Datalog and data trees

Filip Mazowiecki Filip Murlak Adam Witkowski

University of Warsaw

Forum Informatyki Teoretycznej 2015

Datalog

A datalog program is a set of **rules**:

$$P(X) :- \downarrow(X, Y), b(X), Q(Y)$$
$$\underbrace{Q(X)}_{head} :- \underbrace{c(X)}_{body}$$

Datalog

A datalog program is a set of **rules**:

$$P(X) :- \downarrow(X, Y), b(X), Q(Y)$$
$$\underbrace{Q(X)}_{head} :- \underbrace{c(X)}_{body}$$

- **extensional** predicates ($\downarrow$, b, c);

Datalog

A datalog program is a set of **rules**:

$$P(X) :- \downarrow(X, Y), b(X), Q(Y)$$
$$\underbrace{Q(X)}_{head} :- \underbrace{c(X)}_{body}$$

- **extensional** predicates ($\downarrow$, b, c);
- **intensional** predicates (P, Q);

Datalog

A datalog program is a set of **rules**:

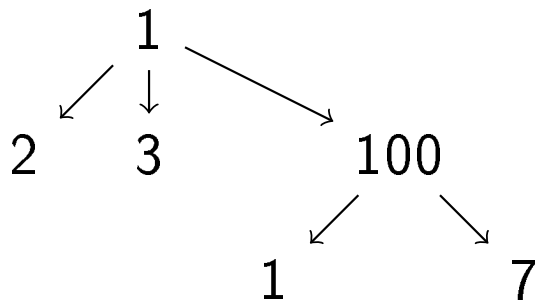
$$P(X) :- \downarrow(X, Y), b(X), Q(Y)$$
$$\underbrace{Q(X)}_{head} :- \underbrace{c(X)}_{body}$$

- **extensional** predicates ($\downarrow$, b, c);
- **intensional** predicates (P, Q);
- one designated **goal** predicate (P).

Trees

Trees over infinte alphabet Σ.
The extensional relations are

- $a(X)$ for each $a \in \Sigma$;
- the label equality $\sim (X, Y)$;
- navigational relations $\downarrow(X, Y)$ and $\downarrow_+(X, Y)$.



Datalog rules can be

1. **monadic**: head has exactly 1 variable;

Datalog rules can be

1. **monadic**: head has exactly 1 variable;
2. **linear**: at most one intensional predicate in the body;

Datalog rules can be

1. **monadic**: head has exactly 1 variable;
2. **linear**: at most one intensional predicate in the body;
3. **connected**: all used variables are connected;

Datalog rules can be

1. **monadic**: head has exactly 1 variable;
2. **linear**: at most one intensional predicate in the body;
3. **connected**: all used variables are connected;

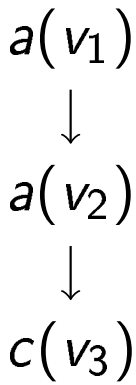
Datalog rules can be

1. **monadic**: head has exactly 1 variable;
2. **linear**: at most one intensional predicate in the body;
3. **connected**: all used variables are connected;

$$P(X) :- \downarrow(X, Y), \downarrow(Y, Z), P(Z), P(Y) \quad \text{monadic, connected}$$
$$Q(X, Y) :- a(Z), \downarrow_+(X, Y) \quad \text{linear}$$

How to evaluate P in a tree?

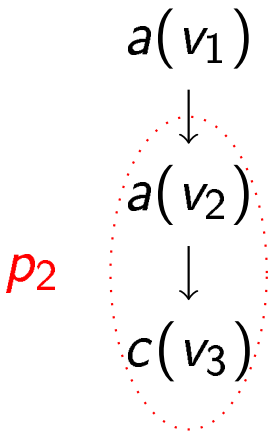
$$P(X) :- a(X), \downarrow(X, Y), a(Y), P(Y) \quad (p_1)$$
$$P(X) :- \downarrow(X, Y), c(Y) \quad (p_2)$$



$$P = \emptyset$$

How to evaluate P in a tree?

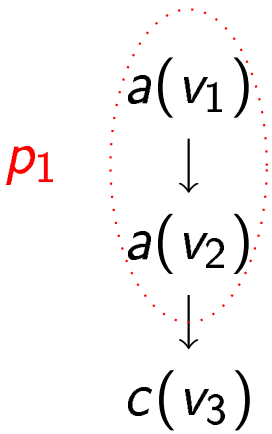
$$P(X) :- a(X), \downarrow(X, Y), a(Y), P(Y) \quad (p_1)$$
$$P(X) :- \downarrow(X, Y), c(Y) \quad (p_2)$$



$$P = \{v_2\}$$

How to evaluate P in a tree?

$$P(X) :- a(X), \downarrow(X, Y), a(Y), P(Y) \quad (p_1)$$
$$P(X) :- \downarrow(X, Y), c(Y) \quad (p_2)$$



$$P = \{v_1, v_2\}$$

Containment

Containment

Given two datalog programs, $\mathcal{P}$ and $\mathcal{Q}$, is $\mathcal{P} \subseteq \mathcal{Q}$?
That is, does $\forall$ tree t, $P(t) \neq \emptyset \implies Q(t) \neq \emptyset$ hold?

Background

Containment on general structures

- general case undecidable (Shmueli 1993);

Background

Containment on general structures

- general case undecidable (Shmueli 1993);
- in UCQ's decidable (Chaudhuri, Vardi; 1993);

Background

Containment on general structures

- general case undecidable (Shmueli 1993);
- in UCQ's decidable (Chaudhuri, Vardi; 1993);
- of monadic programs decidable (Cosmadakis et al; 1988);

Background

Containment on general structures

- general case undecidable (Shmueli 1993);
- in UCQ's decidable (Chaudhuri, Vardi; 1993);
- of monadic programs decidable (Cosmadakis et al; 1988);

Containment on trees

- Finite alphabet, monadic programs: decidable (Gottlob, Koch; 2004);

Background

Containment on general structures

- general case undecidable (Shmueli 1993);
- in UCQ's decidable (Chaudhuri, Vardi; 1993);
- of monadic programs decidable (Cosmadakis et al; 1988);

Containment on trees

- Finite alphabet, monadic programs: decidable (Gottlob, Koch; 2004);
- Infinite alphabet, linear monadic programs: undecidable (Abiteboul, Bourhis, Muscholl; 2013)

Our contribution

We identified decidable fragments:

- **child-only** programs (no $\downarrow_+$ relation);
- **downward** programs = each variable in a rule is a descendant of the head variable.

Our contribution

We identified decidable fragments:

- **child-only** programs (no $\downarrow_+$ relation);
- **downward** programs = each variable in a rule is a descendant of the head variable.

For example:

$$P(X) :- \downarrow(X, Y), a(Y) \qquad \text{downward}$$
$$P(X) :- \downarrow(Y, X), a(Y) \qquad \text{not downward}$$

Summary of the results

	Unranked trees		Ranked trees	
	linear	non-linear	linear	non-linear
D-Datalog($\downarrow, \downarrow_+$)	ExpSpace	2-ExpTime	Undec.	Undec.
Datalog($\downarrow$)	in 3-ExpTime	Undec.	2-ExpTime	in 3-ExpTime

Table : Complexity of containment for datalog fragments.

Conclusion

Thank you for your attention!