Computational Logic

Datalog Evaluation

Free University of Bozen-Bolzano, 2010

Werner Nutt

(Based on slides by Thomas Eiter and 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 minimal model of P on $\mathbf{I}$.

What is the complexity? How many $\mathbf{K}$s are there?

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:

$$P = \{ \quad tc(X,Y) \leftarrow arc(X,Y).$$
$$\qquad\quad tc(X,Y) \leftarrow arc(X,Z), tc(Z,Y). \}$$

We have:

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

Thus,

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

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 and multiple occurrences of the same variable by adding new variables and equality atoms)
2. The rule head yields the projection
3. Remove constants in ordinary body atoms (use new variables and equality atoms)
4. Remove multiple variable occurrences in ordinary body atoms (use new variables and equality atoms)
5. The conjunction of ordinary atoms 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 $idb(P) = \{R_1, \ldots, R_n\}$ and a database instance $\mathbf{I}$, compute $P(\mathbf{I})(R_i)$ as follows.
- Suppose we have n relation instances $S_1, \ldots, S_n$ such that $S_j \subseteq P(\mathbf{I})(R_j)$, $j = 1, \ldots, n$
- Let $E_i[S_1, \ldots, S_n]$ be the result of evaluating $Expr(R_i, P)$, where $S_1, \ldots, S_n$ are used in place of $R_1, \ldots, R_n$, respectively.
- Using $E_i[S_1, \ldots, S_n]$ as a basic 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 $R_1, \ldots, R_n, S$; Integer i; Boolean fixpoint;
  begin
    for i := 1 **to** n **do** $R_i := \emptyset$;
    repeat
      fixpoint := true;
      for i := 1 **to** n **do**
        begin
          $S := R_i$;
          $R_i := E_i[R_1, \ldots, R_n]$;
          if $R_i \neq S$ **then** fixpoint := false;
        end
    until fixpoint;
    return $\langle R_1, \ldots, R_n \rangle$
  end

Example

$$
\begin{aligned}
P: \quad & black(X) \leftarrow start(X). \\
& black(X) \leftarrow white(Y), arc(Y, X). \\
& white(X) \leftarrow black(Y), arc(Y, X). \\
& black(X) \leftarrow white(Y), arc(X, Y). \\
& white(X) \leftarrow black(Y), arc(X, Y).
\end{aligned}
$$

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

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

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

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 that 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 $R_1, \ldots, R_n, D_1, \ldots, D_n$; 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 := $E_i[D_1, \ldots, D_n]$;
          $R_i := D_i \cup R_i$;
          if $D_i \neq \emptyset$ **then** fixpoint := false;
        end
    until fixpoint;
    return $\langle R_1, \ldots, R_n \rangle$
  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 $R_1, \ldots, R_n, D_1, \ldots, D_n$; 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
          $D_i := E_i[D_1, \ldots, D_n] \setminus R_i$;
          $R_i := D_i \cup R_i$;
          if $D_i \neq \emptyset$ **then** fixpoint := false;
        end
    until fixpoint;
    return $\langle R_1, \ldots, R_n \rangle$
  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*:

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

for each $i \in \{1, \ldots, n\}$. *Why?*

Example of non-linear program:

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

$$\begin{aligned} \mathbf{I}(arc) &= \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\} \\ R_1(= tc) &= arc \cup \pi_{1,4}(\sigma_{2=3}(tc \times tc)) \end{aligned}$$

Only for Linear Programs?

$$\begin{aligned} \mathbf{I}(arc) &= \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\} \\ R_1(= tc) &= arc \cup \pi_{1,4}(\sigma_{2=3}(tc \times tc)) \end{aligned}$$

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

- Assumption: No rule body has multiple occurrence of same idb relation
- Introduce "alias" auxiliary relation in case.
- Call programs with this property "cleaned"

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

Semi-Naive Evaluation for Nonlinear Recursion /2

- **Method**: "Linearize" direct recursion: Replace each rule (arguments omitted)

$$R_i \leftarrow Q_1, \ldots, Q_k, R_i^{(1)}, \ldots, R_i^{(n)},$$

$n > 1$, where $Q_1, \ldots, Q_k$, are the relations different from R, by

$$R_i \leftarrow Q_1, \ldots, Q_k, R_i^{(1)}, S_1.$$
$$S_1 \leftarrow R_i^{(2)}, S_2.$$
$$\ldots$$
$$S_{n-1} \leftarrow R_i^{(n)}.$$

where $S_1, \ldots S_{n-1}$ are fresh relations.

