

# **Foundations of Databases**

## **Datalog Evaluation**

**Free University of Bozen – Bolzano, 2004–2005**

**Thomas Eiter**

**Institut für Informationssysteme**

**Arbeitsbereich Wissensbasierte Systeme (184/3)**

**Technische Universität Wien**

**`http://www.kr.tuwien.ac.at/staff/eiter`**

**(Slides based on material by Wolfgang Faber)**

## Simple Evaluation Algorithms

Methods to evaluate datalog program  $P$  on database instance  $\mathbf{I}$ , derived from the different equivalent definitions of the semantics:

- **Model-theoretic definition:**

Enumerate all subsets  $\mathbf{J} \subseteq \mathbf{B}(P, \mathbf{I})$  and check modelhood  
pick smallest such  $\mathbf{J}$

- **Fixpoint definition:**

Augment  $\mathbf{I}$  using operator  $\mathbf{T}_P$  until a fixpoint is reached.

- **Proof-theoretic definition:**

Use SLD resolution (bottom-up or top-down)

## Very Naive Evaluation

Given datalog program  $P$ , database instance  $\mathbf{I}$

1. Enumerate all  $\mathbf{K} \in inst(sch(P))$  with  $\emptyset \subseteq \mathbf{K} \subseteq \mathbf{B}(P, \mathbf{I})$ , by increasing cardinality.
2. Check if every rule in  $P$  is satisfied by  $\mathbf{K}$ .
3. If Yes  $\Rightarrow \mathbf{K}$  is the minimum model of  $P$  on  $\mathbf{I}$ .

Note: Exponential complexity, even if  $P$  is fixed.

## Classes of Datalog Evaluation

More sophisticated algorithms relate to proof trees.

Two major classes of evaluation approaches:

- **Bottom-Up, Forward Chaining**

Proceed in the proof tree from the leaves to the root

Apply the datalog rules “from body to head” (forward)

- **Top-Down, Backward Chaining**

Proceed in the proof tree from the root to the leaves

Apply the datalog rules “from head to body” (backward)

## Naive Evaluation

- Follow the bottom-up approach
- Compute the minimum fixpoint of  $\mathbf{T}_P$  containing  $\mathbf{I}$  ( $=\mathbf{T}_P^\omega(\mathbf{I})$ ):

**Function** LeastFixpoint(program  $P$ ; db instance  $\mathbf{I}$ ): db instance

**var** db instance  $\mathbf{J}, \mathbf{K}$ ;

**begin**

$\mathbf{J} := \mathbf{I}$ ;

**repeat**

$\mathbf{K} := \mathbf{J}$ ;

$\mathbf{J} := \mathbf{T}_P(\mathbf{K})$ ;

**until**  $\mathbf{K} = \mathbf{J}$ ;

**return**  $\mathbf{K}$ ;

**end**

## Datalog as Relational Algebra

- A rule  $r$  can be viewed as (partial) definition of the relation  $R$  occurring in  $H(r)$ .
- The union of all such definitions describes  $R$ .

Example:

$$P = \{ \begin{array}{l} tc(X, Y) \leftarrow arc(X, Y). \\ tc(X, Y) \leftarrow arc(X, Z), arc(Z, Y). \end{array} \}$$

We have:

$$arc \subseteq tc$$

$$\pi_{1,4}(\sigma_{2=3}(arc \times arc)) \subseteq tc$$

Thus,

$$tc = (arc \cup \pi_{1,4}(\sigma_{2=3}(arc \times arc)))$$

## Datalog as Relational Algebra /2

Important step: translate each datalog rule  $r$  into an SPJ expression  $Expr(r)$

1. Normalize the rule head (remove constants, multiple occurrence of the same variable by adding new variables and equality atoms)
2. rule head yields the projection
3. remove constants in ordinary body atoms (use new variables and equality atoms)
4. remove multiple variable occurrence in ordinary body atoms (use new variables and equality atoms)
5. conjunction yields the Cartesian product.

**Datalog as Relational Algebra /3**

For each relation  $R \in idb(P)$ , we can produce an equation in relational algebra

$$Expr(R, P) = \bigcup_{r \in HR(R, P)} Expr(r)$$

where  $HR(R, P)$  is the set of rules in  $P$  with relation  $R$  in the head.

Note:

- $Expr(r)$  may contain  $R$
- Optimizations may be applied to simplify  $Expr(R, P)$



## Bottom-Up Evaluation

- Given a program  $P$  with  $idb(P) = \{R_1, \dots, R_n\}$  and a database instance  $\mathbf{I}$ , compute  $P(\mathbf{I})(R_i)$  as follows.
- Suppose we have  $n$  relation instances  $S_1, \dots, S_n$  such that  $S_j \subseteq P(\mathbf{I})(R_j)$ ,  $1 \leq j \leq n$
- Let  $E_i[S_1, \dots, S_n]$  be the result of evaluating  $Expr(R_i, P)$ , where  $S_1, \dots, S_n$  are used in place of  $R_1, \dots, R_n$ , respectively.
- Using  $E_i[S_1, \dots, S_n]$  as a basis operation, different algorithms can be formulated which compute the  $P(\mathbf{I})(R_i)$  as fixpoint of a system of equations in relational algebra.

