Resolution and Unification

Using Prover9 and Maze4

Luciano Serafini

Fondazione Bruno Kessler, Trento, Italy

April 17, 2018

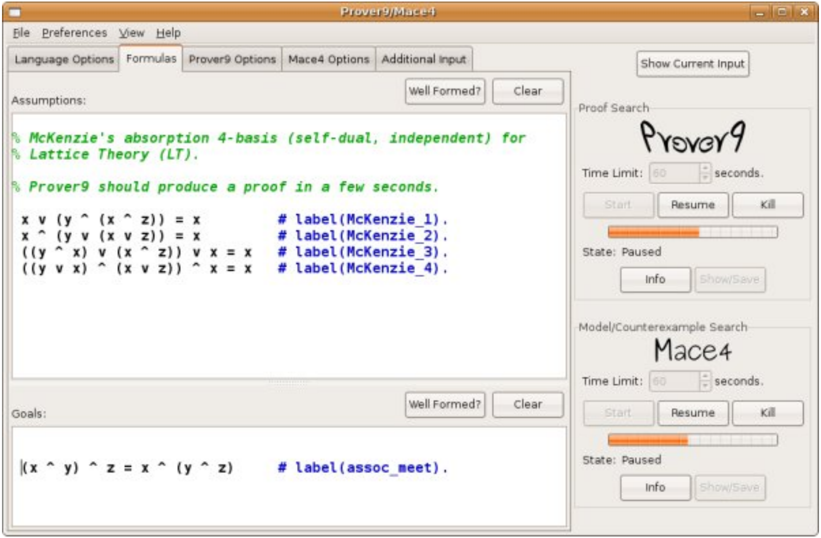
Prover9 Home Page

http://www.cs.unm.edu/ mccune/prover9/

Prover9 and Mace4

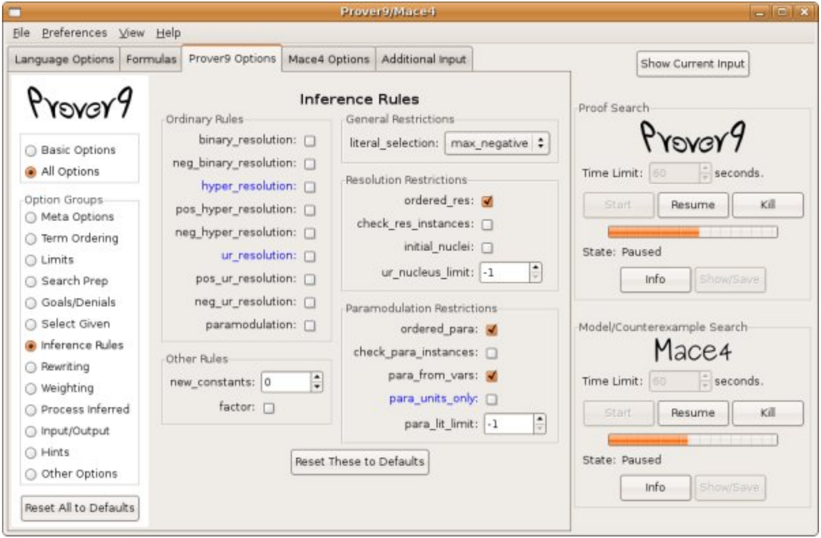
- Prover9 is an automated theorem prover for first-order and equational logic,
- Mace4 searches for finite models and counterexamples

Prover9 GUI



Prover9/Mace4
File Preferences View Help
Language Options
Formulas
Prover9 Options
Mace4 Options
Additional Input
Show Current Input
Assumptions:
Well Formed?
Clear
% McKenzie's absorption 4-basis (self-dual, independent) for
% Lattice Theory (LT).
% Prover9 should produce a proof in a few seconds.
x v (y ^ (x ^ z)) = x # label(McKenzie_1).
x ^ (y v (x v z)) = x # label(McKenzie_2).
((y ^ x) v (x ^ z)) v x = x # label(McKenzie_3).
((y v x) ^ (x v z)) ^ x = x # label(McKenzie_4).
Goals:
Well Formed?
Clear
(x ^ y) ^ z = x ^ (y ^ z) # label(assoc_meet).
Proof Search
Prover9
Time Limit: 60 seconds.
Start
Resume
Kill
State: Paused
Info
Show/Save
Model/Counterexample Search
Mace4
Time Limit: 60 seconds.
Start
Resume
Kill
State: Paused
Info
Show/Save

