Resolution

We want to automatically establish validity of one-sided sequents.

The most currently succesful approach are resolution and superposition.

The formulas in the sequents are first brought into clausal normal form.

After that, the sequent is extended by applying resolution. If the sequent is valid, this process eventually result in the formula $\perp$.

If the sequent is not valid, resolution may last forever.

Clause

A clause is a first-order formula of form

$$\forall x_1 \cdots x_n \ \pm A_1 \vee \cdots \vee \pm A_m,$$

where each each $\pm A_i$ is a literal, i.e. an atom or a negated atom with form $p(t_1, \ldots, t_k)$, with each t_j a term.

In general p can be equality as well, but for the moment not.

Clauses are usually written with the following conventions:

The quantifier $\forall$ is omitted. In order to recognize variables, they are written with capitals.

Since disjunction is conjunctive and idempotent, one can write $\pm A_1 \vee \cdots \vee \pm A_m$ either as set $\{\pm A_1, \ldots, \pm A_m\}$ or as multiset $[\pm A_1, \ldots, \pm A_m]$.

Multisets are useful in the completeness proof. In implementations, sets are better.

Examples

Consider the (provable) sequent

$$p(0),\ \forall x\ \neg p(x) \vee p(s(x)),\ \neg p(s^3(0)) \vdash$$

It can be written as

$$\{p(0)\},\ \{\neg p(X) \vee p(s(X))\},\ \{\neg p(s^3(0))\} \vdash$$

Sequent

$$\forall x\ R(x,x),\ \forall xy\ \neg R(x,y) \vee R(y,x),\ \forall xyz\ R(x,y) \wedge R(y,z) \to R(x,z) \vdash$$

can be written as

$$\{R(X,X)\},\ \{\neg R(X,Y), R(Y,X)\},\ \{\neg R(X,Y), \neg R(Y,Z), R(X,Z)\} \vdash$$

Resolution (Robinson)

Pick two clauses from the sequent. Rename them so that they are variable disjoint. If the resulting clauses can be written in the form $c_1 = \{A_1, \ldots, A_p\} \cup R_1$ and $c_2 = \{\neg B_1, \ldots, \neg B_q\} \cup R_2$, s.t. $A_1, \ldots, A_p, B_1, \ldots, B_q$ have a simultaneous most general unifier Θ, then $R_1\Theta \cup R_2\Theta$ is a resolvent of c_1 and c_2.

The resolvent can be added to the sequent.

Resolution and Factoring (Modern)

Pick two clauses from the sequent. Rename them so that they are variable disjoint. If the resulting clauses can be written in the form $c_1 = \{A\} \cup R_1$ and $c_2 = \{\neg B\} \cup R_2$, s.t. that A and B have a unifier Θ, then $R_1\Theta \cup R_2\Theta$ is a resolvent of c_1 and c_2.

Let $c = \{A, B\} \cup R$ be a clause that is present in the sequent. If A and B have a most general unifier Θ, then $\{A\Theta\} \cup R\Theta$ is a factor of c.

Resolvents and factors can be added to the sequent.

Correctness and Completeness

Theorem: Adding a resolvent or factor to a sequent does not make a non-provable sequent provable.

Theorem: If the sequent is provable, then resolution will eventually derive the empty clause $\perp$.

Correctness of Resolution

Proving correctness is more subtle than it seems at first, because of disappearing variables. Let $z_1, \ldots, z_k$ be the variables that occur in $A\Theta, R_1\Theta, B\Theta, R_2\Theta$.

In order to prove correctness, one first proves the sequent

$$\forall x_1 \cdots x_n \ A \vee R, \quad \forall y_1 \cdots y_m \ B \vee S \vdash \forall z_1 \cdots z_k \ R\Theta \vee S\Theta.$$

If some of the z_i does not occur in $R\Theta \vee S\Theta$ they must be substituted away. This results in the sequent

$$\forall z_1 \cdots z_k \ R\Theta \vee S\Theta \vdash \forall z'_1 \cdots z'_{k'} \ R\Theta \vee S\Theta.$$

Finding proper instantiations may be tricky when the terms are typed. After that, cut can be used.

