First Order Logic - Resolution

1   Resolution for first-order logic

Resolution is not only a proof system for propositional logic but it is a sound and complete proof system for first order logic too:

- If Σ is a set of first order formulae and ϕ is a first order formula such that:

$$\text{for every interpretation } M, M = \Sigma \Rightarrow M = \phi$$

  then the proof system resolution will answer "yes" (completeness)

- If the proof system resolution answers "yes" on input (Σ, ϕ) then for every interpretation $M, M = \Sigma \Rightarrow M = \phi$ (soundness)

We shall only consider Resolution for first order logic very briefly.
We are given a set of first-order formulae Σ and another first order formula ϕ and we want to know whether for every interpretation M, if M satisfies each formula in Σ then M necessarily satisfies ϕ

That is, we want to know whether the conjunction of the formula in Σ, denoted Σ too, is such that $\Sigma \Rightarrow \phi$ is valid, i.e. holds in all interpretations

We describe how the Resolution proof system proceeds and given an example (but omit a more detailed discussion)

2   How we proceed

1. Form the conjunction of all formula in Σ and $\neg\phi$

2. Reduce the resulting formula to prenex normal form $Q_1 x_1 Q_2 x_2 \ldots Q_k x_k \psi$

3. For any existentially quantified variable x_i, replace each occurrence of x_i in ψ with the term $F_i\left(x_{j_1}, x_{j_2}, \ldots, x_{j_r}\right)$, where $X_{j_1}, X_{j_2}, \ldots, x_{jr}$ are the universally quantified variables appearing in the list $x_1, x_2, ..., x_{i-1}$ and F_i is a new function symbol

4. Delete the prefix of quantifiers $Q_1 x_1 Q_2 x_2 ... Q_k x_k$

5. Reduce the remaining quantifier-free formula to c.n.f and apply the generalised resolution rule to the resulting set of clauses

6. If we resolve the empty clause then we answer "yes". If we get to the point where we can resolve no more new clauses then we answer "no"
   We may keep on resolving for ever

2.1   Substitutions

Suppose that we have the two clauses:

$$A(F(x)) \vee L(G(x), x) \text{ and } \neg L(u, v) \vee \neg K(u, v)$$

We can unify the terms and variables $\{u/G(x), v/x\}$, make these substitutions throughout the clauses, and then resolve the resulting clauses:

$$A(F(x)) \vee L(G(x), x) \text{ and } \neg L(G(x), x) \vee \neg K(G(x), x)$$

resolve to yield $A(F(x)) \vee \neg K(G(x), x)$

2.2   An illustration

Suppose we know the following:

- Everyone who loves all animals is loved by someone

- Anyone who kills an animal is loved by no-one

- Jack loves all animals

- Either Jack or Curiosity killed the cat, whose name is Tuna

And we want to know: Did Curiosity kill the cat?
Translate to first order logic:

$$\forall x(\forall y(\text{ Animal }(y) \Rightarrow \text{ Loves }(x,y)) \Rightarrow \exists y(\text{ Loves }(y,x))) \tag{1}$$

$$\forall x(\exists y(\text{ Animal }(y) \wedge \text{ Kills }(x,y)) \Rightarrow \forall z(\neg \text{Loves}(z,x))) \tag{2}$$

$$\forall x(\text{ Animal }(x) \Rightarrow \text{ Loves }(\text{Jack},x)) \tag{3}$$

$$\text{Kills}(\text{Jack, Tuna}) \vee \text{Kills}(\text{Curiosity, Tuna}) \tag{4}$$

$$\text{Animal}(\text{Tuna}) \tag{5}$$

$$\text{Kills}(\text{Curiosity, Tuna}) \tag{6}$$

Convert to prenex normal form with the quantifier-free part in conjunctive normal form: $\forall x \exists y \exists z \forall u \forall v \forall w \forall t$

$$((\text{ Animal }(y) \vee \text{ Loves }(z,x)) \wedge (\neg \text{ Loves }(x,y) \vee \text{ Loves }(z,x))) \tag{7}$$

$$(\neg \text{ Animal}(v) \vee \neg \text{ Kills }(u,v)) \vee \neg \text{ Loves }(w,u)) \tag{8}$$

$$(\text{Animal}(t) \vee \text{Loves}(\textit{Jack},t)) \tag{9}$$

$$\text{Kills}(\textit{Jack},\textit{Tuna}) \vee \text{Kills}(\textit{Curiosity},\textit{Tuna}) \tag{10}$$

$$\text{Animal}(\textit{Tuna}) \tag{11}$$

$$\neg(\text{Kills}(\textit{Curiosity},\textit{Tuna})) \tag{12}$$

Replace existential quantifiers and remove universal quantifiers:

$$((\text{Animal}(F(x)) \vee \text{Loves}(G(x),x)) \wedge (\neg \text{Loves}(x,F(x)) \vee \text{Loves}(G(x),x))) \tag{13}$$

$$\neg \text{Animal}(v) \vee \neg \text{Kills}(u,v) \vee \neg \text{Loves}(w,u) \tag{14}$$

$$\text{Animal}(t) \vee \text{Kills}(\textit{Curiosity},\textit{Tuna}) \tag{15}$$

$$\text{Animal}(\textit{Tuna}) \tag{16}$$

$$\neg(\text{Kills}(\textit{Curiosity},\textit{Tuna})) \tag{17}$$

Apply Resolution

$$((\text{ Animal }(F(x)) \vee \text{Loves}(G(x),x)) \wedge (\neg \text{Loves}(x,F(x)) \vee \text{Loves}(G(x),x))) \tag{18}$$

$$(\neg \text{ Animal }(v) \vee \neg \text{ Kills }(u,v)) \vee \neg \text{ Loves }(w,u)) \tag{19}$$

$$(\text{Animal}(t) \vee \text{Loves}(Jack,t)) \tag{20}$$

$$\text{Kills}(Jack,Tuna) \vee \text{Kills}(Curiosity,Tuna) \tag{21}$$

$$\text{Animal}(Tuna) \tag{22}$$

$$\neg(\text{Kills}(Curiosity,Tuna)) \tag{23}$$

$$v/Tuna \rightarrow \neg Kills(u,Tuna) \vee \neg \text{Loves}(w,u) \tag{24}$$

$$\rightarrow \text{Kills}(Jack,Tuna) \tag{25}$$

$$u/Jack \rightarrow \neg Loves(w,Jack) \tag{26}$$

$$x/Jack, t/F(Jack) \rightarrow \neg \text{ Animal}(F(Jack)) \vee \text{Loves}(G(Jack),Jack) \tag{27}$$

$$x/Jack \rightarrow \text{Loves}(G(Jack),Jack) \tag{28}$$

$$w/G(Jack) \Rightarrow \{\} \tag{29}$$

So curiosity Killed the cat

3   Some comments on resolution

There might appear to be a mismatch between the undecidability of the validity problem for first order logic:

"There is no computer program that computes whether any given first-order formula is valid; that is, is satisfied by all interpretations."

and the soundness and completeness of the Resolution proof system for first order logic:

"If a first-order formula is valid then Resolution answers 'yes', and if Resolution answers 'yes' then the first-order formula is valid."

However, this is not the case; for unlike in propositional logic Resolution in first-order logic might not halt when a formula is not valid. Resolution is sometimes said to be semi-decidable.

The Resolution proof system lies at the heart of many modern-day theorem provers.