General case of rules:

positiveheadatom :- possible EDB atoms, negative EDB atoms, positive IDB atoms, negative IDB atoms.

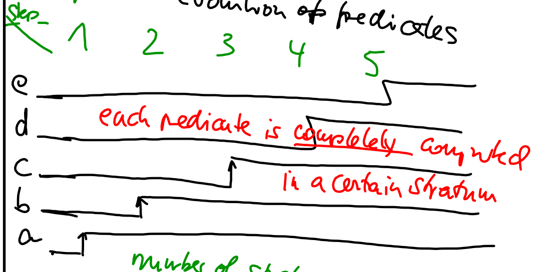
Jun 24-10:09

recall Stratification: (a is EDB)

a(1).
a(3).
b(x) ← ¬a(x)
c(x) ← ¬b(x)
d(x) ← ¬c(x)
e(x) ← ¬d(x)

S_1: a
S_2: b
S_3: c
S_4: d
S_5: e

computation: evolution of predicates



number of strata can be precomputed from the program
= the length of the longest (negative) path in the dependency graph

For this simple example, each T_P^{ω} actually is only T_P^1.

Jun 24-10:55

recall <u>positive</u> programs:

$a(1).\quad a(2).$

$b(X) \leftarrow a(X).$

$c(X) \leftarrow b(X).$

$d(X) \leftarrow c(X).$

computation T_P^{ω} , in this simple example $= T_P^4$

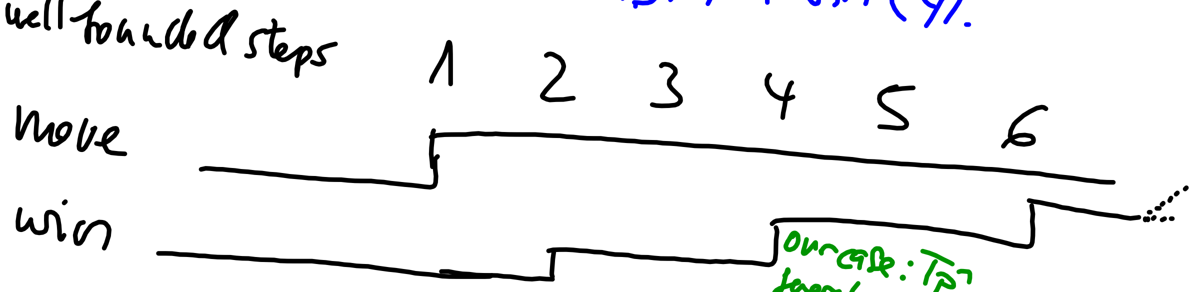
with <u>a recursive</u> rule, recall reachability (transitive closure)

$\Longrightarrow T_P^{\omega}$ is necessary

Jun 24-11:03

So back to <u>well founded</u>

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



our case: T_P^{7}

general recursive case T_P^{ε} [illegible]

incremental knowledge for individual predicates

Jun 24-11:06

Consider the following extended game prg:

win(X) :- move(X, Y), ¬win(Y).

red(X) :- win(X), move(X,Y), win(Y).

~~green(X) :- win(X), ¬∃Y:(move(X,Y), win(Y))~~

not in Datalog!

⇒ 2 T_P-steps in each well-founded round

Jun 24-11:11

Consider the reduct for that extended program

$\mathcal{X}_0 = \emptyset$; ie. win = $\emptyset$

P = { move(a,f), move(b,k), ... ≙ rule move(i,j) :- true.
the two rules from the previous slide }

P_g = { move(a,f) :- true,
move(b,k) :- true,
win(a) :- move(a,b), ~~¬win(b)~~.
win(a) :- move(a, f), ~~¬win(f)~~.
win(i) :- move(b,a), ~~¬win(a)~~.
win(i) :- move(b,k), ~~¬win(j)~~.
from the second rule:
red(a) :- win(a), move(a,b), win(b).
}

Side-deletes:
· 2nd item does not delete anything
⇒ $P_{\mathcal{X}_0}$

Jun 24-11:23

$$T^{1}_{P_{\mathcal{H}_0}}(\emptyset) = \{move(a,f) \dots move(b,k), \dots\}$$

only the moves = EDB

$$T^{2}_{P_{\mathcal{H}_0}}(\emptyset) = T_{P_{\mathcal{H}_0}}(\ \ ) = \{move(a,f) \dots move(b,k), win(a), \dots win(b)\}$$

$$T^{3}_{P_{\mathcal{H}_0}}(\emptyset) = \{\dots\} \cup \{\underline{red(a)}\}$$

$\Rightarrow$ we need T_P^{ω} here!

