14/06/2017 EX2

lunedì 4 gennaio 2021 11:22

Exercise 2 Given the following ASP program P:

```
r(x,y) :- p1(y), p2(x).
s(x,y) :- q(x,y).
s(x,y) :- q(x,z), s(z,y).
t(x,y) :- r(x,y), not s(x,y).
t(x,y) :- s(x,y), not r(x,y).
v(x,y) :- t(x,y).
v(x,y) :- t(y,x).
w(x,y) :- v(x,y), not r(x,y).
p1(c). p1(d). p2(a). p2(b). p2(c).
q(a,b). q(b,c). q(c,d). q(d,e). q(e,d).
```

(a) tell whether P is stratified;

(b) compute the answer sets of P.

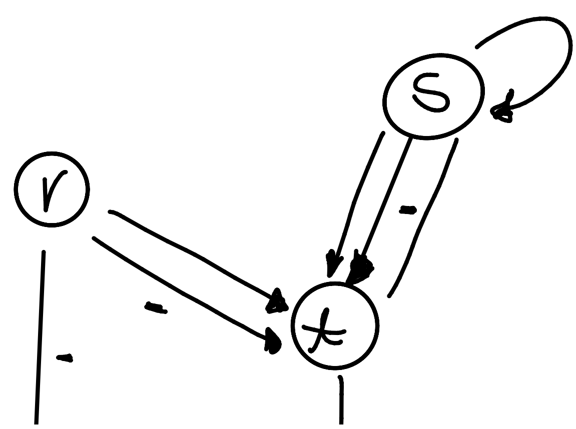
a)

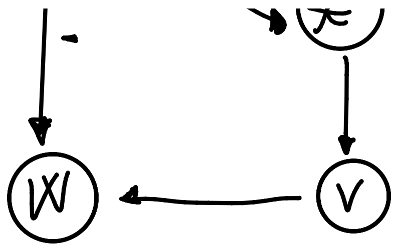
P is stratified if in the precedence graph of the program P there are no cycles with negated edges.

In particular, the precedence graph of program P is a graph composed by a set of vertices V, a set of edges E and L that it is labelling of the edges. It is composed in this way;

- There is a vertex for every IDB predicates in the Datalog program P
- There exists an edge (s,t) in E (without label) if there exist a rule in which s,t are IDB predicates and t appears as head and s appears as positive atom in the body of the rule
- There exists a negated edge (s,t) in E with label "-" if there exist a rule in which s,t are IDB predicates and t appears as head and s appears as negated atom in the body of the rule.

IDB = {r/2, s/2, t/2, v/2, w/2}
EDB = {p1/1, p2/1, q/2}



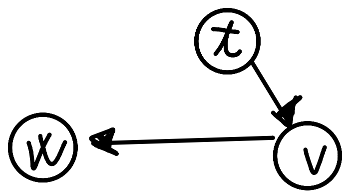


We can see that the precedence graph has no cycles with negated edge, the only cycle that it has, is the positive cycle on "s".

b)

We know that if a program is stratified we have an unique answer set, and this unique answer set is exaclty the minimal model of the program.
To compute the minimal model we need to consider the stratification of a stratified program.
A stratification return a set of stratum in which we compute this stratum looking at the precedence graph. At each iteration we consider only vertices that has no negative dependency ans we take in out stratum, after that we consider these vertices as EDB predicates and so we remove from the graph and also we remove their edges. We repeat this untile the graph is empty.
A vertices v has negative dependency if there exists another vertices v' and there exists a path from v' to v that passing through a negated edge.

In our case



S1 = {r, s}

S2 = {t,v,w}

Ok, now we are to compute the minimal model.
But first we need to say that stratification is very important because in this way when we compute the MM(P) we only take care about IDB predicates of the particular stratum that we take. For instance, in S1 we have only r and s IDB predicates, so in that iteration to compute the MM(P) we only consider rules that have as head r and s and this means that in the body we have only negated atom that are EDB predicate because we have already compute all of them in the previous strata and this means also the negation, so we can evaluate. It is not possible to have a negated IDB predicates because otherwise the program is not stratified.

We start with
MM0 = EDB(P) = {p1(c),p1(d), p2(a), p2(b), p2(c), q(a,b), q(b,c), q(c,d), q(d,e), q(e,d)}

P(S1){
    r(x,y) :- p1(y), p2(x).

```
    s(x,y) :- q(x,y).
    s(x,y) :- q(x,z), s(z,y).
}
```

I= MM0

I' = Tp(I) = {r(a,c),r(b,c),r(c,c),r(a,d),r(b,d),r(c,d), s(a,b),s(b,c),s(d,e),s(c,d),s(e,d)}

Delta I' = {Delta'r(a,c),Delta'r(b,c),Delta'r(c,c),Delta'r(a,d),Delta'r(b,d),Delta'r(c,d), Delta's(a,b),Delta's(b,c),Delta' s(c,d), Delta's(d,e),Delta's(e,d)}

```
Delta P ={
  Delta' s(x,y) :- q(x,z), Delta s(z,y).
}
```

1-iteration

I = I union {r(a,c),r(b,c),r(c,c),r(a,d),r(b,d),r(c,d), s(a,b),s(b,c), s(c,d), s(d,e),s(e,d)}

Delta I = {Delta r(a,c), Delta r(b,c), Delta r(c,c), Delta r(a,d), Delta r(b,d), Delta r(c,d), Delta s(a,b),Delta s(b,c),Delta s(c,d), Delta s(d,e),Delta s(e,d)}