Prover9 GUI


Prover9/Mace4
File Preferences View Help
Language Options
Formulas
Prover9 Options
Mace4 Options
Additional Input
Show Current Input
Prover9
Basic Options
All Options
Option Groups
Meta Options
Term Ordering
Limits
Search Prep
Goals/Denials
Select Given
Inference Rules
Rewriting
Weighting
Process Inferred
Input/Output
Hints
Other Options
Reset All to Defaults
Inference Rules
Ordinary Rules
binary_resolution:
neg_binary_resolution:
hyper_resolution:
pos_hyper_resolution:
neg_hyper_resolution:
ur_resolution:
pos_ur_resolution:
neg_ur_resolution:
paramodulation:
Other Rules
new_constants: 0
factor:
General Restrictions
literal_selection: max_negative
Resolution Restrictions
ordered_res:
check_res_instances:
initial_nuclei:
ur_nucleus_limit: -1
Paramodulation Restrictions
ordered_para:
check_para_instances:
para_from_vars:
para_units_only:
para_lit_limit: -1
Reset These to Defaults
Proof Search
Prover9
Time Limit: 60 seconds.
Start
Resume
Kill
State: Paused
Info
Show/Save
Model/Counterexample Search
Mace4
Time Limit: 60 seconds.
Start
Resume
Kill
State: Paused
Info
Show/Save

Prover9's Proof Method

- The primary mode of inference used by Prover9 is resolution. It repeatedly makes resolution inferences with the aim of detecting inconsistency
- Prover9 will first do some preprocessing on the input file to convert it into the form it uses for inferencing.
  1. First it negates the formula given as a goal
  2. It then translates all formulae into clausal form.
  3. In some cases it will do some further pre-processing, (but you do not need to worry about this)
- Then it will compute inferences and by default write these standard output. Unless the input is very simple it will often generate a large number of inferences.
- If it detects an inconsistency it will stop and print out a proof consisting of the sequence of resolution rules that generated the inconsistency.
- It will also print out various statistics associated with the proof.

Simple example

Example (Reasoning in proposition logic)

Check if $p \wedge s, p \supset q, q \supset r \models r \vee t$ holds

Prover9 simple input file

```
formulas(assumptions).
p & s.                  % "&" symbol is for conjunction "and"
p -> q.                 % "->" symbol is for implication "implies"
q -> r.
end_of_list.

formulas(goals).
r | t.                  % "|" symbol is for distunction "or"
end_of_list.
```

Output of Prover9

```
============================== prooftrans ============================
Prover9 (32) version Dec-2007, Dec 2007.
Process 71916 was started by luciano on coccobill.local,
Fri Nov 22 11:36:46 2013
The command was "/Users/luciano/Applications/Prover9-Mace4-v05B.app/Contents/Resources/bin-mac-intel/prove
============================== end of head ===========================

============================== end of input ==========================

============================== PROOF =================================

% -------- Comments from original proof --------
% Proof 1 at 0.00 (+ 0.00) seconds.
% Length of proof is 11.
% Level of proof is 3.
% Maximum clause weight is 2.
% Given clauses 5.

1 p & s # label(non_clause).  [assumption].
2 p -> q # label(non_clause).  [assumption].
3 q -> r # label(non_clause).  [assumption].
4 r | t # label(non_clause) # label(goal).  [goal].
5 p.  [clausify(1)].
7 -p | q.  [clausify(2)].
8 -q | r.  [clausify(3)].
9 -r.  [deny(4)].
11 q.  [ur(7,a,5,a)].
12 -q.  [resolve(9,a,8,b)].
13 $F.  [resolve(12,a,11,a)].

============================== end of proof ==========================
```

A slightly more complex example using quantifiers

Example (Transitivity of subset relation)

