: is there a (finite) database D such that $Q(D)$ is non-empty?

EQUIV(**DATALOG**)

Input: two queries $Q_1 \in$ **DATALOG** and $Q_2 \in$ **DATALOG**

Question: $Q_1 \equiv Q_2$? or $Q_1(D) = Q_2(D)$ for every database D?

CONT(**DATALOG**)

Input: two queries $Q_1 \in$ **DATALOG** and $Q_2 \in$ **DATALOG**

Question: $Q_1 \subseteq Q_2$? or $Q_1(D) \subseteq Q_2(D)$ for every database D?

What About Optimization of Datalog?

SAT(**DATALOG**)

Input: a query $Q \in$ **DATALOG**

Question: is there a (finite) database D such that $Q(D)$ is non-empty?

EQUIV(**DATALOG**)

Input: two queries $Q_1 \in$ **DATALOG** and $Q_2 \in$ **DATALOG**

Question: $Q_1 \equiv Q_2$? or $Q_1(D) = Q_2(D)$ for every database D?

UNDECIDABLE

CONT(**DATALOG**)

Input: two queries $Q_1 \in$ **DATALOG** and $Q_2 \in$ **DATALOG**

Question: $Q_1 \subseteq Q_2$? or $Q_1(D) \subseteq Q_2(D)$ for every database D?

What About Optimization of Datalog?

SAT(**DATALOG**)

Input: a query Q ∈ **DATALOG**

Question: is there a (finite) database D such that Q(D) is non-empty?

Theorem: SAT(**DATALOG**) is in EXPTIME

Lemma: Given a Datalog query Q = (P, Answer), Q is satisfiable iff $Q(D_P) \neq \emptyset$, where $D_P = \{R(b_1,\ldots,b_m) \mid R \in EDB(P) \text{ and } b_i \in \{\star,a_1,\ldots,a_n\}\}$, with $a_1,\ldots,a_n$ being the constants occurring in the rules of P, and $\star$ being a new constant not in $\{a_1,\ldots,a_n\}$

Recap

- Recursive queries are not expressible via relational algebra or calculus
- Adding recursion to CQs → Datalog
- Fixpoint semantics of Datalog based on the immediate consequence operator
- Evaluating Datalog queries is EXPTIME-complete in combined complexity and PTIME-complete in data complexity
- We can check for satisfiability of Datalog queries, but equivalence and containment are undecidable (perfect query optimization not possible)

MAI4CAREU | Master programmes in Artificial Intelligence 4 Careers in Europe

University of Cyprus

MAI649: PRINCIPLES OF ONTOLOGICAL DATABASES

Thank You!

Andreas Pieris

Spring 2022-2023

Co-financed by the European Union
Connecting Europe Facility

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