## Naive Evaluation – Gauss-Seidel Algorithm

**Function** Gauss-Seidel(program  $P$ , db instance  $\mathbf{I}$ ) : Sequence of Relations

**var** Relation  $R_1, \dots, R_n, S$ ; Integer  $i$ ; Boolean fixpoint;

**begin**

**for**  $i := 1$  **to**  $n$  **do**  $R_i := \emptyset$ ;

**repeat**

    fixpoint := true;

**for**  $i := 1$  **to**  $n$  **do**

**begin**

$S := R_i$ ;

$R_i := E_i[R_1, \dots, R_n]$ ;

**if**  $R_i \neq S$  **then** fixpoint := false;

**end**

**until** fixpoint;

**return**  $\langle R_1, \dots, R_n \rangle$

**end**

**Example**

$P : \text{black}(X) \leftarrow \text{start}(X).$

$\text{black}(X) \leftarrow \text{white}(Y), \text{arc}(Y, X).$

$\text{white}(X) \leftarrow \text{black}(Y), \text{arc}(Y, X).$

$\text{black}(X) \leftarrow \text{white}(Y), \text{arc}(X, Y).$

$\text{white}(X) \leftarrow \text{black}(Y), \text{arc}(X, Y).$

where  $\text{edb}(P) = \{\text{start}, \text{arc}\}$ ,  $\text{idb}(P) = \{\text{black}(= R_1), \text{white}(= R_2)\}$

$$\text{Expr}(R_1, P) = \text{start} \cup \pi_3(\sigma_{1=2}(R_2 \times \text{arc})) \cup \pi_2(\sigma_{1=3}(R_2 \times \text{arc}))$$

$$\text{Expr}(R_2, P) = \pi_3(\sigma_{1=2}(R_1 \times \text{arc})) \cup \pi_2(\sigma_{1=3}(R_1 \times \text{arc}))$$

**Example /2**

$$\mathbf{I} = \{start(a), arc(d, a), arc(e, a), arc(a, b), arc(a, c), arc(b, f), arc(c, f)\}$$

$$Expr(R_1, P) = start \cup \pi_3(\sigma_{1=2}(R_2 \times arc)) \cup \pi_2(\sigma_{1=3}(R_2 \times arc))$$

$$Expr(R_2, P) = \pi_3(\sigma_{1=2}(R_1 \times arc)) \cup \pi_2(\sigma_{1=3}(R_1 \times arc))$$

Iteration	$R_1$	$R_2$
0	$\{\}$	$\{\}$
1	$\{\langle a \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$
2	$\{\langle a \rangle, \langle f \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$
3	$\{\langle a \rangle, \langle f \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$

## Naive Evaluation – Disadvantages

1. Relations have to be computed always from scratch
2. Relations must be copied
3. Iteration on *all* relations, even if they do not recursively depend on each other

$$tc(X, Y) \leftarrow arc(X, Y).$$

$$tc(X, Y) \leftarrow tc(X, Z), tc(Z, Y).$$

$$utc(X, Y) \leftarrow tc(X, Y).$$

$$utc(X, Y) \leftarrow utc(X, Z), utc(Y, Z).$$

## Semi-Naive Evaluation

- **Goal:** Remove Disadvantages 1 and 2
- **Observation:** “new” tuples can only be derived from “new” tuples.
- **Idea:** Use only tuples, which have been generated in the previous step.

## Semi-naive Evaluation – First Attempt

```
Function Linear-Semi-Naive-WRONG(Linear_Program  $P$ , db instance  $\mathbf{I}$ ) : Sequence of Relations
  var Relation  $R_1, \dots, R_n, D_1, \dots, D_n$ ; Integer  $i$ ; Boolean fixpoint;
  begin
    for  $i := 1$  to  $n$  do  $R_i := \emptyset$ ;
    for  $i := 1$  to  $n$  do  $D_i := \emptyset$ ;
    repeat
      fixpoint := true;
      for  $i := 1$  to  $n$  do
        begin
           $D_i := E_i[D_1, \dots, D_n]$ ;
           $R_i := D_i \cup R_i$ ;
          if  $D_i \neq \emptyset$  then fixpoint := false;
        end
      until fixpoint;
      return  $\langle R_1, \dots, R_n \rangle$ 
    end
```

## Why the algorithm fails

$$\mathbf{I} = \{start(a), arc(d, a), arc(e, a), arc(a, b), arc(a, c), arc(b, f), arc(c, f)\}$$

$$Expr(R_1, P) = start \cup \pi_3(\sigma_{1=2}(R_2 \times arc)) \cup \pi_2(\sigma_{1=3}(R_2 \times arc))$$

$$Expr(R_2, P) = \pi_3(\sigma_{1=2}(R_1 \times arc)) \cup \pi_2(\sigma_{1=3}(R_1 \times arc))$$

Iteration	$D_1$	$D_2$	$R_1$	$R_2$
0	$\{\}$	$\{\}$	$\{\}$	$\{\}$
1	$\{\langle a \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$	$\{\langle a \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$
2	$\{\langle a \rangle, \langle f \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$	$\{\langle a \rangle, \langle f \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$
3	$\{\langle a \rangle, \langle f \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$	$\{\langle a \rangle, \langle f \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$
...	...	...	...	...