Show that the containment relation between sets is transitive. I.e., For any set A, B, and C

$$A \subseteq B \wedge B \subseteq C \rightarrow A \subseteq C$$

Where $A \subseteq B$ is defined as $\forall x (x \in A \rightarrow x \in B)$

Prover9 input file

```
formulas(assumptions).
all x all y (subset(x,y) <-> (all z (member(z,x) -> member(z,y)))).
end_of_list.

formulas(goals).
all x all y all z (subset(x,y) & subset(y,z) -> subset(x,z)).
end_of_list.
```

Output of Prover9

```
============================== prooftrans ============================
Prover9 (32) version Dec-2007, Dec 2007.
Process 71873 was started by luciano on coccobill.local,
Fri Nov 22 11:32:23 2013
The command was "/Users/luciano/Applications/Prover9-Mace4-v05B.app/Contents/Resources/bin-mac-intel/prove
============================== end of head ===========================

============================== end of input ==========================

============================== PROOF =================================

% -------- Comments from original proof --------
% Proof 1 at 0.00 (+ 0.00) seconds.
% Length of proof is 14.
% Level of proof is 4.
% Maximum clause weight is 6.
% Given clauses 6.

1 (all x all y (subset(x,y) <-> (all z (member(z,x) -> member(z,y))))) # label(non_clause).  [assumption]
2 (all x all y all z (subset(x,y) & subset(y,z) -> subset(x,z))) # label(non_clause) # label(goal).  [goa
3 subset(x,y) | member(f1(x,y),x).  [clausify(1)].
4 -subset(x,y) | -member(z,x) | member(z,y).  [clausify(1)].
5 subset(x,y) | -member(f1(x,y),y).  [clausify(1)].
6 subset(c1,c2).  [deny(2)].
7 subset(c2,c3).  [deny(2)].
8 -subset(c1,c3).  [deny(2)].
11 -member(x,c1) | member(x,c2).  [resolve(6,a,4,a)].
12 -member(x,c2) | member(x,c3).  [resolve(7,a,4,a)].
13 member(f1(c1,c3),c1).  [resolve(8,a,3,a)].
14 -member(f1(c1,c3),c3).  [resolve(8,a,5,a)].
15 member(f1(c1,c3),c2).  [resolve(13,a,11,a)].
18 $F.  [ur(12,b,14,a),unit_del(a,15)].
```

Problem solving with Propositional Resolution

Six sculptures $\{C, D, E, F, G, H\}$ are to be exhibited in rooms $\{1, 2, 3\}$ of an art gallery.

1. Sculptures C and E may not be exhibited in the same room.
2. Sculptures D and G must be exhibited in the same room.
3. If sculptures E and F are exhibited in the same room, no other sculpture may be exhibited in that room.
4. At least one sculpture must be exhibited in each room, and
5. no more than three sculptures may be exhibited in any room.

1. If sculpture D is exhibited in room 1 and sculptures E and F are exhibited in room 2, which of the following must be true?
    1. Sculpture C must be exhibited in room 1.
    2. Sculpture H must be exhibited in room 3.
    3. Sculpture G must be exhibited in room 1.
    4. Sculpture H must be exhibited in room 2.
    5. Sculptures C and H must be exhibited in the same room.

Problem solving with Propositional Resolution

Six sculptures $\{C, D, E, F, G, H\}$ are to be exhibited in rooms $\{1, 2, 3\}$ of an art gallery.

$$P = \{Exhibits(X, n) \mid X \in \{C, \dots, H\},\ n \in \{1, 2, 3\}\}$$

$$\bigwedge_{\substack{X \in \{C,\dots,H\} \\ n \in \{1,2,3\}}} Exhibits(X, n) \equiv \neg Exhibits(X, (n \mod 3)+1) \wedge \neg Exhibits(X, (n \mod 3)+2)$$

1. Sculptures C and E may not be exhibited in the same room.

   no formalization = no information

2. Sculptures D and G must be exhibited in the same room.

$$\bigwedge_{n \in \{1,2,3\}} Exhibits(D, n) \equiv Exhibits(G, n)$$

3. If sculptures E and F are exhibited in the same room, no other sculpture may be exhibited in that room.

$$\bigwedge_{n\in\{1,2,3\}} \left(Exhibits(E,n) \wedge Exhibits(F,n) \supset \bigwedge_{X\in\{C,\dots,H\}\setminus\{E,F\}} \neg Exhibits(X,n) \right)$$

