Resolution for First-Order Logic

Viktor Kuncak, EPFL

https://lara.epfl.ch/w/fv

First-Order Logic Syntax and Terminology

A first-order *signature* (akka language) specifies a set of function symbols (constants are functions symbols that take zero arguments), and predicate symbols.
Syntax of formulas (F) and terms (t) in first-order logic with equality:

$$F ::= p(t_1, \ldots, t_n) \mid t_1 = t_2 \mid \top \mid \bot \mid \neg F \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid F_1 \rightarrow F_2 \mid F_1 \longleftrightarrow F_2$$
$$t ::= x \mid c \mid f(t_1, \ldots, t_n)$$

where $x \in Var$ denotes variables, c denotes constants, f are function symbols and p are predicate symbols.
ar denotes arity of functions and predicate symbols; e.g. $ar(f) = 2$ means f takes two arguments, so it is allowed to form a term $f(t_1, t_2)$, and also $ar(p) = 2$ for predicate symbol p means that it is allowed to form formula $p(t_1, t_2)$.

We call $p(t_1, \ldots, t_n)$ an *atomic formula* (contains no logical connectives or quantifiers).
A *literal* is an atomic formula or its negation.
A *clause* is a disjunction of literals, e.g. $\neg p(x, f(y)) \vee q(y) \vee \neg r(x, z)$

Semantics

A first-order *interpretation* is $I = (D, e)$ where $D \neq \emptyset$ and e maps constants, function and predicate symbols as follows:

- each constant c into element of D, i.e. $e(c) \in D$
- each function symbol f with $ar(f) = n$ into a total function of n arguments, $e(f) : D^n \to D$, i.e. a subset of D^{n+1} such that for all $d_1, \ldots, d_n \in D$ there exists exactly one $d \in D$ with $(d_1, \ldots, d_n, d) \in e(f)$.
- each predicate symbol p with $ar(p) = n$ into an n-ary relation $e(p) \subseteq D^n$

We have defined $I \models F$ to mean that formulas F is true in interpretation I and $I \not\models F$ to mean that it is not the case that $I \models F$. We may also use notation $[\![F]\!]_I = 1$ to mean $I \models F$ and $[\![F]\!]_I = 0$ to mean $I \not\models F$, so that, e.g.,

$$[\![F_1 \wedge F_2]\!]_I = [\![F_1]\!]_I \wedge [\![F_2]\!]_I$$

as for propositional logic. For quantifiers we have, e.g.,

$$[\![\forall x.F]\!]_{(D,e)} = \forall d \in D.\ [\![F]\!]_{(D,e[x:=d])}$$

What Makes Logic First-Order

$$[\![\forall x.F]\!]_{(D,e)} = \forall d \in D.\ [\![F]\!]_{(D,e[x:=d])}$$

We *can* quantify over variables $\forall x.F$, $\exists x.F$, which are interpreted over D, and we can nest quantifiers, e.g. $\forall x.\exists y.\ (p(x,y) \land q(y,x))$.

We *cannot* write a FOL formula that quantifies over function and relation symbols.

The meaning of function and relation symbols is fixed in e of interpretation $I = (D,e)$.

To make general statements, we use concepts of *satisfiability* and *validity*:

- F is **valid** if, **for all interpretations** (D,e) (for arbitrarily large sets D and all possible choices of e), $[\![F]\!]_{(D,e)} = 1$
- F is **satisfiable** if **there exists an interpretation** (D,e) such that $[\![F]\!]_{(D,e)} = 1$

Satisfiability and Validity Illustration

Take first-order logic (FOL) formula

$$\forall x.\exists y.\ (p(x,y) \land q(y,x))$$

Its **satisfiability** is a statement:

$$\exists \mathbf{D} \neq \emptyset.\ \exists \mathbf{p},\mathbf{q} \subseteq \mathbf{D}^2.\ \forall x \in D.\exists y \in D.\ (x,y) \in p \land (y,x) \in q$$

Its **validity** is a statement:

$$\forall \mathbf{D} \neq \emptyset.\ \forall \mathbf{p},\mathbf{q} \subseteq \mathbf{D}^2.\ \forall x \in D.\exists y \in D.\ (x,y) \in p \land (y,x) \in q$$

So, the domain, functions, and relations are either all existentially quantified (if we ask about satisfiability) or all universally quantified (if we ask about validity).

Observation: F is valid if and only if $\neg F$ is not satisfiable.

Example

Consider this formula

$$
\begin{aligned}
&(\forall x.\exists y.\ R(x,y)) \land \\
&(\forall x.\forall y.\ R(x,y) \rightarrow \forall z.\ R(x,f(y,z))) \land \\
&(\forall x.\ P(x) \lor P(f(x,a))) \\
&\rightarrow \forall x.\exists y.\ R(x,y) \land P(y)
\end{aligned}
$$

We are interested in checking its validity of this formula.
We will check the satisfiability of its negation:

$$
\begin{aligned}
&(\forall x.\exists y.\ R(x,y)) \land \\
&(\forall x.\forall y.\ R(x,y) \rightarrow \forall z.\ R(x,f(y,z))) \land \\
&(\forall x.\ P(x) \lor P(f(x,a))) \land \\
&\neg \forall x.\exists y.\ R(x,y) \land P(y)
\end{aligned}
$$

Negation Normal Form for FOL

Observation: If $F \leftrightarrow G$ is a valid FOL formula, then inside any other FOL formula H we can replace a sub-formula F with G without changing the truth value of the formula: $H[F] \rightsquigarrow H[G]$.