## Semi-Naive Evaluation – Second Attempt

**Function** Linear-Semi-Naive(Linear\_Program  $P$ , db instance  $\mathbf{I}$ ) : Sequence of Relations  
**var** Relation  $R_1, \dots, R_n, D_1, \dots, D_n$ ; Integer  $i$ ; Boolean  $\text{fixpoint}$ ;  
**begin**  
  **for**  $i := 1$  **to**  $n$  **do**  $R_i := \emptyset$ ;  
  **for**  $i := 1$  **to**  $n$  **do**  $D_i := \emptyset$ ;  
  **repeat**  
     $\text{fixpoint} := \text{true}$ ;  
    **for**  $i := 1$  **to**  $n$  **do**  
      **begin**  
         $D_i := E_i[D_1, \dots, D_n] - R_i$ ;  
         $R_i := D_i \cup R_i$ ;  
        **if**  $D_i \neq \emptyset$  **then**  $\text{fixpoint} := \text{false}$ ;  
      **end**  
    **until**  $\text{fixpoint}$ ;  
  **return**  $\langle R_1, \dots, R_n \rangle$   
**end**

**Example**

$$\mathbf{I} = \{start(a), arc(d, a), arc(e, a), arc(a, b), arc(a, c), arc(b, f), arc(c, f)\}$$

$$Expr(R_1, P) = start \cup \pi_3(\sigma_{1=2}(R_2 \times arc)) \cup \pi_2(\sigma_{1=3}(R_2 \times arc))$$

$$Expr(R_2, P) = \pi_3(\sigma_{1=2}(R_1 \times arc)) \cup \pi_2(\sigma_{1=3}(R_1 \times arc))$$

Iteration	$D_1$	$D_2$	$R_1$	$R_2$
0	$\{\}$	$\{\}$	$\{\}$	$\{\}$
1	$\{\langle a \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$	$\{\langle a \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$
2	$\{\langle f \rangle\}$	$\emptyset$	$\{\langle a \rangle, \langle f \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$
3	$\emptyset$	$\emptyset$	$\{\langle a \rangle, \langle f \rangle\}$	$\{\langle b \rangle, \langle c \rangle, \langle d \rangle, \langle e \rangle\}$

## Linear Recursion

- The algorithm presented above does not work in general. It works for *linear recursion*, i.e., at most one atom in the rule body is an *idb* relation.
- An algebraic condition that ensures correctness is *linearity*:

$$E_i[R'_1 \cup R''_1, \dots, R'_n \cup R''_n] = E_i[R'_1, \dots, R'_n] \cup E_i[R''_1, \dots, R''_n]$$

for each  $i \in \{1, \dots, n\}$ .

Example of non-linear program:

$$tc(X, Y) \leftarrow arc(X, Y).$$

$$tc(X, Y) \leftarrow tc(X, Z), tc(Z, Y).$$

$$\mathbf{I}(arc) = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$$

$$R_1(= tc) = arc \cup \pi_{1,4}(\sigma_{2=3}(tc \times tc))$$

**Only for linear programs?**

$$\mathbf{I}(arc) = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$$

$$R_1(= tc) = arc \cup \pi_{1,4}(\sigma_{2=3}(tc \times tc))$$

Iteration	$D_1$	$R_1$
1	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$
2	$\{\langle 1, 3 \rangle, \langle 2, 4 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 3 \rangle, \langle 2, 4 \rangle\}$
3	$\emptyset$	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 3 \rangle, \langle 2, 4 \rangle\}$

$\langle 1, 4 \rangle$  was not computed!

**Semi-Naive Evaluation for Nonlinear Recursion**

“Linearization” of direct recursion (arguments omitted):

$$R \leftarrow B, R^{(1)}, \dots, R^{(n)}.$$

becomes

$$R \leftarrow B, R^{(1)}, S_1.$$

$$S_1 \leftarrow R^{(2)}, S_2.$$

...

$$S_{n-1} \leftarrow R^{(n)}.$$

and modify the algorithm.

