Logic ch 3.2

Sherri Shulman
TESC, CS

Mar 9, 2011

Chapter 3.2: Esercises p 171: 1,3,5(abcdef), 8(abc), 10

1 An SLD refultation of $P \cup \{G\}$ via R is an LD refutation, $\langle G_0, C_0\rangle, \cdots, \langle G_n, C_n\rangle$ of $P \cup \{G\}$ in which $R(G_i)$ is the literal resolved on at step i of the proof. If no R is mentioned we assume tat te standard one of choosing the leftmost literal is intended.) This is Definition 1.6.

So now we'll apply this to P_5, on page 170, with the goal $\{equivalent(c, a)\}$.

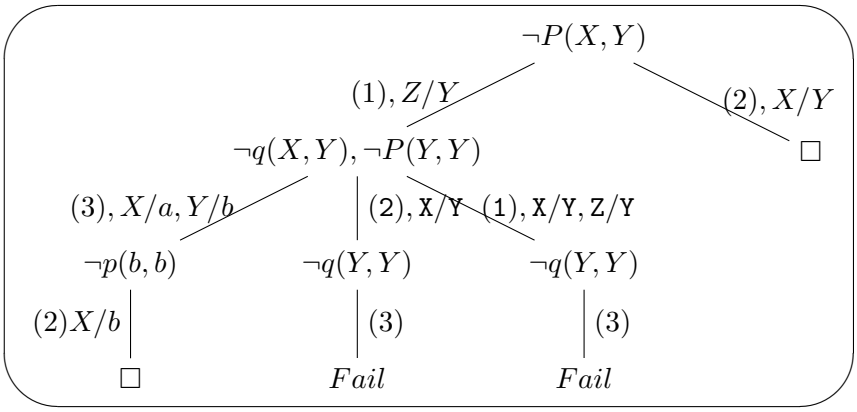
I'll use e for equivalent to reduce the typing. Here is P_5 in clausal form:

$$
\begin{aligned}
&(1)[e(X,Y), \neg e(Y,X)]\\
&(2)[e(X,Z), \neg e(X,Y), \neg e(Y,Z)]\\
&(3)[e(a,b)]\\
&(4)[e(b,c)]\\
&(G)[\neg e(c,a)]
\end{aligned}
$$

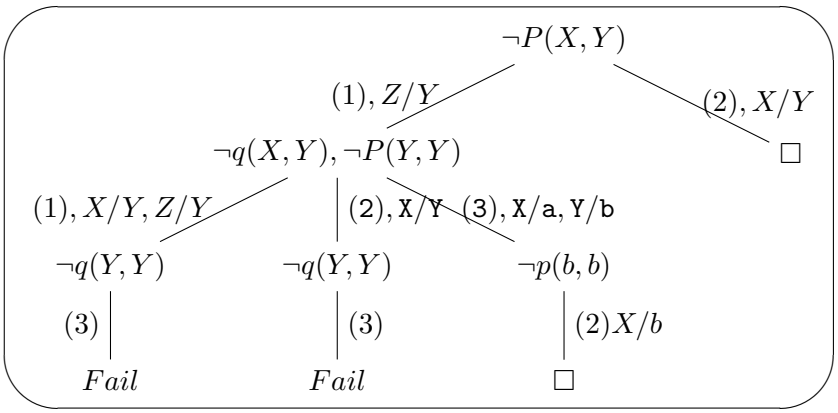
I'm going to resolve using clauses (1), (2), (3), (4) in that order

Goal, Program clause Substitution	Resolvent
$[\neg e(c,a)], [e(X,Y), \neg e(Y,X)][X/C, Y/a]$	$[\neg e(a,c)]$
$[\neg e(a,c)], [e(X,z), \neg e(X,Y), \neg e(Y,Z)][X/a][Z/c]$	$[\neg e(a,Y), \neg e(Y,c)]$
$[\neg e(a,Y), \neg e(Y,c)], [e(a,b)][Y/b]$	$[\neg e(b,c)]$
$[\neg e(b,c)], [e(b,c)]$	$\square$

3 The standard selection rule will choose the leftmost literal from a clause to resolve on. Notice that because we are using program clauses, the choice is in the goal clause: we will only ever have one positive literal to choose from in the program clause.