We can transform formulas to negation normal using these transformations:

$$
\begin{array}{rcl}
F_1 \leftrightarrow F_2 & \rightsquigarrow & (F_1 \rightarrow F_2) \wedge (F_2 \rightarrow F_1) \\
F_1 \rightarrow F_2 & \rightsquigarrow & \neg F_1 \vee F_2 \\
\neg\neg F & \rightsquigarrow & F \\
\neg(F_1 \wedge F_2) & \rightsquigarrow & \neg F_1 \vee \neg F_2 \\
\neg(F_1 \vee F_2) & \rightsquigarrow & \neg F_1 \wedge \neg F_2 \\
\neg \forall x. F & \rightsquigarrow & \exists x. \neg F \\
\neg \exists x. F & \rightsquigarrow & \forall x. \neg F \\
\neg \bot & \rightsquigarrow & \top \\
\neg \top & \rightsquigarrow & \bot
\end{array}
$$

In negation normal form, negation applies only to atomic formulas and the only other propositional connectives are $\wedge$, $\vee$.

Compute Negation Normal Form

$$
\begin{array}{l}
(\forall x.\exists y.\ R(x,y)) \wedge \\
(\forall x.\forall y.\ R(x,y) \rightarrow \forall z.\ R(x,f(y,z))) \wedge \\
(\forall x.\ P(x) \vee P(f(x,a))) \wedge \\
\neg \forall x.\exists y.\ R(x,y) \wedge P(y)
\end{array}
$$

Introducing a Skolem Function

Observe that e.g. the following formula is valid:

$$(\forall x.p(x,f(x))) \to (\forall x.\exists y.p(x,y))$$

Indeed, fix any interpretation (D,e) and assume $\forall x.p(x,f(x))$. To prove $\forall x.\exists y.p(x,y)$, assume x to be arbitrary and let y be equal to $f(x)$.

A sort of converse is also true. Take any interpretation (D,e) in which $\forall x.\exists y.p(x,y)$ is true. Then, for every $x_d \in D$ there exists $y_d \in D$ such that $(x_d,y_d) \in e(p)$. Construct (by axiom of choice) a set $\bar{f}$ that contains, for every element $x_d \in D$ exactly one pair (x_d,y_d) where $y_d \in D$, picking y_d such that $(x_d,y_d) \in e(p)$. Thus $\bar{f}$ is a total function, $\bar{f} : D \to D$. Extend the signature with a **new function symbol** f (Skolem function, from (W) Thoralf Skolem) that does not appear in the formula. Define a new interpretation $I' = (D,e')$ (on the same domain) where $e' = e \cup \{(f,\bar{f})\}$, that is, e' behaves like e but maps a new symbol f to the function $\bar{f}$.
Then $[\![\forall x.p(x,f(x))]\!]_{I'} = 1$. Thus, two formulas have same satisfiability.

Prenex Normal Form

Once in negation-normal form, we can pull quantifiers to the top level of the formula.

Skolemization

In a formula that is in prenex and negation normal form, replace a subformula

$$\forall x_1,\ldots,x_n.\exists y.\ F(x_1,\ldots,x_n,y)$$

with

$$\forall x_1,\ldots,x_n.\ F(x_1,\ldots,x_n,g(x_1,\ldots,x_n))$$

where g is a new function symbol (Skolem function) of arity n.

Optimization: we do not need formula to be in prenex form. If it is in negation-normal form, just introduce Skolem function whose arguments are the variables that are free in F and are universally quantified.

$$\begin{array}{rcl}
FV(c) & = & \emptyset,\ \ FV(x) = \{x\} \\
FV(f(t_1,\ldots,t_n)) & = & FV(t_1)\cup\ldots\cup FV(t_n) = FV(p(t_1,\ldots,t_n)) \\
FV(F_1 \land F_2) & = & FV(F_1)\cup FV(F_2) \\
FV(\neg F) & = & FV(F) \\
FV(\forall x.F) & = & FV(F)\setminus\{x\} = FV(\exists x.F)
\end{array}$$

Compute Skolem Normal Form

Conjunctive Normal Form for FOL

Given formula with only $\forall$ quantifiers in prenex form, we can transform quantifier free formula into conjunctive normal form, as for propositional logic:

$$\forall x_1, \ldots, x_n.(C_1 \wedge \ldots \wedge C_m)$$

The quantifiers can be moved to each C_i and those that do not occur in C_i can be dropped:

$$(\forall x_1, \ldots, x_n.C_1) \wedge \ldots \wedge (\forall x_1, \ldots, x_n.C_n)$$