$\vdots$

$$=: \mathcal{H}_1$$

$\vdots$

Jun 24-11:33

Consider a differently extended program:

win(x) :- move(x,y), ¬win(y).

lose(y) :- ¬win(y).

consider evaluation:

let's guess some (incomplete) intermediate stage:

$\mathcal{H}_2^*$: { win(a), win(b), win(i),

lose(f), lose(k), lose(h), lose(j) }

obvious!

(obvious) agree

Jun 24-11:43

reduct $P_{\mathcal{H}_2}$:

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

~~win(a) :- move(a,b), ¬win(b)~~,
win(a) :- move(a,f), ~~¬win(f)~~,
win(b) :- move(b,k), ~~¬win(k)~~,
~~win(e) :- move(e,a), ¬win(a)~~, (! cover our intuition that e is not won)
⋮
win(c) :- move(c,d), ~~¬win(d)~~,
win(d) :- move(d,e), ~~¬win(e)~~,
~~lose(a) :- not win(a)~~, a is not lost! because it is known to be won
⋮
lose(c) :- ~~not win(c)~~ true
lose(f) :- ~~not win(f)~~ true
lose(h) :- ~~not win(h)~~ true
win(i) :- move(i,j), ~~not win(j)~~
lose(m) :- ~~not win(m)~~ true
} ⋮ $=: P_{\mathcal{H}_2}$

win(h) :- move(h,m), ~~not win(m)~~.
win(m) :- move(m,h), ~~not win(h)~~.

Jun 24-11:44

evaluate $P_{\mathcal{H}_2}$:

$T_{P_{\mathcal{H}_2}}(\cancel{\mathcal{H}_2}\ \emptyset)$ it will rederive everything that was safely correctly derived before

= { move(a,b) ..... move(...), win(h)?, win(m)? ?
win(a)(v) win(d)✓
lose(c)(?), win(c) (?) win(b)✓, win(i)✓ WHY is THIS ok?
lose(h)(?), lose(m) (?)
lose(f)✓, lose(k)✓, lose(n)✓, lose(j)✓
..... }

$T_{P_{\mathcal{H}_2}}$ again and again... no changes, fixpoint,

$=: \mathcal{H}_3^*$

see: $\mathcal{H}_3$ contains all (possible) atoms that we know that they must be true, but also some more.....

Jun 24-11:52

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

~~win(a) :- move(a,b), ¬win(b)~~,

win(a) :- move(a,f), ~~¬win(f)~~,

win(b) :- move(b,k), ~~¬win(k)~~,

~~win(e) :- move(e,a), ¬win(a)~~;

⋮

~~win(c) :- move(c,d), ¬win(d).~~

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

⋮

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

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

~~lose(b) :- not win(b)~~

win(i). move(i,j), ~~not win(j)~~.

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

⋮

$\} =: P_{\mathcal{X}_3}$

now copy

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

$T_{P_{\mathcal{X}_3}}(\mathcal{X}_3)$

Jun 26-14:08

$T_{P_{\mathcal{X}_3}}(\cancel{\emptyset}) = \{$ move(...), are all there

win(a) ✓, win(b) ✓, win(i) ✓, ...

lose(f)

⋮

$\}$

much less than before!

$= \mathcal{X}_4$

Jun 26-15:05

more abstract:

$\mathcal{H}_2^*$ was a small set of "guaranteed" positive atoms.

