EXERCISE 11

Andrea Crotti (#299466), Can Liu (#286105), Sebastian Roidl (#281941)

20 Gennaio 2010

Exercise 11.1

Datalog program D without negation, prove the following:

1. If M1 and M2 are models ofD,then $M1 \bigcap M2$ is also a model of D.

We assume M1 is not a model. Then there is a ground rule r: A
leftarrow A1, A2,...An, which is false. A and Ai are positive literals. r is
false means that Ai is in $M1 \bigcap M2$ while A is not.
So we get $Ai \in M1$ and $Ai \in M2$.
Since M1 is a model, A should also be in M1. So the rule r is true in M1.
The same we get $A \in M2$ too. There is a contradiction with the assumption
that $A \notin M1 \bigcap M2$, which means $M1 \bigcap M2$ is a model.

2. There exists a unique minimal 2 model of D (we call it the least model).

We assume there are more than one minimal models, Let's say $M1M2$.
According to 1, we get $M = M1 \bigcap M2$ is also a model. So we could get that
$M \subset M1$ or $M \subset M2$ since $M1M2$. Thus M1 or M2 is not minimal. There
is a contradiction.

3. Each fixpoint of TD^3 is a model. Give an example that the converse does not hold.

We assume F is a fixpoint, but not a model. So TD(F) = F and there is a
ground rule r: $A \leftarrow A1, A2, ...An$, that is false in F. Then $F \vDash Ai \land F \nvDash A$.
Consider TD(F) and r, by definition of immediate consequence operator we
get that $TD(F) \vDash A$. $TD(F)F$, this is a contradiction.

There are some models which are not fixpoints. For example, we have the facts $F : q(a,b), q(b,a), t(b)$ and rule R: $p(X) \leftarrow q(X,Y), t(Y)$. We could have a model $M : q(a,b), q(b,a), q(b,b), t(b), p(a), p(b)$.
This model is not a fixpoint because fixpoint is generated by recursively calculation of the facts by the rule.
And q(b,b) is a component of the Herbrand base, but can not be generated by the current F and R.

4. If $TD(M) \subset M$, then M is a model.

We assume M is not a model. It indicates there is a rule r: $A \leftarrow A1, A2, ...An$, which is false.
This means $A \notin M$ while $Ai \in M$. According to the immediate consequence, A must be in TD(M). This leads to TD(M)is not a subset of M, contradiction.

5. If M is a model, then $TD(M) \subset M$.

We assume TD(M) is not a subset of M.
Then there must exists an f which is generated by a rule, but not in M. This means this rule r: $A \leftarrow A1, A2, ...An$, is false. Then by definition, M is not a Model. This is a contradiction.

6. If M is a minimal model, then M is a fixpoint.

We assume that M is not a fixpoint. So there may exist some facts which are not generated by TD(S). If we delete any of these facts, M may still be a model according to the definition. So M is not the minimal model. This is the contradiction.

7. The least model is equal to the least fixpoint, i.e., $MD = lfp(TD)$.

According to 3, lfp(TD) is a model. According to 6, the minimal model M is also a fixpoint. We assume M is not the least fixpoint. Then there exists a M', which is also a model, but it is a smaller fixpoint. Because of the monotony of TD, we can see that M' is contained in M. So M is not the least model. This leads to contradiction.

Exercise 11.2

Given the following Datalog program and the query q(X) : rel(12,X).

```
%% Rule system
rel(X,U) :- link(X,U).
rel(X,U) :- link(X,Z),rel(Z,U).
tie(X,Y) :- rel(X,U),rel(Y,U),x<>y.
%% Facts
link(1,12) link(2,12) link(2,23) link(3,23) link(12,127) link(7,127) link(0,1)
```

1. Evaluate the query using semi-naive bottom-up approach.

We generate also all implicit facts that are not useful.
rel (1,12), rel (1,127), rel (2,12), rel (2,127), rel (2,23), rel (3,23), rel (12, 127), rel (7,127), rel (0,1), rel (0,12), rel (0,127).
tie (1,2), tie (0,1), tie (0,2), tie (2,1), tie (1,0), tie (2,0), (12,7), tie (7,12), tie (0,7), tie (7,0), tie (1,7), tie (7,1), tie (2,7), tie (7,2), tie (2,3), tie (3,2).
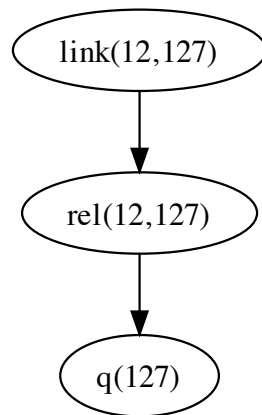
2. Evaluate the query using top-down approach. Draw the proof tree

