Normalization for Resolution

As in the case of propositional logic, resolution for first-order predicate logic is a refutation procedure which requires that the formulas be in conjunctive normal form (CNF).

However, there are additional complications which arise when dealing with first-order formulas.
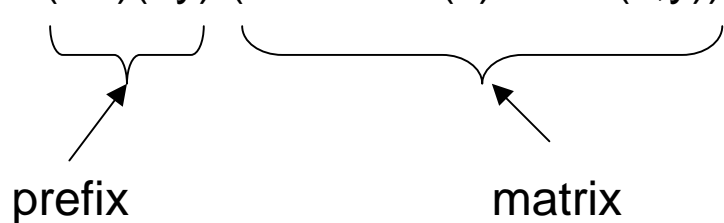
These slides address the preliminary process of converting an arbitrary first-order *sentence* into a form suitable for resolution.

Important: These techniques work only on sentences.

Prenex normal form:

Definition: A first order formula is in *prenex normal form* (PNF) if it consists of a string of quantifiers followed by a formula which does not contain any quantifiers.

Example:

$$(\forall x)(\exists y) \ (On\_table(x) \lor On(x,y))$$

prefix   matrix

The string of quantifiers is called the *prefix*, and the quantifier-free formula is called the *matrix.*

Example: The following formula is not in prenex normal form:

$$(\forall x)(On\_table(x) \lor (\exists y)On(x,y))$$

Example: The following formula is not in prenex normal form,

$$(\forall x)On\_table(x) \lor On(P1,B1)$$

while the following is:

$$(\forall x)(On\_table(x) \lor On(P1,B1))$$

Algorithm *Normalize*: Convert any first-order formula to an equivalent one which is in prenex normal form and fully prepared for resolution.

The algorithm consists of eight steps, which are best illustrated via example.

The running example is

$$(\forall x)(P(x) \rightarrow (((\forall y)((P(y) \rightarrow P(f(x,y))) \wedge$$
$$(\neg(\forall y)(Q(x,y) \rightarrow (P(y) \wedge R(c,y))))))$$

Note the following:
- x and y are variables.
- P is a unary relation symbol.
- Q and R are binary relation symbols.
- c is a constant symbol.
- f is a binary function symbol.