Choosing the right most literal in this case doesn't make a lot of different. Two of the three clauses have only a signal literal. Rule (1) has 3 literals, so when we resolve on the positive literal, we remove one literal from the goal clause, but add one as well, so after using rule (1), we would be able to choose to resolve on the rightmost instead of the leftmost.



5 These are pretty straight forward. I didn't use difference lists, so if you are curious, try them out.

```
second_element(X,[_,X|_]).

substitute_for_second_element(Y,[X,_|Xs], [X,Y|Xs]).

switch([X|Xs], M):- split_last(Xs,Y, Ys), append([Y|Ys], [X], M).

split_last([X], X, []).
split_last([X|Xs], Y, [X|Ys]) :- split_last(Xs, Y, Ys).
```

8 a) We want to prove that the program:

$$(1) add(X, 0, X).$$
$$(2) add(X, s(Y), s(Z)) : -add(X, Y, Z).$$

defines addition on the natural numbers in the sense that $add(s^n(0), s^m(o), s^r(0))$ is a consequence of the program iff $n + m = r$.

We'll use induction.

The base case is when we evaluate $add(s^n, 0, R)$. Because it makes our proof look more consistent, $0 = s^0(0)$ (ie 0 is 0 applications of the s function.) According to rule (1), $R = s^n$ and $n + 0 = n \; \forall n$

So, now suppose the proposition is true for some $k \geq 0$: $add(s^n(0), s^k(0), s^r(0)) \leftrightarrow n + k = r$. We'll prove that it is true for $k + 1$.

First we'll characterize a reduction by its length. A reduction of length 0 is $add(s^n(0), 0, s(n))$. A reduction of length 1 is $add(s^n(0), s(0), s(Z)), add(s^n(0), 0, Z), Z = s^n(0)$. In a reduction of length 0, the third argument has the same number of applications of s as the first argument.

Claim: everytime we use rule (2), we add one s to the third argument. Assume it is true for some k, then $add(s^n(0), s^{k+1}(0), s(Z)) : -add(s^n(0), s^k(0), Z)$. This is a reduction of length $k + 1$, and it adds one s to the third argument produced in the reduction of length k.

$$add(s^n(0), s^{k+1}(0), s(Z)) = add(s^n(0), s^k(0), Z)$$

So the rhs corresponds to a reduction of length k. The right hand side is an instance of the IH, so it correctly sets $Z = s^r(0)$ where $r = n + k$. And because the lefthand side is a reduction of length $k + 1$, it adds one s, to the third argument: $s(Z)$ so that since $add(s^n(0), s^k(0), s^r(0))$ sets its third argument $s^{n+k}(0) = s^r(0)$, $add(s^n(0), s^{k+1}(0), s(s^r(0))$ sets its thrid argument to $s^{n+k+1}(0)$. Since $add(s^n(0), s^k(0), s^r(0)) \leftrightarrow n+k = r$, then when we extend the reduction by one we add one s. So $add(s^n(0), s^{k+1}(0), s(s^r(0)))$ extends the k length reduction by 1, resulting in the third argument being set to $s^{n+k+1}(0)$ as required.

b)
```
add(X,0,X).
add(X, s(Y), s(Z)):-add(X,Y,Z).

mult(X,0,0).
mult(X,s(0), X).
mult(X,s(Z), Y):- mult(X, Z, W), add(X,W,Y).
```

c) This proof is just like the add proof, except that now we will have $s(Z) - 1$ adds instead of applications of $s(Z)$.

10 This is flatten, which I assigned on the final. I'll post the answer with the final answers.