Normal Form Transformation

Skolemization

Skolemization is called after Thoralf Skolem (1887-1963).

Let F be a formula that contains an existially quantified subformula. Write $F = F[\ \exists y{:}Y\ \ P\ ]$. Assume that $\exists y{:}Y\ \ P$ occurs only in the scope of $\vee, \wedge, \forall$.

Assume that $\exists y{:}Y\ \ P$ is in the scope of universal quantifiers $\forall x_1, \ldots, \forall x_n$.

Invent a new function symbol f with arity n and type $X_1 \times \cdots \times X_n \rightarrow Y$.

Replace $F[\ \exists y{:}Y\ \ P\ ]$ by $F[\ P[y := f(x_1, \ldots, x_n)]\ ]$.

Skolemization can be repeated until all $\exists$ are gone.

Skolemization (Improved)

If some variable x_i is not free in $\exists y{:}Y\ P$, then it can be omitted from $f(x_1, \ldots, x_n)$ on the condition that one is guaranteed to find an instance for it.

Write F in the form $F[\ \forall x_i{:}X_i\ Q\ ]$.

If $F[\ \exists x_i{:}X_i\ \top]$ holds, one can remove x_i as argument from the Skolem function.

Skolemization (3)

Theorem: Let F' be obtained by Skolemization of F.

Sequent $A_1, \ldots, A_p, F \vdash$ is provable (valid) iff $A_1, \ldots, A_p, F' \vdash$ is provable (valid).

The technicalities of the proof are very tricky. The intuition is that the function f can be defined in such a way that if $\exists y{:}Y\ P$ is true, then $f(x_1, \ldots, x_n)$ chooses a possible y.

One can also analyze the proofs, which is nicer, but even harder.

Subformula Replacement

Let F be a formula that contains a subformula A. Write $F = F[\ A\ ]$. Let $x_1, \ldots, x_n$ be the free variables of A. Let $X_1, \ldots, X_n$ be their types.

Invent a new predicate symbol p with arity n. Replace $F[\ A\ ]$ by two formulas

$$F[\ p(x_1, \ldots, x_n)\ ], \qquad \forall x_1{:}X_1 \cdots x_n{:}X_n \quad p(x_1, \ldots, x_n) \leftrightarrow A.$$

Subformula Replacement (Positive)

In case A occurs only in the scope of $\vee, \wedge, \forall, \exists$, then $F[\ A\ ]$ can be replaced by

$$F[\ p(x_1, \ldots, x_n)\ ], \qquad \forall x_1{:}X_1 \cdots x_n{:}X_n \quad p(x_1, \ldots, x_n) \rightarrow A.$$

Subformula Replacement (2)

Theorem: Let F_1, F_2 be obtained by subformula replacement in F.

The sequent $A_1, \ldots, A_p, F \vdash$ is provable (or valid) iff the sequent $A_1, \ldots, A_p,\ F_1, F_2 \vdash$ is provable (or valid).

Intuition is that predicate p can be defined in such a way that it agrees with A.

Antiprenexing

We can now give a complete CNF transformation for sequents:

Let $A_1, \ldots, A_p \vdash$ be a one sided sequent.

If one of the A_i contains a subformula of form $A \leftrightarrow B$, it can be replaced by $(\neg A \vee B) \wedge (A \vee \neg B)$.

If there are nested $\leftrightarrow$s, this may cause exponential increase of formula size.

This can be avoided by proper subformula replacement (the first version) (Example: $A \leftrightarrow (B \leftrightarrow (C \leftrightarrow D))$.)

Antiprenexing

After removal of $\leftrightarrow$, the negation rules (of one sided sequent calculus) can be applied without problems. The size of the formulas stays the same.

