First-order resolution

Till Mossakowski

WiSe 2009/10
First-order resolution

- generalises propositional resolution to first-order logic
- is a proof system that is well-suited for efficient implementation
- many automated first-order provers are based on resolution: SPASS, Prover9, Vampire
- also interactive provers for higher-order logic are based on resolution: Isabelle, HOL, HOL-light
Logical consequence can be reduced to (un)satisfiability:

The logical consequence $\mathcal{T} \models S$ holds if and only if $\mathcal{T} \cup \{\neg S\}$ is unsatisfiable.

Note: Resolution is about satisfiability.
The sentence
\[ \forall x \exists y \text{Neighbor}(x, y) \]
is logically equivalent to the second-order sentence
\[ \exists f \forall x \text{Neighbor}(x, f(x)) \]
In first-order logic, we have the Skolem normal form
\[ \forall x \text{Neighbor}(x, f(x)) \]
Theorem about Skolem normal form

**Theorem**
A sentence $S \equiv \forall x \exists y P(x, y)$ is satisfiable iff its Skolem normal form $\forall x P(x, f(x))$ is.

Every structure satisfying the Skolem normal form also satisfies $S$. Moreover, every structure satisfying $S$ can be turned into one satisfying the Skolem normal form. This is done by interpreting $f$ by a function which picks out, for any object $b$ in the domain, some object $c$ such that they satisfy $P(x, y)$. 

Till Mossakowski
Logic
Unification of terms

\{ P(f(a)), \forall x \neg P(f(g(x))) \}\}

is satisfiable, but

\{ P(f(g(a))), \forall x \neg P(f(x)) \}\}

is not. This can be seen with unification.
Terms $t_1, \ldots, t_n$ are unifiable, if there is a substitution of terms for some or all the variables in $t_1, \ldots, t_n$ such that the terms that result from the substitution are syntactically identical terms.
Example

\[ f(g(z), x), \quad f(y, x), \quad f(y, h(a)) \]

are unifiable by substituting \( h(a) \) for \( x \) and \( g(z) \) for \( y \).
Prenex Normal Form

Goal: shift all quantifiers to the top-level

\[(\forall x P) \land Q \leadsto \forall x (P \land Q)\]  \[(\exists x P) \land Q \leadsto \exists x (P \land Q)\]

\[P \land (\forall x Q) \leadsto \forall x (P \land Q)\]  \[P \land (\exists x Q) \leadsto \exists x (P \land Q)\]

\[(\forall x P) \lor Q \leadsto \forall x (P \lor Q)\]  \[(\exists x P) \lor Q \leadsto \exists x (P \lor Q)\]

\[P \lor (\forall x Q) \leadsto \forall x (P \lor Q)\]  \[P \lor (\exists x Q) \leadsto \exists x (P \lor Q)\]

\[\neg \forall x P \leadsto \exists x (\neg P)\]  \[\neg \exists x P \leadsto \forall x (\neg P)\]

\[(\forall x P) \rightarrow Q \leadsto \exists x (P \rightarrow Q)\]  \[(\exists x P) \rightarrow Q \leadsto \forall x (P \rightarrow Q)\]

\[P \rightarrow (\forall x Q) \leadsto \forall x (P \rightarrow Q)\]  \[P \rightarrow (\exists x Q) \leadsto \exists x (P \rightarrow Q)\]

\[P \leftrightarrow Q \leadsto (P \rightarrow Q) \land (Q \rightarrow P)\]
The Prenex normal form algorithm assumes that all variables in a formula are distinct. This can be achieved by \( \alpha \)-renaming:

\[
\forall x P(x) \sim \forall y P(y) \\
\exists x P(x) \sim \exists y P(y)
\]
Suppose that we have a set $\mathcal{T}$ of sentences and want to show that they are not simultaneously first-order satisfiable.

1. Put each sentence in $\mathcal{T}$ into prenex form, say

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 \ldots P(x_1, y_1, x_2, y_2, \ldots)$$

2. Skolemize each of the resulting sentences, say

$$\forall x_1 \forall x_2 \ldots P(x_1, f_1(x_1), x_2, f_2(x_1, x_2), \ldots)$$

using different Skolem functions for different sentences.

3. Put each quantifier free matrix $P$ into conjunctive normal form, say

$$P_1 \land P_2 \land \ldots \land P_n$$

where each $P_i$ is a disjunction of literals.

4. Distribute the universal quantifiers in each sentence across the conjunctions and drop the conjunction signs, ending with a set of sentences of the form

$$\forall x_1 \forall x_2 \ldots P_i$$
5. Change the bound variables in each of the resulting sentences so that no variable appears in two of them.

6. Turn each of the resulting sentences into a set of literals by dropping the universal quantifiers and disjunction signs. In this way we end up with a set of resolution clauses.

7. Use resolution and unification to resolve this set of clauses

\[
\begin{align*}
\{C_1, \ldots, C_m\}, \{\neg D_1, \ldots, D_n\} \\
\{C_2\theta, \ldots C_m\theta, D_2\theta, \ldots, D_n\theta\}
\end{align*}
\]

if \( C_1\theta = D_1\theta \) (\( \theta \) is a unifier of \( C_1 \) and \( D_1 \))