Construct the reduct:

- remove all rules such that they have a negative literal that is not satisfied
- remove all negative literals that are assumed to be satisfied by negation as default
  - -> for all "unknown things" the literals are removed
  - -> remove many negative atoms without actually checking them!

these rules denote their head

-> result is a large set of positive atoms $\mathcal{H}_2$

⋮

next set ($\mathcal{H}_3$) will be small

next set ($\mathcal{H}_4$) will be large

Jun 26-15:11

Consider the whole process:

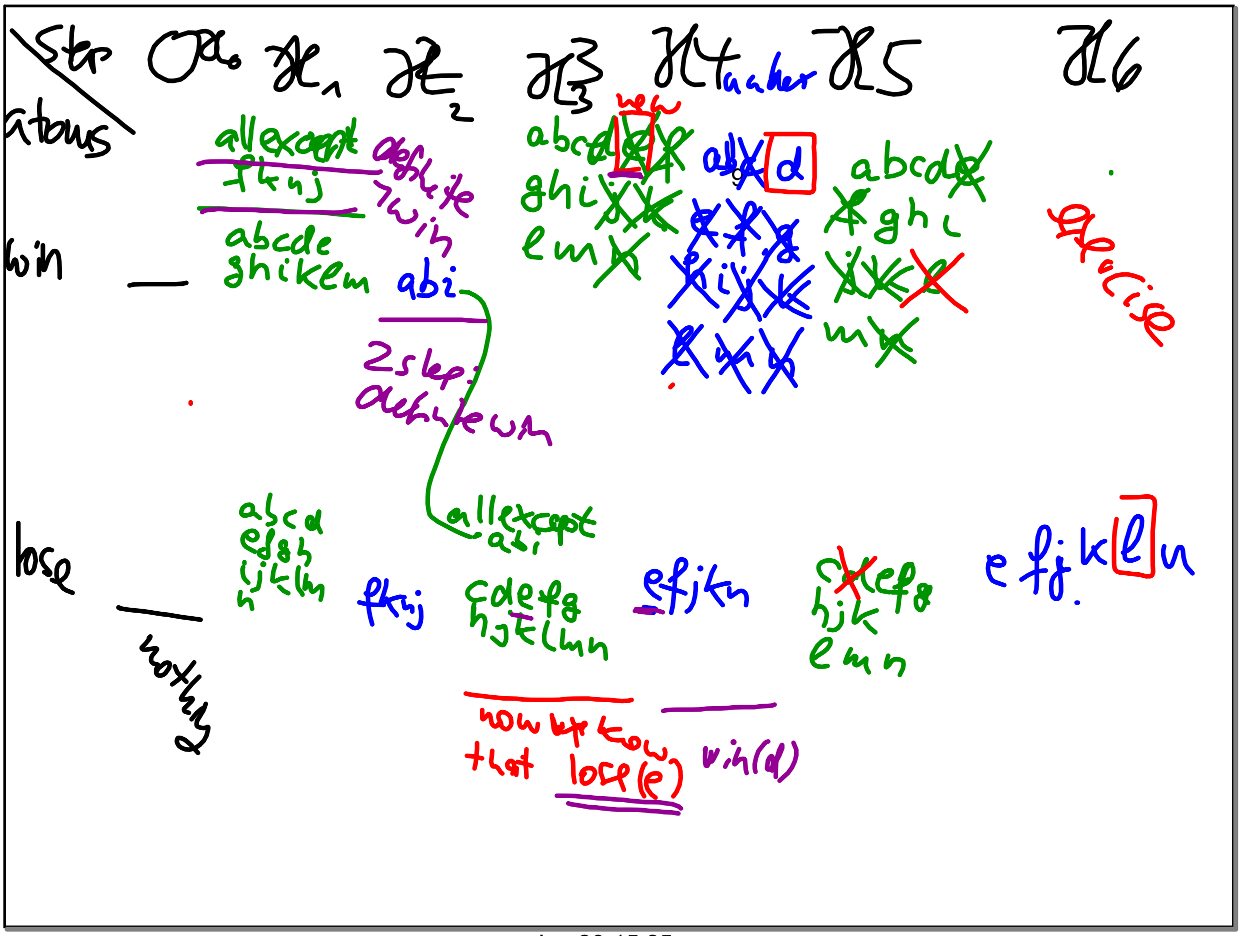
start with $\emptyset$

= a very small set of positive atoms

$T_{P_\emptyset}(\emptyset)$ is a large set $=: \mathcal{H}_1$

$T_{P_{\mathcal{H}_1}}(\emptyset)$ ... the $P_{\mathcal{H}_1}$ is smaller... $=: \mathcal{H}_2$ is small

