The procedure for conversion to CNF is similar to the propositional case, which we saw on page 253. The principal difference arises from the need to eliminate existential quantifiers. We illustrate the procedure by translating the sentence "Everyone who loves all animals is loved by someone," or

\[ V x \forall y \text{Animal}(y) \Rightarrow Loves(x, y) \].

The steps are as follows:

- **Eliminate implications:**
  
  \[ V x \lnot \text{Animal}(y) \lor Loves(x, y) \lor \exists y Loves(y, x) \].

- **Move inwards:** In addition to the usual rules for negated connectives, we need rules for negated quantifiers. Thus, we have
  
  - \( p \) becomes \( \exists x \lnot p \)
  - \( p \) becomes \( \forall x \)

  Our sentence goes through the following transformations:

  \[ V x \forall y \lnot \text{Animal}(y) \lor Loves(x, y) \lor \exists y Loves(y, x) \]

  \[ V x \forall y \lnot \text{Animal}(y) \land \lnot Loves(x, y) \lor \exists y Loves(y, x) \]

  \[ V x \exists z \text{Animal}(y) \land 
  \lnot Loves(x, z) \lor Loves(z, x) \]

  Notice how a universal quantifier \( \forall y \) in the premise of the implication has become an existential quantifier. The sentence now reads "Either there is some animal that a doesn't love, or (if this is not the case) someone loves x." Clearly, the meaning of the original sentence has been preserved.

- **Standardize variables:** For sentences like \( (P(x)) \lor Q(y) \) which use the same variable name twice, change the name of one of the variables. This avoids confusion later when we drop the quantifiers. Thus, we have

  \[ V x \exists z \text{Animal}(y) \land 
  \lnot Loves(x, z) \lor Loves(z, x) \]

  Here \( F \) and \( C \) are Skolem functions. The general rule is that the arguments of the Skolem function are all the universally quantified variables in whose scope the existential quantifier appears. As with Existential Instantiation, the Skolemized sentence is satisfiable exactly when the original sentence is satisfiable.
• Drop universal quantifiers: At this point, all remaining variables must be universally quantified. Moreover, the sentence is equivalent to one in which all the universal quantifiers have been moved to the left. We can therefore drop the universal quantifiers:

\[ \text{Animal}(F(x)) \lor \neg \text{Loves}(x, F(x)) \lor \text{Loves}(G(z)) \]

• Distribute \( \lor \) over \( \land \):

\[ \text{Animal}(F(x)) \lor \text{Loves}(G(z), x) \land [\neg \text{Loves}(x, F(x)) \lor \text{Loves}(G(z), x)] \]

This step may also require flattening out nested conjunctions and disjunctions.

The sentence is now in CNF and consists of two clauses. It is quite unreadable. (It may help to explain that the Skolem function \( F(x) \) refers to the animal potentially unloved by \( z \), whereas \( G(2) \) refers to someone who might love \( x \).) Fortunately, humans seldom need look at CNF sentences—the translation process is easily automated.

### 9.5.2 The resolution inference rule

The resolution rule for first-order clauses is simply a lifted version of the propositional resolution rule given on page 253. Two clauses, which are assumed to be standardized apart so that they share no variables, can be resolved if they contain complementary literals. Propositional literals are complementary if one is the negation of the other; first-order literals are complementary if one unifies with the negation of the other. Thus, we have

\[ \text{SUBST}(\theta, \ell_k, \ell_{k-1} \lor \cdots \lor \ell_1 \lor m_1 \lor \cdots \lor m_n) \]

where \( \text{UNIFY}(\ell_i, \ell_j) = \theta \). For example, we can resolve the two clauses

\[ \text{Animal}(F(x)) \lor \text{Loves}(G(x), y) \]

\[ \neg \text{Loves}(u, v) \lor \neg \text{Kills}(u) \]

by eliminating the complementary literals \( \text{Loves}(G(x), y) \) and \( \neg \text{Loves}(u, v) \), with unifier \( \theta = \{u/G(x), v/x\} \), to produce the resolvent clause

\[ \text{Animal}(F(x)) \lor \text{Loves}(G(x), x) \]

This rule is called the binary resolution rule because it resolves exactly two literals. The binary resolution rule by itself does not yield a complete inference procedure. The full resolution rule resolves subsets of literals in each clause that are unifiable. An alternative approach is to extend factoring—the removal of redundant literals—to the first-order case. Propositional factoring reduces two literals to one if they are identical; first-order factoring reduces two literals to one if they are unifiable. The unifier must be applied to the entire clause. The combination of binary resolution and factoring is complete.

### 9.5.3 Example proofs

Resolution proves that \( KB \) is unsatisfiable, that is, by deriving the empty clause. The algorithmic approach is identical to the propositional case, described in