- Modify the algorithm for such "linearized" programs.

Semi-Naive Evaluation: Algorithm

Function Semi-Naive(Linearized_Program P; db instance $\mathbf{I}$) : Sequence of Relations
var Relation $R_1, \ldots, R_n, D_1, \ldots, D_n$; 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
$D_i := \bigcup_{j=1}^{n} E_i[R_1, \ldots, R_{j-1}, D_j, R_{j+1} \ldots, R_n] \setminus R_i$;
$R_i := D_i \cup R_i$;
if $D_i \neq \emptyset$ **then** fixpoint := false;
end
until fixpoint;
return $\langle R_1, \ldots, R_n \rangle$ **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

$$\mathbf{I}(arc) = \{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle\} \quad R_1(= tc) = arc \cup \pi_{1,4}(\sigma_{2=3}(tc \times aux))$$
$$R_2(= aux) = tc$$

Iteration	D_1	R_1
0	$\emptyset$	$\emptyset$
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	$\{\langle 1,4\rangle\}$	$\{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle, \langle 1,3\rangle, \langle 2,4\rangle, \langle 1,4\rangle\}$
4	$\emptyset$	$\{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle, \langle 1,3\rangle, \langle 2,4\rangle, \langle 1,4\rangle\}$

Iteration	D_2	R_2
0	$\emptyset$	$\emptyset$
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	$\{\langle 1,4\rangle\}$	$\{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle, \langle 1,3\rangle, \langle 2,4\rangle, \langle 1,4\rangle\}$
4	$\emptyset$	$\{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle, \langle 1,3\rangle, \langle 2,4\rangle, \langle 1,4\rangle\}$

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

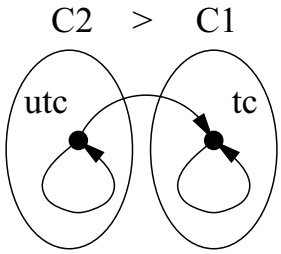
- Associate with datalog program P a directed graph *DEP*$(P) = (N, E)$, called *Dependency Graph*, as follows:
  - $N = sch(P)$, i.e., the nodes are the relations in P.
  - $E = \{\langle R, R' \rangle \mid \exists r \in P : H(r) = R \wedge R' \in B(r)\}$, i.e., arcs $R \rightarrow R'$ from the relations 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 basic 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

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



Example /2

$$P_{C1} = \{\ tc(X,Y) \leftarrow arc(X,Y)$$
$$\qquad tc(X,Y) \leftarrow tc(X,Z), tc(Z,Y)\}$$
$$P_{C2} = \{\ utc(X,Y) \leftarrow tc(X,Y)$$
$$\qquad utc(X,Y) \leftarrow utc(X,Z), utc(Y,Z)\}$$

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 pushing 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

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

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

$$\begin{array}{llll} rsg^{bf}(X,Y) \leftarrow & up(X,X1), & rsg^{fb}(Y1,X1), & down(Y1,Y) \quad . \\ & \uparrow & \uparrow & \uparrow \quad \uparrow \\ & sup_0^2[X] & sup_1^2[X,X1] & sup_2^2[X,Y1] \quad sup_3^2[X,Y] \end{array}$$

Example: Supplementary Relations /2

$$rsg^{fb}(X,Y) \leftarrow \underset{\uparrow}{flat(X,Y)} \quad .$$
$$\underset{sup_0^3[Y]}{\uparrow} \qquad \underset{sup_1^3[X,Y]}{\uparrow}$$

$$rsg^{fb}(X,Y) \leftarrow \underset{\uparrow}{down(Y1,Y)}, \quad \underset{\uparrow}{rsg^{bf}(Y1,X1)}, \quad \underset{\uparrow}{up(X,X1)} \quad .$$
$$\underset{sup_0^4[Y]}{\uparrow} \qquad \underset{sup_1^4[Y,Y1]}{\uparrow} \qquad \underset{sup_2^4[Y,X1]}{\uparrow} \quad \underset{sup_3^4[X,Y]}{\uparrow}$$

QSQ Templates and Algorithms

- For each rule r in the Reachable Adorned Program, $\langle sup_0, \ldots, sup_n \rangle$ 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^γ in the Reachable Adorned Program, two relation variables are needed:
  - $input\_R^\gamma$ (whose arity is the number of b's in the adornment γ), and
  - ans_R^γ (same arity as R).

