Resolution: Rules of Inference

Rule 1: If a literal appears more than once in a CNF sentence, all but the first may be dropped.
Example: From $p \lor \neg q \lor p$, infer $p \lor \neg q$.

Rule 2: If a CNF sentence contains both an atom and its negation, then the sentence is useless, and may be ignored.
Example: The sentence $p \lor \neg q \lor \neg p$ is useless.

Rule 3: Let $S_1$ be the sentence $\alpha_1 \lor \ldots \lor \alpha_k$ and let $S_2$ be the sentence $\beta_1 \lor \ldots \lor \beta_n$, where each $\alpha$ and each $\beta$ is a literal. Suppose that for some particular $i$ and $j$, $\alpha_i$ is the negation of $\beta_j$. Then it is possible to infer a new sentence $S_3$ which is the disjunction of all the $\alpha$’s except $\alpha_i$ with all the $\beta$’s except $\beta_j$.

Examples:

If $S_1$ is $p \lor q \lor r$ and $S_2$ is $\neg q \lor \neg s$, infer $p \lor r \lor \neg s$.
If $S_1$ is $\neg p \lor s$ and $S_2$ is $r \lor p$, infer $r \lor s$.
If $S_1$ is $p$ and $S_2$ is $\neg p \lor q$, infer $q$.
If $S_1$ is $p$ and $S_2$ is $\neg p$, infer the empty sentence.

Resolution: Proof Technique

To prove sentence $\phi$ from a set of axioms $\Gamma$:

begin Set $\Delta = \Gamma \cup \{\neg \phi\}$;
Convert $\Delta$ to CNF;
For each sentence $S \in \Delta$, apply rule 1 or rule 2 if applicable.
Loop Find two sentences $S_1$ and $S_2$ in $\Delta$ where rule 3 applies, but has not been previously used;
  If there are no such two sentences, then return “$\phi$ cannot be proven.”
  Apply rule 3 to get a new sentence $S_3$;
  If $S_3$ is the empty sentence, then return “$\phi$ has been proven.”
  If either rule 1 or rule 2 applies to $S_3$, then apply it;
  Add $S_3$ to $\Delta$
Endloop
end.
Example:

Given: 1. $p \iff (r \lor s)$.
   2. $r \Rightarrow \neg p$.
   3. $s \Rightarrow \neg p$.
Prove: $\neg p \land \neg r \land \neg s$.

Negation of H: 4. $\neg (\neg p \land \neg r \land \neg s)$.

 Converted to CNF.
1a. $\neg p \lor r \lor s$.
1b. $\neg r \lor p$.
1c. $\neg s \lor p$.
2. $\neg r \lor \neg p$.
3. $\neg s \lor \neg p$.
4. $p \lor r \lor s$.

From 4 and 1a, infer 5. $r \lor s \lor r \lor s$.
From 5 using rule 1, infer 6. $r \lor s$.
From 6 and 1b, infer 7. $p \lor s$.
From 7 and 1c, infer 8. $p \lor p$.
From 8, using rule 1, infer 9. $p$.
From 9 and 2, infer 10. $\neg r$.
From 9 and 3, infer 11. $\neg s$.
From 11 and 6, infer 12. $r$.
From 12 and 10, infer 13. $\emptyset$.

Syntax of Predicate Calculus

The predicate calculus uses the following types of symbols:

**Constants:** A constant symbol denotes a particular entity. E.g. “john”, “muriel” “1”.

**Functions:** A function symbol denotes a mapping from a number of entities to a single entity. E.g. “father_of” is a function with one argument. “plus” is a function with two arguments. “father_of(john)” is some person. “plus(2,7)” is some number.

**Predicates:** A predicate denotes a relation on a number of entities. E.g. “married” is a predicate with two arguments. “odd” is a predicate with one argument. “married(john, sue)” is a sentence that is true if the relation of marriage holds between the people John and Sue. ‘odd(plus(2,7))” is a true sentence.
Variables: These represent some undetermined entity. Examples: “X” “S1” . . .

Boolean operators: ¬, ∨, ∧, ⇒, ⇔.

Quantifiers: The symbols ∀ (for all) and ∃ (there exists).