## Semi-Naive Evaluation: Algorithm

**Function** Semi-Naive(Linearized\_Program  $P$ ; db instance  $\mathbf{I}$ ) : Sequence of Relations  
**var** Relation  $R_1, \dots, R_n, D_1, \dots, D_n$ ; Integer  $i$ ; Boolean fixpoint;  
**begin**  
  **for**  $i := 1$  **to**  $n$  **do**  $R_i := \emptyset$ ;  
  **for**  $i := 1$  **to**  $n$  **do**  $D_i := \emptyset$ ;  
  **repeat**  
    fixpoint := true;  
    **for**  $i := 1$  **to**  $n$  **do**  
      **begin**  
         $D_i := E_i[R_1, \dots, D_i, \dots, R_n] - R_i$ ;  
         $R_i := D_i \cup R_i$ ;  
        **if**  $D_i \neq \emptyset$  **then** fixpoint := false;  
      **end**  
    **until** fixpoint;  
  **return**  $\langle R_1, \dots, R_n \rangle$  **end**

**Example**

$$P : \quad \{ tc(X, Y) \leftarrow arc(X, Y) \\ tc(X, Y) \leftarrow tc(X, Z), tc(Z, Y) \}$$

where  $edb(P) = \{arc\}$ ,  $idb(P) = \{tc\}$

$$P' : \quad \{ tc(X, Y) \leftarrow arc(X, Y) \\ tc(X, Y) \leftarrow tc(X, Z), aux(Z, Y) \\ aux(Z, Y) \leftarrow tc(Z, Y) \}$$

where  $edb(P') = \{arc\}$ ,  $idb(P') = \{tc, aux\}$

**Example /2**

$$\mathbf{I}(arc) = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$$

$$R_1(= tc) = arc \cup \pi_{1,4}(\sigma_{2=3}(tc \times aux))$$

$$R_2(= aux) = tc$$

Iteration	$D_1$	$R_1$
1	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$
2	$\{\langle 1, 3 \rangle, \langle 2, 4 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 3 \rangle, \langle 2, 4 \rangle\}$
3	$\{\langle 1, 4 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 3 \rangle, \langle 2, 4 \rangle, \langle 1, 4 \rangle\}$

Iteration	$D_2$	$R_2$
1	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$
2	$\{\langle 1, 3 \rangle, \langle 2, 4 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 3 \rangle, \langle 2, 4 \rangle\}$
3	$\{\langle 1, 4 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 3 \rangle, \langle 2, 4 \rangle, \langle 1, 4 \rangle\}$



## Componentwise Evaluation

Disadvantage 3 (always all relations at once) is still present

- **Observation:** The program can be split and evaluated incrementally
- **Idea:** Partition the program such that there is no mutually recursive dependency between the components

Utilize a *dependency graph*

## Componentwise Evaluation /2

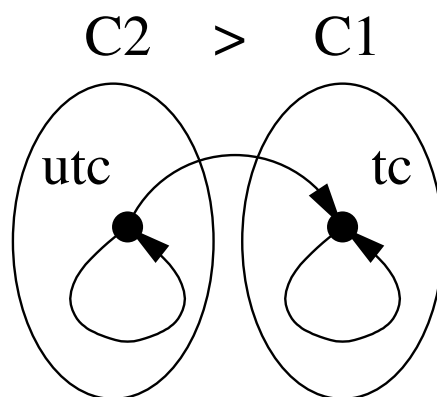
- Associate with datalog program  $P$  a directed graph  $DEP(P) = (N, E)$ , called *Dependency Graph*, as follows:
  - $N = sch(P)$ , i.e., the nodes are the relations in  $P$ .
  - $E = \{\langle R, R' \rangle \mid \exists r \in P : H(r) = R \wedge R' \in B(r)\}$ , i.e., arcs  $R \rightarrow R'$  from the relations in rule heads to the relations in the body.
- *Component of  $P$* : maximal subset  $C \subseteq N$ , such that for each  $R, R' \in C$  a path from  $R$  to  $R'$  exists within  $C$  (“strongly connected component”)
- *partial order  $\leq$*  on set of all components  $Comp(P)$ :  $C \leq C'$  iff some node  $R \in C'$  reaches some node  $R' \in C$

## Componentwise Evaluation /3

- Apply the basis algorithm only to the subprogram  $P_C$  corresponding to a component of  $P$ .
- Proceed along the partial order  $\leq$  on  $Comp(P)$ , starting with a component  $C$  which does not depend on other components
- All relations in already processed components can be viewed as completely evaluated, and treated like *edb* relations

**Example**

$P :$

$$tc(X, Y) \leftarrow arc(X, Y)$$
$$tc(X, Y) \leftarrow tc(X, Z), tc(Z, Y)$$
$$utc(X, Y) \leftarrow tc(X, Y)$$
$$utc(X, Y) \leftarrow utc(X, Z), utc(Y, Z)$$


**Example /2**

$$P_{C1} = \{ tc(X, Y) \leftarrow arc(X, Y) \\ tc(X, Y) \leftarrow tc(X, Z), tc(Z, Y) \}$$

$$P_{C2} = \{ utc(X, Y) \leftarrow tc(X, Y) \\ utc(X, Y) \leftarrow utc(X, Z), utc(Y, Z) \}$$

Evaluate  $P_{C1}$ , followed by  $P_{C2}$ .