4. At least one sculpture must be exhibited in each room

$$\bigwedge_{n\in\{1,2,3\}} \bigvee_{X\in\{C,\dots,H\}} Exhibits(X,n)$$

5. no more than three sculptures may be exhibited in any room.

$$\bigwedge_{n\in\{1,2,3\}} \bigwedge_{\substack{\mathcal{S}\subset\{C,\dots,H\}\\ |\mathcal{S}|=4}} \neg \left(\bigwedge_{X\in E} Exhibits(X,n) \right)$$

Problem solving with Propositional Resolution

1. If sculpture D is exhibited in room 1 and sculptures E and F are exhibited in room 2, which of the following must be true?

$$Exhibites(D, 1) \wedge Exhibites(E, 2) \wedge Exhibites(F, 3) \supset \phi$$

   1. Sculpture C must be exhibited in room 1. $\phi = Exhibits(C, 1)$
   2. Sculpture H must be exhibited in room 3. $\phi = Exhibits(B, 3)$
   3. Sculpture G must be exhibited in room 1. $\phi = Exhibits(G, 1)$
   4. Sculpture H must be exhibited in room 2. $\phi = Exhibits(H, 2)$
   5. Sculptures C and H must be exhibited in the same room. $\phi = \bigvee_{n \in \{1,2,3\}} Exhibits(C, n) \equiv Exhibits(H, n)$

Problem solving with Propositional Resolution

$$CNF\left(\bigwedge_{\substack{X\in\{C,\dots,H\}\\ n\in\{1,2,3\}}} Exhibits(X,n) \equiv \left(\begin{array}{r} \neg Exhibits(X,(n \mod 3)+1) \wedge \\ \neg Exhibits(X,(n \mod 3)+2) \end{array}\right)\right) =$$

$$\left\{ \begin{array}{l} \{\neg Exhibits(X,n), \neg Exhibits(X,m)\}, \\ \{Exhibits(X,1), Exhibits(X,2), Exhibits(X,3)\} \end{array} \middle| \begin{array}{l} X\in\{C,\dots,H\} \\ n\neq m\in\{1,2,3\} \end{array} \right\}$$

$$CNF\left(\bigwedge_{n\in\{1,2,3\}} Exhibits(D,n) \equiv Exhibits(G,n)\right) =$$

$$\left\{ \begin{array}{l} \{\neg Exhibits(D,n), Exhibits(G,n)\} \\ \{\neg Exhibits(G,n), Exhibits(D,n)\} \end{array} \middle| n\in\{1,2,3\} \right\}$$

Problem solving with Propositional Resolution

$$CNF\left(\bigwedge_{n\in\{1,2,3\}}\left(Exhibits(E,n)\wedge Exhibits(F,n)\supset\bigwedge_{\substack{X\in\{C,\dots,H\}\\X\notin\{E,F\}}}\neg Exhibits(X,n)\right)\right)=$$

$$\left\{\left\{\begin{array}{c}\neg Exhibits(E,n),\neg Exhibits(F,n),\\ \neg Exhibits(X,n)\end{array}\right\}\;\middle|\;\begin{array}{l}n\in\{1,2,3\}\\ X\in\{C,\dots,H\}\setminus\{E,F\}\end{array}\right\}$$

Problem solving with Propositional Resolution

$$CNF\left(\bigwedge_{n\in\{1,2,3\}}\bigvee_{X\in\{C,\ldots,H\}} Exhibits(X,n)\right) =$$

$$\{\{Exhibits(X,n) \mid X \in \{C,\ldots,H\}\} \mid n \in \{1,2,3\}\} =$$

$$\left\{\begin{array}{c}
\{Exhibits(C,1), Exhibits(C,2), Exhibits(C,3)\} \\
\{Exhibits(D,1), Exhibits(D,2), Exhibits(D,3)\} \\
\vdots \\
\{Exhibits(H,1), Exhibits(H,2), Exhibits(H,3)\}
\end{array}\right\}$$