Jun 26-15:20



Jun 26-15:25

1.7.2015:

what we saw before (previous slide)

- even ones: $\mathcal{H}_0, \mathcal{H}_2, \mathcal{H}_4 \ldots$

  increasing sequence of true atoms

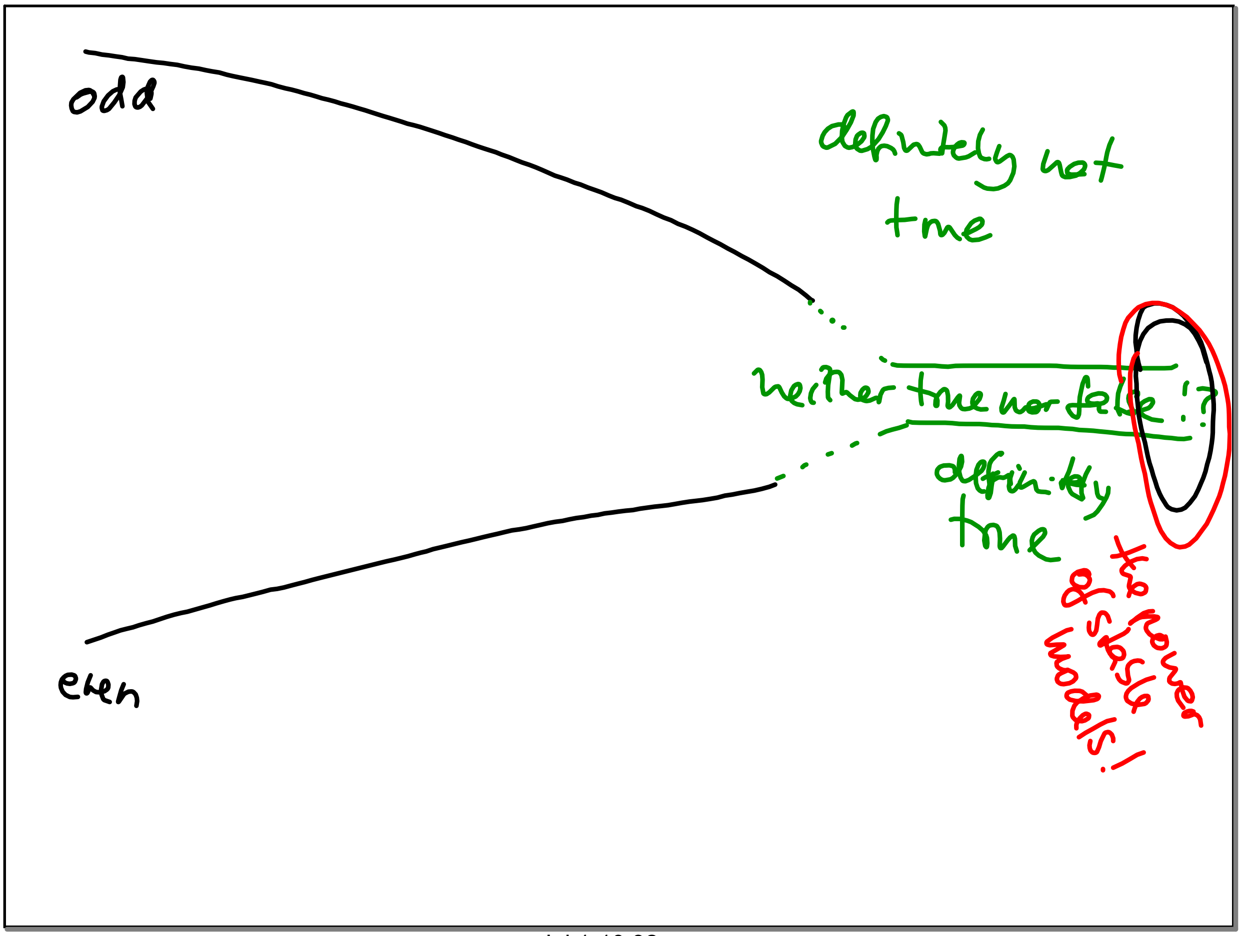
- odd ones: $\mathcal{H}_1, \mathcal{H}_3, \mathcal{H}_5$

