

Jun 8-14:03

2.1 for every algebra expression, a Datalog prog. can be given
 BY STRUCTURAL INDUCTION $e \rightarrow P_e$

Base cases:

$e = \text{name: } v \Rightarrow P_e = \{ \text{res}_e(X) :- X = v \}$

name
v

$e = \tau \begin{array}{c|c|c} \tau & & \tau \\ \hline c_1 & \dots & c_n \\ \hline \vdots & & \vdots \\ \hline \vdots & & \vdots \end{array} \Rightarrow P_e = \{ \text{res}(X_1, \dots, X_n) :- \tau(X_1, \dots, X_n) \}$

Jun 8-14:23

induction step: e is a complex expression

• $e = e_1 \cup e_2 \rightarrow P_{e_1}, P_{e_2}$ with n -ary result predicates
 $res_{e_1}(\dots), res_{e_2}(\dots)$

$$= P_e = P_{e_1} \cup P_{e_2} \cup \left\{ res_e(X_1 \dots X_n) :- res_{e_1}(X_1 \dots X_n), res_{e_2}(X_1 \dots X_n) \right\} \cup \left\{ res_e(X_1 \dots X_n) :- \neg res_{e_1}(X_1 \dots X_n), \neg res_{e_2}(X_1 \dots X_n) \right\}$$

• $e = e_1 \setminus e_2$

$$P_e = P_{e_1} \cup P_{e_2} \cup \left\{ res_e(X_1 \dots X_n) :- res_{e_1}(X_1 \dots X_n), \neg res_{e_2}(X_1 \dots X_n) \right\}$$

safe? yes

Jun 8-14:29

• $e = \pi[c_1 \dots c_k](e_1)$ e_1 Format $(c_1 \dots c_n)$
 $\pi[c_{i_1} \dots c_{i_k}](e_1)$ $i_1 \dots i_k \in \{1 \dots n\}$

$$P_e = P_{e_1} \cup \left\{ res_e(X_{i_1} \dots X_{i_k}) :- res_{e_1}(X_1 \dots X_n) \right\}$$

• $e = \delta[cond](e_1)$ ↙ conjunctive

$$P = P_{e_1} \cup \left\{ res_e(X_1 \dots X_n) :- res_{e_1}(X_1 \dots X_n), cond(X_1 \dots X_n) \right\}$$

if cond e.g. disjunctive: $e = \delta[cond_1]e_1 \cup \delta[cond_2]e_1$ and analogous...

• $e = \rho[c_{n_i} \rightarrow d_{n_i}](e_1)$

not applicable since Datalog is positional,
 no column names

Jun 8-14:35

$e = e_1 \bowtie e_2$ $e_1(A_1 \dots A_n)$ $e_2(A_{n+1} \dots A_{i+k}, B_{k+1} \dots B_{k+m})$
 $\{i \dots i+k\} \subseteq \{1 \dots n\}$
 $P_e = P_{e_1} \cup P_{e_2} \cup$
 $\{res_e(X_1 \dots X_n, Y_1 \dots Y_m) :-$
 $res_{e_1}(X_1 \dots X_n), res_{e_2}(X_{i+1} \dots X_{i+k}, Y_1 \dots Y_m)\}$
 (Note: $k \geq 0, m \geq 0$ in blue; $\{i \dots i+k\} \subseteq \{1 \dots n\}$ in green; res_{e_2} in red with "vars" written below it)

\Rightarrow prog is stratified,
 non-recursive

Jun 8-14:43

Ex. 2.2: Nonrecursive Stratified Datalog? \subseteq Algebra (safe!)

Individual rules: positive negative
 $B \leftarrow C_1 \wedge \dots \wedge C_m \wedge \neg D_1 \wedge \dots \wedge \neg D_n$

C_i, D_j atom $p_i(\dots)$
 and there is a program $P_1 \dots P_m$ for $C_1 \dots C_m$
 \Rightarrow algebra expr: $E_1 \dots E_m$
 Consider $C_1 \wedge \dots \wedge C_m$ (positive)
 $\hat{=} E_1 \wedge \dots \wedge E_m \supseteq$ result set
 $\hat{=} C$

2 possibilities:
 1) Consider one $D_j(X_{i_1} \dots X_{i_k})$ $\neg C_j$ D_j $\neg C_j \wedge D_j$
 (Note: $\neg C_j$ is circled in red, D_j is circled in green)

$\pi_j(C \wedge (\neg C_j) \wedge D_j) \wedge (\pi_j(C) \setminus \pi_j(D_j)) \wedge \dots \wedge \pi_j(C_m)$
 (Note: $\pi_j(C \wedge (\neg C_j) \wedge D_j)$ is circled in red, $\pi_j(C) \setminus \pi_j(D_j)$ is circled in green)

algebra expr $\hat{=} body$ of the rule
 \Rightarrow if head contains any subset of vars