Problem solving with Propositional Resolution

$$CNF\left(\bigwedge_{n\in\{1,2,3\}}\ \bigwedge_{\substack{\mathcal{S}\subset\{C,\dots,H\}\\|\mathcal{S}|=4}} \neg\left(\bigwedge_{X\in E} Exhibits(X,n)\right)\right) =$$

$$\left\{\left\{\begin{array}{l}\neg Exhibits(X_1,n), \neg Exhibits(X_2,n),\\ \neg Exhibits(X_3,n), \neg Exhibits(X_4,n),\end{array}\right\}\middle|\ \begin{array}{l}\{X_1,X_2,X_3,X_4\}\subset\{C,\dots,H\}\\ X_i\neq X_j \text{ for } i\neq j,\quad n\in\{1,2,3\}\end{array}\right\} =$$

$$\left\{\begin{array}{c}\{\neg Exhibits(C,1), \neg Exhibits(D,1), \neg Exhibits(E,1), \neg Exhibits(F,1)\}\\ \{\neg Exhibits(C,1), \neg Exhibits(D,1), \neg Exhibits(E,1), \neg Exhibits(G,1)\}\\ \{\neg Exhibits(C,1), \neg Exhibits(D,1), \neg Exhibits(E,1), \neg Exhibits(H,1)\}\\ \vdots\\ \{\neg Exhibits(E,1), \neg Exhibits(F,1), \neg Exhibits(G,1), \neg Exhibits(H,1)\}\end{array}\right\}$$

$$CNF(\neg(Exhibites(D,1) \wedge Exhibites(E,2) \wedge Exhibites(F,3) \supset \phi) =$$

$$\{\{Exhibites(D,1)\}, \{Exhibites(E,2)\}, \{Exhibites(F,3)\}, \{\neg\phi\}\}$$

where ϕ is one of the following formulas

1. $Exhibits(C,1)$ NO
2. $Exhibits(B,3)$ NO
3. $Exhibits(G,1)$ YES
4. $Exhibits(H,2)$ NO
5. We consider the last case separately

Problem solving with Propositional Resolution

$Exhibits(D,1) \equiv Exhibits(G,1)$	assumption	(1)
$Exhibits(D,1) \wedge Exhibits(E,2) \wedge Exhibits(F,2) \supset$		
$Exhibits(G,1)$	goal	(2)
$\neg Exhibits(D,1), Exhibits(G,1)$	clausify (1)	(3)
$Exhibits(D,1)$	deny (10)	(4)
$\neg Exhibits(G,1)$	deny (10)	(5)
$Exhibits(G,1)$	RES (3), (4)	(6)
$\perp$	RES (6), (5)	(7)

5. Sculptures C and H must be exhibited in the same room.

$$\bigvee_{n\in\{1,2,3\}} Exhibits(C,n) \equiv Exhibits(H,n)$$

$$CNF\left(\neg\left(\begin{array}{c} Exhibites(D,1) \wedge Exhibites(E,2) \wedge Exhibites(F,3) \supset \\ \bigvee_{n\in\{1,2,3\}} Exhibits(C,n) \equiv Exhibits(H,n) \end{array}\right)\right) =$$

$$\left\{\begin{array}{c} \{Exhibites(D,1)\}, \{Exhibites(E,2)\}, \{Exhibites(F,3)\} \\ \{Exhibites(C,1), Exhibites(H,1)\}, \{\neg Exhibites(C,1), \neg Exhibites(H,1)\}, \\ \{Exhibites(C,2), Exhibites(H,2)\}, \{\neg Exhibites(C,2), \neg Exhibites(H,2)\}, \\ \{Exhibites(C,3), Exhibites(H,3)\}, \{\neg Exhibites(C,3), \neg Exhibites(H,3)\} \end{array}\right\}$$