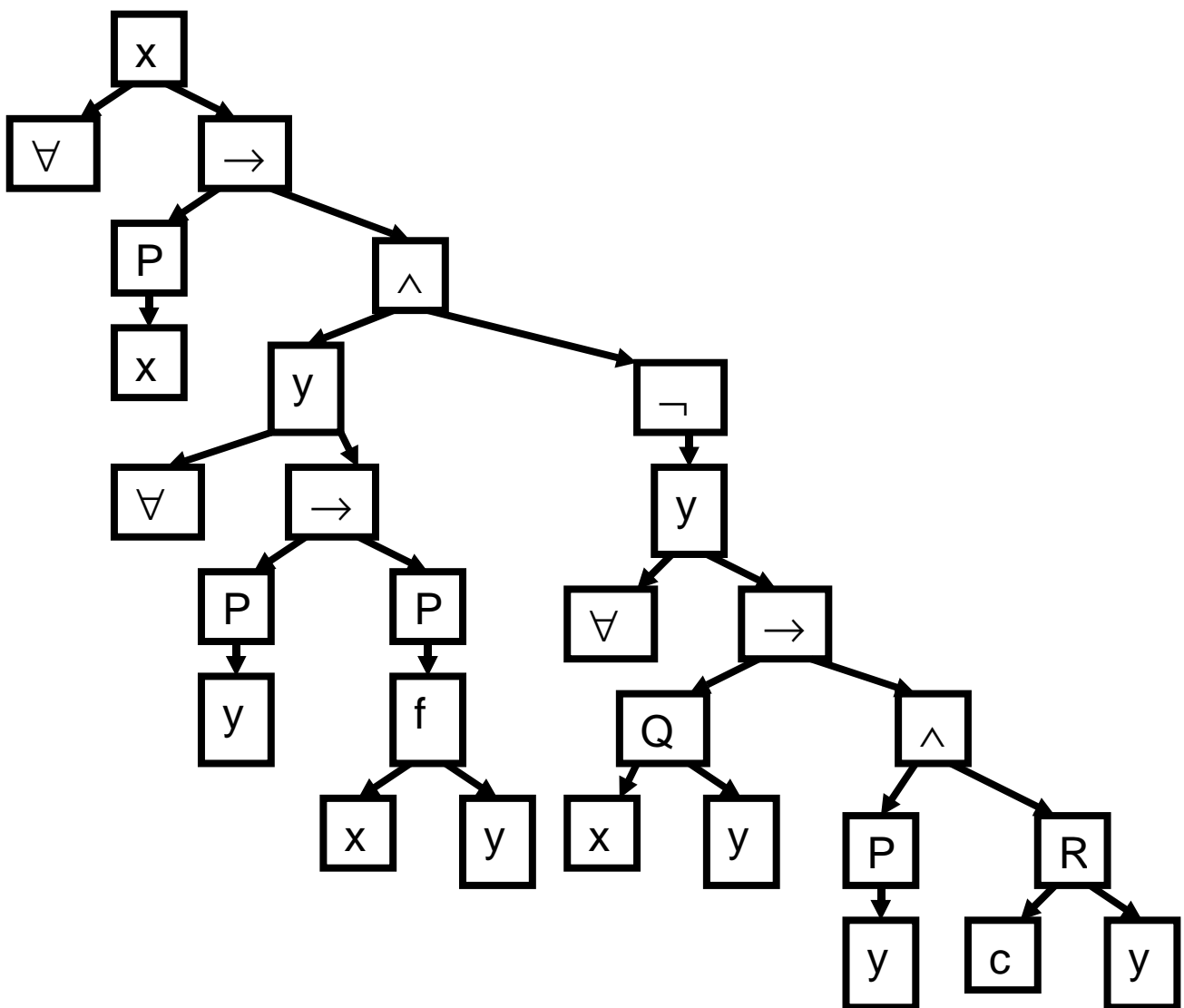It is helpful to begin by drawing the parse tree for this formula.

$$(\forall x)(P(x) \rightarrow ((\forall y)(P(y) \rightarrow P(f(x,y))) \wedge$$
$$(\neg(\forall y)(Q(x,y) \rightarrow (P(y) \wedge R(c,y))))))$$

x
∀   →
P   ∧
x   y   ¬
∀   →   y
P   P   ∀   →
y   f   Q   ∧
x y x y   P   R
y   c   y

Step Normalize1: Eliminate all $\rightarrow$ and $\leftrightarrow$ ($\equiv$) connectives, by using the appropriate identities.

$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge$
$\qquad (\neg(\forall y)(\neg Q(x,y) \vee (P(y) \wedge R(c,y)))))$

Step Normalize2: Reduce the scope of all negations to atoms.

Substep 1:

$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge$
$\qquad ((\exists y)(\neg(\neg Q(x,y) \vee (P(y) \wedge R(c,y))))))$

Substep 2:

$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge$
$\qquad ((\exists y)((Q(x,y) \wedge \neg(P(y) \wedge R(c,y))))))$

Substep 3:

$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge$
$\qquad ((\exists y)((Q(x,y) \wedge (\neg P(y) \vee \neg R(c,y))))))$

Simplify:

$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge$
$\qquad (\exists y)((Q(x,y) \wedge (\neg P(y) \vee \neg R(c,y)))))$

Step Normalize3: Standardize all variables. This means that variables are renamed so that none occurs in more than one quantifier.