## Top-Down Techniques

Advantages / Desiderata:

- Start from a *query*, i.e., a pair  $(P, q)$  of a datalog program  $P$  and a rule  $q$  of form

$$query(\vec{x}) \leftarrow R(\vec{v})$$

where *query* is a new relation and  $R \in idb(P)$

- *set-at-a-time* processing
- Involve only facts *relevant* to answering the query

## Top-Down – Query-Subquery

### Basic Elements:

- Framework of SLD Resolution, but set-at-a-time. Permits use of (optimized versions of) relational algebra operations
- “Push” constants from goals to subgoals (similar to push selections into joins)
- Pass constant binding information from one atom to the next in subgoals
- Use an efficient global flow-of-control strategy

## Reachable Adorned Rules

- Annotate *idb* relations with a string of *b*'s and *f*'s
- *b* represents bound arguments
- *f* represents free arguments
- string length = arity



## Reachable Adorned Rules /2

*Bindings* are created through:

- constants
- variables which occur in a bound argument in the rule head
- variables which occur repeatedly in the body (from 2nd occurrence on)

Here, the order of atoms in the body matters!

**Example**

$$rsg(X, Y) \leftarrow flat(X, Y).$$

$$rsg(X, Y) \leftarrow up(X, X1), rsg(Y1, X1), down(Y1, Y).$$

$$\text{Query} \rightarrow query(Y) \leftarrow rsg(a, Y).$$

Reachable Adorned Program (*up*, *down*, *flat* are *edb*):

$$rsg^{bf}(X, Y) \leftarrow flat(X, Y).$$

$$rsg^{bf}(X, Y) \leftarrow up(X, X1), rsg^{fb}(Y1, X1), down(Y1, Y).$$

$$rsg^{fb}(X, Y) \leftarrow flat(X, Y).$$

$$rsg^{fb}(X, Y) \leftarrow down(Y1, Y), rsg^{bf}(Y1, X1), up(X, X1).$$

## Subqueries

*Idea:* Utilize adornments for defining and evaluating restricted subqueries

- Each atom in the body represents a relation
- These relations are constrained by propagated bindings
- Compute subrelations for the already bound variables or variables that are to be bound.

## Supplementary Relations

*Idea:* identify for each position in the rule body the “interesting” variable bindings

- “Interesting” are variables which are already bound at the respective position and are either used in the remainder of the body, or are from the head.
- At the beginning (leftmost), bindings might result from the rule head.
- At the end (rightmost), the variables in the head must have been used (safety).

**Example: Supplementary Relations**

$$rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad .$$

$$\begin{array}{ccc} & \uparrow & \uparrow \\ & sup_0^1[X] & sup_1^1[X, Y] \end{array}$$

$$rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad .$$

$$\begin{array}{cccc} & \uparrow & \uparrow & \uparrow & \uparrow \\ & sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \end{array}$$

**Example: Supplementary Relations /2**

$$rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad .$$

$$\begin{array}{ccc} \uparrow & & \uparrow \\ sup_0^3[Y] & & sup_1^3[X, Y] \end{array}$$

$$rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad .$$

$$\begin{array}{cccc} \uparrow & & \uparrow & & \uparrow & & \uparrow \\ sup_0^4[Y] & & sup_1^4[Y, Y1] & & sup_2^4[Y, X1] & & sup_3^4[X, Y] \end{array}$$

## QSQ Templates and Algorithms

- For each rule  $r$  in the Reachable Adorned Program,  $\langle sup_0, \dots, sup_n \rangle$  is called the *QSQ Template* for  $r$ .
- These QSQ Templates are used as a temporary memory for computing and storing intermediate results.
- In addition, for every *idb* relation  $R^\gamma$  in the Reachable Adorned Program, two relation variables are needed:
  - *input* $_R^\gamma$  (whose arity is the number of  $b$ 's in the adornment  $\gamma$ ), and
  - *ans* $_R^\gamma$  (same arity as  $R$ ).

