Resolution for Predicate Logic

The connection between general satisfiability and Herbrand satisfiability provides the basis for a refutational approach to first-order theorem proving.

Validity of a first-order sentence ϕ can be checked as follows.

1. First convert the negated formula $\neg\phi$ into a prenex form $Q_1 x_1 \ldots Q_n x_n \, \psi$.

2. Next skolemize to obtain a universal formula of the form $\forall x_1 \ldots \forall x_n \, \psi'$.

3. Then convert ψ' into conjunctive form, $\psi \wedge \ldots \wedge \psi$, where each formula ψ is a disjunction of literals.

   The formula $\forall x_1 \ldots \forall x_n \, \psi'$ is equivalent to

   $$(\forall x_1 \ldots \forall x_n \, \psi_1) \wedge \ldots \wedge (\forall x_1 \ldots \forall x_n \, \psi_m).$$

4. Finally, apply suitable "resolution" inferences to

   $$\psi_1, \ldots, \psi_m,$$

   interpreting each formula ψ as a clause.

   If a contradiction is derived, then $\neg\phi$ is unsatisfiable, which implies that ϕ is valid.

Renaming

A substitution is said to be a *renaming* if it is a one-to-one and onto function on the set of variables.

For example, the substitution

$$\sigma \quad = \quad [x \mapsto y, y \mapsto x]$$

is a renaming (in this case of the variables x and y).

We also call a substitution ρ a *renaming for a formula* ϕ if $\rho(x)$ and $\rho(y)$ are different variables, whenever x and y are different variables in ϕ.

If C is a clause and ρ a renaming for C, then C and $C\rho$ obviously have the same ground instances. Thus the Herbrand semantics of a clause does not change under renaming.

We will employ renaming substitutions, as well as (most general) unifiers, in defining inferences on clauses.

Binary Resolution

The following inference rule provides a strating point for a refutational theorem proving method for clauses.

Binary Resolution

$$\frac{C \vee A \qquad D \vee \neg B}{C\rho\sigma \vee D\sigma}$$

where ρ is a renaming substitution, such that the renamed first premise has no variables in common with the second premise, and σ is a most general unifier of $A\rho$ and B.

For example,

$$\frac{P(fx, x) \qquad \neg P(x, fy) \vee \neg P(x, z)}{\neg P(ffy, z)}$$

is a binary resolution inference.

Example – Resolution

Consider the set of the following clauses:

$$\neg P(x, y) \vee P(y, x) \tag{1}$$
$$\neg P(x, y) \vee \neg P(y, z) \vee P(x, z) \tag{2}$$
$$P(x, fx) \tag{3}$$
$$\neg P(a, a) \tag{4}$$

We use resolution to show that this set is unsatisfiable.

From clauses (3) and (1) we obtain

$$P(fx_1, x_1) \tag{5}$$

whereas from (3) and (2) we get

$$\neg P(fx_2, z) \vee P(x_2, z) \tag{6}$$

Resolving clauses (5) and (6) yields

$$P(x_3, x_3) \tag{7}$$

which can be resolved with (4) to yield a contradiction.

Factoring

Binary resolution by itself is not sufficient to obtain a contradiction from every unsatisfiable set of clauses, but needs to be supplemented by the following rule.

Positive Factoring

$$\frac{C \vee A \vee B}{C\sigma \vee A\sigma}$$

if σ is a most general unifier of the positive literals A and B.

The following inference rule is not necessary (for refutational completeness), but can also be used.

Negative Factoring

$$\frac{C \vee \neg A \vee \neg B}{C\sigma \vee \neg A\sigma}$$

if σ is a most general unifier of A and B.

Resolution

Factoring can also be integrated into resolution inferences.

General Resolution

$$\frac{C \vee A_1 \vee \cdots \vee A_k \qquad D \vee \neg B_1 \vee \cdots \vee \neg B_l}{C\rho\sigma \vee D\sigma}$$

where ρ is a renaming substitution for the first premise such that the renamed clause has no variables in common with the second premise, and σ is a most general unifier of $\{A_1\rho, \ldots, A_k\rho, B_1, \ldots, B_l\}$.

For example,

$$\frac{R(ffx, x) \qquad \neg R(y, z) \vee R(fy, fz)}{R(fffx, fx)}$$

is a resolution inference. In this example, no renaming is necessary (and hence ρ may be the identity function) and the atoms to be unified are $R(ffx, x)$ and $R(y, z)$, which have most general unifier $\sigma = [y \mapsto ffx, z \mapsto x]$.

Another Example

Recursively define terms $f^n(x)$ by: (i) $f^1(x) = f(x)$ and (ii) $f^n(x) = f(f^{n-1}(x))$, if $n > 1$.

Let S^n denote the set of two clauses,

$$\begin{array}{l} P(x) \vee P(f(x)) \\ \neg P(y) \vee \neg P(f^n(y)) \end{array}$$

For example, S^1 consists of these clauses:

$$\begin{array}{l} P(x) \vee P(f(x)) \\ \neg P(y) \vee \neg P(fy) \end{array}$$

This set is satisfiable, as there is a Herbrand interpretation in which all ground instances of both clauses are true.