Input formula:

$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge$
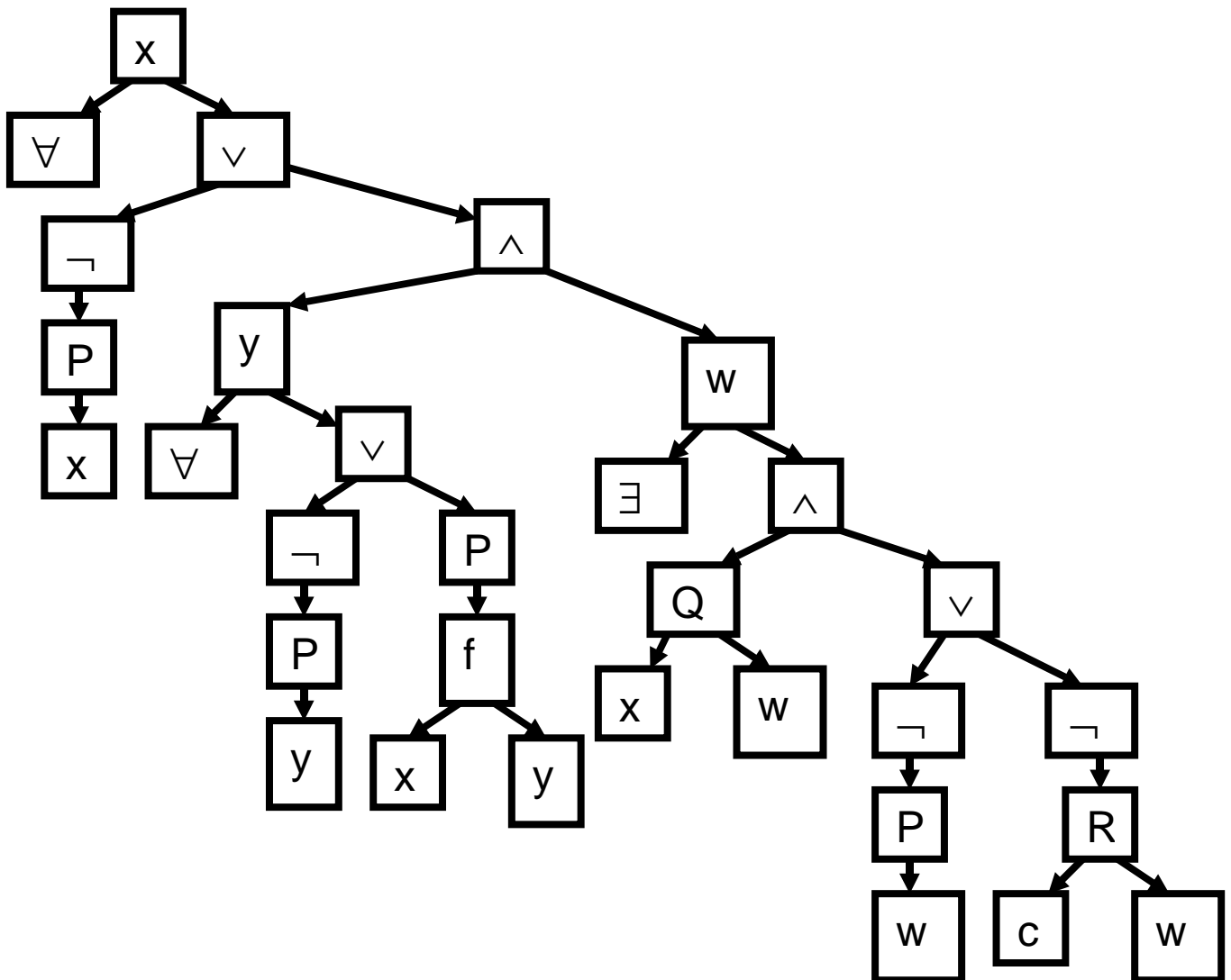$(\exists y)((Q(x,y) \wedge (\neg P(y) \vee \neg R(c,y)))))$

Result:

$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge$
$(\exists w)((Q(x,w) \wedge (\neg P(w) \vee \neg R(c,w)))))$

Step Normalize4: Pull all quantifiers out to the front.

