Chapter 6

Preprocessing formulae: Herbrand's theorem

1. Introduction

In this chapter we first describe two types of preprocessing of formulae of first-order predicate calculus which together can remove all the quantifiers. These are the operations of putting the formula first into prenex form (Section 2) and then into Skolem form (Section 3).

This procedure is a necessary preliminary to the application of Herbrand's theorem (Section 4), which states that the question of the existence of a model for a set of formulae is equivalent to that of the existence of a syntactic model. A syntactic model is one where the domain is a set of terms derived from the terms appearing in the formulae. The theorem makes it possible to give a semi-decision procedure for questions of the type 'is B a consequence of $\{B_1,B_2,\ldots,B_n\}$' (Section 5). This algorithm, while giving no more than would a simple enumeration of the deductions, is more efficient than the latter and is the first important algorithm developed for automated theorem proving. A similar algorithm was given by Gilmore in 1960 and later improved by Davies and Putnam (Chang and Lee, 1937). Gilmore's algorithm, however, can handle only very simple problems and the method of resolution, described in Chapter 7, or one of its many variants has to be used for problems of any greater scale.

2. Prenex form

A formula of predicate calculus is said to be in prenex form when all the quantifiers are at the head, that is, it has the form:

$Q_1x_1, Q_2x_2, \ldots, Q_nx_nB$

where each Q_i is either $\forall$ or $\exists$ and B contains no quantifier.

Proposition 1: for any formula A there is an equivalent formula A′ in prenex form

Proof

First recall from the exercises in Chapter 5 that two formulae are equivalent if in any interpretation their associated functions are identical; that is, the formulae 'have the same meaning.' Further, that if a sub-formula A of

a formula B is replaced by an equivalent A′, then the resulting B′ is equivalent to B.

We establish the proposition by giving two methods by which any given formula can be reduced to prenex form in a finite number of steps.

Method 1

1. Remove all $\leftrightarrow$, v and $\wedge$ by the use of the following, working from left to right:
   (a) $(A \vee B) \equiv (\neg A \rightarrow B)$
   (b) $(A \wedge B) \equiv \neg (A \rightarrow \neg B)$
   (c) $(A \longleftrightarrow B) \equiv \neg ((A \rightarrow B) \rightarrow \neg (B \rightarrow A))$
2. Change the names of some of the bound variables so as to have no variable quantified twice; for this, use the following:
   (a) $\forall x\, A(x) \equiv \forall y\, A(y)$
   (b) $\exists x\, A(x) \equiv \exists y\, A(y)$
3. Bring all the quantifiers to the head of the formula, using the following from left to right:
   (a) $\neg \exists x\, A(x) \equiv \forall x \neg A(x)$ (here $x \notin vf(C)$)
   (b) $\neg \forall x\, A(x) \equiv \exists x \neg A(x)$
   (c) $\neg \neg A \equiv A$
   (d) $C \rightarrow \forall x\, A(x) \equiv \forall x\, (C \rightarrow A(x))$
   (e) $C \rightarrow \exists x\, A(x) \equiv \exists x\, (C \rightarrow A(x))$
   (f) $(\forall x\, A(x) \rightarrow C) \equiv \exists x\, (A(x) \rightarrow C)$
   (g) $(\exists x\, A(x) \rightarrow C) \equiv \forall x\, (A(x) \rightarrow C)$