It is sometimes useful to do antiprenexing:

The following replacements can always be made when variable x is not free in A :

$$\begin{array}{rcl}
\exists x{:}\, X\ (A \wedge B) & \Rightarrow & A \wedge (\exists x{:}\, X\ \ B) \\
\exists x{:}\, X\ (B \wedge A) & \Rightarrow & (\exists x{:}\, X\ \ B) \wedge A \\
\forall x{:}\, X\ (A \vee B) & \Rightarrow & A \vee (\forall x{:}\, X\ \ B) \\
\forall x{:}\, X\ (B \vee A) & \Rightarrow & (\forall x{:}\, X\ \ B) \vee A
\end{array}$$

Antiprenexing (2)

In case $F[\ \exists x{:}\, X\ \top\ ]$ is true, the following replacements can be made in context $F[\ ]$:

$$
\begin{array}{rcl}
\forall x{:}\, X\ (A \wedge B) & \Rightarrow & A \wedge (\forall x{:}\, X\ \ B) \\
\forall x{:}\, X\ (B \wedge A) & \Rightarrow & (\forall x{:}\, X\ \ B) \wedge A \\
\exists x{:}\, X\ (A \vee B) & \Rightarrow & A \vee (\exists x{:}\, X\ \ B) \\
\exists x{:}\, X\ (B \vee A) & \Rightarrow & (\exists x{:}\, X\ \ B) \vee A
\end{array}
$$

The first two replacements are incorrect when type X is empty. The last two replacements can cause incompleteness when type X is empty.

CNF Transformation

When the formula has been antiprenexed, it can be Skolemized.

After that, apply the following rewrite rules:

$$
\begin{array}{lcl}
\forall x{:}\,X\ \ (A \wedge B) & \Rightarrow & (\forall x{:}\,X\ \ A) \wedge (\forall x{:}\,X\ \ B) \\
A \vee (B \wedge C) & \Rightarrow & (A \wedge B) \vee (A \wedge C) \\
(A \wedge B) \vee C & \Rightarrow & (A \wedge C) \vee (B \wedge C) \\
(\forall x{:}\,X\ \ A) \vee B & \Rightarrow & \forall x{:}\,X\ \ (A \vee B) \\
A \vee (\forall x{:}\,X\ \ B) & \Rightarrow & \forall x{:}\,X\ \ (A \vee B)
\end{array}
$$

Remove top level $\wedge$ from the sequent, and replace explicitly quantified variables by special variable symbols.

The last two rules can be applied in context $F[\ ]$ only when $F[\exists x{:}\,X\ \ \top]$ is true. Otherwise, subformula replacement must be used.

Example: Drinker

Let us try to prove the drinker paradox:

$$\exists x\ \top \vdash \exists x\ (\ D(x) \rightarrow \forall x\ D(x)\ ).$$

Assuming that there are people in the bar, there is somebody, s.t. if he drinks, then everyone drinks.

One sided sequent:

$$\exists x\ \top,\ \neg\exists x\ (\ D(x) \rightarrow \forall x\ D(x)\ ) \vdash .$$

NNF:

$$\exists x\ \top,\ \forall x\ (\ D(x) \wedge \exists x\ \neg D(x)\ ) \vdash$$

Drinker (2)

Antiprenexing is possible, because we have $\exists x\ \top$.

$$\exists x\ \top,\ \forall x\ D(x)\ \land\ \exists x\ \neg D(x) \vdash$$

Skolemization results in:

$$\exists x\ \top,\ \forall x\ D(x) \land \neg D(c) \vdash$$

These are the clauses:

$$\top,\ \{D(X)\},\ \{\neg D(c)\} \vdash$$

Without antiprenexing, unimproved Skolemization would have resulted in:

$$\top,\ \{D(X)\},\ \{\neg D(f(X))\} \vdash$$

Resolution-Based Proof Search

Once the sequent has been transformed into CNF, resolution can be applied until either the empty clause is obtained, no more clauses can be derived, or the stars stop shining.

Subsumption

Let c_1 and c_2 be two clauses.

We say that c_1 subsumes c_2 if there exists a substitution Θ, s.t. $c_1\Theta \subseteq c_2$, and some other restrictions apply.