Grouping symbols: The open and close parentheses and the comma.

A term is either

1. A constant symbol; or
2. A variable symbol; or
3. A function symbol applied to terms.

Examples: “john”, “X”, “father_{of}(john)”, “plus(X,plus(1,3))”.

An atomic formula is a predicate symbol applied to terms.

Examples: “odd(X)” “odd(plus(2,2))”, “married(sue,father_{of}(john))”.

A formula is either

1. An atomic formula; or
2. The application of a Boolean operator to formulas; or
3. A quantifier followed by a variable followed by a formula.

Examples: “odd(X),” “odd(X) ∨ ¬odd(plus(X, X)),” “∃X odd(plus(X, Y)),”
“∀X odd(X) ⇒ ¬odd(plus(X,3)).”

A sentence is a formula with no free variables. (That is, every occurrence of every
variable is associated with some quantifier.)

Clausal Form and Skolemization

A literal is either an atomic formula or the negation of an atomic formula.

Examples: odd(3). ¬odd(plus(X,3)). married(sue,Y).

A clause is the disjunction of literals. Variables in a clause are interpreted as
universally quantified with the largest possible scope.

Example: odd(X) ∨ odd(Y) ∨ ¬odd(plus(X, Y)) is interpreted as
∀X,Y odd(X) ∨ odd(Y) ∨ ¬odd(plus(X, Y)).
Converting a sentence to clausal form

1. Replace every occurrence of $\alpha \Leftrightarrow \beta$ by
   $(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$. When this is complete, the sentence will have no occurrence of $\Leftrightarrow$.

2. Replace every occurrence of $\alpha \Rightarrow \beta$ by $\neg \alpha \lor \beta$. When this is complete, the only Boolean operators will be $\lor$, $\neg$, and $\land$.

3. Replace every occurrence of $\neg (\alpha \lor \beta)$ by $\neg \alpha \land \neg \beta$; every occurrence of $\neg (\alpha \land \beta)$ by $\neg \alpha \lor \neg \beta$; and every occurrence of $\neg \neg \alpha$ by $\alpha$.
   
   New step: Also, replace every occurrence of $\neg \exists \mu \alpha$ by $\forall \mu \neg \alpha$ and every occurrence of $\neg \forall \mu \alpha$ by $\exists \mu \neg \alpha$.
   Repeat as long as applicable. When this is done, all negations will be next to an atomic sentence.

4. (New Step: Skolemization). For every existential quantifier $\exists \mu$ in the formula, do the following:
   If the existential quantifier is not inside the scope of any universal quantifiers, then
   
   i. Create a new constant symbol $\gamma$.
   ii. Replace every occurrence of the variable $\mu$ by $\gamma$.
   iii. Drop the existential quantifier.

   If the existential quantifier is inside the scope of universal quantifiers with variables $\Delta_1 \ldots \Delta_k$, then
   
   i. Create a new function symbol $\gamma$.
   ii. Replace every occurrence of the variable $\mu$ by the term $\gamma(\Delta_1 \ldots \Delta_k)$
   iii. Drop the existential quantifier.

   Example. Change “$\exists_X \text{blue}(X)$” to “blue(sk1)”.
   Change $\forall_X \exists_Y \text{odd}(\text{plus}(X,Y))$ to $\forall_X \text{odd}(\text{plus}(X,\text{sk2}(X)))$.
   Change $\forall_X \forall_Y \exists_A \exists_B \text{p}(X,Y,Z,A,B)$ to $\text{p}(X,Y,\text{sk3}(X,Y),A,\text{sk4}(X,Y,A))$.

5. New step: Elimination of universal quantifiers:
   Part 1. Make sure that each universal quantifier in the formula uses a variable with a different name, by changing variable names if necessary.
   Part 2. Drop all universal quantifiers.

   Example. Change $[\forall_X \text{p}(X)] \lor [\forall_X \text{q}(X)]$ to $\text{p}(X) \lor \text{q}(X1)$.
6. (Same as step 4 of CNF conversion.) Replace every occurrence of \((\alpha \land \beta) \lor \gamma\)
by \((\alpha \lor \gamma) \land (\beta \lor \gamma)\), and every occurrence of \(\alpha \lor (\beta \land \gamma)\) by \((\alpha \lor \beta) \land (\alpha \lor \gamma)\). Repeat as long as applicable. When this is done, all conjunctions will be at top level.