(Consider a language with a single constant a. Then the Herbrand universe consists of the terms

$$a, fa, ffa, \ldots$$

Define P^I in such a way that $P^I(t)$ is true if, and only if, t is a term $f^n(a)$, for which n is an odd number. This Herbrand model I satisfies S^1.)

Example (cont.)

Next take the set S^2:

$$P(x) \vee P(f(x)) \tag{1}$$
$$\neg P(y) \vee \neg P(fy) \tag{2}$$

Resolving (1) and (2) yields

$$\neg P(x) \vee P(f(x)) \tag{3}$$

From (3) and (2) we obtain

$$\neg P(x) \vee \neg P(f(x)) \tag{4}$$

Resolving (3) and (4) yields

$$\neg P(x) \vee \neg P(x) \tag{5}$$

From (1) and (5) we obtain

$$P(x) \tag{6}$$

Clauses (5) and (6) yield a contradiction, which implies that the set S^2 is unsatisfiable.

(Resolvents have been consistently renamed so as to simplify the presentation.)

Example (cont.)

The set S^6, it turns out, is also unsatisfiable. Here is a refutation by resolution.

$$
\begin{array}{lr}
P(x) \vee P(f(x)) & (1) \\
\neg P(x) \vee \neg P(f^6(x)) & (2) \\
\neg P(x) \vee P(f^5(x)) & (1),(2) : (3) \\
\neg P(x) \vee \neg P(f(x)) & (2),(3) : (4) \\
\neg P(x) \vee P(x) & (1),(4) : (5) \\
\neg P(x) \vee \neg P(f^4(x)) & (3),(4) : (6) \\
\neg P(x) \vee P(f^3(x)) & (1),(6) : (7) \\
\neg P(x) \vee \neg P(f(x)) & (3),(6) : (8) \\
\neg P(x) \vee \neg P(f^3(x)) & (2),(7) : (9) \\
\neg P(x) \vee \neg P(f^2(x)) & (4),(7) : (10) \\
\neg P(x) \vee \neg P(f(x)) & (6),(7) : (11) \\
\neg P(x) \vee P(f^2(x)) & (1),(9) : (12) \\
\neg P(x) \vee \neg P(f^2(x)) & (3),(9) : (13) \\
\neg P(x) & (7),(9) : (14) \\
P(x) & (1),(14) : (15) \\
\perp & (14),(15) : (16)
\end{array}
$$

Soundness of Resolution

The inferences we have described above–binary and general resolution and factoring—–are logically sound.

Theorem [Soundness]

> The conclusion of a resolution or factoring inference is true in every Herbrand model that satisfies the premises of the inference.

Instances of Inferences

An *instance* of a resolution inference

$$\frac{C \qquad\qquad D}{C'\rho\sigma \vee D'\sigma}$$

is any inference

$$\frac{C\rho\sigma\tau \qquad\qquad D\sigma\tau}{C'\rho\sigma\tau \vee D'\sigma\tau}$$

Similarly, by an instance of a factoring inference

$$\frac{C \vee L \vee L'}{C\sigma \vee L\sigma}$$

we mean any inference

$$\frac{C\sigma\tau \vee L\sigma\tau \vee L'\sigma\tau}{C\sigma\tau \vee L\sigma\tau}$$

If neither premises nor conclusion contain any variables, we speak of a *ground instance*.

Note that instances of a resolution inference need not themselves be resolution inferences (as inferences employ only most general unifiers).

Projection

The connection between (a) ground instances of inferences from arbitrary clauses and (b) inferences from ground instances of clauses, forms the basis of the refutational completeness of resolution plus factoring.

Lifting Lemma

> Let C and D be two clauses with no variables in common and $C\tau$ and $D\tau$ be corresponding ground instances.
>
> (i) If E' is the conclusion of a resolution inference with premises $C\tau$ and $D\tau$, then E' is a ground instance of a clause E that is the conclusion of a resolution inference with premises C and D.
>
> (ii) If E' is the conclusion of a factoring inference with premise $C\tau$, then E' is a ground instance of a clause E that is the conclusion of a factoring inference with premise C.

Refutational Completeness

A set of clauses N is *saturated* (with respect to given inference rules) if N contains the conclusion of every inference, the premises of which are elements of N.

The following lemma follows from the Lifting Lemma.

Lemma

> If a set of clauses N is saturated with respect to resolution and factoring, then the set of all ground instances of clauses in N is saturated with respect to ground resolution and factoring.

Observing that the empty clause is an instance of itself, but of no other clause, we obtain:

Corollary [Refutation Completeness]

> A set of clauses is unsatisfiable if and only if the empty clause can be derived from it by resolution and factoring.

Subsumption

Some clauses may be redundant in the sense that they are not needed for deriving a contradiction.

We say that a clause C *subsumes* another clause D if D can be written as $C\sigma$ or $C\sigma \vee C'$, for same substitution σ.

For example, the ground clause $P \vee Q$ subsumes $P \vee Q \vee \neg R$, and $P(x)$ subsumes $P(f(y)) \vee Q(y)$.

A clause C is redundant with respect to a set N if N contains a(nother) clause that subsumes C. Redundant clauses may be ignored in derivations.