Ex 1, Sheet 4

Algebra

Algebra || Algebra + tc

Difference — Stratified Datalog¬

Union $\cup$
Projection π
Selection σ
(~~Renaming~~ ρ)
Join $\bowtie$

positive Datalog

can be ignored

recursion

Union $u = p \cup q$
$\Rightarrow$ p and q must have the same format

$u(X_1 \dots X_n) \text{ :- } p(X_1 \dots X_n).$
$u(X_1 \dots X_n) \text{ :- } q(X_1, \dots X_n).$

Difference $d = p \setminus q$ $\Rightarrow$ p, q have the same format

$d(X_1 \dots X_n) \text{ :- } p(X_1 \dots X_n), \neg q(X_1 \dots X_n).$ ← Datalog¬

is this rule safe? – yes: all vars in $\neg q(\dots)$ also occur positively in p

Projection $pr(X_{i1} \dots X_{ik}) = \pi[X_{i1} \dots X_{ik}]\, p(X_1 \dots X_n)$

$pr(X_{i1} \dots X_{ik}) \text{ :- } p(X_1 \dots X_n).$

Selection $s(X_1 \dots X_n) = \sigma[\alpha](p(X_1 \dots X_n))$

$s(X_1 \dots X_n) \text{ :- } p(X_1 \dots X_n).\ \alpha \dots$ ← obvious for conjunctive cond, otherwise substitution....

Join $j(X_1 \dots X_m, X_{m+1} \dots X_n, Y_1 \dots Y_k) = p(X_1 \dots X_m, X_{m+1} \dots X_n) \bowtie q(X_1 \dots X_m, Y_1 \dots Y_k)$

$j(X_1 \dots X_m, X_{m+1} \dots X_n, Y_1 \dots Y_k) \text{ :- } p(X_1 \dots X_m, X_{m+1} \dots X_n), q(X_1 \dots X_m, Y_1 \dots Y_k).$

$\Rightarrow$ resulting Datalog¬-program is stratifiable

and program is nonrecursive, because each head atom represents a level in the algebra tree and uses only lower predicates

lower: wrt tree, wrt stratification

Jan 22-10:10

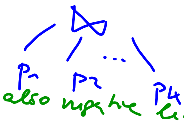
Ex 2

consider first a rule with only positive body literals:

$B(X_1 \dots X_n, Y_1 \dots Y_m, Z_1 \dots Z_k) \text{ :- } p_1(X_1 \dots X_n), p_2(Y_1 \dots Y_m), \dots, p_k(Z_1 \dots Z_k).$

omit Y that are among the X "

$\Rightarrow$ Algebra



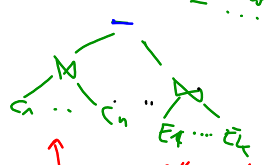
general case: also negative literals in the body

$B(\quad) \text{ :- } C_1 \dots C_n, \underbrace{D_1 \dots D_k}.$ rule is safe!

$\bowtie$ with $-$ negative literals

each $D_i = \neg E_i$

idea

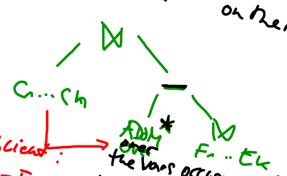


! for "$-$" both sides must have the same format

all vars/column names here must also occur

all vars occurring in the left side must also occur on the right side!

$\Rightarrow$



ADom* over the vars occurring in $E_1 \dots E_k = V$

sufficient: take $\pi[V](C_1 \bowtie \dots C_n)$ = Vars

same problem as for RANF

Jan 22-10:43

done : individual rules ✓

2b) other necessary construct .

