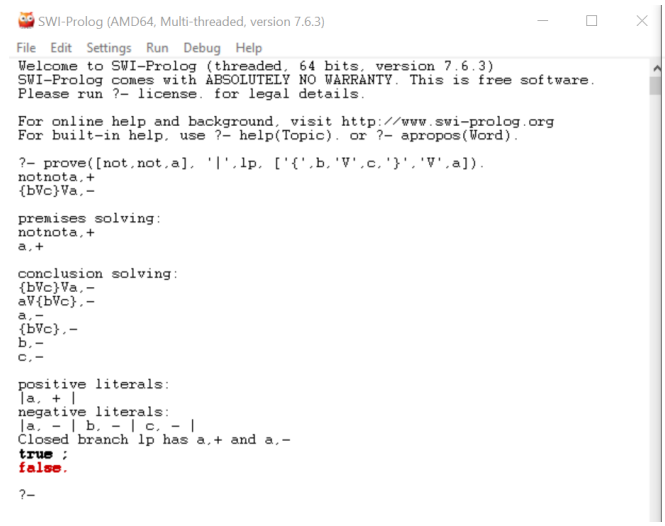
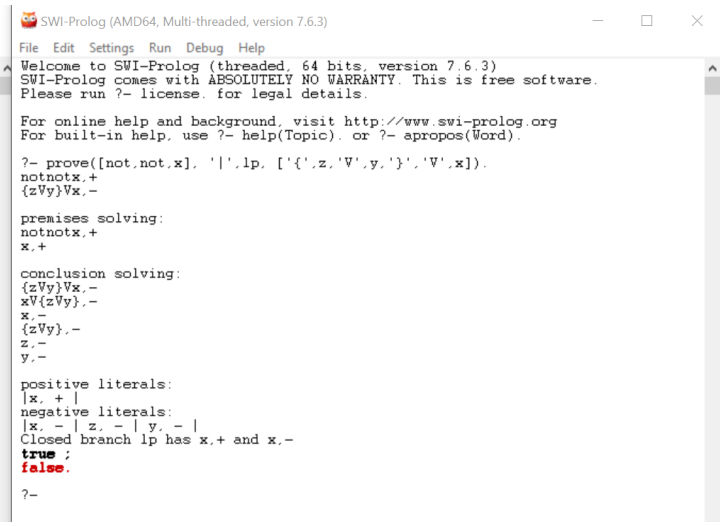
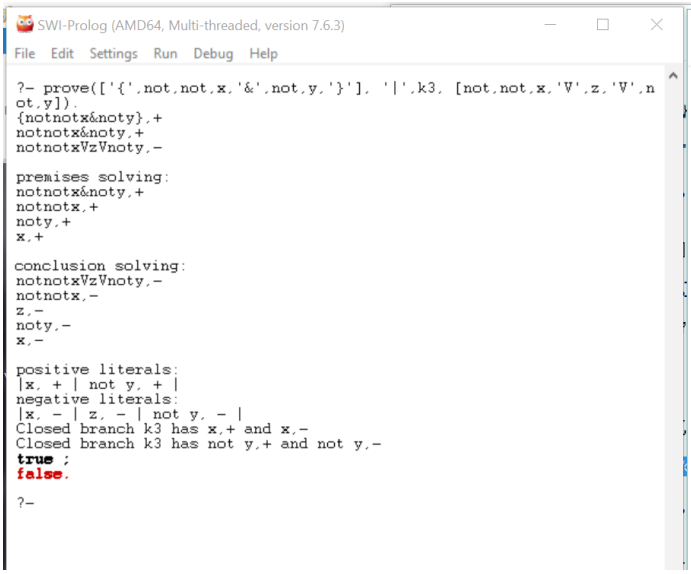