Example 1

$$
\begin{aligned}
& (\forall x\, A(x) \vee \exists x\, (B(x) \wedge \exists t\, C(x,t))) \\
\equiv\ & (\neg \forall x\, A(x) \rightarrow \exists x\, (B(x) \wedge \exists t\, C(x,t))) && \text{(1a)} \\
\equiv\ & (\neg \forall x\, A(x) \rightarrow \exists x \neg (B(x) \rightarrow \neg \exists t\, C(x,t))) && \text{(1b)} \\
\equiv\ & (\neg \forall x\, A(x) \rightarrow \exists y \neg (B(y) \rightarrow \neg \exists t\, C(y,t))) && \text{(2b)} \\
\equiv\ & (\exists x \neg A(x) \rightarrow \exists y \neg (B(y) \rightarrow \neg \exists t\, C(y,t))) && \text{(3b)} \\
\equiv\ & (\exists x \neg A(x) \rightarrow \exists y \neg (B(y) \rightarrow \forall t \neg C(y,t))) && \text{(3a)} \\
\equiv\ & (\exists x \neg A(x) \rightarrow \exists y \neg \forall t\, (B(y) \rightarrow \neg C(y,t))) && \text{(3d)} \\
\equiv\ & (\exists x \neg A(x) \rightarrow \exists y \exists t \neg (B(y) \rightarrow \neg C(y,t))) && \text{(3b)} \\
\equiv\ & \forall x\, (\neg A(x) \rightarrow \exists y \exists t \neg (B(y) \rightarrow \neg C(y,t))) && \text{(3g)} \\
\equiv\ & \forall x \exists y\, (\neg A(x) \rightarrow \exists t \neg (B(y) \rightarrow \neg C(y,t))) && \text{(3e)} \\
\equiv\ & \forall x \exists y \exists t\, (\neg A(x) \rightarrow \neg (B(y) \rightarrow \neg C(y,t))) && \text{(3e)}
\end{aligned}
$$

In order to minimize the number of quantifiers in the final formula it can

be an advantage not to take Stage 2 as far as it could be taken, and in Stage 3 to use the additional equivalence:

$$(\forall\ x\ A(x) \rightarrow \exists\ x\ B(x)) \equiv \exists\ x\ (A(x) \rightarrow B(x)) \qquad (3h)$$

Example 2

Using (3h):

$$\begin{aligned}
& (\forall\ x\ A(x) \wedge \forall\ x\ B(x)) \\
\equiv\ & \neg\ (\forall\ x\ A(x) \rightarrow \neg\ \forall\ x\ B(x)) && (1b)\\
\equiv\ & \neg (\forall\ x\ A(x) \rightarrow \exists\ x\ \neg B(x)) && (3b)\\
\equiv\ & \neg \exists\ x\ (A(x) \rightarrow \neg\ B(x)) && (3h)\\
\equiv\ & \forall\ x\ \neg\ (A(x) \rightarrow \neg\ B(x)) && (3a)
\end{aligned}$$

and for the same problem, without using (3h):

$$\begin{aligned}
& (\forall\ x\ A(x) \wedge \forall\ x\ B(x)) \\
\equiv\ & \neg(\forall\ x\ A(x) \rightarrow \neg\ \forall x\ B(x)) && (1b)\\
\equiv\ & \neg\ (\forall x\ A(x) \rightarrow \neg\ \forall y\ B(y)) && (2a)\\
\equiv\ & \neg\ (\forall x\ A(x) \rightarrow \exists\ y\ \neg B(y)) && (3b)\\
\equiv\ & \neg\ \exists\ x\ (A(x) \rightarrow \exists\ y\ \neg B(y)) && (3f)\\
\equiv\ & \neg \exists x\ \exists\ y\ (A(x) \rightarrow \neg\ B(y)) && (3e)\\
\equiv\ & \forall\ x\ \neg\ \exists\ y\ (A(x) \rightarrow \neg\ B(y)) && (3a)\\
\equiv\ & \forall x\ \forall\ y\ \neg\ (A(x) \rightarrow \neg\ B(y)) && (3a)
\end{aligned}$$

Method 2

1. Remove all $\rightarrow$ and $\leftrightarrow$ by using the following:
   (a) $(A \rightarrow B) \equiv (\neg\ A \vee B)$
   (b) $(A \leftrightarrow B) \equiv ((A \wedge B) \vee (\neg\ A \wedge \neg\ B))$
2. Change the names of some of the bound variables so as to have no variable quantified twice; for this, use the following:
   (a) $\forall\ x\ A(x) \equiv \forall\ y\ A(y)$
   (b) $\exists\ x\ A(x) \equiv \exists\ y\ A(y)$
3. Bring all the quantifiers to the head of the formula, using the following:

   (a) $\neg\ \exists\ x\ A(x) \equiv \forall\ x\ \neg\ A(x)$
   (b) $\neg\ \forall\ x\ A(x) \equiv \exists\ x \neg\ A(x)$
   (c) $\neg\ \neg\ A \equiv A$
   (d) $(C \vee \forall\ x\ A(x)) \equiv \forall x\ (C \vee A(x))$
   (e) $(C \vee \exists\ x\ A(x)) \equiv \exists\ x\ (C \vee A(x))$
   (f) $(\forall\ x\ A(x) \vee C) \equiv \forall\ x\ (A(x) \vee C)$
   (g) $(\exists\ x\ A(x) \vee C) \equiv \exists\ x\ (A(x) \vee C)$
   (h) $C \wedge \forall\ x\ A(x) \equiv \forall\ x\ (C \wedge A(x))$

(i) $C \wedge \exists x\, A(x) \equiv \exists x\, (C \wedge A(x))$
(j) $(\forall x\, A(x) \wedge C) \equiv \forall x\, (A(x) \wedge C)$
(k) $(\exists x\, A(x) \wedge C) \equiv \exists x\, (A(x) \wedge C)$

where again $x \notin vf(C)$

Example

$(\forall x\, A(x) \rightarrow (\exists t\, B(t) \vee \exists t\, C(t)))$
$\equiv (\neg \forall x\, A(x) \vee (\exists t\, B(t) \vee \exists t\, C(t)))$ (1a)
$\equiv (\neg \forall x\, A(x) \vee (\exists t\, B(t) \vee \exists y\, C(y)))$ (2b)
$\equiv (\exists x \neg A(x) \vee (\exists t\, B(t) \vee \exists y\, C(y)))$ (3b)
$\equiv \exists x\, (\neg A(x) \vee (\exists t\, B(t) \vee \exists\, y\, C(y)))$ (3g)
$\equiv \exists x\, (\neg A(x) \vee \exists t\, (B(t) \vee \exists y\, C(y)))$ (3g)
$\equiv \exists x \exists t\, (\neg A(x) \vee (B(t) \vee \exists y\, C(y)))$ (3e)
$\equiv \exists x \exists t\, (\neg A(x) \vee \exists y\, (B(t) \vee C(y)))$ (3e)
$\equiv \exists x \exists t \exists y\, (\neg A(x) \vee (B(t) \vee C(y)))$ (3e)

As in Method 1, it can be advantageous to curtail the use of Stage 2 and in Stage 3 to use the following:

$(\forall x\, A(x) \wedge \forall x\, B(x) \equiv \forall x\, (A(x) \wedge B(x)))$ (3l)
$(\exists x\, A(x) \vee \exists x\, B(x) \equiv \exists x\, (A(x) \vee B(x)))$ (3m)

For the same example this gives:

$(\forall x\, A(x) \rightarrow (\exists t\, B(t) \vee \exists t\, C(t)))$
$\equiv (\neg \forall x\, A(x) \vee (\exists t\, B(t) \vee \exists t\, C(t)))$ (1a)
$\equiv (\neg \forall x\, A(x) \vee \exists t\, (B(t) \vee C(t)))$ (3m)
$\equiv (\exists x \neg A(x) \vee \exists t\, (B(t) \vee C(t)))$ (3b)
$\equiv \exists x\, (\neg A(x) \vee \exists x\, (B(x) \vee C(x)))$ (2b)
$\equiv \exists x\, (\neg A(x) \vee (B(x) \vee C(x)))$ (3m)

Δ

3. Skolem's theorem

If a formula A has been put into prenex form $Q_1x_1\, Q_2x_2 \ldots Q_mx_mB$, the Skolem form of A, written A^S, is the formula that results when all the quantifiers $\exists$ are removed by replacing all the variables x_i thus quantified by $f_i(x_{j1}, x_{j2}, \ldots, x_{jk})$, where $x_{j1}, x_{j2}, \ldots, x_{jk}$ are the variables quantified by the $\forall$ preceding the $\exists\, x_i$. The functional symbols f_i must of course be different from any others occurring elsewhere in the formula. If there is no $\forall$ before an $\exists\, x_i$, a constant symbol is used: a constant is simply an 0-ary function. The following examples illustrate the process:

1. $A = \exists x\, p(x, f(x))$ $\quad A^S = p(a, f(a))$
2. $A = \forall x \exists y\, p(x, f(y))$ $\quad A^S = \forall x\, p(x, f(f_1(x)))$

3. $A = \exists x_1 \forall x_2 \exists x_3 \exists x_4 (p(x_1, x_2) \rightarrow r(x_3, x_4))$
   $A^S = \forall x_2 (p(a, x_2) \rightarrow r(f_1(x_2) f_2(x_2)))$
4. $A = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 p(x_1, x_2, x_3, x_4, x_5)$
   $A^S = \forall x_2 \forall x_4 p(a, x_2, f_1(x_2), x_4, f_2(x_2, x_4))$

Proposition 2: Let $\{A_1, A_2, \ldots, A_n\}$ be a set of formulae and $\{A_1^S, A_2^S, \ldots, A_n^S\}$ the corresponding Skolem forms, then the first has a model with domain S if and only if the second has a model with this domain (Skolem's theorem)

Proof

We prove this for a single formula A, which we assume has the form $\forall x \exists y p(x, y)$.

Apart from some technical complications the same method can be used to prove the general theorem.

We have $A^S = \forall x p(x, f(x))$. There are two cases to consider:

1. Suppose $i = (S, \bar{p})$, where $\bar{p} = S \times S \rightarrow \{\mathbf{T}, \mathbf{F}\}$, is a model of A with domain S. By definition, $i[\exists y p(x, y)]$ is a mapping of S on to $\{\mathbf{T}, \mathbf{F}\}$ that takes only the value **T**; so for all $s \in S$ there is a $s' \in S$ such that $p(s, s') = \mathbf{T}$.

   If we now introduce the function f and put $\bar{f}(s) = s'$ for all $s \in S$, where s' is defined as above, we have $i' = (S, \bar{p}, \bar{f})$ as a model of A^S. Thus the model $(S, \bar{p})$ of A has been 'extended' to a model $(S, \bar{p}, \bar{f})$ of A^S, and the condition is shown to be sufficient.
2. If $(S, \bar{p}, \bar{f})$ is a model of A^S, it follows that $(S, \bar{p})$ is a model of A, so the condition is necessary.

Δ

Automated theorem proving

The general problem here can be expressed as a question: 'Is it true that the formula A is a theorem of the formal theory for which the non-logical axioms are the set $\mathscr{A}$?' The same question can be put in other ways:

'Is it true that the formula A is a consequence of $\mathscr{A}$?'

or, using Proposition 4 of Chapter 5

'Is it true that the set $\mathscr{A} \cup \{\neg A\}$ has no model?'

We are thus led to the problem of deciding whether or not a given set of formulae $\mathscr{A}_0$ has a model. To solve this, we put the formulae into standard form by applying the following processes in succession:

1. Put into prenex form, giving a set $\mathscr{A}_1$ (see Section 2).
2. Put formula in $\mathscr{A}_1$ into Skolem form, giving $\mathscr{A}_2$.

3. Remove the universal quantifiers ∀ from formula in $\mathscr{A}_2$ giving $\mathscr{A}_3$.
4. Express $\mathscr{A}_3$ as a set of clauses, giving $\mathscr{A}_4$ (see Proposition 8 in Chapter 4).

The necessary and sufficient condition for the existence of a model of $\mathscr{A}_0$ is that there should be a model for each set $\mathscr{A}_i$, in particular for $\mathscr{A}_4$.