Delta' I = {Delta' s(a,c), Delta' s(b,d), Delta' s(c,e), Delta' s(d,d), Delta' s(e,e)}

2-iteration

I = I union{s(a,c),s(b,d),s(c,e),s(d,d), s(e,e)}

Delta I = {Delta s(a,c), Delta s(b,d), Delta s(c,e), Delta s(d,d), Delta s(e,e)}

Delta' I = {Delta' s(a,d), Delta' s(b,e)}

3-iteration

I = I union {s(a,d), s(b,e)}

Delta I = {Delta s(a,d), Delta s(b,e)}

Delta' I = {Delta' s (a,e)}

4- iteration

I = I union {s(a,e)}

Delta I = {Delta s(a,e)}

Delta I' = {}

Stop

MM1 = MM0 union {r(a,c),r(b,c),r(c,c),r(a,d),r(b,d),r(c,d), s(a,b),s(b,c), s(c,d), s(d,e),s(e,d),s(a,c),s(b,d),s(c,e),s(d,d), s(e,e),s(a,d), s(b,e),s(a,e)}

```
P(S2) = {
    t(x,y) :- r(x,y), not s(x,y).
    t(x,y) :- s(x,y), not r(x,y).
    v(x,y) :- t(x,y).
    v(x,y) :- t(y,x).
    w(x,y) :- v(x,y), not r(x,y).
```

}

I = MM1
I' = Tp(I) = {t(c,c),t(d,e), t(e,d), t(c,e), t(d,d), t(e,e), t(b,e), t(a,e), t(a,b)}
Delta' I ={Delta' t(c,c), Delta' t(d,e),Delta' t(e,d),Delta' t(c,e),Delta' t(d,d),Delta' t(e,e),Delta' t(b,e),Delta' t(a,e), Delta' t(a,b)}

Delta P = {
    Delta' v(x,y) :- Delta t(x,y).
    Delta 'v(x,y) :- Delta t(y,x).
    Delta 'w(x,y) :- Delta v(x,y), not r(x,y).
    Delta 'w(x,y) :- v(x,y), Delta not r(x,y).
}

1-iteration
I = I union {t(c,c),t(d,e), t(e,d), t(c,e), t(d,d), t(e,e), t(b,e), t(a,e), t(a,b)}
Delta I = {Delta t(c,c), Delta t(d,e),Delta t(e,d),Delta t(c,e),Delta t(d,d),Delta t(e,e),Delta t(b,e),Delta t(a,e), Delta t(a,b)}
Delta' I = Tp(I union Delta I) = {Delta' v(c,c), Delta' v(d,e),Delta' v(e,d),Delta' v(c,e),Delta' v(d,d),Delta' v(e,e),Delta' v(b,e),Delta' v(a,e), Delta' v(e,c), Delta' v(e,b), Delta' v(e,a), Delta' v(a,b), v(b,a)}

2-iteration
I = I union {v(c,c), v(d,e),v(e,d), v(c,e),v(d,d), v(e,e), v(b,e), v(a,e), v(e,c), v(e,b), v(e,a), v(a,b), v(b,a)}
Delta I = {Delta v(c,c), Delta v(d,e),Delta v(e,d),Delta v(c,e),Delta v(d,d),Delta v(e,e),Delta v(b,e),Delta v(a,e), Delta v(e,c), Delta v(e,b), Delta v(e,a), Delta v(a,b), Delta v(b,a)}
Delta' I = {Delta' w(d,e), Delta' w(e,d), Delta' w(c,e), Delta 'w(d,d), Delta' w(e,e), Delta'w(b,e), Delta'w(a,e), Delta'w(e,c), Delta' w(e,b), Delta' w(e,a), Delta' w(a,b), Delta' w(b,a)}

3-iteration
I = I union {w(d,e), w(e,d), w(c,e), w(d,d), w(e,e), w(b,e), w(a,e), w(e,c), w(e,b), w(e,a), w(a,b), w(b,a)}
Delta I = {Delta w(d,e), Delta w(e,d), Delta w(c,e), Delta w(d,d), Delta w(e,e), Delta w(b,e), Delta w(a,e), Delta w(e,c), Delta w(e,b), Delta w(e,a), Delta w(a,b), Delta w(b,a)}
Delta'I = {}

Stop

MM2 = MM1 union {t(c,c),t(d,e), t(e,d), t(c,e), t(d,d), t(e,e), t(b,e), t(a,e),v(c,c), v(d,e),v(e,d), v(c,e),v(d,d), v(e,e), v(b,e), v(a,e), v(e,c), v(e,b), v(e,a), w(d,e), w(e,d), w(c,e), w(d,d), w(e,e), w(b,e), w(a,e), w(e,c), w(e,b), w(e,a), w(a,b), w(b,a)}

MM(P) = {p1(c),p1(d), p2(a), p2(b), p2(c), q(a,b), q(b,c), q(c,d), q(d,e), q(e,d), r(a,c),r(b,c),r(c,c),r(a,d),r(b,d),r(c,d), s(a,b),s(b,c), s(c,d), s(d,e),s(e,d),s(a,c),s(b,d),s(c,e),s(d,d), s(e,e),s(a,d), s(b,e),s(a,e), t(c,c),t(d,e), t(e,d), t(c,e), t(d,d), t(e,e), t(b,e), t(a,e),v(c,c), v(d,e),v(e,d), v(c,e),v(d,d), v(e,e), v(b,e), v(a,e), v(e,c), v(e,b), v(e,a), w(d,e), w(e,d), w(c,e), w(d,d), w(e,e), w(b,e), w(a,e), w(e,c), w(e,b), w(e,a), w(a,b), w(b,a)}