Input formula:

$(\forall x)(\neg P(x) \lor ((\forall y)(\neg P(y) \lor P(f(x,y))) \land$
$(\exists w)((Q(x,w) \land (\neg P(w) \lor \neg R(c,w)))))$

It is convenient to examine the parse tree at this point:

We may bring out the quantifiers in any order which respects the quantifier nesting. In this case, there are two possibilities:

$(\forall x)(\forall y)(\exists w)$ and $(\forall x)(\exists w)(\forall y)$, so the two possible formulas are:

$(\forall x)(\forall y)(\exists w)(\neg P(x) \vee ((\neg P(y) \vee P(f(x,y)) \wedge$
$\qquad (Q(x,w) \wedge (\neg P(w) \vee \neg R(c,w)))))$

$(\forall x)(\exists w)(\forall y)(\neg P(x) \vee ((\neg P(y) \vee P(f(x,y)) \wedge$
$\qquad (Q(x,w) \wedge (\neg P(w) \vee \neg R(c,w)))))$

These are equivalent formula in prenex normal form.

Conjunctive normal form:

It is an easy matter to convert a prenex normal form formula to conjunctive normal form. The same techniques which work for propositional logic apply here, as the prefix is not involved. Continuing with the example.

Input formulas:

$(\forall x)(\forall y)(\exists w)(\neg P(x) \lor ((\neg P(y) \lor P(f(x,y)) \land$
$\qquad (Q(x,w) \land (\neg P(w) \lor \neg R(c,w)))))$

$(\forall x)(\exists w)(\forall y)(\neg P(x) \lor ((\neg P(y) \lor P(f(x,y)) \land$
$\qquad (Q(x,w) \land (\neg P(w) \lor \neg R(c,w)))))$

Step Normalize5: Convert to CNF equivalents.

$(\forall x)(\forall y)(\exists w)$
$\qquad ((\neg P(x) \lor \neg P(y) \lor P(f(x,y))) \land$
$\qquad\quad (\neg P(x) \lor Q(x,w)) \land$
$\qquad\quad (\neg P(x) \lor \neg P(w) \lor \neg R(c,w))$

$(\forall x)(\exists w)(\forall y)$
$\qquad ((\neg P(x) \lor \neg P(y) \lor P(f(x,y))) \land$
$\qquad\quad (\neg P(x) \lor Q(x,w)) \land$
$\qquad\quad (\neg P(x) \lor \neg P(w) \lor \neg R(c,w)))$

Skolemization:

Recall that universal quantifiers may be distributed over conjunction.

Thus, a formula such as

$$(\forall x)(\forall y)((A(x) \vee B(y)) \wedge (C(x) \vee D(y)))$$

is equivalent to

$$(\forall x)(\forall y)(A(x) \vee B(y)) \wedge (\forall x)(\forall y)(C(x) \vee D(y)),$$

which after renaming of variables is also equivalent to

$$(\forall x)(\forall y)(A(x) \vee B(y)) \wedge (\forall z)(\forall w)(C(z) \vee D(w)).$$

Thus, we have two clauses, each with their own quantifiers. They are not coupled by variables in any way.

However, if we have a formula such as

$$(\exists x)(\forall y)((A(x) \vee B(y)) \wedge (C(x) \vee D(y))),$$

the situation is more complex, because this formula is **not** equivalent to either

$$(\exists x)(\forall y)(A(x) \vee B(y)) \wedge (\exists x)(\forall y)(C(x) \vee D(y)),$$

or

$$(\exists x)(\forall y)(A(x) \vee B(y)) \wedge (\exists z)(\forall w)(C(z) \vee D(w)).$$

To obtain clauses which are not coupled by quantifiers, an ingenious device known as a Skolem function is used. (It is named after the Norwegian logician, Thoralf Skolem.)

In Skolemization, an existentially quantified variable is replaced by a function which identifies a specific instance of that variable. For example, to Skolemize

$$(\exists x)(\forall y)(A(x) \vee B(y)),$$

the variable x is replaced with a new constant symbol c_x. The resulting formula is

$$(\forall y)(A(c_x) \vee B(y)).$$

Think of c_x as "selecting" a domain element for which the formula is true. If there exists an x for which the formula is true, then in a model M, c_x^M is just such an element.

The following points are important:

- The Skolem constant is a *new* constant symbol which is added to the existing language. It is never an existing symbol.

- It can be shown that the Skolemized formula is satisfiable iff the original formula is.

