3.5 Normal Forms and Skolemization

Study of normal forms motivated by

- reduction of logical concepts,
- efficient data structures for theorem proving.

The main problem in first-order logic is the treatment of quantifiers. The subsequent normal form transformations are intended to eliminate many of them.

Prenex Normal Form (Traditional)

Prenex formulas have the form

$$Q_1 x_1 \dots Q_n x_n \; F,$$

where F is quantifier-free and $Q_i \in \{\forall, \exists\}$; we call $Q_1 x_1 \dots Q_n x_n$ the *quantifier prefix* and F the *matrix* of the formula.

Computing prenex normal form by the reduction system $\Rightarrow_P$:

$$\begin{array}{rcl}
H[(F \leftrightarrow G)]_p & \Rightarrow_P & H[(F \rightarrow G) \wedge (G \rightarrow F)]_p \\
H[\neg QxF]_p & \Rightarrow_P & H[\overline{Q}x \neg F]_p \\
H[((QxF) \; \rho \; G)]_p & \Rightarrow_P & H[Qy(F\{x \mapsto y\} \; \rho \; G)]_p, \\
 & & \quad \rho \in \{\wedge, \vee\} \\
H[((QxF) \rightarrow G)]_p & \Rightarrow_P & H[\overline{Q}y(F\{x \mapsto y\} \rightarrow G)]_p, \\
H[(F \; \rho \; (QxG))]_p & \Rightarrow_P & H[Qy(F \; \rho \; G\{x \mapsto y\})]_p, \\
 & & \quad \rho \in \{\wedge, \vee, \rightarrow\}
\end{array}$$

Here y is always assumed to be some fresh variable and $\overline{Q}$ denotes the quantifier *dual* to Q, i. e., $\overline{\forall} = \exists$ and $\overline{\exists} = \forall$.

Skolemization

Intuition: replacement of $\exists y$ by a concrete choice function computing y from all the arguments y depends on.

Transformation $\Rightarrow_S$
(to be applied outermost, *not* in subformulas):

$$\forall x_1, \dots, x_n \exists y \, F \quad \Rightarrow_S \quad \forall x_1, \dots, x_n \, F\{y \mapsto f(x_1, \dots, x_n)\}$$

where f/n is a new function symbol (*Skolem function*).

Together: $F \Rightarrow_P^* \underbrace{G}_{\text{prenex}} \Rightarrow_S^* \underbrace{H}_{\text{prenex, no } \exists}$

Theorem 3.7 *Let F, G, and H as defined above and closed. Then*

(i) F and G are equivalent.

(ii) $H \models G$ but the converse is not true in general.

(iii) G satisfiable (w. r. t. Σ-Alg) $\Leftrightarrow$ H satisfiable (w. r. t. Σ'-Alg) where $\Sigma' = (\Omega \cup SKF, \Pi)$ if $\Sigma = (\Omega, \Pi)$.

The Complete Picture

$$
\begin{array}{llll}
F & \Rightarrow_P^* & Q_1 y_1 \ldots Q_n y_n \, G & (G \text{ quantifier-free}) \\
 & \Rightarrow_S^* & \forall x_1, \ldots, x_m \, H & (m \leq n,\ H \text{ quantifier-free}) \\
 & \Rightarrow_{OCNF}^* & \underbrace{\underbrace{\forall x_1, \ldots, x_m}_{\text{leave out}} \ \bigwedge_{i=1}^{k} \ \underbrace{\bigvee_{j=1}^{n_i} L_{ij}}_{\text{clauses } C_i}}_{F'} &
\end{array}
$$

$N = \{C_1, \ldots, C_k\}$ is called the *clausal (normal) form* (CNF) of F.
Note: The variables in the clauses are implicitly universally quantified.

Theorem 3.8 *Let F be closed. Then $F' \models F$. (The converse is not true in general.)*

Theorem 3.9 *Let F be closed. Then F is satisfiable iff F' is satisfiable iff N is satisfiable*

Optimization

The normal form algorithm described so far leaves lots of room for optimization. Note that we only can preserve satisfiability anyway due to Skolemization.

- size of the CNF is exponential when done naively; the transformations we introduced already for propositional logic avoid this exponential growth;
- we want to preserve the original formula structure;
- we want small arity of Skolem functions (see next section).