Example

Consider the problem for which

$A = \exists x \exists y\ q(x, y)$
$\mathscr{A} = \{\forall x\ (p(x) \rightarrow \exists y(r(y) \wedge q(x,y))), \exists x\ p(x)\}$

The set $\mathscr{A}_0$ is:

$$\mathscr{A}_0 = \{\forall x\ (p(x) \rightarrow \exists y(r(y) \wedge q(x, y))), \exists x\ p(x), \neg \exists x \exists y\ q(x, y)\}$$

and we get in succession:

$$\begin{aligned}
\mathscr{A}_1 &= \{\forall x \exists y\ (\neg p(x) \vee (r(y) \wedge q(x, y))),\\
&\qquad \exists x\ p(x),\\
&\qquad \forall x \forall z \neg q(x, y)\}\\
\mathscr{A}_2 &= \{\forall x\ (\neg p(x) \vee (r(f(x)) \wedge q(x, f(x)))),\\
&\qquad p(a),\\
&\qquad \forall x \forall y \neg q(x, y)\}\\
\mathscr{A}_3 &= \{(\neg p(x) \vee (r(f(x)) \wedge q(x, f(x)))),\\
&\qquad p(a),\\
&\qquad \neg q(x, y)\}\\
\mathscr{A}_4 &= \{\neg p(x) \vee q(x, f(x)),\\
&\qquad \neg p(x) \vee r(f(x)),\\
&\qquad p(a),\\
&\qquad \neg q(x, y)\}
\end{aligned}$$

There are methods for putting into standard form that minimize the complexity of the resulting $\mathscr{A}_4$. These methods make the best possible use of the equivalences given in Section 2, combining Stages (1) and (2) so as to limit the number of arguments of the Skolem functions introduced there (Loveland, 1978) and whenever possible simplifying the clauses obtained in Stage (4).

We base our decision on $\mathscr{A}_4$: if a model exists for this set then A is not a theorem, but if there is no model then A is a theorem.

4. Herbrand's theorem

It would appear at first sight than an infinite number of tests would have to be made in order to decide whether or not a model existed for a given set of formulae: first, if there was a model with a domain of a single element, if not then with two elements and so on without limit; and each of these tests would

itself involve a large number of tests. The importance of Herbrand's theorem is that it enables us to reduce all this to a single test: to decide whether or not the given set of formulae has a syntactic model, that is, a model constructed by a standard process from the vocabulary of the set. Further, the question of the existence of such a model is reduced to the study of formulae of propositional calculus, for which, as we have seen, 'everything is decidable'.

Let {A1, A2, ..., An} be a finite set of first-order predicate calculus formulae $\mathscr{A}_4$ above for example, the set resulting from putting a given set into clause form. We need some definitions which we now give, illustrating these with the formulae of $\mathscr{A}_4$ in the above example:

A1 = ¬ p(x) v q(x, f(x))
A2 = ¬ p(x) v r(f(x))
A3 = p(a)
A4 = ¬ q(x, y)

The *Herbrand universe* associated with {A1, A2, ..., An} written $HU_{\{A1,...,An\}}$ is the set of all terms, not having variables, constructed from the vocabulary of the formulae A_i (written Ai here). To ensure that this is never an empty set, if this vocabulary does not contain a constant one is introduced.

In our example:

HU = {a, f(a), f(f(a)), ..., f(f(...f(a)...)),...}

Note that if there is any function symbol in the vocabulary the Herbrand universe is an infinite set.

The *Herbrand atoms* associated with {A1, A2, ..., An} written $HA_{\{A1,...,An\}}$ is the set of all atoms, not having variables, constructed from the vocabulary of the formulae of Ai; that is, of the formulae of the form $p(t_1, t_2, ..., t_r)$ where p is a predicate symbol appearing in one of the formulae Ai and t_j are elements of the associated Herbrand universe U.

In our example:

HA = {p(a), r(a), q(a, a), p(f(a)), ..., q(a, f(a)), p(f(f(a))) ...}

The order here is that we first list the atoms involving only the term a, then those involving also f(a), then f(f(a)) and so on.

The *Herbrand system* associated with {A1, A2, ..., An} written $HS_{\{A1,...,An\}}$ is the set of all formulae obtained from the Ai by replacing the variables by elements of the Herbrand universe. If the Ai are all clauses, the formulae of the Herbrand system are all disjunctions of Herbrand atoms and negations of these.