## QSQ Algorithms

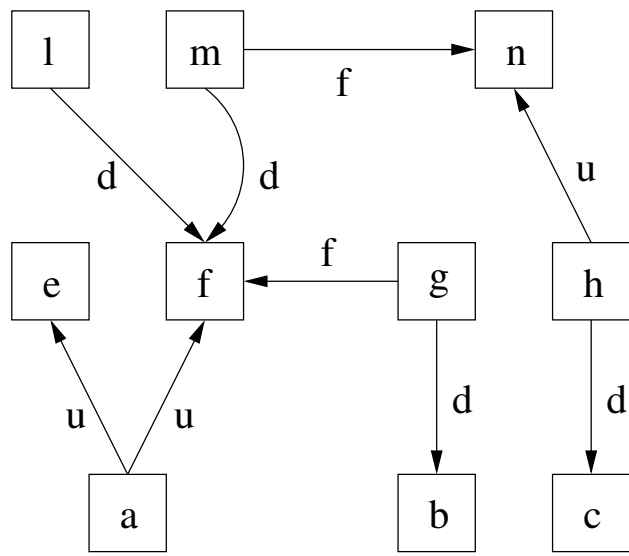
Different algorithms can be imagined, depending on the control strategy:

- Search strategy
  - breadth first
  - depth first
- Processing
  - recursive
  - iterative

The formulation of the algorithms is involved. We skip details (see literature) and consider merely an example.



# QSQ – Example



<i>up</i>		
	<i>a</i>	<i>e</i>
	<i>a</i>	<i>f</i>
	<i>h</i>	<i>n</i>

<i>flat</i>		
	<i>g</i>	<i>f</i>
	<i>m</i>	<i>n</i>

<i>down</i>		
	<i>l</i>	<i>f</i>
	<i>m</i>	<i>f</i>
	<i>g</i>	<i>b</i>
	<i>h</i>	<i>c</i>

## QSQ – Example /2

Results after some steps of the QSQ approach (init: put  $\langle a \rangle$  into  $input\_rsg^{bf}$ )

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^1[X] & sup_1^1[X, Y] \\
 \hline
 a & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a & \begin{array}{c} a \ e \\ a \ f \end{array} & a \ g & a \ b
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^3[Y] & sup_1^3[X, Y] \\
 \hline
 e \ f & g \ f
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e \ f & \begin{array}{c} f \ l \\ f \ m \end{array} & \dots & \dots
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a} & \frac{input\_rsg^{fb}}{e \ f} & \frac{ans\_rsg^{bf}}{a \ b} & \frac{ans\_rsg^{fb}}{g \ f}
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^1[X] & sup_1^1[X, Y] \\
 \hline
 a & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a & \begin{array}{c} a \ e \\ a \ f \end{array} & a \ g & a \ b
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^3[Y] & sup_1^3[X, Y] \\
 \hline
 e \ f & g \ f
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e \ f & \begin{array}{c} f \ l \\ f \ m \end{array} & \dots & \dots
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a \ 1} & \frac{input\_rsg^{fb}}{e \ f} & \frac{ans\_rsg^{bf}}{a \ b} & \frac{ans\_rsg^{fb}}{g \ f}
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{ccc}
 \uparrow & & \uparrow \\
 sup_0^1[X] & & sup_1^1[X, Y] \\
 \hline
 a & \mathbf{2} & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a & \mathbf{2} & a & e \\
 & & a & f \\
 & & a & g \\
 & & & a & b
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{ccc}
 \uparrow & & \uparrow \\
 sup_0^3[Y] & & sup_1^3[X, Y] \\
 \hline
 e & & g & f \\
 f & & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e & & f & l \\
 f & & f & m \\
 & & \dots & \dots
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a} & \mathbf{1} & \frac{input\_rsg^{fb}}{e} & f \\
 \frac{ans\_rsg^{bf}}{a} & b & \frac{ans\_rsg^{fb}}{g} & f
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^1[X] & sup_1^1[X, Y] \\
 \hline
 a \quad 2
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a \quad 2 & a \quad e \quad 3 & a \quad g & a \quad b \\
 & a \quad f \quad 3 & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^3[Y] & sup_1^3[X, Y] \\
 \hline
 e & g \quad f \\
 f & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e & f \quad l & \dots & \dots \\
 f & f \quad m & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a \quad 1} & \frac{input\_rsg^{fb}}{e} & \frac{ans\_rsg^{bf}}{a \quad b} & \frac{ans\_rsg^{fb}}{g \quad f}
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^1[X] & sup_1^1[X, Y] \\
 \hline
 a \quad 2
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a \quad 2 & a \quad e \quad 3 & a \quad g & a \quad b \\
 & a \quad f \quad 3 & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^3[Y] & sup_1^3[X, Y] \\
 \hline
 e & g \quad f \\
 f & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e & f \quad l & \dots & \dots \\
 f & f \quad m & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a \quad 1} & \frac{input\_rsg^{fb}}{e \quad 4} & \frac{ans\_rsg^{bf}}{a \quad b} & \frac{ans\_rsg^{fb}}{g \quad f} \\
 & f \quad 4 & & 
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^1[X] & sup_1^1[X, Y] \\
 \hline
 a \quad 2 & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a \quad 2 & \begin{array}{c} a \quad e \quad 3 \\ a \quad f \quad 3 \end{array} & a \quad g & a \quad b
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^3[Y] & sup_1^3[X, Y] \\
 \hline
 e \quad 5 & g \quad f \\
 f \quad 5 & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e \quad 5 & \begin{array}{c} f \quad l \\ f \quad m \end{array} & \dots & \dots \\
 f \quad 5 & & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a \quad 1} & \frac{input\_rsg^{fb}}{e \quad 4 \\ f \quad 4} & \frac{ans\_rsg^{bf}}{a \quad b} & \frac{ans\_rsg^{fb}}{g \quad f}
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^1[X] & sup_1^1[X, Y] \\
 \hline
 a \quad 2 &
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a \quad 2 & \begin{array}{c} a \quad e \quad 3 \\ a \quad f \quad 3 \end{array} & a \quad g & a \quad b
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^3[Y] & sup_1^3[X, Y] \\
 \hline
 e \quad 5 \\
 f \quad 5 & g \quad f \quad 6
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e \quad 5 & \begin{array}{c} f \quad l \\ f \quad m \end{array} & \dots & \dots \\
 \hline
 \begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a \quad 1} & \frac{input\_rsg^{fb}}{e \quad 4 \\ f \quad 4} & \frac{ans\_rsg^{bf}}{a \quad b} & \frac{ans\_rsg^{fb}}{g \quad f}
 \end{array}
 \end{array}
 \end{array}$$



## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{ccc}
 \uparrow & & \uparrow \\
 sup_0^1[X] & & sup_1^1[X, Y] \\
 \hline
 a \quad \mathbf{2}
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a \quad \mathbf{2} & \begin{array}{c} a \quad e \quad \mathbf{3} \\ a \quad f \quad \mathbf{3} \end{array} & a \quad g & a \quad b
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{ccc}
 \uparrow & & \uparrow \\
 sup_0^3[Y] & & sup_1^3[X, Y] \\
 \hline
 e \quad \mathbf{5} \\
 f \quad \mathbf{5} & & g \quad f \quad \mathbf{6}
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e \quad \mathbf{5} & \begin{array}{c} f \quad l \quad \mathbf{7} \\ f \quad m \quad \mathbf{7} \end{array} & \dots & \dots
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \hline
 \frac{input\_rsg^{bf}}{a \quad \mathbf{1}} & \frac{input\_rsg^{fb}}{e \quad \mathbf{4} \\ f \quad \mathbf{4}} & \frac{ans\_rsg^{bf}}{a \quad b} & \frac{ans\_rsg^{fb}}{g \quad f} \\
 \hline
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{ccc}
 \uparrow & & \uparrow \\
 sup_0^1[X] & & sup_1^1[X, Y] \\
 \hline
 a \quad \mathbf{2}
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a \quad \mathbf{2} & \begin{array}{c} a \quad e \quad \mathbf{3} \\ a \quad f \quad \mathbf{3} \end{array} & a \quad g & a \quad b
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{ccc}
 \uparrow & & \uparrow \\
 sup_0^3[Y] & & sup_1^3[X, Y] \\
 \hline
 e \quad \mathbf{5} \\
 f \quad \mathbf{5} & & g \quad f \quad \mathbf{6}
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e \quad \mathbf{5} & \begin{array}{c} f \quad l \quad \mathbf{7} \\ f \quad m \quad \mathbf{7} \end{array} & \dots & \dots
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a \quad \mathbf{1}} & \frac{input\_rsg^{fb}}{e \quad \mathbf{4} \\ f \quad \mathbf{4}} & \frac{ans\_rsg^{bf}}{a \quad b} & \frac{ans\_rsg^{fb}}{g \quad f \quad \mathbf{8}}
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^1[X] & sup_1^1[X, Y] \\
 \hline
 a \mathbf{2} & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a \mathbf{2} & \begin{array}{c} a \ e \ \mathbf{3} \\ a \ f \ \mathbf{3} \end{array} & a \ g \ \mathbf{9} & a \ b
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^3[Y] & sup_1^3[X, Y] \\
 \hline
 e \ \mathbf{5} \\
 f \ \mathbf{5} & g \ f \ \mathbf{6}
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e \ \mathbf{5} & \begin{array}{c} f \ l \ \mathbf{7} \\ f \ m \ \mathbf{7} \end{array} & \dots & \dots \\
 \hline
 \begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a \ \mathbf{1}} & \frac{input\_rsg^{fb}}{e \ \mathbf{4} \\ f \ \mathbf{4}} & \frac{ans\_rsg^{bf}}{a \quad b} & \frac{ans\_rsg^{fb}}{g \quad f \ \mathbf{8}}
 \end{array}
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^1[X] & sup_1^1[X, Y] \\
 \hline
 a \quad 2 & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a \quad 2 & a \quad e \quad 3 & a \quad g \quad 9 & a \quad b \quad 10 \\
 & a \quad f \quad 3 & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{cc}
 \uparrow & \uparrow \\
 sup_0^3[Y] & sup_1^3[X, Y] \\
 \hline
 e \quad 5 & g \quad f \quad 6 \\
 f \quad 5 & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e \quad 5 & f \quad l \quad 7 & \dots & \dots \\
 f \quad 5 & f \quad m \quad 8 & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a \quad 1} & \frac{input\_rsg^{fb}}{e \quad 4} & \frac{ans\_rsg^{bf}}{a \quad b} & \frac{ans\_rsg^{fb}}{g \quad f \quad 8} \\
 & f \quad 4 & & 
 \end{array}$$

