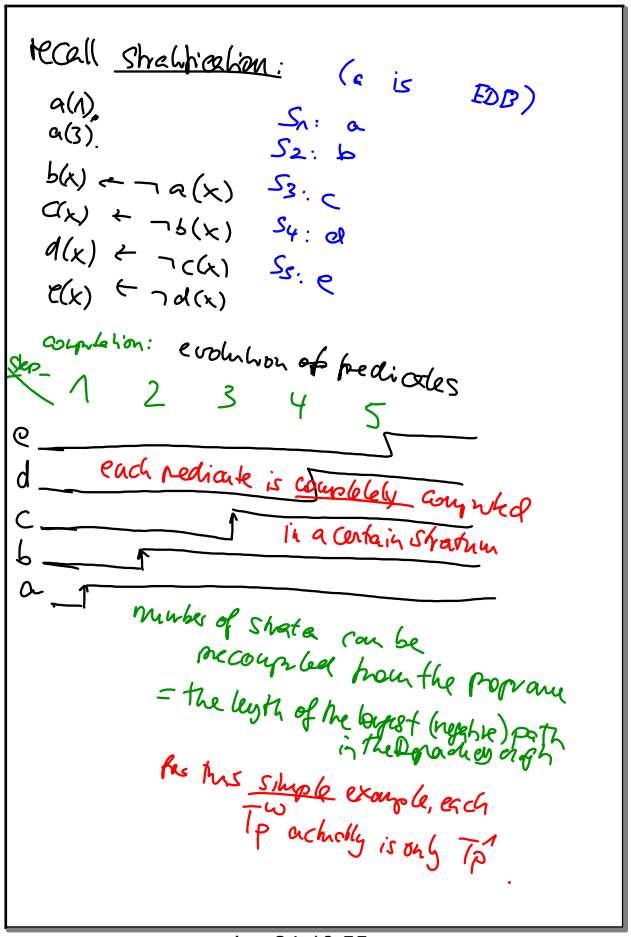
General case of rules:

positive head atom :- possible EDB atoms, negative EDB atoms, possible IDB atoms, negative IDB atoms.

Jun 24-10:09



Jun 24-10:55

recall positive programs:

a(1). a(2).

b(x) ← a(x).

c(x) ← b(x).

d(x) ← c(x).

computation $T_P \uparrow \omega$, in this simple example $= T_P^4$

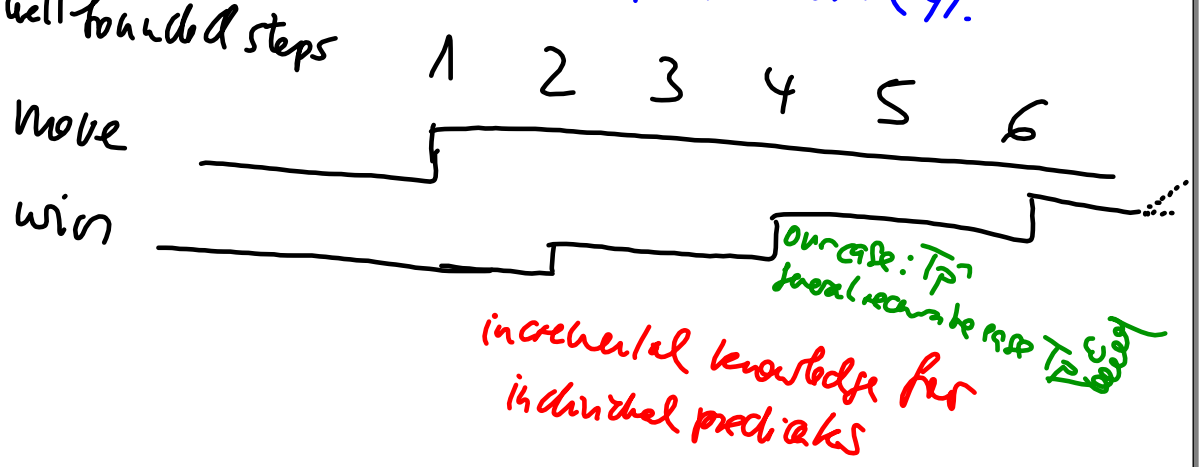
with a recursive rule, recall reachability (monadic)

$\Rightarrow T_P \uparrow \omega$ is necessary

Jun 24-11:03

So back to well-founded

win(X) :- move(X,Y), ¬win(Y).



Jun 24-11:06

Consider the following extended game prg:

$win(x) :- move(x,y), \neg win(y).$

$red(x) :- win(x), move(x,y), win(y).$

~~$seen(x) :- win(x), \neg \exists y (move(x,y), win(y))$~~ not in Datalog!

$\Rightarrow$ 2 T_P-steps in each well-founded round

Jun 24-11:11

Consider the reduct for that extended program

$\gamma_0 = \emptyset$; ie. $win = \emptyset$

$P = \{ move(a,f), move(b,k), \ldots\ldots\ldots,$ $\hat{=}$ rule `move(i;j) :- true.`
the two rules from the previous slide $\}$

$P_\emptyset = \{ move(a,f) :- true,$
$move(b,k) :- true,$
$win(a) :- move(a,b), \neg win(b).$
$win(c) :- move(a,f), \neg win(f).$
$win(i) :- move(b,a), \neg win(a).$
$win(i) :- move(b,k), \neg win(j).$
Rem the second rule:
$red(a) :- win(a), move(a,b), win(b).$
$\}$

Side-deletes:
- 2nd item does not delete anything

$\Rightarrow P_\emptyset$

Jun 24-11:23

$T^1_{P_{\text{game}}}(\emptyset) = \{move(a,b) \dots move(b,k), \dots\}$

only the moves = EDB

$T^2_{P_{\text{game}}}(\emptyset) = T_{P_{\text{game}}}(\downarrow) = \{move(a,b) \dots move(b,k), win(a), \dots win(b)\}$

$T^3_{P_{\text{game}}}(\emptyset) = \{\dots\} \cup \{red(a)\}$

$\Rightarrow$ we need $T_P \uparrow \omega$ here!

$=: \mathcal{H}_1$

Jun 24-11:33

Consider a differently extended program:

$win(X) :- move(X,Y), \neg win(Y).$

$lose(Y) :- \neg win(Y).$

Consider evaluation ....

$\mathcal{H}_1$: { win(a), win(b), win(i), lose(f), lose(k), lose(h), lose(j) } (obvious) agree

obvious!

Jun 24-11:43

reduct P_{χ_1} :

$P_g = \{$ move(...) ....

~~win(a) :- move(a,b), ¬win(b),~~

win(a) :- move(a,c), ~~¬win(c)~~,

~~win(e) :- move(e,a), ¬win(a).~~

win(c) :- move(c,d), ~~¬win(d)~~,

~~lose(a) :- not win(a).~~

lose(c) :- ~~not win(c)~~ true

lose(h) :- ~~not win(h)~~ true

lose(m) :- ~~not win(h)~~ true

$\}$ $=: P_{\chi_1}$

(! cover our intuition that e is not won)

a is not lost! because it is known to be won

Jun 24-11:44

evaluate P_{χ_1} :

$T_{P_{\chi_1}}(\emptyset)$

it will recreate everything that was safely correctly derived before

$= \{$ move(a,b) ..... (move ..)

lose(c), lose(m) lose(h) ....

$\}$

FRIDAY

Jun 24-11:52