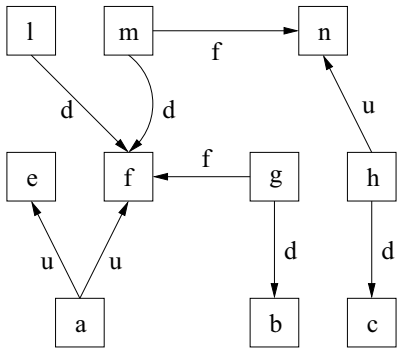
QSQ Algorithms

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



up	
a	e
a	f
h	n

$flat$	
g	f
m	n

$down$	
l	f
m	f
g	b
h	c

QSQ – Example /2

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

$$rsg^{bf}(X,Y) \leftarrow \underset{\substack{\uparrow \\ sup_0^1[X] \\ \hline a}}{} \quad flat(X,Y) \quad . \qquad \underset{\substack{\uparrow \\ sup_1^1[X,Y] \\ \hline}}{}$$

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

$$rsg^{fb}(X,Y) \leftarrow \underset{\substack{\uparrow \\ sup_0^3[Y] \\ \hline e \\ f}}{} \quad flat(X,Y) \quad . \qquad \underset{\substack{\uparrow \\ sup_1^3[X,Y] \\ \hline g\ f}}{}$$

$$rsg^{fb}(X,Y) \leftarrow \underset{\substack{\uparrow \\ sup_0^4[Y] \\ \hline e \\ f}}{} \quad down(Y1,Y), \quad \underset{\substack{\uparrow \\ sup_1^4[Y,Y1] \\ \hline f\ l \\ f\ m}}{} \quad rsg^{bf}(Y1,X1), \quad \underset{\substack{\uparrow \\ sup_2^4[Y,X1] \\ \hline \ldots}}{} \quad up(X,X1) \quad . \quad \underset{\substack{\uparrow \\ sup_3^4[X,Y] \\ \hline \ldots}}{}$$

$$\underset{\substack{input\_rsg^{bf} \\ \hline a}}{} \qquad \underset{\substack{input\_rsg^{fb} \\ \hline e \\ f}}{} \qquad \underset{\substack{ans\_rsg^{bf} \\ \hline a\quad b}}{} \qquad \underset{\substack{ans\_rsg^{fb} \\ \hline g\quad f}}{}$$

QSQ – Example /2

Results after some steps of the QSQ approach

$$rsg^{bf}(X,Y) \leftarrow \underset{\substack{\uparrow \\ sup_0^1[X] \\ \hline a}}{} \quad flat(X,Y) \quad . \qquad \underset{\substack{\uparrow \\ sup_1^1[X,Y] \\ \hline}}{}$$

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

$$rsg^{fb}(X,Y) \leftarrow \underset{\substack{\uparrow \\ sup_0^3[Y] \\ \hline e \\ f}}{} \quad flat(X,Y) \quad . \qquad \underset{\substack{\uparrow \\ sup_1^3[X,Y] \\ \hline g\ f}}{}$$

$$rsg^{fb}(X,Y) \leftarrow \underset{\substack{\uparrow \\ sup_0^4[Y] \\ \hline e \\ f}}{} \quad down(Y1,Y), \quad \underset{\substack{\uparrow \\ sup_1^4[Y,Y1] \\ \hline f\ l \\ f\ m}}{} \quad rsg^{bf}(Y1,X1), \quad \underset{\substack{\uparrow \\ sup_2^4[Y,X1] \\ \hline \ldots}}{} \quad up(X,X1) \quad . \quad \underset{\substack{\uparrow \\ sup_3^4[X,Y] \\ \hline \ldots}}{}$$

$$\underset{\substack{input\_rsg^{bf} \\ \hline a\ \mathbf{1}}}{} \qquad \underset{\substack{input\_rsg^{fb} \\ \hline e \\ f}}{} \qquad \underset{\substack{ans\_rsg^{bf} \\ \hline a\quad b}}{} \qquad \underset{\substack{ans\_rsg^{fb} \\ \hline g\quad f}}{}$$

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^1[X,Y]}{}$$

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

$$\frac{sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^2[X,X1]}{\begin{array}{l}a\ e\\ a\ f\end{array}} \qquad \frac{sup_2^2[X,Y1]}{a\ g} \qquad \frac{sup_3^2[X,Y]}{a\ b}$$

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

$$\frac{sup_0^3[Y]}{\begin{array}{l}e\\ f\end{array}} \qquad \frac{sup_1^3[X,Y]}{g\ f}$$

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

$$\frac{sup_0^4[Y]}{\begin{array}{l}e\\ f\end{array}} \qquad \frac{sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\\ f\ m\end{array}} \qquad \frac{sup_2^4[Y,X1]}{\ldots} \qquad \frac{sup_3^4[X,Y]}{\ldots}$$