2) $\pi_j((C - (C \wedge D_j)) - (C \wedge E_1) - \dots - (C \wedge E_m))$
 (Note: $(C - (C \wedge D_j))$ is circled in red, $(C \wedge E_1) - \dots - (C \wedge E_m)$ is circled in green)

More inductive of SQL
 Select from C
 where not exists... in D_1
 and not exists... in D_2
 and not exists in D_m

another algebra operators: $R_n \triangleright R_e$: "anti-join"
 $\pi_j(C \triangleright E_1 \triangleright E_2 \triangleright \dots \triangleright E_m)$

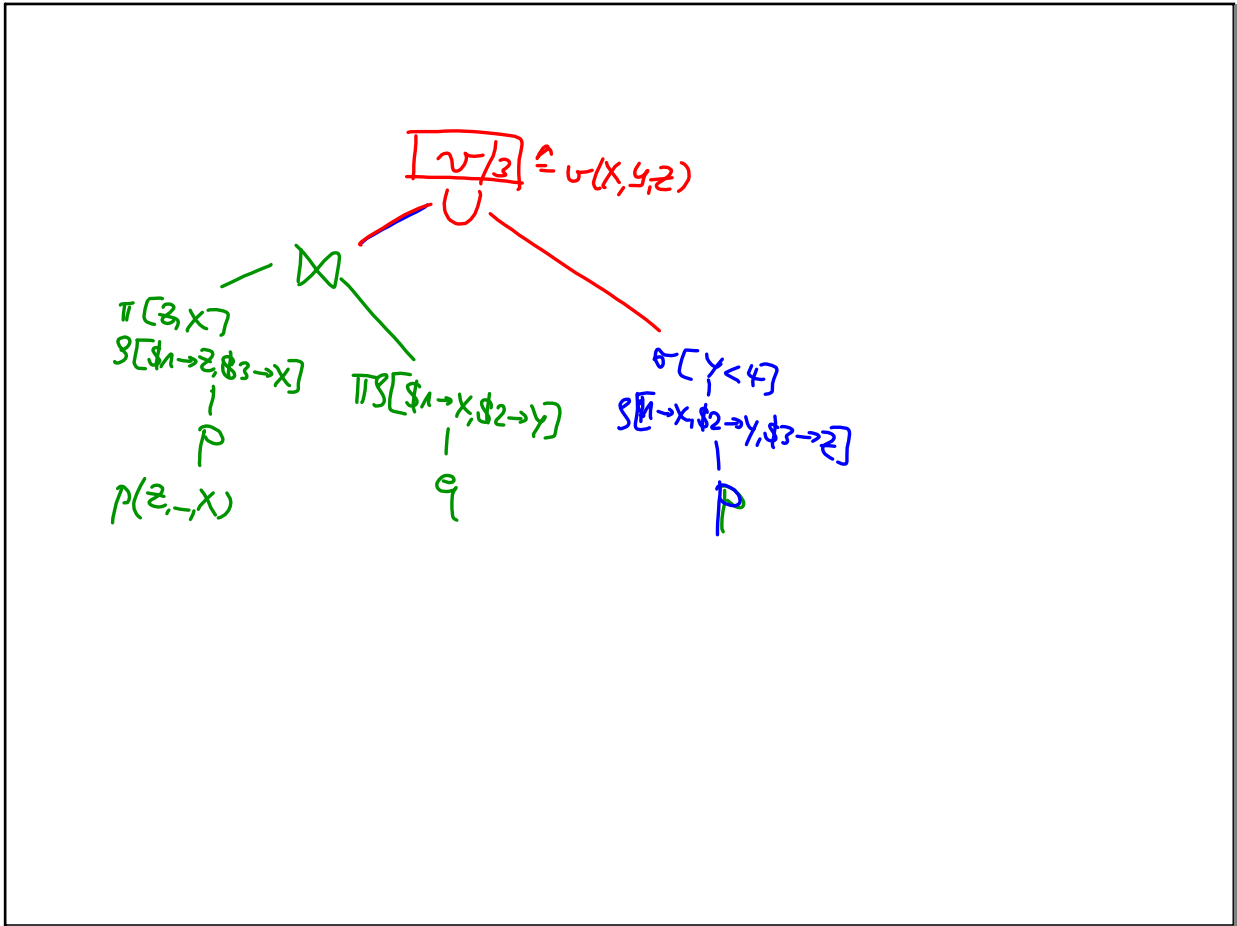
Jun 8-14:51

- individual Datalog rules (over relation symbols) ✓
- $B \leftarrow C_1 \dots C_n, \neg D_{n+1}, \dots, \neg D_{n+m}, \underbrace{X=Y, Y_2=a}_{\text{(safe) Comparisons}}$
 $\Rightarrow \sigma[X=Y], \sigma[Y_2=a] \dots$
- $P(\dots) \leftarrow \dots \rightarrow e_p$
- $P(\dots) \leftarrow \dots \rightarrow e_p \Rightarrow e_p \cup e_p$