7. (Same as step 5 of CNF conversion.) Break up the top-level conjunctions into separate sentences. That is, replace \(\alpha \land \beta\) by the two sentences \(\alpha\) and \(\beta\). When this is done, the set will be in CNF.

Example:

Start. \(\forall X \ [even(X) \Leftrightarrow [\forall Y \ even(times(X,Y))]\]

After Step 1: \(\forall X \ [\lbrack even(X) \Rightarrow [\forall Y \ even(times(X,Y))] \land \lbrack [\forall Y \ even(times(X,Y))] \Rightarrow even(X) \rbrack]\).

After step 2: \(\forall X \ [\lbrack \neg even(X) \lor [\forall Y \ even(times(X,Y))] \land \lbrack [\forall Y \ even(times(X,Y))] \lor even(X) \rbrack]\).

After step 3: \(\forall X \ [\lbrack \neg even(X) \lor [\forall Y \ even(times(X,Y))] \land \lbrack [\exists Y \ \neg even(times(X,Y))] \lor even(X) \rbrack]\).

After step 4: \(\forall X \ [\lbrack \neg even(X) \lor [\forall Y \ even(times(X,Y))] \land \lbrack \neg even(times(X,sk1(X))) \lor even(X) \rbrack]\).

After step 5: \(\lbrack \neg even(X) \lor even(times(X,Y)) \rbrack \land \lbrack \neg even(times(X,sk1(X))) \lor even(X) \rbrack\).

Step 6 has no effect.

After step 7: \(\neg even(X) \lor even(times(X,Y)).\)
\(\neg even(times(X,sk1(X))) \lor even(X).\)

Resolution

A \textit{substitution} is an association of variables with terms;

Example: \(\sigma = \{ \ X \rightarrow a, Y \rightarrow f(Z) \ \}\) is a substitution.

The \textit{application} of a substitution \(\sigma\) to a clause \(\phi\), written \(\phi\sigma\), is the clause that is obtained when each occurrence in \(\phi\) of a variable in \(\sigma\) is replaced by the associated term.
Example: If $\phi$ is the clause $p(X,Y) \lor \neg q(Y,Z)$, and $\sigma$ is the substution above, then $\phi\sigma$ is $p(a,f(Z)) \lor q(f(Z),Z)$.

Fact: If $\phi$ is true, then $\phi\sigma$ is true.

Let $\alpha$ and $\beta$ be atomic formulas. $\alpha$ and $\beta$ are unifiable if there are substitutions $\sigma_A$ and $\sigma_B$ such that $\alpha\sigma_A = \beta\sigma_B$.

Examples. “$p(a,b)$” is unifiable with “$p(X,Y)$” under the substitution $\sigma_B = \{ X \rightarrow a, Y \rightarrow b \}$

“$p(a,b)$” is not unifiable with “$p(X,X)$”.

“$p(a,Z)$” is unifiable with “$p(Z,b)$” under the substitutions $\sigma_A = \{ Z \rightarrow b \}$, $\sigma_B = \{ Z \rightarrow a \}$.

“$p(f(X),W)$” is unifiable with “$p(Z,Z)$” under the substitutions $\sigma_A = \{ W \rightarrow f(X) \}$, $\sigma_B = \{ Z \rightarrow f(X) \}$.

“$p(f(X),X)$” is not unifiable with “$p(Z,Z)$”.

There may be more than one set of substitutions that unifies two formulas. For example “$p(a,f(a),X)$” can be unified with “$p(a,f(a),Y)$” by substituting $X$ for $Y$, or by substituting “a” for both $X$ and $Y$, or by substituting $f(a)$ for both $X$ and $Y$, or by substituting “$f(W)$” for both $X$ and $Y$ ... However, the best way to unify them is to substitute $X$ for $Y$ (or vice versa), because all the other substitutions can be derived by further substitutions from it. It is called the most general unifier (mgu).

