2011-02-10

Watch Watson do action on TV next week. Root for the machine!

Binary resolution, as defined in Ch. 1 of Schöning’s book is refutation complete for the propositional calculus.

Non-binary resolution (on which
all pairs of complementary literals are removed from the resolvent during a resolution step) is also refutation complete.

Factoring (i.e., the removal of complementary literals in one clause) may also be used. Previews: binary resolution is sound but not complete for First-Order logic.
(for) $e_i$, either non-binary resolution or binary resolution and factorily are reflection complete.
Example

Show that $F = (P \lor Q) \land (P \lor \neg Q) \land (\neg P \lor Q)$ is unsatisfiable.

Proof 1: $\{P, Q\} \vdash P, \neg Q \vdash \neg P, \neg Q \vdash \neg P, \neg Q$
Proof 2: \{p, \neg q\} \not\subseteq \{p, \neg p, \neg q\}

Note:
This is a refutation graph (Shöning)
(Leveled uses
the single and
the short edges)
Exercise 33 [Schnitzen]

Show that $A \land B \land C$ is a consequence of the clause set $F = \{ \neg A, B \}, \{ \neg B, C \}, \{ A, \neg C \}, \{ A, B, C \}$

$F = A \land B \land C$

Let $F' = (\neg A \lor B) \land (\neg B \lor C) \land (A \lor \neg C) \land (A \lor B \lor C)$

$F' \land \neg (A \land B \land C)$ is unsatisfiable.

$F \cup \{ \neg A, \neg B, \neg C \}$ is unsatisfiable.
Exercise 34  Show that the following formula is a tautology:

\[ F = (\neg B \land C \land D) \lor (\neg B \land \neg D) \lor (C \land D) \lor B \]

A formula is a tautology iff its negation is unsatisfiable.

\[ \neg F = \neg (\neg B \land C \land D) \land \neg (\neg B \land \neg D) \land (C \land D) \land \neg B \]

\[ \uparrow (B \lor C \lor D) \land (B \lor D) \land (\neg C \lor \neg D) \land \neg B \]
\{B, C, \neg D\} \quad \{B, 0\} \quad \{\neg C, \neg 0\} \quad \{\neg B\}

- \{B, \neg 0\} \quad \{B, 0\}
- \{B\}
Goal: \( l_{1-b} \); Show \( KB \models \neg l_{1-b} \) is unsatisfiable.

\[ \neg l_{1-b}, \neg sw_{-up}, \neg pow, \neg un_{-l2} \]

In computer:

\[ l_{1-b} \leftarrow sw_{-up} \]
\[ \land power \land unlit_{-light1}. \]
\[ sw_{-up}. \]
\[ power \leftarrow lit_{-light2}. \]
\[ unlit_{-light1}. \]
\[ lit_{-light2}. \]

Conclusion: \( l_{1-b} \)

- The computer doesn't know the meaning of the symbols
- The user can interpret the symbol using their meaning