CS3750 – FALL 1999 – MIDTERM EXAM

Malek Mouhoub
(10/25/99)

Exercise 1

Write a predicate reverse_lst(L1,L2) which reverses L1 in L2.

Example :
`reverse_lst([a,b,c,d],[d,c,b,a]).` $\rightarrow$ `true`
`reverse_lst([a,b,c],X).` $\rightarrow$ `X=[c,b,a]`

Solution

```
reverse_lst([],[]).
reverse_lst([H|L1],L2) :- reverse_lst(L1,L), append(L,[H],L2).
```

Exercise 2

Write a recursive predicate equi(L1,L2) which is true when the lists L1 and L2 differ only by the occurrences of the character $*$.

Example :
`equi([a,b,c,d],[*, a, *, b, c, d]).` $\rightarrow$ `true`
`equi([*, a, b, *,c],[a, *,b ,c ,*,*]).` $\rightarrow$ `true`
`equi([*,a,*,*,b,c],[a,b,c]).` $\rightarrow$ `true`
`equi([a,b,c],[a,*,d,b,c]).` $\rightarrow$ `false`

Consider the following cases :

- The two lists are empty.
- The two lists have the same head.
- The first list starts with an $*$.
- The second list starts with an $*$.

Solution

```
equi([],[]).\\
equi([H|T],[H1|T1]) :- equi(T,T1).
equi([*|T],L) :- equi(T,L).
equi(L,[*|T]) :- equi(L,T).
```

Exercise 3 : Derivation tree in PROLOG

Let's consider the following PROLOG program :

```
p(Y) :- q(X,Y), r(Y).
p(X) :- q(X,X).
q(X,X) :- s(X).
r(b).
s(a).
s(b).
```

1. Develop the derivation tree corresponding to the goal : `?- p(X).`
2. What's happen if we replace the first clause by :
   `p(Y) :- q(X,Y), !, r(Y).`

Solution

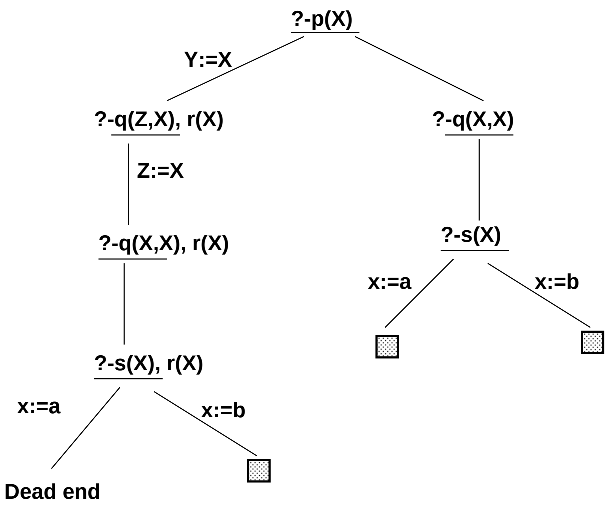

Figure 1: Derivation tree for question 1.

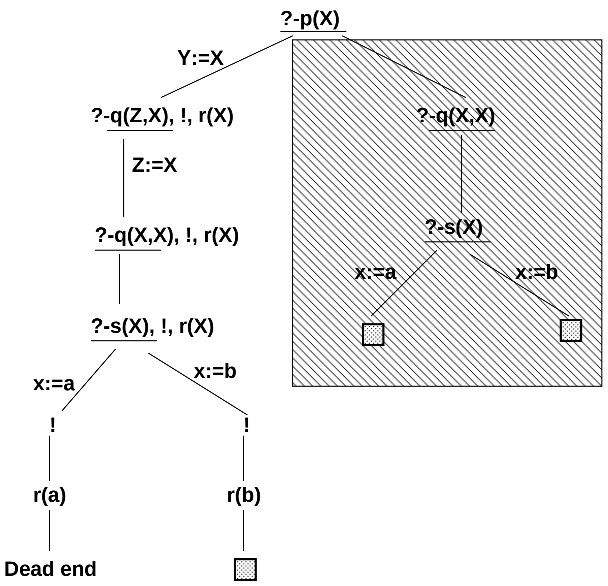

Figure 2: Derivation tree for question 2.