  decreasing sequence of true atoms

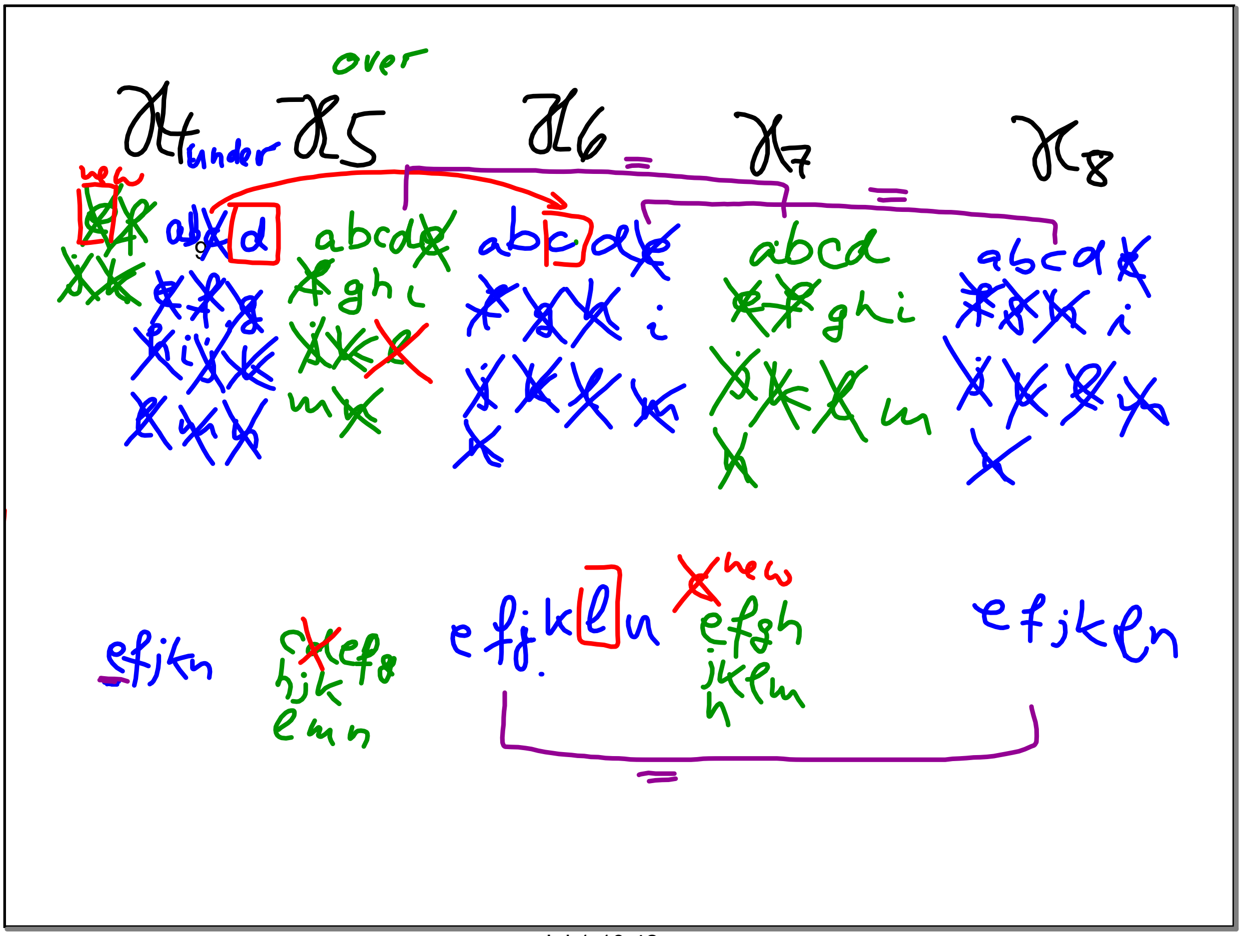
- the odd ones are always a superset of the even ones

