First-Order CNF Transformation

Similar to the propositional case, first-order resolution and superposition operate on clauses. In this section I show how any first-order sentence can be efficiently transformed into a CNF, preserving satisfiability. To this end all existentially quantified variables are replaced with so called Skolem functions. Similar to the renaming of subformulas this replacement preserves satisfiability only. Eventually, all variables in clauses are implicitly universally quantified.

More concretely, the acnf CNF transformation is algorithm from Section 2.5.3 is generalized to first-order logic with equality. The adiitional complications are:

(i) additional rules for the quantifiers,

(ii) the formula renaming technique is extended to cope with variables, and

(iii) removal of existential quantifiers through the introduction of *Skolem* functions.

The first two extra rules eliminate $\top$ and $\bot$ from first-order formula starting with a quantifier.

ElimTB13 $\chi[\{\forall,\exists\}x.\top]_p \Rightarrow_{\text{ACNF}} \chi[\top]_p$

ElimTB14 $\chi[\{\forall,\exists\}x.\bot]_p \Rightarrow_{\text{ACNF}} \chi[\bot]_p$

Next, in order to obtain a negation normal form with negation symbols in front of atoms only, the respective rules for pushing negations over the quantifiers are needed as well.

PushNeg4 $\chi[\neg\forall x.\phi]_p \Rightarrow_{\mathrm{ACNF}} \chi[\exists x.\neg\phi]_p$

PushNeg5 $\chi[\neg\exists x.\phi]_p \Rightarrow_{\mathrm{ACNF}} \chi[\forall x.\neg\phi]_p$

where $\{\forall, \exists\}x.\phi$ covers both cases $\forall x.\phi$ and $\exists x.\phi$.

The next step is to rename all variables such that different quantifiers bind different variables. This step is necessary to prevent a later on confusion of variables, once the quantifiers are dropped.

RenVar $\phi \Rightarrow_{\text{ACNF}} \phi\sigma$

for $\sigma = \{\}$

In first-order logic, the renaming of subformulas has to take care of variables as well. The notion of an obvious position remains unchanged. Therefore, the basic mechanism of renaming and the concept of a beneficial subformula is exactly the same as in propositional logic. The only difference is that renaming does introduce an atom in the free variables of the respective subformula.

When some formula ψ is renamed at position p an atom $P(\vec{x}_n)$, $\vec{x}_n = x_1, \ldots, x_n$ replaces $\psi|_p$ where $\text{fvars}(\psi|_p) = \{x_1 \ldots, x_n\}$.

The respective definition of $P(\vec{x}_n)$ becomes

$$\mathrm{def}(\psi, p, P(\vec{x}_n)) := \begin{cases} \forall \vec{x}_n.(P(\vec{x}_n) \rightarrow \psi|_p) & \text{if } \mathrm{pol}(\psi, p) = 1 \\ \forall \vec{x}_n.(\psi|_p \rightarrow P(\vec{x}_n)) & \text{if } \mathrm{pol}(\psi, p) = -1 \\ \forall \vec{x}_n.(P(\vec{x}_n) \leftrightarrow \psi|_p) & \text{if } \mathrm{pol}(\psi, p) = 0 \end{cases}$$

SimpleRenaming is changed accordingly.

SimpleRenaming $\phi \;\Rightarrow_{\text{ACNF}}$
$\phi[P_1(\vec{x_1}_{,j_1})]_{p_1}[P_2(\vec{x_2}_{,j_2})]_{p_2}\dots[P_n(\vec{x_n}_{,j_n})]_{p_n} \;\wedge\; \text{def}(\phi, p_1, P_1(\vec{x_1}_{,j_1})) \;\wedge$
$\dots \wedge$
$\text{def}(\phi[P_1(\vec{x_1}_{,j_1})]_{p_1}[P_2(\vec{x_2}_{,j_2})]_{p_2}\dots[P_{n-1}(\vec{x_{n-1}}_{,j_{n-1}})]_{p_{n-1}}, p_n, P_n(\vec{x_n}_{,j_n}))$
provided $\{p_1, \dots, p_n\} \subset \text{pos}(\phi)$ and for all $i, i+j$ either $p_i \parallel p_{i+j}$ or $p_i > p_{i+j}$ and where $\text{fvars}(\phi|_{p_i}) = \{x_{i,1}, \dots, x_{i,j_i}\}$ and all P_i are different and new to ϕ

In first-order logic the existential quantifiers are eliminated first by the introduction of Skolem functions. In order to receive Skolem functions with few arguments, the quantifiers are first moved inwards as far as passible. This step is called *mini-scoping*.