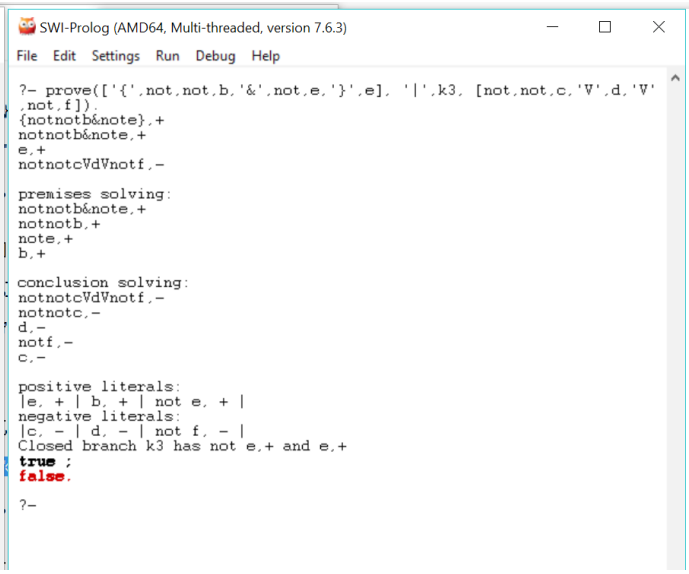
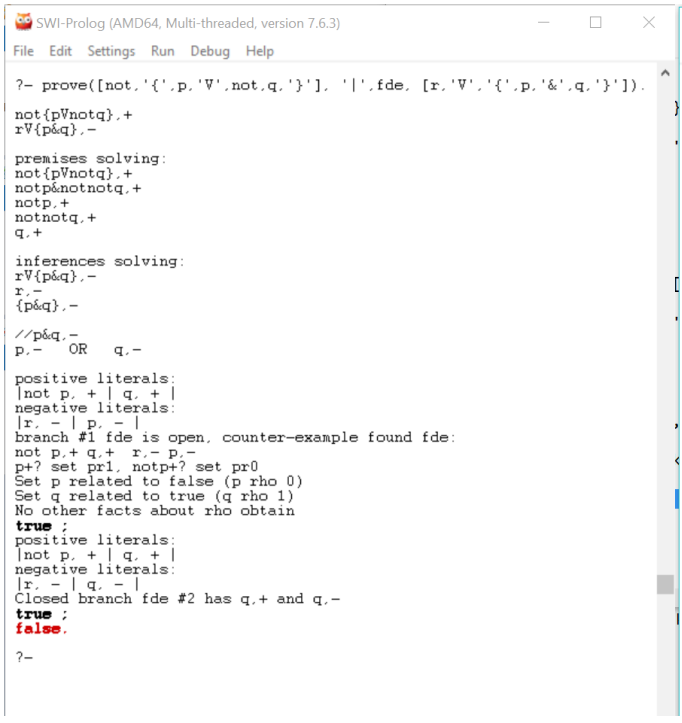
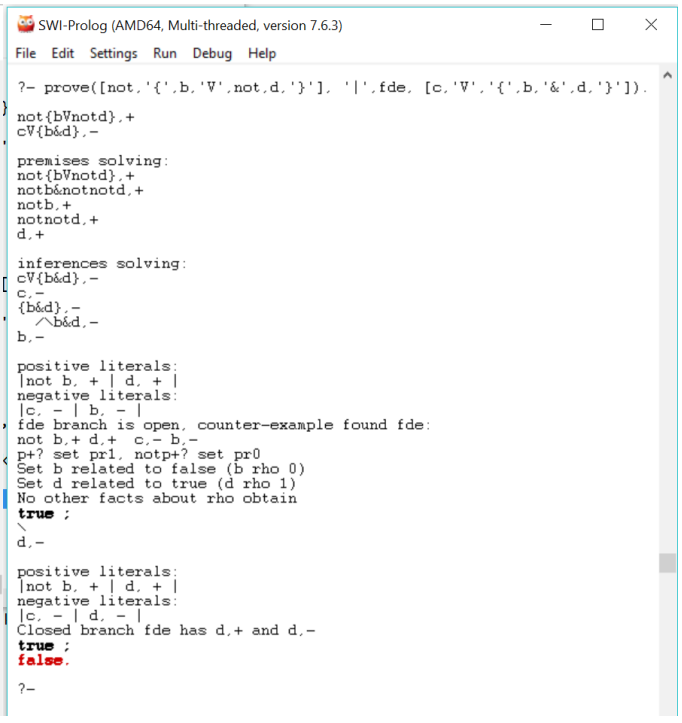

SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- prove([not,not,a], '|',lp, ['{',b,'V',c,'}','V',a]).
notnota,+
{bVc}Va,-
premises solving:
notnota,+
a,+
conclusion solving:
{bVc}Va,-
aV{bVc},-
a,-
{bVc},-
b,-
c,-
positive literals:
|a, + |
negative literals:
|a, - | b, - | c, - |
Closed branch lp has a,+ and a,-
true ;
false.
?-

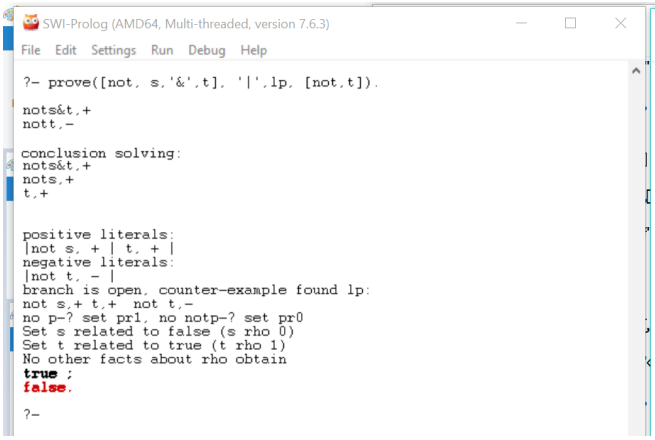

SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- prove([not,not,x], '|',lp, ['{',z,'V',y,'}','V',x]).
notnotx,+
{zVy}Vx,-
premises solving:
notnotx,+
x,+
conclusion solving:
{zVy}Vx,-
xV{zVy},-
x,-
{zVy},-
z,-
y,-
positive literals:
|x, + |
negative literals:
|x, - | z, - | y, - |
Closed branch lp has x,+ and x,-
true ;
false.
?-

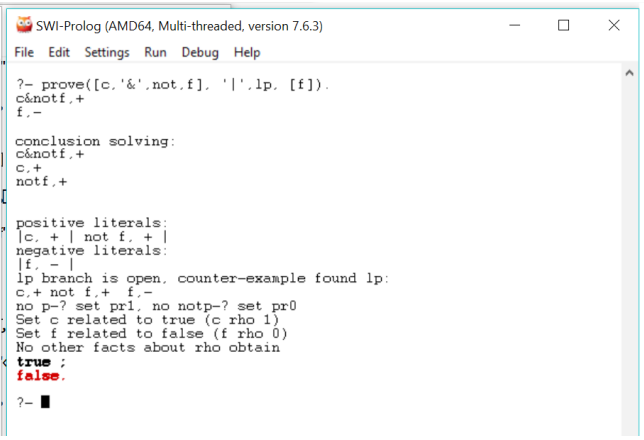

SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove(['{',not,not,x,'&',not,y,'}'], '|',k3, [not,not,x,'V',z,'V',not,y]).
{notnotx&noty},+
notnotx&noty,+
notnotxVzVnoty,-
premises solving:
notnotx&noty,+
notnotx,+
noty,+
x,+
conclusion solving:
notnotxVzVnoty,-
notnotx,-
z,-
noty,-
x,-
positive literals:
|x, + | not y, + |
negative literals:
|x, - | z, - | not y, - |
Closed branch k3 has x,+ and x,-
Closed branch k3 has not y,+ and not y,-
true ;
false.
?-

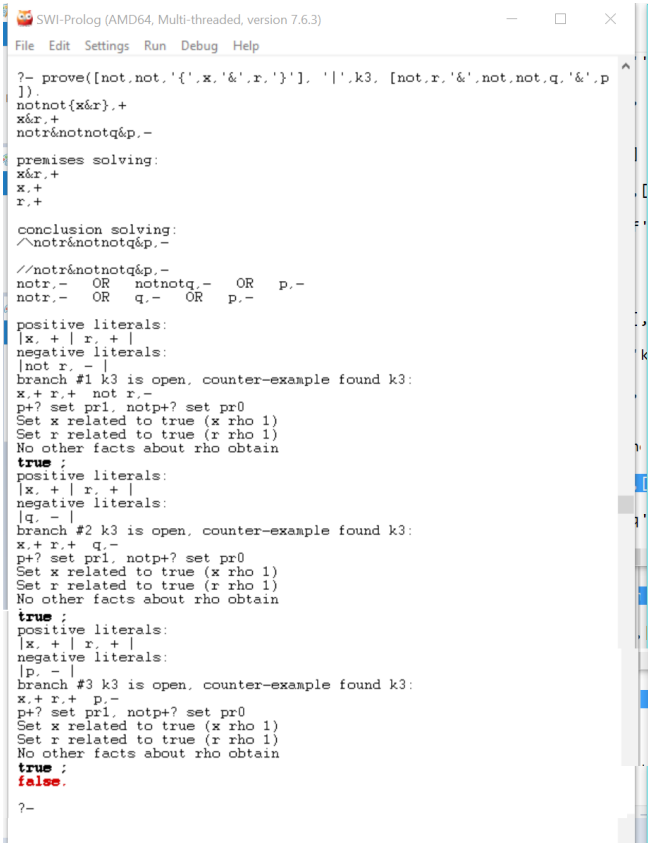

SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove(['{',not,not,b,'&',not,e,'}',e], '|',k3, [not,not,c,'V',d,'V',not,f]).
{notnotb&note},+
notnotb&note,+
e,+
notnotcVdVnotf,-
premises solving:
notnotb&note,+
notnotb,+
note,+
b,+
conclusion solving:
notnotcVdVnotf,-
notnotc,-
d,-
notf,-
c,-
positive literals:
|e, + | b, + | not e, + |
negative literals:
|c, - | d, - | not f, - |
Closed branch k3 has not e,+ and e,+
true ;
false.
?-