In our example:

HS = {¬p(a) v q(a,f(a)), ¬ p(a) v r(f(a)), p(a), ¬ q(a, a),
p(f(a), ¬ p(f(a)) v q(f(a), f(f(a))), ¬ p(f(a)) v r(f(f(a))),
¬ q(a, f(a)), ¬ q(f(a), a), ¬ q(f(a), f(a)), ...}

As with the atoms, the ordering here is first the formula derived from Ai by substituting a for the variables, then by substituting a or f(a), etc.

Proposition 3: let {A1,A2,...,An} be a finite set of clauses, then the three following statements are equivalent (Herbrand's theorem):

(a) {A1, A2, ..., An} has a model;
(b) {A1, A2, ..., An} has a model for which the domain is the Herbrand universe for the set;
(c) The Herbrand system $HS_{\{A1,A2,...,An\}}$, considered as a set of formulae of first-order propositional calculus for which the atoms are $HA_{\{A1,A2,...,An\}}$, has a model.

Note

Any model corresponding to case (c) is called a *Herbrand model* for the set. Giving such a model is from the definition giving a mapping i from $HA_{\{A1,A2,...,An\}}$ to {T, F} such that i[A] = T for all A ϵ $HS_{\{A1,A2,...,An\}}$, which is equivalent to giving a subset I of the atoms $HA_{\{A1,A2,...,An\}}$ such that i[A] = T for an A ϵ I. For this reason the Herbrand model is sometimes defined as a subset of the Herbrand atoms.

Proof

To prove the theorem we first prove:

(a) ⇒ (c)

If i is any model of {A1, A2, ..., An}, then for every formula B of $HS_{\{A1,A2,...,An\}}$ we have:

i[B] = T

So if we define i′ by:

i′[A] = i[A]

for every Herbrand atom A we have an interpretation i′ of propositional calculus domain where the set of propositions is the set of Herbrand atoms for {A1, A2, ..., An}, and i′ is a model of:

$HS_{\{A1,A2,...,An\}}$

Next, to prove:

(c) ⇒ (b)

let i′ be [any] model of $HS_{\{A1,A2,...,An\}}$. Given this we can define a model i of {A1, A2, ..., An} whose domain is the Herbrand universe HU as follows:

1. each function symbol is given its natural interpretation:

$$\bar{f}(t_1, t_2, \ldots, t_p) = f(t_1, t_2, \ldots, t_p)$$

2. for each predicate symbol we set:

$$\bar{p}(t_1, t_2, \ldots, t_p) = i'[p(t_1, t_2, \ldots, t_p)]$$

The interpretation thus defined:

$i = [HU, \bar{f}, \bar{g}, \ldots, \bar{p}, \bar{r}, \ldots]$,

is a model of {A1, A2, ..., An} because i', from which it was derived, is a model of

$HS_{\{A1,A2,\ldots,An\}}$

The proof that (b) ⇒ (a) is trivial, for if the set has a model whose domain is the Herbrand universe then it has a model.

△

Proposition 4: let $\{B_0, B_1, \ldots, B_m, \ldots\}$ be an enumeration of the formulae of $HS_{\{A1,A2,\ldots,An\}}$. Then the set {A1, A2, ..., An} is inconsistent (has no model) if and only if one of the formulae $B_0 \wedge B_1 \wedge \ldots \wedge B_m$ for m = 0, 1, 2, ..., m, ... is unsatisfiable

Proof

The proof follows immediately from Proposition 3 and the compactness theorem of propositional calculus (Proposition 10 in Chapter 4).

△

As an illustration, consider the previous example. The enumeration is:

$B_0 = \neg\, p(a) \vee q(a, f(a))$
$B_1 = \neg\, p(a) \vee r(f(a))$
$B_2 = p(a)$
$B_3 = \neg\, q(a, a)$
$B_4 = \neg\, p(f(a)) \vee q(f(a), f(f(a)))$
$B_5 = \neg\, p(f(a)) \vee r(f(f(a)))$
$B_6 = \neg\, q(a, f(a))$
$B_7 = \neg\, q(f(a), a)$
etc.