  $\Rightarrow$

Jul 1-10:28



Jul 1-10:32



Jul 1-10:42

$\ldots \mathcal{H}_8 = \mathcal{H}_6$ $\quad = \mathcal{H}_{10} = \mathcal{H}_{12} \ldots$

$\Rightarrow P_{\mathcal{H}_8} = P_{\mathcal{H}_6}$

$\mathcal{H}_7 = T_{P_{\mathcal{H}_8}}(\emptyset) = T_{P_{\mathcal{H}_6}}(\emptyset) =: \mathcal{H}_9 \quad = \mathcal{H}_{11} = \mathcal{H}_{13} \ldots$

$\Rightarrow$ stop the process,

$\mathcal{H}_8$: underestimate of true atoms

$\cap|$

$\mathcal{H}_7$: overestimate of the atoms
underestimate of the false atoms

going back to well-founded argumentation:

$\mathcal{H}_8$: everything which is true in $\mathcal{H}_8$ is forced

$\mathcal{H}_6$: everything which is not true in $\mathcal{H}_7$ cannot be true in any "solution"

Jul 1-10:56

NOTE: $\mathcal{H}_8$ and $\mathcal{H}_7$ are

- $\mathcal{H}_7$ (overestimate) is a model, but in general not stable ("not supported")
  (here: lose(h) and lose(m))
- $\mathcal{H}_8$ (underestimate) is in general even NOT A MODEL
  (h and m neither win, nor lose)

$\Rightarrow$ "difference" up to interpretation of the result
human/semantic

Jul 1-11:02

- consider three-valued models:
  true, false, unknown
- other way: consider stable models
  → there are maybe several ones

Jul 1-11:07

consider what we did:

$\mathcal{H}_8 = T^{\omega}_{P_{\mathcal{H}_7}}(\emptyset)$

consider again $P \cup \{\text{redwin}(x) :- \text{move}(x,y), \text{win}(y), \text{win}(x)\}$,

then, the result will be a possible program that needs several T_P steps

$\mathcal{H}_7 = T^{\omega}_{P_{\mathcal{H}_6}}(\emptyset)$

$\vdots$

$\mathcal{H}_1 = T^{\omega}_{P_{\mathcal{H}_0}}(\emptyset)$

Jul 1-11:10

$$\mathcal{X}_8 = T^{\omega}_{P_{\left(T^{\omega}_{P_{\left(T^{\omega}_{P_{\cdots \left(T^{\omega}_{P_{\emptyset}}(\emptyset)\right)}}(\emptyset)\right)}}(\emptyset)\right)}}(\emptyset)$$

Jul 1-11:15

Slide 600, bottom:

positive program P, arbitrary $\mathcal{X}_0$:

notice $P_{\mathcal{X}_0} = P$

$$\rightsquigarrow \mathcal{X}_1 = T^{\omega}_{P_{\mathcal{X}_0}}(\emptyset) = T^{\omega}_{P}(\emptyset) \text{ is the minimal model}$$

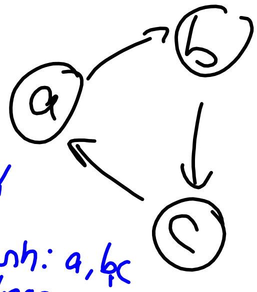
$$\parallel$$

$$\mathcal{X}_2 = T^{\omega}_{P_{\mathcal{X}_1}}(\emptyset) = T^{\omega}_{P}(\emptyset)$$

$$(\mathcal{X}_3 = \mathcal{X}_1)$$

Jul 1-11:24

win-move with 3-cycle



underestimate: $\emptyset$

overestimate: win: a, b, c
lose: a, b, c

$\overbrace{\text{no well}}^{\text{two valued}}$-founded-model

three-valued model $\mathcal{M}$:

$$\left(\overset{T}{\emptyset} \ , \ \overset{F}{\emptyset} \right)$$

$\mathcal{M}$: $val_{\mathcal{M}}(win(a)) = undef$,
lose(c):

Jul 1-11:43