Other restrictions can be $\|c_1\| \leq \|c_2\|$, or $\max(c_1) < \max(c_2)$.

If two clauses are equal (renamings of each other), they subsume each other.

If a sequent contains two clauses c_1, c_2 such that c_1 subsumes c_2, then the clause c_2 can be removed.

Implementation

Resolution is usually implemented by the given clause algorithm. (Invented by William McCune.)

The sequent $\Gamma \vdash$ is split into two parts:

- P passive clauses are not (yet) used for resolution.
- A active clauses can be used in resolution.

Initially, $A = \emptyset$, and $P = A$.

Given Clause Algorithm (2)

- If P is empty, then the original sequent $\Gamma \vdash$ has no proof.
- Otherwise, pick the lightest or oldest clause in P. Call this clause g, the given clause and remove it from P.
- Construct all possible resolvents between g and A, and possibly between g and itself.

  If this results in non-subsumed resolvents, then add them to P.
- Insert g into A.

Backward Subsumption

If g subsumes a clause in A or P, this clause can be removed. Applying subsumption in this way is called backward subsumption.

Doing this only for A is called the Kaiserslautern approach.

Applying backward subsumption A and P is called the Otter approach.

Complexity of Subsumption

Subsumption testing in general is NP complete.

Suppose one wants to prove the following sequent by resolution:

$$\forall x\ (\ p(x) \rightarrow q(c)\ ) \vdash [\ \neg \forall x (\neg\ p(x) \wedge \neg\ q(x))\ ] \rightarrow \exists y\ q(y).$$

Make the sequent one-sided:

$$\forall x\ (p(x) \rightarrow q(c)),\ [\ \neg \forall x (\neg\ p(x) \wedge \neg\ q(x))\ ] \wedge \neg \exists y\ q(y)\ \ \vdash$$

NNF:

$$\forall x\ (\ \neg p(x) \vee q(c)\ ),\ \exists x\ (\ p(x) \vee q(x)\ ) \wedge \forall y\ \neg q(y) \vdash$$

Skolemization:

$$\forall x\ (\ \neg p(x) \vee q(c)\ ),\ (\ p(d) \vee p(d)\ ) \wedge \forall y\ \neg\ q(y) \vdash$$

Clauses:

$$
\begin{array}{ll}
(1) & \{\neg\, p(X), q(c)\} \\
(2) & \{p(d), q(d)\} \\
(3) & \{\neg\, q(Y)\}.
\end{array}
$$

Then, there is the following resolution refutation: We give both the ground refutation, and the non-ground refutation:

(1)	$\{\neg\, p(d), q(c)\}$	$\{\neg p(X), q(c)\}$	(initial clause)
(2)	$\{p(d), q(d)\}$	$\{p(d), q(d)\}$	(initial clause)
(3)	$\{\neg\, q(d)\}$	$\{\neg\, q(Y)\}$	(initial clause)
(4)	$\{\neg\, q(c)\}$	$\{\neg\, q(Y)\}$	(initial clause)
(5)	$\{\neg\, p(d)\}$	$\{\neg\, p(Y)\}$	(from 1 and 4)
(6)	$\{q(d)\}$	{ q(d) }	(from 2 and 5)
(7)	$\{\}$	$\{\}$	(from 3 and 6)

Refinements

An L-order $\prec$ is an order on literals with the following property:

$$A \preceq B \text{ implies } A\Theta \preceq B\Theta, \text{ for all substitutions } \Theta.$$

This property is called liftability.

An A-order is an L-order with the following property:

$$A \prec B \text{ implies } \pm A \prec \pm B.$$

L-ordered Resolution

Let $\prec$ be an L-order. A literal A is maximal in its clause c if $A \in c$, and there is no literal $A' \in c$ with $A \preceq A'$

To the (modern) definition of resolution, add the following restrictions:

Literal A is maximal in c_1, literal B is maximal in c_2.

To factoring, add the following restriction: Literal A is maximal in c.

Theorem: L-ordered resolution + factoring is complete.