Datenbanktheorie
Sommersemester 2012
Prof. Dr. W. May

5. Übungsblatt: Well-founded and Stable Semantics

Besprechung voraussichtlich am 12./13.7.2012

Aufgabe 1 (Well-Founded Model) a) Zeigen Sie dass es nicht-stratifizierbare Datalog$^\neg$-Programme gibt, die ein totales wohlfundiertes Modell (d.h., keine undefinierten Atome) haben.

b) Gibt es nicht-stratifizierbare Datalog$^\neg$-Programme, die für *jede* EDB-Instanz ein (eindeutiges) totales stabiles Modell besitzen?

---

a) Take a simple win-move game that has only won and lost positions, no drawn ones:

```
pos(a). pos(b). pos(c).
move(a,b).
move(b,c).
win(X) :-  move(X,Y), not win(Y).
```

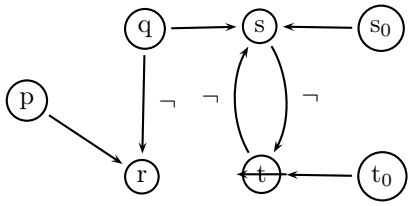
The well-founded model is

({pos(a). pos(b). pos(c). move(a,b). move(b,c). win(b)},
{move(a,a). move(a,c). move(b,a). move(b,a). move(c,a). move(c,b). move(c,c). win(a). win(c).})

b) Consider EDB relations $p/1$, $q/1$, $s_0/1$, $t_0/1$. The program P is as follows:

```
r(x) :- p(x), not q(x).
s(x) :- s0(x).
s(x) :- q(x), not t(x).
t(x) :- t0(x).
t(x) :- r(x), not s(x).
```

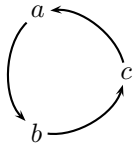
The dependency graph is



For each EDB instance that defines $I(p)$, $I(q)$, $I(s_0)$, $I(t_0)$, the well-founded model is total since either the $s \to t$ rule or the $t \to s$ rule body evaluate to false.

Aufgabe 2 (Well-Founded Model) Geben Sie eine Instanz des win-move-Games an, die kein totales stabiles Modell besitzt.

Ein Zyklus aus drei Knoten:



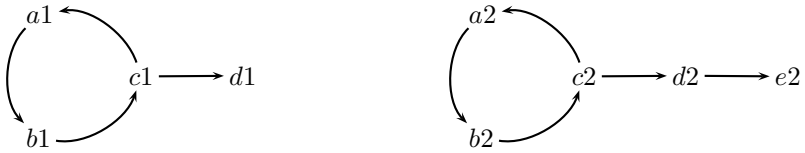
```
win(X) :-  move(X,Y), not win(Y).
lose(X) :- pos(X), not win(X).
pos(a).
pos(b).
pos(c).
move(a,b).
move(b,c).
move(c,a).
% lparse -n 0 -d none --partial winmovenontotal1.s |smodels
```

[Filename: winmovenontotal1.s]

Das einziges stabiles Modell ist M mit

$$val_M(win(a)) = val_M(win(b)) = val_M(win(b)) = u,$$
$$val_M(lose(a)) = val_M(lose(b)) = val_M(lose(b)) = u.$$

Bzw allgemein: ein Zyklus aus einer ungeraden Anzahl von Knoten, in dem kein Knoten aufgrund eines Ausweges aus dem Knoten "verloren" ist.




```
win(X) :-  move(X,Y), not win(Y).
lose(X) :- pos(X), not win(X).
pos(a1).
pos(b1).
pos(c1).
pos(d1).
move(a1,b1).
move(b1,c1).
move(c1,a1).
move(c1,d1).

pos(a2).
pos(b2).
pos(c2).
pos(d2).
pos(e2).
move(a2,b2).
move(b2,c2).
move(c2,a2).
move(c2,d2).
move(d2,e2).
% lparse -n 0 -d none --partial winmovenontotal2.s |smodels
```

[Filename: winmovenontotal2.s]

Im "1"-Spiel führt der "Ausgang" dazu, dass d_1 verloren ist, und damit und c_1 gewonnen ist (wer auf c_1 am Zug ist, zieht nach d_1). Damit ist b_1 verloren und a_1 gewonnen.

Im "2"-Spiel führt der "Ausgang" dazu, dass zwar e_1 verloren ist, und d_1 gewonnen ist, aber c_1 ist nicht verloren, da derjenige, der dort am Zug ist, nach a_1 ziehen wird, und der Spielverlauf den Zyklus nicht verlässt.