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

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^1[X,Y]}{}$$

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

$$\frac{sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^2[X,X1]}{\begin{array}{l}a\ e\ \mathbf{3}\\ a\ f\ \mathbf{3}\end{array}} \qquad \frac{sup_2^2[X,Y1]}{a\ g} \qquad \frac{sup_3^2[X,Y]}{a\ b}$$

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

$$\frac{sup_0^3[Y]}{\begin{array}{l}e\\ f\end{array}} \qquad \frac{sup_1^3[X,Y]}{g\ f}$$

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

$$\frac{sup_0^4[Y]}{\begin{array}{l}e\\ f\end{array}} \qquad \frac{sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\\ f\ m\end{array}} \qquad \frac{sup_2^4[Y,X1]}{\ldots} \qquad \frac{sup_3^4[X,Y]}{\ldots}$$

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

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^1[X,Y]}{}$$

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

$$\frac{sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^2[X,X1]}{\begin{array}{l}a\ e\ \mathbf{3}\\ a\ f\ \mathbf{3}\end{array}} \qquad \frac{sup_2^2[X,Y1]}{a\ g} \qquad \frac{sup_3^2[X,Y]}{a\ b}$$

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

$$\frac{sup_0^3[Y]}{\begin{array}{l}e\\ f\end{array}} \qquad \frac{sup_1^3[X,Y]}{g\ f}$$

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

$$\frac{sup_0^4[Y]}{\begin{array}{l}e\\ f\end{array}} \qquad \frac{sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\\ f\ m\end{array}} \qquad \frac{sup_2^4[Y,X1]}{\ldots} \qquad \frac{sup_3^4[X,Y]}{\ldots}$$

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

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^1[X,Y]}{}$$

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

$$\frac{sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^2[X,X1]}{\begin{array}{l}a\ e\ \mathbf{3}\\ a\ f\ \mathbf{3}\end{array}} \qquad \frac{sup_2^2[X,Y1]}{a\ g} \qquad \frac{sup_3^2[X,Y]}{a\ b}$$

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

$$\frac{sup_0^3[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^3[X,Y]}{g\ f}$$

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

$$\frac{sup_0^4[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\\ f\ m\end{array}} \qquad \frac{sup_2^4[Y,X1]}{\ldots} \qquad \frac{sup_3^4[X,Y]}{\ldots}$$

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

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{\uparrow sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{\uparrow sup_1^1[X,Y]}{}$$

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

$$\frac{\uparrow sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{\uparrow sup_1^2[X,X1]}{\begin{array}{l}a\ e\ \mathbf{3}\\ a\ f\ \mathbf{3}\end{array}} \qquad \frac{\uparrow sup_2^2[X,Y1]}{a\ g} \qquad \frac{\uparrow sup_3^2[X,Y]}{a\ b}$$

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

$$\frac{\uparrow sup_0^3[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{\uparrow sup_1^3[X,Y]}{g\ f\ \mathbf{6}}$$

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

$$\frac{\uparrow sup_0^4[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{\uparrow sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\\ f\ m\end{array}} \qquad \frac{\uparrow sup_2^4[Y,X1]}{\ldots} \qquad \frac{\uparrow sup_3^4[X,Y]}{\ldots}$$

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

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{\uparrow sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{\uparrow sup_1^1[X,Y]}{}$$

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

$$\frac{\uparrow sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{\uparrow sup_1^2[X,X1]}{\begin{array}{l}a\ e\ \mathbf{3}\\ a\ f\ \mathbf{3}\end{array}} \qquad \frac{\uparrow sup_2^2[X,Y1]}{a\ g} \qquad \frac{\uparrow sup_3^2[X,Y]}{a\ b}$$

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

$$\frac{\uparrow sup_0^3[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{\uparrow sup_1^3[X,Y]}{g\ f\ \mathbf{6}}$$

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

$$\frac{\uparrow sup_0^4[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{\uparrow sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\ \mathbf{7}\\ f\ m\ \mathbf{7}\end{array}} \qquad \frac{\uparrow sup_2^4[Y,X1]}{\ldots} \qquad \frac{\uparrow sup_3^4[X,Y]}{\ldots}$$

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

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^1[X,Y]}{}$$

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

$$\frac{sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^2[X,X1]}{\begin{array}{l}a\ e\ \mathbf{3}\\ a\ f\ \mathbf{3}\end{array}} \qquad \frac{sup_2^2[X,Y1]}{a\ g} \qquad \frac{sup_3^2[X,Y]}{a\ b}$$