MiniScope1 $\chi[\forall x.(\psi_1 \circ \psi_2)]_p \Rightarrow_{\text{ACNF}} \chi[(\forall x.\psi_1) \circ \psi_2]_p$
provided $\circ \in \{\wedge, \vee\}$, $x \notin \text{fvars}(\psi_2)$

MiniScope2 $\chi[\exists x.(\psi_1 \circ \psi_2)]_p \Rightarrow_{\text{ACNF}} \chi[(\exists x.\psi_1) \circ \psi_2]_p$
provided $\circ \in \{\wedge, \vee\}$, $x \notin \text{fvars}(\psi_2)$

MiniScope3 $\chi[\forall x.(\psi_1 \wedge \psi_2)]_p \Rightarrow_{\text{ACNF}} \chi[(\forall x.\psi_1) \wedge (\forall x.\psi_2)\sigma]_p$
where $\sigma = \{\}$, $x \in (\text{fvars}(\psi_1) \cap \text{fvars}(\psi_2))$

MiniScope4 $\chi[\exists x.(\psi_1 \vee \psi_2)]_p \Rightarrow_{\text{ACNF}} \chi[(\exists x.\psi_1) \vee (\exists x.\psi_2)\sigma]_p$
where $\sigma = \{\}, x \in (\text{fvars}(\psi_1) \cap \text{fvars}(\psi_2))$

Skolemization replaces all existentially quantified variables by shallow Skolem function terms.

Skolemization $\chi[\exists x.\phi]_p \Rightarrow_{\mathrm{ACNF}} \chi[\phi\{x \mapsto f(y_1, \ldots, y_n)\}]_p$
provided there is no q, $q < p$ with $\phi|_q = \exists x'.\psi'$,
$\mathrm{fvars}(\exists x.\psi) = \{y_1, \ldots, y_n\}$, $f : \mathrm{sort}(y_1) \times \ldots \times \mathrm{sort}(y_n) \rightarrow \mathrm{sort}(x)$
is a new function symbol

3.9.1 Theorem (Skolemization Preserves Satisfiability)

A formula $\chi[\exists x.\phi]_p$ is satisfiable iff the formula $\chi[\phi\{x \mapsto f(y_1, \dots, y_n)\}]_p$ is, where χ is in negation normal form, p the maximal position of an existential quantifier, $\mathrm{fvars}(\exists x.\psi) = \{y_1, \dots, y_n\}$, and $\mathrm{arity}(f) = n$ is a new function symbol to ϕ, $f : \mathrm{sort}(y_1) \times \dots \times \mathrm{sort}(y_n) \to \mathrm{sort}(x)$.

Algorithm: 11 acnf(ϕ)

Input : A first-order formula ϕ.

Output: A formula ψ in CNF satisfiability preserving to ϕ.

whilerule *(***ElimTB1**(ϕ)*,...,***ElimTB14**(ϕ)*)* **do** ;

RenVar(ϕ);

SimpleRenaming(ϕ) on obvious positions;

whilerule *(***ElimEquiv1**(ϕ)*,***ElimEquiv2**(ϕ)*)* **do** ;

whilerule *(***ElimImp**(ϕ)*)* **do** ;

whilerule *(***PushNeg1**(ϕ)*,...,***PushNeg5**(ϕ)*)* **do** ;

whilerule *(***MiniScope1**(ϕ)*,...,***MiniScope4**(ϕ)*)* **do** ;

whilerule *(***Skolemization**(ϕ)*)* **do** ;

whilerule *(***RemForall**(ϕ)*)* **do** ;

whilerule *(***PushDisj**(ϕ)*)* **do** ;

return ϕ;

3.9.3 Theorem (Properties of the ACNF Transformation)

Let ϕ be a first-order sentence, then

1. acnf(ϕ) terminates
2. ϕ is satisfiable iff acnf(ϕ) is satisfiable