3 Graph Query Language Semantics

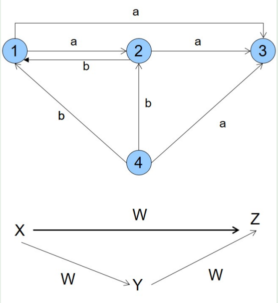
Graph Query Languages: SPARQL, Cypher, and Gremlin.

Most widely used query languages in practice but offer significant differences:

- SPARQL operates over RDF graphs, i.e. edge-labelled graphs;
- Cypher is designed to operate over property graphs;
- Gremlin is more imperative in nature than the other two, geared more towards graph traversal than graph pattern matching.

Our interest: semantic issues of graph pattern matching and graph traversal

Given an ELG G and query Q:



Two matches:

- $h_1 = \{X \to 1, Y \to 2, Z \to 3, W \to A\}$
- $h_2 = \{X \to 4, Y \to 2, Z \to 1, W \to B\}$

ELG representation by relation $edge(from, label, to)$

Graph G as instance of ternary relation $edge$:

from	*label*	*to*
1	*a*	2
2	*a*	3
1	*a*	3
4	*a*	3
4	*b*	2
2	*b*	1
4	*b*	1

Query Q as boolean *Conjunctive Query* (CQ):

$$\forall X, Y, Z, W\ (\ true \leftarrow edge(X, W, Y) \wedge edge(Y, W, Z) \wedge edge(X, W, Z)\ )$$

3.1 Conjunctive Queries

Definition

A *conjunctive Query* Q over a database schema $\mathcal{R}$ is given as

$$ans(\vec{U}) \leftarrow R_1(\vec{U_1}), \ldots, R_n(\vec{U_n}),$$

such that for $1 \leq i \leq n$

- R_i a relation name in $\mathcal{R}$ and
- $\vec{U}$ and $\vec{U_i}$ vectors of variables and constants;
- any variable appearing in $\vec{U}$ appears also in some $\vec{U_i}$.
- Left to $\leftarrow$ is the *head* of the query, and to the right there is the *body*. The atoms in the body are also called *subgoals*.

Example 1 - quantifiers and connectives are explicitly given

$$\forall X, Y, Z, W \; (\; ans(W) \leftarrow edge(X, W, Y) \wedge edge(Y, W, Z) \wedge edge(X, W, Z) \;)$$

Example 2 - quantifiers and connectives are implicitly given

Sales(Part, Supplier, Customer),
Part(PName, Type),
Cust(CName, CAddr),
Supp(SName, SAddr).

$$Q: \qquad ans(T) \leftarrow Sales(P, S, C), Part(P, T), Cust(C, A), Supp(S, A)$$

$$ans(\vec{U}) \leftarrow R_1(\vec{U_1}), \dots, R_n(\vec{U_n}).$$

Answer

- The set of answers Q w.r.t. an instance $\mathcal{I}$ of the given relations is denoted $Q(\mathcal{I})$.
- If there is a *substitution (match, mapping)* σ from the variables in $\vec{U_1}, \dots, \vec{U_n}$ to the constants in *dom*, such that $\sigma(R_1(\vec{U_1})), \dots, \sigma(R_n(\vec{U_n})) \in \mathcal{I}$, then by applying the same substitution σ to $\vec{U}$, we say that $\sigma(ans(\vec{U}))$ is an answer in $Q(\mathcal{I})$.
- Substitutions are functions - a constant is mapped into itself.

Problemes

Let Q, Q_1, Q_2 be conjunctive queries.

Containment: $Q_1 \sqsubseteq Q_2$, i.e., $Q_1(\mathcal{I}) \subseteq Q_2(\mathcal{I})$ for any instance $\mathcal{I}$?

Equivalence: $Q_1 \equiv Q_2$, i.e., $Q_1 \sqsubseteq Q_2$ and $Q_2 \sqsubseteq Q_1$?

Minimization: Given Q_1, construct an equivalent query Q_2, which has as most as many subgoals in its body as Q_1 and is minimal in the sense, that any query Q_3 being equivalent to Q_2 has at least as many subgoals in the body as Q_2.

Q_2 is called *minimal*.