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

$$\frac{sup_0^3[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^3[X,Y]}{g\ f\ \mathbf{6}}$$

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

$$\frac{sup_0^4[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\ \mathbf{7}\\ f\ m\ \mathbf{7}\end{array}} \qquad \frac{sup_2^4[Y,X1]}{\ldots} \qquad \frac{sup_3^4[X,Y]}{\ldots}$$

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

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^1[X,Y]}{}$$

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

$$\frac{sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^2[X,X1]}{\begin{array}{l}a\ e\ \mathbf{3}\\ a\ f\ \mathbf{3}\end{array}} \qquad \frac{sup_2^2[X,Y1]}{a\ g\ \mathbf{9}} \qquad \frac{sup_3^2[X,Y]}{a\ b}$$

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

$$\frac{sup_0^3[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^3[X,Y]}{g\ f\ \mathbf{6}}$$

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

$$\frac{sup_0^4[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\ \mathbf{7}\\ f\ m\ \mathbf{7}\end{array}} \qquad \frac{sup_2^4[Y,X1]}{\ldots} \qquad \frac{sup_3^4[X,Y]}{\ldots}$$

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

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^1[X,Y]}{}$$

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

$$\frac{sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^2[X,X1]}{\begin{array}{l}a\ e\ \mathbf{3}\\ a\ f\ \mathbf{3}\end{array}} \qquad \frac{sup_2^2[X,Y1]}{a\ g\ \mathbf{9}} \qquad \frac{sup_3^2[X,Y]}{a\ b\ \mathbf{10}}$$

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

$$\frac{sup_0^3[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^3[X,Y]}{g\ f\ \mathbf{6}}$$

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

$$\frac{sup_0^4[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\ \mathbf{7}\\ f\ m\ \mathbf{8}\end{array}} \qquad \frac{sup_2^4[Y,X1]}{\ldots} \qquad \frac{sup_3^4[X,Y]}{\ldots}$$

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

QSQ – Example /2

Results after some steps of the QSQ approach

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

$$\frac{sup_0^1[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^1[X,Y]}{}$$

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

$$\frac{sup_0^2[X]}{a\ \mathbf{2}} \qquad \frac{sup_1^2[X,X1]}{\begin{array}{l}a\ e\ \mathbf{3}\\ a\ f\ \mathbf{3}\end{array}} \qquad \frac{sup_2^2[X,Y1]}{a\ g\ \mathbf{9}} \qquad \frac{sup_3^2[X,Y]}{a\ b\ \mathbf{10}}$$

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

$$\frac{sup_0^3[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^3[X,Y]}{g\ f\ \mathbf{6}}$$

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

$$\frac{sup_0^4[Y]}{\begin{array}{l}e\ \mathbf{5}\\ f\ \mathbf{5}\end{array}} \qquad \frac{sup_1^4[Y,Y1]}{\begin{array}{l}f\ l\ \mathbf{7}\\ f\ m\ \mathbf{8}\end{array}} \qquad \frac{sup_2^4[Y,X1]}{\ldots} \qquad \frac{sup_3^4[X,Y]}{\ldots}$$

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

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

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

$$rsg^{bf}(X,Y) \leftarrow supmagic_2^2(X,Y1), down(Y1,Y).$$
$$supmagic_2^2(X,Y1) \leftarrow supmagic_1^2(X,X1), rsg^{fb}(Y1,X1).$$
$$supmagic_1^2(X,X1) \leftarrow magic\_rsg^{bf}(X), up(X,X1).$$

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

$$rsg^{fb}(X,Y) \leftarrow supmagic_2^4(Y,X1), up(X,X1).$$
$$supmagic_2^4(Y,X1) \leftarrow supmagic_1^4(Y,Y1), rsg^{bf}(Y1,X1).$$
$$supmagic_1^4(Y,Y1) \leftarrow magic\_rsg^{fb}(Y), down(Y1,Y).$$

$$magic\_rsg^{bf}(a).$$
$$magic\_rsg^{bf}(X1) \leftarrow supmagic_1^4(X,X1).$$

$$magic\_rsg^{fb}(Y1) \leftarrow supmagic_1^2(Y,Y1).$$

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

Magic Sets – Generation

- The *magic* relations prevent generating *irrelevant* atoms by building "cones" starting from the query.
- The transformation algorithm is involved but comparativley easy to implement.
- In some cases, the transformation just creates overhead and no gain is achied. 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.