Advanced Data Modeling

6: Datalog

Steffen Staab
with
Simon Schenk

Fix a database schema (input schema) and a relation schema (output schema).

Query mapping: mapping from the set of instances of the input schema to the set of instances of the output schema.

Query: any expression whose semantics is a query mapping.

Query language: any collection of queries.

1. **Expressive power**: the set of query mappings expressible in a language.

2. **Complexity**: How difficult it is to answer queries in a language.

3. **Succinctness**: How long is a query to express particular query mappings.

Query: expression built from input relation names using the relational algebra operations.

Let C be a set of clauses.

Its **dependency graph** consists of pairs $p \rightarrow r$ such that p occurs in the body of a clause which defines r in C.

A set of clauses is **non-recursive** if the dependency graph contains no cycles.

- Fix a set of input relation symbols, output relation symbol, and maybe some auxiliary relation symbols.
- Consider a non-recursive set of definitions of the auxiliary and output symbols.
- Use **completion** to get a set of completed definitions.
- It defines a **unique instance** of the output relation from instances of the input relations.

Let C be a set of clauses.

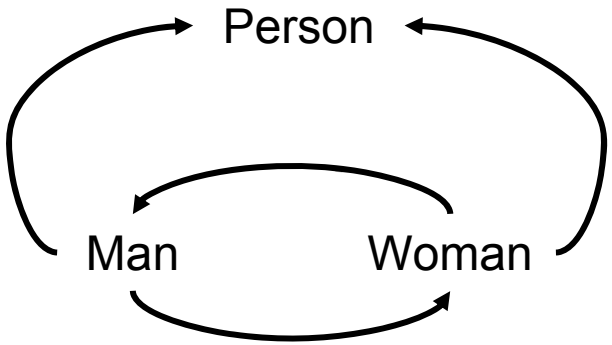
Its **dependency graph** consists of pairs $p \rightarrow r$ such that p occurs in the body of a clause which defines r in C.

A set of clauses is **non-recursive** if the dependency graph contains no cycles.

Person(a).

Woman(X) :- Person(X), Not Man(X).

Man(X) :- Person(X), Not Woman(X).



Query: formula built from input relation symbols. The arities of the formulas should respect the input database schema.

Theorem. First-order logics has a query language has the same expressive power as non-recursive definitions.

Problem: semantics is **unclear**, because it is language-dependent.

Range-restricted clause *C*: every variable of *C* occurs in a positive literal of the body of *C*.

Datalog clause: range-restricted clause without function symbols.

Relational algebra expression: built from input relation names using the relational algebra operations. The input database schema gives us the arity of each input relation.

More exactly, we define relational algebra expressions of arity *k*:

- Input relation *R*, where *R* is any relation name. The arity of this expression is equal to the arity of *R*.
- Constant $\{t_1, \ldots, t_n\}$, where each *t* is a tuple of arity *k*.

Let E_1, E_2 be expressions of arities k_1, k_2, respectively.

- if $k_1 = k_2$, then $R1 \cup R2$ and $R1 - R2$ are expressions of arity k_1.
- $E_1 \times E_2$ is an epression of arity $k_1 + k_2$.
- If $i_1, \ldots, i_m \leq k_1$, then $\pi_{i_1 \ldots i_m}(E_1)$ is an expression of arity m.
- If $i, j \leq k_1$, then $\sigma_{\#i=\#j}(E_1)$ and $\sigma_{\#i\neq\#j}(E_2)$ are expressions of arity k_1

Theorem. Non-recursive datalog is language-independent. That is, the semantics of a definition depends only on the input relation instances, but not on the language.

Theorem. Relational algebra has the same expressive power as non-recursive datalog.

Let $C = (A_1 \vee \ldots \vee A_m$:- $L_1, \ldots, L_n)$ be a clause.

Safe variables of C:

1. Each variable occurring in a positive literal L_i in the body of C whose predicate symbol is not =, is safe;
2. Each variable x occurring in a positive literal $x = c$ in the body of C, where c is a constant, is safe;
3. If one of the variables x, y is safe and $x = y$ is a positive literal in the body of C, then both variables x, y are safe.

A clause C is **range-restricted** if each variable occurring in C is safe in C.

Let C be a definition of r, I be a Herbrand model of the corresponding completed definition, and $r(t_1, \ldots, t_n)$ be a ground atom.

Then

$I \models r(t_1, \ldots, t_n) \Leftrightarrow$

$\exists (r(t_1, \ldots, t_n) :- L_1, \ldots, L_m) \in C^* (I \models L_1 \wedge \ldots \wedge L_n)$:

$T_C(I) :=$ { A | there exists (A :- G) $\in C^*$ such that $I \models G$ }

Fixpoint: an interpretation such that $T_C(I) = I$.

Let C be a set of definite clauses.

Define

$I_0 := \{\}$

$I_{n+1} := T_C(I_n)$, for all $n \geq 0$,

$I_\omega := \cup^\omega_{i=0} I_i$

Then I_ω is the least fixpoint of T_C and also the least Herbrand model of C.

Let C be a non-recursive database and K be an arbitrary interpretation.

Define

$I_0 := \mathbf{K}$

$I_{n+1} := T_C(I_n)$, for all $n \geq 0$,

$I\omega := \cup^{\omega}_{i=0} I_i$

Then I_ω is the only fixpoint of T_C. Moreover, for some n we have $I_\omega = I_n$.

- Dependency graph of C consists of pairs p $\rightarrow$ r such that p occurs in the body of a clause which defines r in C.
- C is non-recursive if the dependency graph contains no cycles.

- A relation p depends on a relation q in C if there exists a path of length ≥ 1 from q to p in the dependency graph of C.
- A set of clauses is non-recursive if and only if no relation depends on itself.

- Level mapping:
  mapping ` from a set of relation symbols to N.
- l(r) is called the level of r.

Theorem. Let C be a finite non-recursive set of clauses.
Then there exists a level mapping l such that for every clause c $\in$ C,

if *q* occurs in the body of c and c defines *r*,
then l(*r*) > *l*(*q*).