Example 1

Relation *edge*:

from	*label*	*to*
1	*a*	2
2	*a*	3
1	*a*	3
4	*a*	3
4	*b*	2
2	*b*	1
4	*b*	1

Containment relationship?

$$Q: \quad ans(X, Z) \leftarrow edge(X, W, Y), edge(Y, W, Z), edge(X, W, Z)$$

$$Q': \quad ans(X, Z) \leftarrow edge(X, W, Y), edge(Y, W, Z), edge(X, W, Z), edge(X', W', Y), edge(Y, W', Z'), edge(X', W', Z')$$

$$Q'': \quad ans(X, X') \leftarrow edge(X, W, Y), edge(Y, W, Z), edge(X, W, Z), edge(X', W', Y), edge(Y, W', Z'), edge(X', W', Z')$$

Example 2

Relation *edge*:

from	*label*	*to*
1	*a*	2
2	*a*	3
1	*a*	3
4	*a*	3
4	*b*	2
2	*b*	1
4	*b*	1

Containment relationship?

$$Q: \quad ans(X, X) \leftarrow edge(X, W, Y), edge(Y, W, Z), edge(X, W, Z), edge(X', W', Y), edge(Y, W', Z'), edge(X', W', Z')$$

$$Q': \quad ans(X, X') \leftarrow edge(X, W, Y), edge(Y, W, Z), edge(X, W, Z), edge(X', W', Y), edge(Y, W', Z'), edge(X', W', Z')$$

Example 3

$$Sales(Part, Supplier, Customer),$$
$$Part(PName, Type),$$
$$Cust(CName, CAddr),$$
$$Supp(SName, SAddr).$$

Containment relationship?

$$Q: \quad ans(T) \leftarrow Sales(P, S, C), Part(P, T), Cust(C, A), Supp(S, A)$$

$$Q': \quad ans(T) \leftarrow Sales(P, S, C), Part(P, T), Cust(C, A), Supp(S, A), Sales(P', S', C'), Part(P', T)$$

$\Longrightarrow$ Minimization!

Lemma

Let

$$\begin{array}{ll} Q_1 : & ans(\vec{U}) \leftarrow R_1(\vec{U_1}), \ldots, R_n(\vec{U_n}) \\ Q_2 : & ans(\vec{U}) \leftarrow S_1(\vec{V_1}), \ldots, S_m(\vec{V_m}) \end{array}$$

be conjunctive queries, where

$$\{R_1(\vec{U_1}), \ldots, R_n(\vec{U_n})\} \supseteq \{S_1(\vec{V_1}), \ldots, S_m(\vec{V_m})\}$$

?

Then $Q_1 \sqsubseteq Q_2$.

?

Having more constraint, you will have less answers

Substitution

- A *substitution* θ over a set of variables $\mathcal{V}$ is a mapping from $\mathcal{V}$ to $\mathcal{V} \cup dom$, where dom a corresponding domain.
- We extend θ to constants $a \in dom$ and relation names $R \in \mathcal{R}$, where $\theta(a) = a$, resp. $\theta(R) = R$.

Note, differently to a *match*, variables may be renamed, i.e. mapped to variables.

Example

Consider

$$Q : \quad ans(T) \leftarrow Sales(P, S, C), Part(P, T), Cust(C, A), Supp(S, A)$$

$$Q' : \quad ans(T) \leftarrow Sales(P, S, C), Part(P, T), Cust(C, A), Supp(S, A), Sales(P', S', C'), Part(P', T)$$

and θ:

X	P	P'	S	S'	C	C'	T	A
$\theta(X)$	P	P	S	S	C	C	T	A

Containment Mapping (Homomorphism)

Given conjunctive queries

$$\begin{array}{ll} Q_1 : & ans(\vec{U}) \leftarrow R_1(\vec{U_1}), \ldots, R_n(\vec{U_n}) \\ Q_2 : & ans(\vec{V}) \leftarrow S_1(\vec{V_1}), \ldots, S_m(\vec{V_m}) \end{array}$$

Substitution θ is called *containment mapping* from Q_2 to Q_1, if Q_2 can be transformed by means of θ to become part of Q_1:

- $\theta(ans(\vec{V})) = ans(\vec{U})$,
- for $i = 1, \ldots, m$ there exists a $j \in \{1, \ldots, n\}$, such that $\theta(S_i(\vec{V_i})) = R_j(\vec{U_j})$.

Example

$$Q: \quad ans(T) \leftarrow Sales(P,S,C), Part(P,T), Cust(C,A), Supp(S,A)$$

$$Q': \quad ans(T) \leftarrow Sales(P,S,C), Part(P,T), Cust(C,A), Supp(S,A), Sales(P',S',C'), Part(P',T)$$

θ:

X	P	P'	S	S'	C	C'	T	A
$\theta(X)$	P	P	S	S	C	C	T	A

θ is a containment mapping.

Theorem

Let

$$
\begin{array}{ll}
Q_1: & ans(\vec{U}) \leftarrow R_1(\vec{U}_1), \ldots, R_n(\vec{U}_n) \\
Q_2: & ans(\vec{V}) \leftarrow S_1(\vec{V}_1), \ldots, S_m(\vec{V}_m)
\end{array}
$$

be conjunctive queries.