3.6 Getting Skolem Functions with Small Arity

A clause set that is better suited for automated theorem proving can be obtained using the following steps:

- replace beneficial subformulas
- produce a negation normal form (NNF)
- apply miniscoping
- rename all variables
- skolemize

Formula Replacement

The functions ν and $\bar{\nu}$ that give us an overapproximation for the number of clauses generated by a formula are extended to quantified formulas by

$$\nu(\forall x\, F) = \nu(\exists x\, F) = \nu(F),$$
$$\bar{\nu}(\forall x\, F) = \bar{\nu}(\exists x\, F) = \bar{\nu}(F).$$

The other cases are defined as for propositional formulas.

Introduce top-down fresh predicates for beneficial subformulas:

$$H[F]_p \ \Rightarrow_{\text{OCNF}} \ H[P(x_1, \dots, x_n)]_p \wedge \text{def}(H, p, P, F)$$

where $\{x_1, \dots, x_n\}$ are the free variables in F, P/n is a predicate new to $H[F]_p$, $\text{def}(H, p, P, F)$ is defined by

$$\begin{array}{ll} \forall x_1, \dots, x_n\ (P(x_1, \dots, x_n) \rightarrow F), & \text{if } \text{pol}(H, p) = 1, \\ \forall x_1, \dots, x_n\ (F \rightarrow P(x_1, \dots, x_n)), & \text{if } \text{pol}(H, p) = -1, \\ \forall x_1, \dots, x_n\ (P(x_1, \dots, x_n) \leftrightarrow F), & \text{if } \text{pol}(H, p) = 0, \end{array}$$

and $\nu(H[F]_p) > \nu(H[P(\dots)]_p \wedge \text{def}(H, p, P, F))$.

Negation Normal Form (NNF)

Apply the reduction system $\Rightarrow_{\text{NNF}}$:

$$H[F \leftrightarrow G]_p \;\Rightarrow_{\text{NNF}}\; H[(F \rightarrow G) \wedge (G \rightarrow F)]_p$$

if $\text{pol}(H,p) = 1$ or $\text{pol}(H,p) = 0$.

$$H[F \leftrightarrow G]_p \;\Rightarrow_{\text{NNF}}\; H[(F \wedge G) \vee (\neg G \wedge \neg F)]_p$$

if $\text{pol}(H,p) = -1$.

$$\begin{array}{rcl}
H[\neg Qx\; F]_p & \Rightarrow_{\text{NNF}} & H[\overline{Q}x\; \neg F]_p \\
H[\neg(F \vee G)]_p & \Rightarrow_{\text{NNF}} & H[\neg F \wedge \neg G]_p \\
H[\neg(F \wedge G)]_p & \Rightarrow_{\text{NNF}} & H[\neg F \vee \neg G]_p \\
H[F \rightarrow G]_p & \Rightarrow_{\text{NNF}} & H[\neg F \vee G]_p \\
H[\neg\neg F]_p & \Rightarrow_{\text{NNF}} & H[F]_p
\end{array}$$

Miniscoping

Apply the reduction system $\Rightarrow_{\text{MS}}$. For the rules below we assume that x occurs freely in F, F', but x does not occur freely in G:

$$\begin{array}{rcl}
H[Qx\,(F \wedge G)]_p & \Rightarrow_{\text{MS}} & H[(Qx\,F) \wedge G]_p \\
H[Qx\,(F \vee G)]_p & \Rightarrow_{\text{MS}} & H[(Qx\,F) \vee G]_p \\
H[\forall x\,(F \wedge F')]_p & \Rightarrow_{\text{MS}} & H[(\forall x\,F) \wedge (\forall x\,F')]_p \\
H[\exists x\,(F \vee F')]_p & \Rightarrow_{\text{MS}} & H[(\exists x\,F) \vee (\exists x\,F')]_p
\end{array}$$

Variable Renaming

Rename all variables in H such that there are no two different positions p, q with $H|_p = Qx\,F$ and $H|_q = Q'x\,G$.

Standard Skolemization

Apply the reduction system:

$$H[\exists x\,F]_p \Rightarrow_{\text{SK}} H[F\{x \mapsto f(y_1,\ldots,y_n)\}]_p$$

where p has minimal length,
$\{y_1,\ldots,y_n\}$ are the free variables in $\exists x\,F$,
and f/n is a new function symbol to H.