two rules with same head

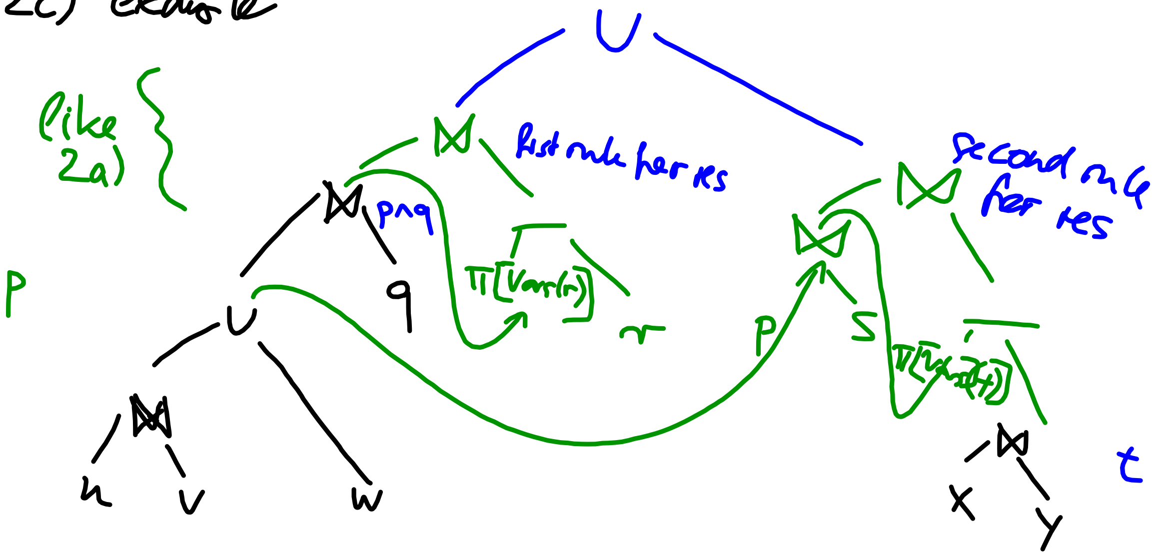
p :- body$_1$. ⇒ from a): body$_1$ equiv to ex$_1$

p :- body$_2$. body$_2$ equiv to ex$_2$

⇒ p is equiv to ex$_1$ ∪ ex$_2$

Jan 22-11:00

2c) example



Jan 22-11:03

Ex 3)

→ example: take the reachable/NotReachable prg from Slide 503

nonmonotonicity:

$\mathcal{S}$(P ∪ MondialDB) ⊨ reachable(D, P)

〃 ⊨ notReachable(D, GB)

$\mathcal{S}$(P ∪ Mondial ∪ {border(GB, F, 100)}) ⊨ reachable(D, GB)

⊭ notReachable(D, GB)

multiple minimal models:

$\mathcal{S}$(P ∪ MondialDB) is minimal

i.e. one cannot remove any atom to obtain a smaller model

$\mathcal{S}$(P ∪ Mondial) \ {notReachable(D, GB)} ∪ {reachable(D, GB)}

is also minimal, and it is a model!

⇒ it is not a "nice" model because reachable is a EDB predicate, and the derived "reachable(D, GB)" is not supported by the prg.

⇒ it is not well-founded!

⇒ the wellfounded semantics coincides with the stratified semantics for stratifiable Datalog¬ programs

Jan 22-11:13

Ex. 4 IDB predicate lang("CH", name, GS)

:- include(mondial). country(C, N, _Pop, _A, _Cap, _CapProv)

:- auto_table.

exA(C) :- language('CH', _X, _),
language(C, _X, _).

exB(C) :- country(C, _, _, _, _, _),
not exA(C).

c)

nonCHLgCountry(C) :- all countries, where some lg. is spoken which is not spoken in CH

lang(C, _L, _),
not lang('CH', _L, _).

... and we again need the remaining countries

onlyCHLgCtry(C) :- country(C, _, _, _, _, _),
not nonCHLgCountry(C).

d)

~~chlg(L) :- language('CH', L).~~

chLgMissing(C) :- if some swiss lg. is missing in C

language('CH', L, _), not lang(C, L, _), country(C, _ _ _ _ _)

exD(C) :- country(C, _ _ _ _ _),
not chLgMissing(C).

Jan 22-11:22

Jan 22-11:35