$Q_1 \sqsubseteq Q_2$ iff there exists a containment mapping θ from Q_2 to Q_1.

$$Q_1 : \quad ans(\vec{U}) \leftarrow R_1(\vec{U_1}), \dots, R_n(\vec{U_n})$$
$$Q_2 : \quad ans(\vec{V}) \leftarrow S_1(\vec{V_1}), \dots, S_m(\vec{V_m})$$
$$Q_1 \sqsubseteq Q_2?$$

Proof "$\Leftarrow$", i.e. there exists a containment mapping θ from Q_2 to Q_1.:

Let $\mathcal{I}$ be a database instance and let $\mu \in Q_1(\mathcal{I})$.

There exists a substitution τ, such that $\tau(\vec{U_j}) \in \mathcal{I}(R_j)$, $j \in \{1, \dots, n\}$ and $\mu = \tau(\vec{U})$.

Consider a substitution $\tau' = \tau \circ \theta$[1] and further $\tau'(S_i(\vec{V_i}))$, $i \in \{1, \dots, m\}$.

There holds $\tau'(\vec{V_i}) \in \mathcal{I}(S_i)$, $i \in \{1, \dots, m\}$ and therefore also $\mu = \tau'(\vec{V})$.

I.e., $\mu \in Q_2(\mathcal{I})$.

□

[1] $\tau'(\cdot) = \tau(\theta(\cdot))$

Proof "$\Rightarrow$" is based on a *canonical instance* of a query Q

Let Q be a conjunctive query $ans(\vec{U}) \leftarrow R_1(\vec{U_1}), \ldots, R_n(\vec{U_n})$ over a database schema $\mathcal{R}$.

The *canonical instance* $\mathcal{I}_Q$ to Q is an instance of $\mathcal{R} = \{R_1, \ldots, R_n\}$ constructed as follows.

Let τ be a substitution, which assigns to any X in Q an unique constant a_X.

- For any subgoal $R(t_1, \ldots, t_n)$ in the body of Q insert a tupel of the form $(\tau(t_1), \ldots, \tau(t_n))$ into $\mathcal{I}_Q(R)$; thus $\tau(R(t_1, \ldots, t_n)) \in \mathcal{I}_Q(R)$.

  No other tuples are inserted into $\mathcal{I}_Q(R)$.

τ is called *canonical substitution*.

Example

$$
\begin{aligned}
Q: \quad & ans(T) \leftarrow Sales(P,S,C), Part(P,T), Cust(C,A), Supp(S,A) \\
Q': \quad & ans(T) \leftarrow Sales(P,S,C), Part(P,T), Cust(C,A), Supp(S,A), \\
& \qquad Sales(P',S',C'), Part(P',T)
\end{aligned}
$$

$\mathcal{I}_Q$:

Substitut.

Sales		
a_P	a_S	a_C

Part	
a_P	a_T

Cust	
a_C	a_A

Supp	
a_S	a_A

$\mathcal{I}_{Q'}$:

Sales		
a_P	a_S	a_C
$a_{P'}$	$a_{S'}$	$a_{C'}$

Part	
a_P	a_T
$a_{P'}$	a_T

Cust	
a_C	a_A

Supp	
a_S	a_A

$$
\begin{aligned}
Q_1 : &\quad ans(\vec{U}) \leftarrow R_1(\vec{U_1}), \ldots, R_n(\vec{U_n}) \\
Q_2 : &\quad ans(\vec{V}) \leftarrow S_1(\vec{V_1}), \ldots, S_m(\vec{V_m})
\end{aligned}
$$

$$Q_1 \sqsubseteq Q_2?$$

Proof "$\Rightarrow$", i.e. we assume $Q_1 \sqsubseteq Q_2$.

Consider $\mathcal{I}_{Q_1}$ and the corresponding canonical substitution τ.

Then $\tau(ans(\vec{U})) \in Q_1(\mathcal{I}_{Q_1})$.

Because of $Q_1 \sqsubseteq Q_2$ further $\tau(ans(\vec{U})) \in Q_2(\mathcal{I}_{Q_1})$.

Thus, there exists a substitution ρ, such that $\rho(S_i(\vec{V_i})) = \tau(R_j(\vec{U_j}))$, $1 \leq i \leq m$, $j \in \{1, \ldots, n\}$ und $\rho(ans(\vec{V})) = \tau(ans(\vec{U}))$.

