Foundations of Logic Programming
Tutorial 4 (on January 5th)

Sarah Gaggl

WS 2014/15

Exercise 4.1:
Show with the help of the Prolog tree how the cut is used in the following program,

```
r(a).
r(b).
 q(a) ← r(X), !, p(a).
q(f(X)) ← r(X).
 p(X) ← r(X).
p(f(X)) ← q(X), !, r(X).
p(g(X)) ← r(X).
```

and where the query `?- p(X).` is taken. What would happen without the cut?

Exercise 4.2:
Take the following program P:

```
p ← .
p ← p.
q ← r.
q ← ¬r, p.
r ← ¬p.
t ← q.
t ← r,¬q.
```

a) Construct the dependency graph D_P of P.

b) Is P *stratified* and/or *hierarchical*?

c) Give a stratification of P.

d) Using your stratification to show how to compute the standard model M_P of P.

Exercise 4.3:
The built-in predicate `fail/0`, fails when Prolog encounters it as a goal. Thus, it can be viewed as an instruction for backtracking. On the other hand, the cut predicate `!`, blocks backtracking.

Define the predicate `neg/1` which allows you to express *negation as failure*.