Analysis of Propositional Logic

To talk about propositional logic, I will use the following assumptions and notation:

A theory F is a set of propositional logic formulas.

Each formula $f \in F$ is a set of literals. A literal is a propositional variable or its negations (e.g., a and $\neg a$ are literals). Any formula that has a literal and its negation is a tautology and can be deleted from F.

Each formula $f \in F$ is interpreted as a disjunction ("or"), and F is interpreted as a conjunction of the formulas. This is called conjunctive normal form (CNF).

A value assignment v assigns a truth value to each propositional variable used in F. It is often useful to think of v as a conjunction of literals.

v satisifies F (equivalently v is a model of F, or $v \models F$) if and only if every formula in F has one literal which is true in v.

For two formulas f_1 and f_2, the resolution inference rule works as follows. Let a be a propositional variable. If $a \in f_1$ and $\neg a \in f_2$, then we can infer $f_3 = (f_1 - \{a\}) \cup (f_2 - \{\neg a\})$.

Theorem 1 *The resolution inference rule is sound.*

This is easily provable using truth tables.

Theorem 2 *Any arbitrary propositional formula F' can be converted to conjunctive normal form.*

Algorithm: The conversion algorithm is as follows. Let F' be an arbitrary propositional formula. Then *convert*(F') proceeds recursively as follows:

If F' is a literal or a disjunction of literals, put the literal(s) into a set f and return $\{f\}$.

Else if F' has the form $\neg\neg F'_1$, then return *convert*(F'_1).

Else if F' has the form $\neg(F'_1 \vee F'_2)$, then return *convert*$(\neg F'_1 \wedge \neg F'_2)$.

Else if F' has the form $\neg(F'_1 \wedge F'_2)$, then return *convert*$(\neg F'_1 \vee \neg F'_2)$.

Else if F' has the form $F'_1 \wedge F'_2$, then return *convert*$(F'_1) \cup$ *convert*(F'_2).

Else if F' has the form $F'_1 \vee F'_2$, then let F_1 be *convert*(F'_1) and let F_2 be *convert*(F'_2). Now create a new propositional variable a. Add a to each formula in F_1 and add $\neg a$ to each formula in F_2 and return $F_1 \cup F_2$.

End Algorithm.

Let $F = convert(F')$. The claim is that $v' \models F'$ if and only if there is a value assignment v such that $v \models v'$ and $v \models F$, i.e., v assigns appropriate values to the extra variables that are added.

The last step in the algorithm is probably the most mysterious. Suppose a value assignment $v \models F_1 \vee F_2$ (before we create a). Now see what happens after we create a. If $v \models F_1$, we can choose false for a to make F_2 true. If $v \models F_2$, then we can choose true for a to make F_1 true, so we really still have $F_1 \vee F_2$. Alternatively, we could have used a distribution law and not add any additional variables, but this could get combinatorially explosive.

So depending on our choice for a, we automatically get either F_1 or F_2.

Theorem 3 *(completeness) If a propositional theory F in conjunctive normal form is inconsistent, then the resolution rule can be used to infer inconsistent literals.*

Proof: We prove this by induction on the number of propositional variables.

Basis: When a theory F contains only one propositional variable a, then F is inconsistent if and only if both $\{a\} \in F$ and $\{\neg a\} \in F$.

Induction: Suppose F contains $n + 1$ propositional variables. If F is inconsistent, then every value assignment v for the variables makes F false, i.e., $v \not\models F$.

Let a be a propositional variable used in F. If both $\{a\} \in F$ and $\{\neg a\} \in F$, then the inconsistency is obvious, and no more work is needed.

Otherwise, create a new theory F' as follows. Remove from F any formula that contains both a and $\neg a$ (these formulas are tautologies, anyway). Let F_a be the subset of formulas that contain a, let $F_{\neg a}$ be the subset of formulas that contain $\neg a$, and let F_{other} be the rest of the formulas. Let F_{new} be the result of resolving a from every pair of formulas $(f_a, f_{\neg a}) \in F_a \times F_{\neg a}$. Let $F' = F_{new} \cup F_{other}$. F' does not use a, so it has n variables (or less).

Suppose F is consistent. Then $v \models F$ for some value assignment v. Because the resolution inference rule is sound, then it must be the case that $v \models F'$.

Suppose F is inconsistent. We show that F' is also inconsistent. Let v' be an arbitrary value assignment for F'. Let v_a be identical to v' except that v_a assigns true to a. Let $v_{\neg a}$ be identical to v' except that $v_{\neg a}$ assigns false to a. Because F is inconsistent, it must be the case that $v_a \not\models F_a$ or $v_a \not\models F_{\neg a}$ or $v_a \not\models F_{other}$. Also, it must be the case that $v_{\neg a} \not\models F_a$ or $v_{\neg a} \not\models F_{\neg a}$ or $v_{\neg a} \not\models F_{other}$. Consider the folowing cases:

Case 1: $v_a \not\models F_a$ or $v_{\neg a} \not\models F_{\neg a}$. Neither of these are possible because, respectively, assigning true to a makes F_a true, and assigning false to a makes $F_{\neg a}$ true.

Case 2: $v_a \not\models F_{other}$ or $v_{\neg a} \not\models F_{other}$. This implies that $v' \not\models F_{other}$ because a is not referenced by F_{other}. Note that $v' \not\models F_{other}$ implies $v' \not\models F'$ because F_{other} is a subset of F'.

Case 3: $v_a \not\models F_{\neg a}$ and $v_{\neg a} \not\models F_a$. First, $v_a \not\models F_{\neg a}$ implies that all the literals in some formula $f' \in F_{\neg a}$ are made false by v_a. Second, $v_{\neg a} \not\models F_a$ implies that all the literals in some formula $f \in F_a$ are made false by $v_{\neg a}$. When resolution is applied to (f, f'), a is eliminated, so it must be that both v_a nor $v_{\neg a}$ makes the new formula false, which implies that v' makes the new formula false. Because the new formula is in F', it follows that $v' \not\models F'$.

So by mathematical induction, if F is inconsistent, then the resolution rule is able to infer inconsistent literals.

End Proof.