## QSQ – Example /2

Results after some steps of the QSQ approach

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{ccc}
 \uparrow & & \uparrow \\
 sup_0^1[X] & & sup_1^1[X, Y] \\
 \hline
 a \quad 2 & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{bf}(X, Y) \leftarrow up(X, X1), \quad rsg^{fb}(Y1, X1), \quad down(Y1, Y) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^2[X] & sup_1^2[X, X1] & sup_2^2[X, Y1] & sup_3^2[X, Y] \\
 \hline
 a \quad 2 & a \quad e \quad 3 & a \quad g \quad 9 & a \quad b \quad 10 \\
 & a \quad f \quad 3 & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow flat(X, Y) \quad . \\
 \begin{array}{ccc}
 \uparrow & & \uparrow \\
 sup_0^3[Y] & & sup_1^3[X, Y] \\
 \hline
 e \quad 5 & & g \quad f \quad 6 \\
 f \quad 5 & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{c}
 rsg^{fb}(X, Y) \leftarrow down(Y1, Y), \quad rsg^{bf}(Y1, X1), \quad up(X, X1) \quad . \\
 \begin{array}{cccc}
 \uparrow & \uparrow & \uparrow & \uparrow \\
 sup_0^4[Y] & sup_1^4[Y, Y1] & sup_2^4[Y, X1] & sup_3^4[X, Y] \\
 \hline
 e \quad 5 & f \quad l \quad 7 & \dots & \dots \\
 f \quad 5 & f \quad m \quad 8 & & 
 \end{array}
 \end{array}$$
  

$$\begin{array}{cccc}
 \frac{input\_rsg^{bf}}{a \quad 1} & \frac{input\_rsg^{fb}}{e \quad 4} & \frac{ans\_rsg^{bf}}{a \quad b \quad 11} & \frac{ans\_rsg^{fb}}{g \quad f \quad 8} \\
 & f \quad 4 & & 
 \end{array}$$

## Magic Sets

- Integrate the advantages of top-down and bottom-up.
- Express elements of the QSQ approach in datalog itself!
- Make supplementary relations explicit (so called *magic* predicates/relations).

Basically, they store the relations

- $input\_R^\gamma$  ( $magic\_R^\gamma$ )
- $sup_j^i$  ( $supmagic_j^i$ ), except the “rightmost” in rule  $i$

- Realized by program transformation, which linearizes the rules bodies (evaluate left to right)
- When evaluated bottom up, it produces only the set of facts produced by top-down approaches

## Magic Sets – Example

$$rsg^{bf}(X, Y) \leftarrow magic\_rsg^{bf}(X), flat(X, Y).$$

$$rsg^{bf}(X, Y) \leftarrow supmagic_2^2(X, Y1), down(Y1, Y).$$

$$supmagic_2^2(X, Y1) \leftarrow supmagic_1^2(X, X1), rsg^{fb}(Y1, X1).$$

$$supmagic_1^2(X, Y1) \leftarrow magic\_rsg^{bf}(X), up(X, X1).$$

$$rsg^{fb}(X, Y) \leftarrow magic\_rsg^{fb}(Y), flat(X, Y).$$

$$rsg^{fb}(X, Y) \leftarrow supmagic_2^4(Y, X1), up(X, X1).$$

$$supmagic_2^4(Y, X1) \leftarrow supmagic_1^4(Y, Y1), rsg^{bf}(Y1, X1).$$

$$supmagic_1^4(Y, Y1) \leftarrow magic\_rsg^{fb}(Y), down(Y1, Y).$$

$$magic\_rsg^{bf}(a).$$

$$magic\_rsg^{bf}(X1) \leftarrow supmagic_1^2(X, X1).$$

$$magic\_rsg^{fb}(Y1) \leftarrow supmagic_1^4(Y, Y1).$$

$$query(Y) \leftarrow rsg^{bf}(a, Y).$$

## Magic Sets – Generation

- The *magic* relations prevent generating *irrelevant* atoms by building “cones” starting from the query.
- The transformation algorithm is involved but comparatively easy to implement.
- In some cases, the transformation just creates overhead and no gain is achieved. In such cases other optimisations may be found (e.g., identical variables within the same atom).



## Bibliography

- [1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [2] S. Ceri, G. Gottlob, and L. Tanca. *Logic Programming and Databases*. Springer, 1990.
- [3] H. Garcia-Molina, J. D. Ullman, and J. Widom. *Database Systems – The Complete Book*. Prentice Hall, 2002.