$\tau^{-1} \circ \rho$ is a containment mapping from Q_2 to Q_1

$\square$

Corollary

Let

$$\begin{array}{ll} Q_1: & ans(\vec{U}) \leftarrow R_1(\vec{U_1}), \ldots, R_n(\vec{U_n}) \\ Q_2: & ans(\vec{V}) \leftarrow S_1(\vec{V_1}), \ldots, S_m(\vec{V_m}) \end{array}$$

be conjunctive queries, $\mathcal{I}_{Q_1}$ the canonical instance to Q_1 with canonical substitution τ.

$$Q_1 \sqsubseteq Q_2, \text{ iff } \tau(ans(\vec{U})) \in Q_2(\mathcal{I}_{Q_1}).$$

Proof: It remains to show, whenever $\tau(ans(\vec{U})) \in Q_2(\mathcal{I}_{Q_1})$, then $Q_1 \sqsubseteq Q_2$.

For any S_j there exists a nonempty R_i, such that $S_j = R_i$.

Further, there exists a substitution ρ, such that for $S_j(\vec{V_j})$ we have $\rho(V_j) \in \mathcal{I}_{Q_1}(R_i)$.

$\tau^{-1} \circ \rho$ is a containment mapping from Q_2 to Q_1.

□

Example

$$ans(a_T) \in Q(\mathcal{I}_{Q'})$$

and

$$ans(a_T) \in Q'(\mathcal{I}_Q).$$

Theorem:

Query containment for conjunctive queries is NP-complete.

Query answering? Possible in polynomial time w.r.t. size of the database (ignoring size of the query).

3.2 Minimization of Conjunctive Queries

Problem

A query Q' is a *subquery* of a query Q, if the body of Q' is a subset of the body of Q.

Given Q_1, construct an equivalent query Q_2, which has as most as many subgoals in its body as Q_1 and is minimal in the sense, that any query Q_3 being equivalent to Q_2 has at least as many subgoals in the body as Q_2.

Can minimization be done by deleting subgoals from Q_1, i.e. the result Q_2 is a subquery of Q_1?

Example:

$$
\begin{aligned}
Q: \quad & ans(T) \leftarrow Sales(P,S,C), Part(P,T), Cust(C,A), Supp(S,A) \\
Q': \quad & ans(T) \leftarrow Sales(P,S,C), Part(P,T), Cust(C,A), Supp(S,A), \\
& \qquad Sales(P',S',C'), Part(P',T)
\end{aligned}
$$

Q is minimal and equivalent to Q'.

Theorem

Let $Q_1 : \; ans(\vec{U}) \leftarrow R_1(\vec{U_1}), \ldots, R_n(\vec{U_n})$ be a conjunctive query.
Then there exists a minimal conjunctive query Q_2 equivalent to Q_1,

$$Q_2 : \; ans(\vec{V}) \leftarrow S_1(\vec{V_1}), \ldots, S_m(\vec{V_m}),$$

such that $\{S_1(\vec{V_1}), \ldots, S_m(\vec{V_m})\} \subseteq \{R_1(\vec{U_1}), \ldots, R_n(\vec{U_n})\}$.

Proof

We can assume the existence of some conjunctive query Q_3 which is minimal and equivalent to Q_1.
Because of equivalence, there exits containment mappings θ from Q_1 to Q_3, and also λ from Q_3 to Q_1.
Let w.o.l.g. $\{S_1(\vec{V_1}), \ldots, S_m(\vec{V_m})\}$ be that subset of subgoals from Q_1, which are images with respect to λ and let Q_2 be a conjunctive query built out of these subgoals and no others.

(i) We have $Q_1 \sqsubseteq Q_2$ as Q_1 may have additional subgoals to the subgoals also being subgoals of Q_2.

(ii) $Q_2 \sqsubseteq Q_1$ as $\lambda \circ \theta$ is a containment mapping, i.e. each subgoal of Q_1 is guaranteed to be mapped on one subgoal of Q_2.

(iii) Minimality follows as, because of λ, Q_2 cannot have more subgoals than Q_3.

□

Query minimization is NP-hard.

We can compute all possible containment mappings over query Q and select one, whose image produces a minimal set of subgoals.

Algorithm *Conjunctive Query Minimization*

- Chose a subgoal from Q and remove it to obtain a new query Q'. We have $Q \sqsubseteq Q'$.
- Check if $Q' \sqsubseteq Q$; if so, then Q' is equivalent and we can continue the process of removing another subgoal.
- If not, try to remove another atom from Q.

Example:

$$Q: \qquad ans(X, Z) \leftarrow R(X, 5, Z_1), R(X_1, 5, Z_2), R(X_1, 5, Z)$$

Q can be minimized to Q'

$$Q': \qquad ans(X, Z) \leftarrow R(X, 5, Z_1), R(X_1, 5, Z)$$

However, not to $Q'' : ans(X, Z) \leftarrow R(X, 5, Z)$, as Q'' and Q, respectively Q'' and Q' are not equivalent.

X -> X
Z1 -> Z
X1 -> X
Z2 -> Z
Z -> Z