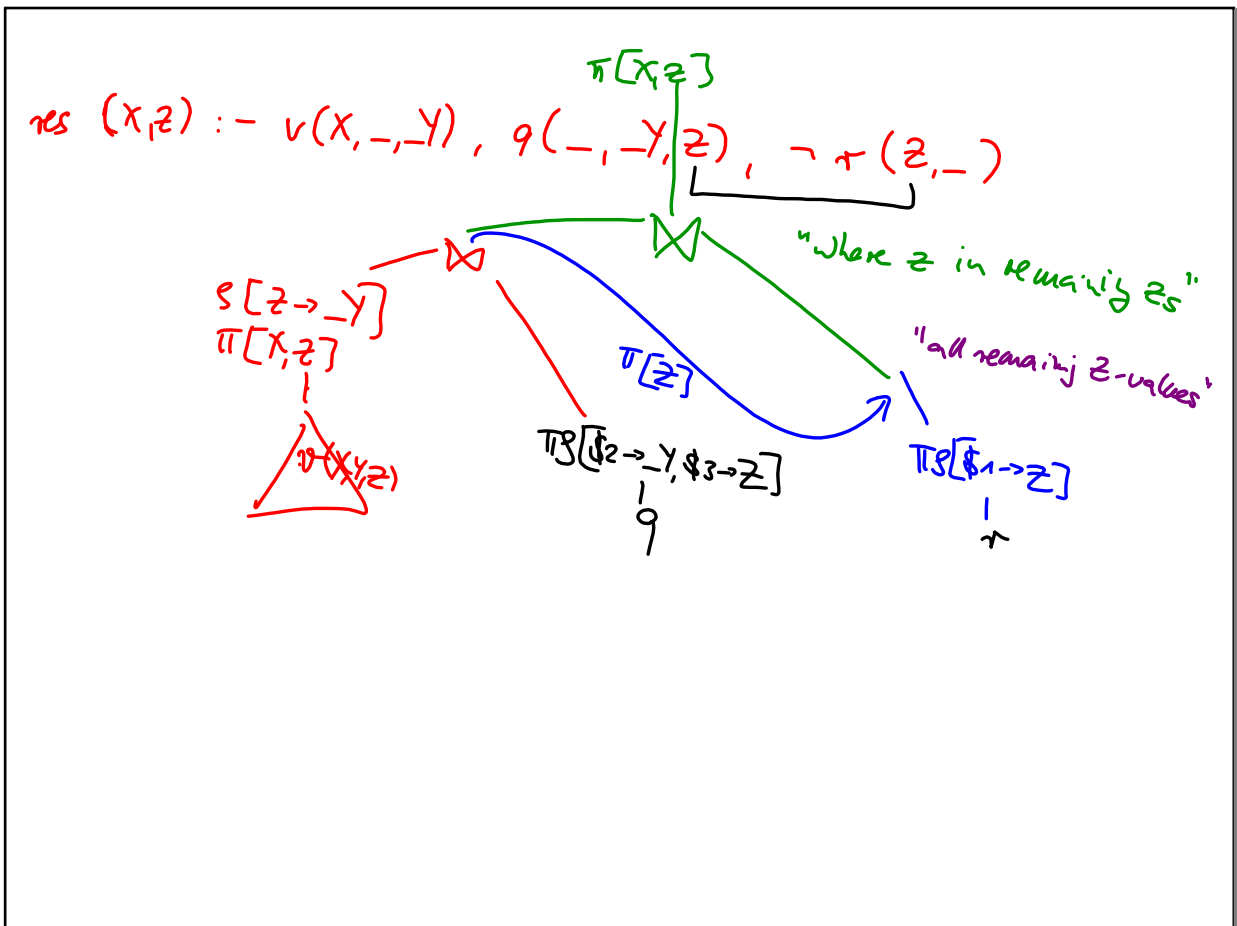
Jun 8-15:16

basic sketch

Jun 8-15:20

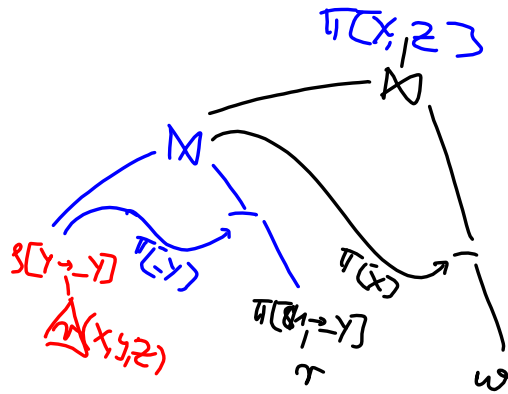


Jun 8-15:25



Jun 8-15:28

$res(x,z) :- v(x, _y, z), \neg r(_y, _), \neg w(x)$



... and (last page) \cup (this page)

Jun 8-15:35

Ex 2.5:
Ex 2.6:

Language (conts, lg, perc)

(no) coast : geo-sea (sea, conts, perc)

- borders borders(c1, c2, length)

Jun 8-15:41