Thus, to convert the following formula to a Skolemized form

$$(\exists x)(\forall y)((A(x) \lor B(y)) \land (C(x) \lor D(y))),$$

a new Skolem constant c_x is introduced. This yields the formula:

$$(\forall y)((A(c_x) \lor B(y)) \land (C(c_x) \lor D(y))),$$

which, after renaming the second y to w, may be viewed as the two independent clauses:

$$\{ (\forall y)(A(c_x) \lor B(y)), (\forall w) (C(c_x) \lor D(w)) \}$$

Although the two clauses are still coupled, it is now via a logical symbol, and not a quantifier. This is extremely important for the resolution process.

The previous technique works only if the existentially quantified variable is not within the scope of a universal quantifier. To motivate this, consider as an example the relation "$\leq$" on the natural numbers

$$\mathbb{N} = \{0, 1, 2, 3, \ldots\}.$$

Suppose that we state that

"$\mathbb{N}$ has a least number under $\leq$."

This may be represented by the formula

$$(\exists x)(\forall y)(\, x \leq y).$$

In a Skolemization of this formula, we need to pick a smallest element, for which the only choice in this case is zero. Thus,

$$(\forall y)(\, c_x \leq y),$$

with c_x identified with 0.

Now, suppose that we state that

"For every natural number, there is a bigger one."

This may be represented by the formula

$$(\forall x)(\exists y)(\ x \leq y).$$

In this case, simple Skolemization does not work. Rather, the Skolem value must depend upon x. The solution is to use a *Skolem function* which is dependent upon y. Let f_y be such a function. The Skolemized formula now becomes

$$(\forall x)(\ x \leq f_y(x)).$$

In the actual example, there are many choices for such an f_y. For example, $f_y(x) = x+1$ would work.

For an example such as

$$(\forall y)(\exists x)((A(x) \vee B(y)) \wedge (C(x) \vee D(y))).$$

the Skolemization is

$$(\forall y)((A(f_x(y)) \vee B(y)) \wedge (C(f_x(y)) \vee D(y))),$$

which yields the following set of clauses, after renaming the second y.

$$\{\ (\forall y)(A(f_x(y)) \vee B(y)),\ (\forall y)(C(f_x(y)) \vee D(y))\ \}$$

In general, a Skolem function must take as many arguments as there are universal quantifiers to its left.

Example:

$$(\forall y_1)(\forall y_2)(\forall y_3)(\exists x_1)(\forall y_4)(\exists x_2)R(y_1, y_2, y_3, y_4, x_1, x_2)$$

The Skolemization is

$$(\forall y_1)(\forall y_2)(\forall y_3)(\forall y_4)$$
$$(R(y_1, y_2, y_3, y_4, f_{x_1}(y_1,y_2,y_3), f_{x_2}(y_1,y_2,y_3,y_4)))$$

A full development of the theory will not be presented here, but the following result is stated.

Theorem: A Skolemization of a wff is satisfiable iff the original formula is. $\square$

Clausal form for resolution:

Now we return to the task of preparing a formula for resolution.

The original example formula was

$(\forall x)(P(x) \rightarrow (((\forall y)((P(y) \rightarrow P(f(x,y))) \wedge$
$(\neg(\forall y)(Q(x,y) \rightarrow (P(y) \wedge R(c,y))))))$

After manipulation, it was placed in CNF and PNF. There were two possibilities.

$(\forall x)(\forall y)(\exists w)$
$\quad ((\neg P(x) \vee \neg P(y) \vee P(f(x,y))) \wedge$
$\quad\quad (\neg P(x) \vee Q(x,w)) \wedge$
$\quad\quad (\neg P(x) \vee \neg P(w) \vee \neg R(c,w))$

$(\forall x)(\exists w)(\forall y)$
$\quad ((\neg P(x) \vee \neg P(y) \vee P(f(x,y))) \wedge$
$\quad\quad (\neg P(x) \vee Q(x,w)) \wedge$
$\quad\quad (\neg P(x) \vee \neg P(w) \vee \neg R(c,w)))$

The next step is to Skolemize.