This tree tries to get to a fact starting from the query and going down using unification. When we reach a fact we find a result for the starting query.

```
                        ┌──────────────┐
                        │   rel(12,X)  │
                        └──────────────┘
                       ╱                ╲
                      ╱                  ╲
         ┌───────────────────────────┐   ┌──────────────────────┐
         │ rel(12, U) :- link(12, U) │   │  link(12,Z), rel(Z,X) │
         │         [X/12]            │   └──────────────────────┘
         └───────────────────────────┘              │
                      │                              │
         ┌───────────────────────────┐   ┌─────────────────────────────────────┐
         │  link(12,127) [U/127]     │   │ link(12,127), rel(127,X)] [Z/127]    │
         └───────────────────────────┘   └─────────────────────────────────────┘
                      │                              │                    │
         ┌───────────────────────────┐   ┌─────────────────────────────────┐   │
         │   verified, so X = 127    │   │ link(127,Z), rel(Z,U)] [X/127]  │   │
         └───────────────────────────┘   └─────────────────────────────────┘   │
                                                    │                           │
                                          ┌────────────────────────────────────────┐
                                          │ dead branch, no more unification possible │
                                          └────────────────────────────────────────┘
```
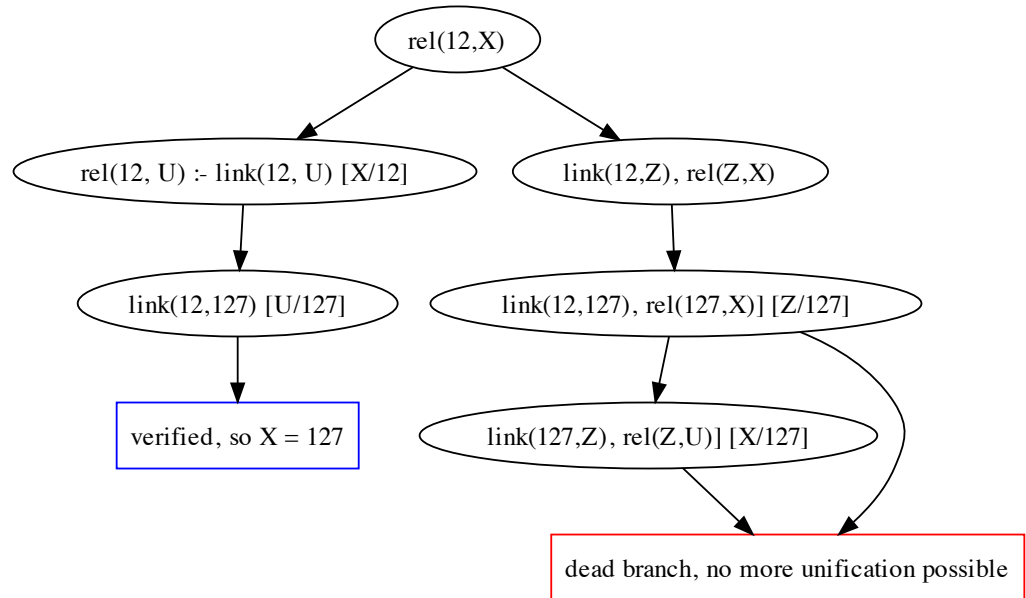
3. What are the pros and cons of top-down evaluation and bottom-up evaluation?

Top-down evaluation starts from the query and try to reach a fact constructing a tree, and using backtracking until all the possible branches have been explored. It unifies variables at every step and it's very efficient if the query is well posed, but it could be very inefficient (or not terminate) in case some subtrees don't lead to a leaf.

Bottom up evaluation instead generates will try a finite number of algebraic steps to derive the goal. This will always terminate but in this process it will generate a large amount of implicit facts that could be avoided.