Sl 596

stratifiable program

$P = \{ \quad p(X) :- r(X) \land \neg q(X).$

$r(a), r(b). \; q(a). \quad \}$

$P_1 = \{ p(X) \leftarrow r(X) \land \neg q(X) \}$

$P_0 = \{ q(a), r(a), r(b) \}$

$I_0 = \emptyset$

$I_1 = \emptyset \cup T^{\omega}_{P_0}(\emptyset) = \{ q(a), r(a), r(b) \}$

$I_2 = I_1 \cup T^{\omega}_{P_1 \cup I_1}(\emptyset) = \{ p(b), q(a), r(a), r(b) \} = \mathfrak{M}(P)$

is a model, is minimal

Consider $\mathcal{H} = \{ q(a), r(a), r(b), q(b) \}$

- is a model
- minimal? try to remove $q(b)$.

$\mathcal{H}' = \{ q(a), r(a), r(b) \}$

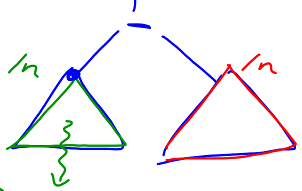
$\Rightarrow$ but this is not a model because the rule would then derive $p(b)$.

Sl 599

Consider an algebra tree like

$\pi[\dots]$

$\sigma[\dots]$



$P_{left} = \{ res_1(X_1, \dots X_n) :- \dots\dots , \text{ more rules } \}$

$P_{right} \{ res_2(X_1, \dots Y_n) :- \dots\dots , \text{ more rules } \}$

r_1: $tmp(X_1 \dots X_n) :- res_1(X_1 \dots X_n), \boxed{\neg res_2(X_1 \dots X_n),}$ $\sigma(X_1 \dots X_n).$ Stratific.

r_2: $res(Y_1, \dots Y_n) :- tmp(\_, Y_1 \_, \dots Y_2, \dots Y_n).$

Stratification

$P_{top} := \{ r_1, r_2 \}$

$P_{right} \to P_{top-1} \rightsquigarrow res_2 :- \dots$

P_0

$P_{left} \to P_{top}, P_{top-1} \rightsquigarrow res_1 :- \dots$

P_0

SL 600 : Stratification

P_2 : { Result(o) :- . . . . }

P_1 : { notResult(O) :- . . . . }

P_0 : base facts ∪ { orgOnCont :- . . . }

```
select abbreviation
from organization o
where not exists
 (select *
  from continent c
  where not exists
   (select  *
    from ismember i, encompasses e
    where i.country=e.country
     and e.continent = c.name
     and i.organization = o.abbreviation))
```

Ex. 5 : grouping + aggregation

→ as an algebra op $\pi[\text{spec}]/1$, $\sigma[\text{spec}]/1$, $\bowtie/2$

→ as a Datalog expr

→ ⋮

Semantics :

– an operation table → table

select . . . . <agg> → colname . . . .

group by <columnnames>

$\gamma[\text{spec}]/1$

$\gamma[\{col_1, \ldots col_n\}, \{$ (all group-by-cols), $\{newcol_1 := aggOp(tuples), \ldots, newcol_k := aggOp(tuples)\} \}]$

Ex : sum of citypop by country :

Algebra :

output: 2ary table/predicate

$\gamma[\text{country}, \text{sum} := \text{sum}(\text{population})]$
|
city

Rel calc :

res(C, S) :- country(_, N, C, _, _, _, _),
S is sum{ $P_{[C]}$ | city(_, C, _, P, _, _, _) }

grp-by-var.