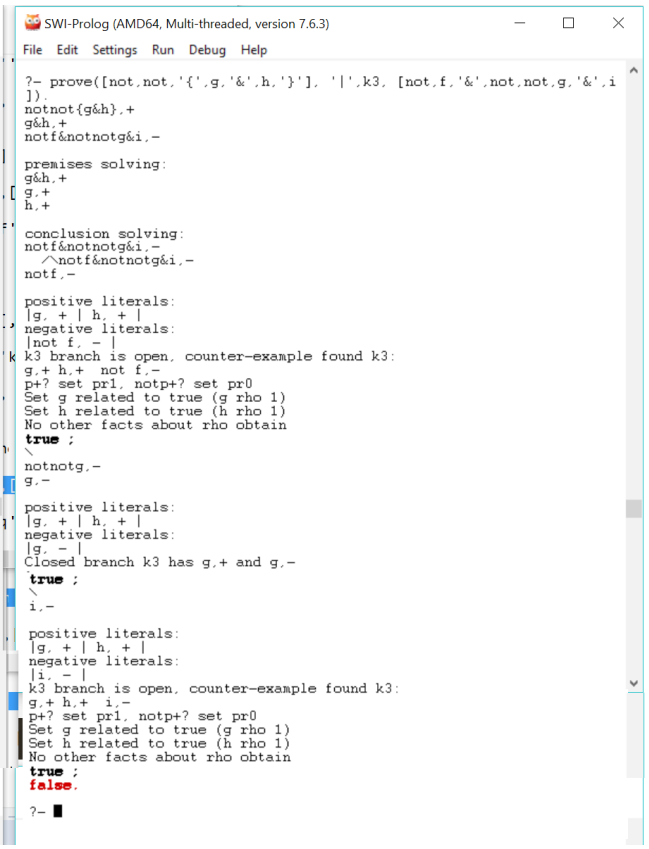

SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove([not,'{',p,'V',not,q,'}'], '|',fde, [r,'V','{',p,'&',q,'}']).
not{pVnotq},+
rV{p&q},-
premises solving:
not{pVnotq},+
notp&notnotq,+
notp,+
notnotq,+
q,+
inferences solving:
rV{p&q},-
r,-
{p&q},-
//p&q,-
p,- OR q,-
positive literals:
|not p, + | q, + |
negative literals:
|r, - | p, - |
branch #1 fde is open, counter-example found fde:
not p,+ q,+ r,- p,-
p+? set pr1, notp+? set pr0
Set p related to false (p rho 0)
Set q related to true (q rho 1)
No other facts about rho obtain
true ;
positive literals:
|not p, + | q, + |
negative literals:
|r, - | q, - |
Closed branch fde #2 has q,+ and q,-
true ;
false.
?-


SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove([not,'{',b,'V',not,d,'}'], '|',fde, [c,'V','{',b,'&',d,'}']).
not{bVnotd},+
cV{b&d},-
premises solving:
not{bVnotd},+
notb&notnotd,+
notb,+
notnotd,+
d,+
inferences solving:
cV{b&d},-
c,-
{b&d},-
/\b&d,-
b,-
positive literals:
|not b, + | d, + |
negative literals:
|c, - | b, - |
fde branch is open, counter-example found fde:
not b,+ d,+ c,- b,-
p+? set pr1, notp+? set pr0
Set b related to false (b rho 0)
Set d related to true (d rho 1)
No other facts about rho obtain
true ;
\
d,-
positive literals:
|not b, + | d, + |
negative literals:
|c, - | d, - |
Closed branch fde has d,+ and d,-
true ;
false.
?-


SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove([not, s,'&',t], '|',lp, [not,t]).
nots&t,+
nott,-
conclusion solving:
nots&t,+
nots,+
t,+
positive literals:
|not s, + | t, + |
negative literals:
|not t, - |
branch is open, counter-example found lp:
not s,+ t,+ not t,-
no p-? set pr1, no notp-? set pr0
Set s related to false (s rho 0)
Set t related to true (t rho 1)
No other facts about rho obtain
true ;
false.
?-


SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove([c,'&',not,f], '|',lp, [f]).
c&notf,+
f,-
conclusion solving:
c&notf,+
c,+
notf,+
positive literals:
|c, + | not f, + |
negative literals:
|f, - |
lp branch is open, counter-example found lp:
c,+ not f,+ f,-
no p-? set pr1, no notp-? set pr0
Set c related to true (c rho 1)
Set f related to false (f rho 0)
No other facts about rho obtain
true ;
false.
?-


SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove([not,not,'{',x,'&',r,'}'], '|',k3, [not,r,'&',not,not,q,'&',p]).
notnot{x&r},+
x&r,+
notr&notnotq&p,-
premises solving:
x&r,+
x,+
r,+
conclusion solving:
/\notr&notnotq&p,-
//notr&notnotq&p,-
notr,- OR notnotq,- OR p,-
notr,- OR q,- OR p,-
positive literals:
|x, + | r, + |
negative literals:
|not r, - |
branch #1 k3 is open, counter-example found k3:
x,+ r,+ not r,-
p+? set pr1, notp+? set pr0
Set x related to true (x rho 1)
Set r related to true (r rho 1)
No other facts about rho obtain
true ;
positive literals:
|x, + | r, + |
negative literals:
|q, - |
branch #2 k3 is open, counter-example found k3:
x,+ r,+ q,-
p+? set pr1, notp+? set pr0
Set x related to true (x rho 1)
Set r related to true (r rho 1)
No other facts about rho obtain
true ;
positive literals:
|x, + | r, + |
negative literals:
|p, - |
branch #3 k3 is open, counter-example found k3:
x,+ r,+ p,-
p+? set pr1, notp+? set pr0
Set x related to true (x rho 1)
Set r related to true (r rho 1)
No other facts about rho obtain
true ;
false.
?-


SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove([not,not,'{',g,'&',h,'}'], '|',k3, [not,f,'&',not,not,g,'&',i]).
notnot{g&h},+
g&h,+
notf&notnotg&i,-
premises solving:
g&h,+
g,+
h,+
conclusion solving:
notf&notnotg&i,-
/\notf&notnotg&i,-
notf,-
positive literals:
|g, + | h, + |
negative literals:
|not f, - |
k3 branch is open, counter-example found k3:
g,+ h,+ not f,-
p+? set pr1, notp+? set pr0
Set g related to true (g rho 1)
Set h related to true (h rho 1)
No other facts about rho obtain
true ;
\
notnotg,-
g,-
positive literals:
|g, + | h, + |
negative literals:
|g, - |
Closed branch k3 has g,+ and g,-
true ;
\
i,-
positive literals:
|g, + | h, + |
negative literals:
|i, - |
k3 branch is open, counter-example found k3:
g,+ h,+ i,-
p+? set pr1, notp+? set pr0
Set g related to true (g rho 1)
Set h related to true (h rho 1)
No other facts about rho obtain
true ;
false.
?-


SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove([not,t,'&','{',not,w,'&',q,'}'], '|',fde, [not,not,'{',not,w,'&',q,'}']).
nott&{notw&q},+
notnot{notw&q},-
premises solving:
nott&{notw&q},+
nott,+
{notw&q},+
notw,+
q,+
conclusion solving:
notnot{notw&q},-
{notw&q},-
/\notw&q,-
//notw&q,-
notw,- OR q,-
positive literals:
|not t, + | not w, + | q, + |
negative literals:
|not w, - |
Closed branch fde #1 has not w,+ and not w,-
true ;
positive literals:
|not t, + | not w, + | q, + |
negative literals:
|q, - |
Closed branch fde #2 has q,+ and q,-
true ;
false.
?-


SWI-Prolog (AMD64, Multi-threaded, version 7.6.3)
File Edit Settings Run Debug Help
?- prove([not,c,'&','{',e,'&',not,f,'}'], '|',fde, [not,not,'{',e,'&',not,f,'}']).
notc&{e&notf},+
notnot{e&notf},-
premises solving:
notc&{e&notf},+
notc,+
{e&notf},+
e,+
notf,+
conclusion solving:
notnot{e&notf},-
{e&notf},-
/\e&notf,-
e,-
positive literals:
|not c, + | e, + | not f, + |
negative literals:
|e, - |
Closed branch fde has e,+ and e,-
true ;
\
notf,-
positive literals:
|not c, + | e, + | not f, + |
negative literals:
|not f, - |
Closed branch fde has not f,+ and not f,-
true ;
false.
?-