Step Normalize6: Skolemize

The two possibilities become:

$(\forall x)(\forall y)$
$\quad$ ($(\neg P(x) \vee \neg P(y) \vee P(f(x,y)))$ $\wedge$
$\quad\quad$ $(\neg P(x) \vee Q(x, f_w(x,y)))$ $\wedge$
$\quad\quad$ $(\neg P(x) \vee \neg P(f_w(x,y)) \vee \neg R(c, f_w(x,y)))$

$(\forall x)(\forall y)$
$\quad$ ($(\neg P(x) \vee \neg P(y) \vee P(f(x,y)))$ $\wedge$
$\quad\quad$ $(\neg P(x) \vee Q(x, f_w(x)))$ $\wedge$
$\quad\quad$ $(\neg P(x) \vee \neg P(f_w(x)) \vee \neg R(c, f_w(x)))$)

Note that the second Skolemization is somewhat simpler than the first. The argument y in the Skolem function f_w of the first formula is superfluous, in that f_w does not really depend upon y.

Heurisitic optimization: When pulling quantifiers out to the front (in step Normalize4), when there is a choice, always place existential quantifiers to the left of universal quantifiers.

In any case, the sets of clauses for the better choice becomes

{ $(\forall x)(\forall y)(\neg P(x) \vee \neg P(y) \vee P(f(x,y)))$,
$\quad (\forall x)(\forall y) (\neg P(x) \vee Q(x, f_w(x)))$,
$\quad (\forall x)(\forall y)(\neg P(x) \vee \neg P(f_w(x)) \vee \neg R(c, f_w(x)))$ }

Step Normalize7: Eliminate quantifiers.

After Skolemization, the only quantifiers which will be left are universal quantifiers. To simplify things, it is customary to drop them, yielding the following set of clauses:

$\{ (\neg P(x) \lor \neg P(y) \lor P(f(x,y))),$
$\quad (\neg P(x) \lor Q(x, f_w(x))),$
$\quad (\neg P(x) \lor \neg P(f_w(x)) \lor \neg R(c, f_w(x))) \}$

Important: The universal quantifiers are dropped as a notational convenience only. We are not converting a sentence to a formula with free variables. The quantifiers are still there; they are just not written down.

Step Normalize8: Rename variables so that none occurs in more than one clause.

Input clauses:

{ $(\neg P(x) \lor \neg P(y) \lor P(f(x,y)))$,
  $(\neg P(x) \lor Q(x, f_w(x)))$,
  $(\neg P(x) \lor \neg P(f_w(x)) \lor \neg R(c, f_w(x)))$ }

Output clauses:

{ $(\neg P(x_1) \lor \neg P(y_1) \lor P(f(x_1,y_1)))$,
  $(\neg P(x_2) \lor Q(x_2, f_w(x_2)))$,
  $(\neg P(x_3) \lor \neg P(f_w(x_3)) \lor \neg R(c, f_w(x_3)))$ }

To see why this step is valid, consider the following simple example:

$$(\forall x)(A(x) \land B(x))$$

is equivalent to

$$(\forall x)A(x) \land (\forall x)B(x)$$

which is equivalent to

$$(\forall x)A(x) \land (\forall y)B(y).$$

Optimal Skolemization:

Although the algorithm just described is correct, it will not always yield optimal Skolemization:

Example: $(\forall x)(\exists y)P(x,y) \lor (\forall w)(\exists z)Q(w,z)$

Steps Normalize1 through Normalize3 are trivial. In step Normalize4, in which the quantifiers are brought out to the front, there are two possibilities:

(1) $(\forall x)(\exists y)(\forall w)(\exists z)(P(x,y) \lor Q(w,z))$

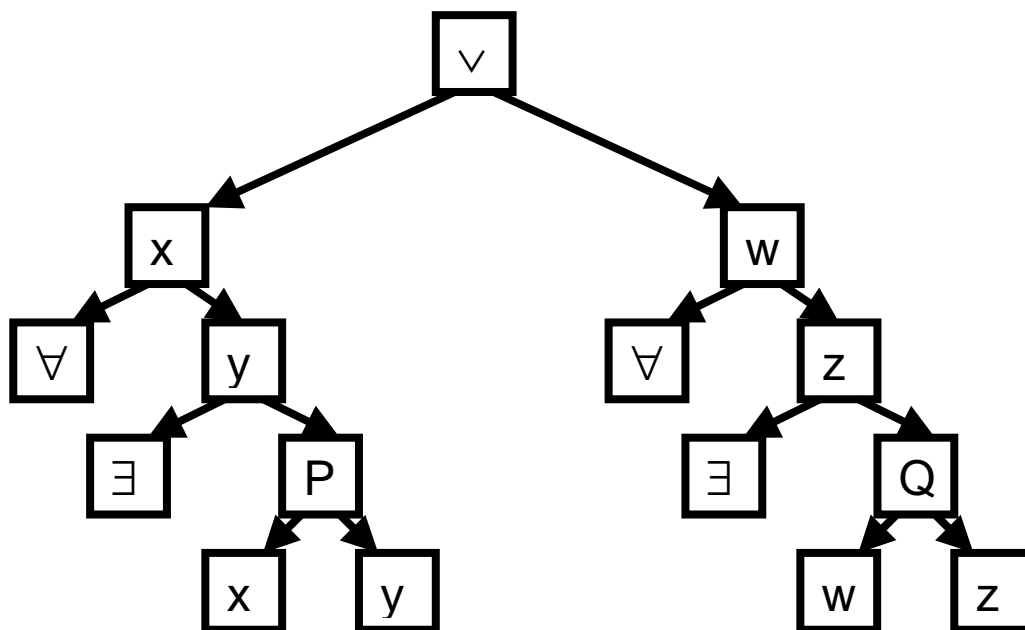(2) $(\forall w)(\exists z)(\forall x)(\exists y)(P(x,y) \lor Q(w,z))$

Step Normalize5 is also trivial, since there is only one clause. In Step Normaliz6 (Skolemize), the two possible results are as follows:

(1) $(\forall x)(\forall w)(P(x,f_y(x)) \lor Q(w,f_z(x,w)))$

(2) $(\forall x)(\forall w)(P(x,f_y(x,w)) \lor Q(w,f_z(w)))$

The problem is that neither of these Skolemizations is optimal. To see this, look at the parse tree.

The parse tree appears as follows.

$\vee$
 x   w
$\forall$ y $\forall$ z
 $\exists$ P $\exists$ Q
 x y   w z

From this, it is clear that the following Skolemization is correct, since y does not depend upon w, nor z upon x.

(3) $(\forall x)(\forall w)(P(x,f_y(x)) \vee Q(w,f_z(w)))$

To realize this within the algorithm, one should Skolemize *before* bringing the quantifiers out to the front. Alternatively, one can Skolemize according to the parse tree, with existentially quantified variables only depending upon those universally quantified variables which are positioned further up the tree.

The optimized normalization algorithm:

Step OptNorm1: (Same as Normalize1.) Eliminate all $\rightarrow$ and $\leftrightarrow$ ($\equiv$) connectives, by using the appropriate identities.

Step OptNorm2: (Same as Normalize 2.) Reduce the scope of all negations to atoms.

Step OptNorm3: (Same as Normalize3.) Standardize all variables. This means that variables are renamed so that none occurs in more than one quantifier.

Step OptNorm4: Skolemize optimally, using the parse tree to determine scoping constraints.

Step OptNorm5: (Same as Normalize4.) Pull all quantifiers out to the front. (Note that since all remaining quantifiers are universal, their order within the prefix does not matter.)

Step OptNorm6: (Same as Normalize5.) Convert to CNF equivalents.

Step OptNorm7: (Same as Normalize7.) Eliminate quantifiers.

Step OptNorm8: (Same as Normalize8.) Rename variables so that none occurs in more than one clause.

Summary:

- The processes identified by steps Normalize1 through Normalize8, or (better) by OptNorm1 through OptNorm8, yield sets of clauses which is equivalent to the original formula, but which are suited for resolution.