Resolution: Rules of Inference

1. (Factoring) Let $\phi$ be the clause $\alpha_1 \lor \alpha_2 \lor \ldots \lor \alpha_k$. Let $\alpha_i$ and $\alpha_j$ be two literals that are either both positive or both negative, and let $\sigma$ be a single substitution that unifies $\alpha_i$ and $\alpha_j$. Then infer $(\phi - \alpha_j)\sigma$.

Example: From “$p(a,X) \lor p(Y,b) \lor q(X,Y,c)$” infer “$p(a,b) \lor q(b,a,c)$”.

2. (Resolution) Let $\phi$ be the clause $\alpha_1 \lor \alpha_2 \lor \ldots \lor \alpha_k$, and let $\psi$ be the clause $\beta_1 \lor \beta_2 \lor \ldots \lor \beta_m$. Suppose that $\alpha_i = \gamma$ and $\beta_j = \neg \delta$, where $\gamma$ and $\delta$ are atomic and where $\gamma$ unifies with $\delta$ under the substitutions $\sigma_A$ and $\sigma_B$. Then infer $(\phi - \alpha_i)\sigma_A \lor (\psi - \beta_j)\sigma_B$.

Examples: From “$p(a,b) \lor q(b,c)$” and “$\neg p(X,Y) \lor r(X,Y)$” infer “$q(b,c) \lor r(a,b)$”.

From “man(socrates)” and “$\neg$man($X$) $\lor$ mortal($X$)”, infer “mortal(socrates)”.

From “man(socrates)” and “$\neg$man($X$)” infer the empty clause.

Fact: $\Delta$ is an inconsistent set of clauses if and only if there is a derivation of the empty clause from $\Delta$ using
Resolution: Proof Technique

To prove sentence $\phi$ from a set of axioms $\Gamma$:

Step 1. Set $\Delta = \Gamma \cup \{\neg \phi\}$;

Step 2. Skolemize $\Delta$;

Step 3. Keep applying rules 1 and 2 to derive new sentences. If you succeed in deriving the empty clause, then $\phi$ is provable from $\Gamma$. If there is no way to derive the empty clause, then $\phi$ is not provable.

Example:

Given: 1. $\forall S_1,S_2 \text{ subset}(S_1, S_2) \leftrightarrow [\forall X \text{ member}(X, S_1) \Rightarrow \text{member}(X, S_2)]$.
Prove: $H. \forall S_1,S_2,S_3 \text{ [subset}(S_1, S_2) \wedge \text{subset}(S_2, S_3)] \Rightarrow \text{subset}(S_1, S_3)$.

Negation of $H$: 2. $\neg[\forall S_1,S_2,S_3 \text{ [subset}(S_1, S_2) \wedge \text{subset}(S_2, S_3)] \Rightarrow \text{subset}(S_1, S_3)]$.

Converted to clausal form:

1a. $\neg \text{subset}(S_1, S_2) \vee \neg \text{member}(X, S_1) \vee \text{member}(X, S_2)$.
1b. $\text{member}(sk0(S_1, S_2), S_1) \vee \text{subset}(S_1, S_2)$.
1c. $\neg \text{member}(sk0(S_1, S_2), S_2) \vee \text{subset}(S_1, S_2)$.
2a. $\text{subset}(sk1, sk2)$.
2b. $\text{subset}(sk2, sk3)$.
2c. $\neg \text{subset}(sk1, sk3)$.

From 2a and 1a infer 3. $\neg \text{member}(X, sk1) \vee \text{member}(X, sk2)$.
From 2b and 1a infer 4. $\neg \text{member}(X, sk2) \vee \text{member}(X, sk3)$.
From 3 and 4 infer 5. $\neg \text{member}(X, sk1) \vee \text{member}(X, sk3)$.
From 2c and 1b infer 6. $\text{member}(sk0(sk1, sk3), sk1)$.
From 2c and 1c infer 7. $\neg \text{member}(sk0(sk1, sk3), sk3)$.
From 6 and 5 infer 8. $\text{member}(sk0(sk1, sk3), sk3)$.
From 7 and 8 infer 9. The empty clause.