$$
\begin{array}{rll}
Exhibits(E,2) \wedge Exhibits(F,2) \supset \neg Exhibits(C,2) & \text{assumption} & (8) \\
Exhibits(E,2) \wedge Exhibits(F,2) \supset \neg Exhibits(H,2) & \text{assumption} & (9) \\
Exhibits(D,I) \wedge Exhibits(E,2) \wedge Exhibits(F,2) \supset & & (10) \\
(Exhibits(C,1) \equiv Exhibits(H,1)) \vee & & \\
(Exhibits(C,2) \equiv Exhibits(H,2)) \vee & & \\
(Exhibits(C,3) \equiv Exhibits(H,3)) & \text{goal} & \\
\{\neg Exhibits(E,2), \neg Exhibits(F,2), \neg Exhibits(C,2) & \text{clausify (8)} & (11) \\
\{\neg Exhibits(E,2), \neg Exhibits(F,2), \neg Exhibits(H,2) & \text{clausify (9)} & (12) \\
Exhibits(E,2) & \text{deny (10)} & (13) \\
Exhibits(F,2) & \text{deny (10)} & (14) \\
Exhibits(C,2), Exhibits(H,2) & \text{deny (10)} & (15) \\
\neg Exhibits(F,2), \neg Exhibits(H,2) & \text{RES (12), (13)} & (16) \\
\neg Exhibits(H,2) & \text{RES (16), (14)} & (17) \\
\neg Exhibits(F,2), \neg Exhibits(C,2) & \text{RES (11), (13)} & (18) \\
\neg Exhibits(C,2) & \text{RES (18), (14)} & (19) \\
Exhibits(H,2) & \text{RES (15), (19)} & (20) \\
\perp & \text{RES (20), (17)} & (21)
\end{array}
$$

Model generation - Mace4

- Prover9 tries to show that $\Gamma \models \phi$ by making attempts to show that the set of formulas $\Gamma \cup \{\neg\phi\}$ is not satisfiable.
- If Prover9 succeeds ok in showing that $\Gamma \cup \{\neg\phi\}$ is not satisfiable, then clearly $\Gamma \models \phi$.
- But what about if Prover9 fails in showing that $\Gamma \cup \{\neg\phi\}$ is not satisfiable? i.e., when $\Gamma \cup \{\neg\phi\}$ is satisfiable?
- Can we have a model for $\Gamma \cup \{\neg\phi\}$?
- Yes, we have to use Mace4.

Mace4

- Mace4 is a program that searches for finite models of first-order formulas.
- For a given domain size, all instances of the formulas over the domain are constructed. The result is a set of ground clauses with equality.
- Then, a decision procedure based on ground equational rewriting is applied. If satisfiability is detected, one or more models are printed.

Mace4 – example

Input file:

```
arc(x,y) -> node(x) & node(y).
exists x1 exists x2 exists x3 (color(x1) & color(x2) & color(x3) &
              x1 != x2 & x2 != x3 & x1 != x3).
color(x1) & color(x2) & color(x3) & color(x4) ->
             x1=x2 | x1=x3 | x1=x4 | x2=x3 | x2=x4 | x3=x4.
hascolor(x,y) -> node(x) & color(y).
color(x) -> -node(x).
color(x) | node(x).
node(x) -> exists y hascolor(x,y).
hascolor(x,y1) & hascolor(x,y2) -> y1=y2.
N1 != N2 & N1 != N3 & N1 != N4 & N2 != N3 & N2 != N4 & N3 != N4.
arc(N1,N2).
arc(N2,N3).
arc(N3,N1).
arc(N1,N4).
arc(N2,N4).
% arc(N3,N4).
arc(x,y) -> arc(y,x)
-arc(x,x).
arc(x,y) & hascolor(x,z) -> -hascolor(y,z).
```

Mace4 – example

Produced model:

```
interpretation( 7, [number = 1,seconds = 0], [
    function(N1, [0]),                  function(c1, [4]),
    function(N2, [1]),                  function(c2, [5]),
    function(N3, [2]),                  function(c3, [6]),
    function(N4, [3]),
    function(f1(_), [4,5,6,6,0,0,0]),
    relation(color(_), [0,0,0,0,1,1,1]),
    relation(node(_), [1,1,1,1,0,0,0]),
    relation(arc(_,_), [               relation(hascolor(_,_), [
        0,1,1,1,0,0,0,                      0,0,0,0,1,0,0,
        1,0,1,1,0,0,0,                      0,0,0,0,0,1,0,
        1,1,0,0,0,0,0,                      0,0,0,0,0,0,1,
        1,1,0,0,0,0,0,                      0,0,0,0,0,0,1,
        0,0,0,0,0,0,0,                      0,0,0,0,0,0,0,
        0,0,0,0,0,0,0,                      0,0,0,0,0,0,0,
        0,0,0,0,0,0,0]),                    0,0,0,0,0,0,0])]).
```