To test for the existence of a model for the original set A1, A2, ..., An we test successively B_0, $B_0 \wedge B_1$, $B_0 \wedge B_1 \wedge B_2$, ... to find whether or not it is satisfiable, and get affirmative answers up to and including $B_0 \wedge \ldots \wedge B_5$.

Here we use, for example, the interpretation:

$i[p(a)] = T$ (for B_2), $i[q(a, f(a))] = T$ (B_0), $i[r(f(a))] = T$ (B_1),
$i[q(a,a)] = F$ (B_3), $i[p(f(a))] = F$ (B_4)

However, we then find that no model can be constructed for $B_0 \wedge B_1 \wedge \ldots \wedge B_6$ because B_0, B_2 and B_6 cannot be satisfied simultaneously. Therefore there is no Herbrand model and so no model for {A1, A2, A3,

A4}, from which it follows, by the general result established in Section 3, that $A = \exists x \exists y\ q(x, y)$ is a theorem of the formal theory whose axioms are:

$\forall x\ p(x) \rightarrow \exists y\ r(y) \wedge q(x, y)))$
$\exists x\ p(x)$

5. An algorithm for automated theorem proving

Herbrand's theorem and also Proposition 4 suggest the following algorithm for automated testing of the truth of a statement of the form $\{B_1, B_2, \ldots, B_n, \neg B\}$. Transform $\{B_1, B_2, \ldots, B_n, \neg B\}$ into a set of clauses $\{A1, A2, \ldots, Ap\}$ using the procedure described in Section 3.

if $HS_{\{A1,A2,\ldots,Ap\}}$ is finite
then do
  if $HS_{\{A1,A2,\ldots,Ap\}}$ is satisfiable
    then do
    print 'B is not a consequence of $\{B_1, B_2, \ldots, B_n\}$'
    stop
  else do
    print 'B is a consequence of $\{B_1, B_2, \ldots, B_n\}$'
    stop
    end if
  else do
    r := 0
    while the first r formulae of $HS_{\{A1,A2,\ldots,Ap\}}$ are satisfiable
    do
    r := r + 1
    end while
    print 'B is a consequence of $\{B_1, B_2, \ldots, B_n\}$'
end if
stop

With the exception of the case in which the Herbrand system is finite, this algorithm will halt after a finite time if B is a consequence of $\{B_1, B_2, \ldots, B_n\}$ but not otherwise. From a purely theoretical point of view it does nothing more than would be done by an algorithm that examines in turn all the possible deductions that can be made from $\{B_1, B_2, \ldots, B_n\}$ and stops only when one of these turns out to be B; but in practice it is easier to implement than such a crude algorithm and is easily applied to simple problems. In a sense, it is the first practical algorithm of this book to be produced for automated theorem proving.

The simplest way to decide whether or not a set of formulae of propositional calculus is satisfiable is to evaluate the truth table for the conjunction of the formulae and examine the set of results: a single value **T** shows that there is an interpretation that satisfies all the formulae simul-

taneously, which means that the set is satisfiable. There are many other methods, for example Gilmore's procedure, developed in 1960 for the first theorem prover and also based on Herbrand's theorem, puts the set of formulae into disjunctive normal form which it then simplifies. However, both Gilmore's method and the truth table method are very inefficient, and it was because of this that Davis and Putnam developed a different procedure. Their procedure is the subject of one of the exercises at the end of this chapter.

A. Use of Semantic Trees

This is a method often resorted to when carrying out a proof procedure by hand. We illustrate it by the following example.

Suppose we wish to find if the formula:

$$B = (\exists x \neg q(x) \rightarrow \forall y\, p(y))$$

is a consequence of the pair of formulae:

$$B_1 = (\exists x\, p(x) \rightarrow \forall y\, p(y))$$
$$B_2 = \forall x\, (p(x) \vee q(x))$$

The negation of B is equivalent to:

$$(\exists x \neg q(x) \wedge \exists y \neg p(y))$$

and putting $\{B_1, B_2, \neg B\}$ into the form of a set of clauses gives:

$$A_1 = \neg p(x) \vee p(y)$$
$$A_2 = p(x) \vee q(x)$$
$$A_3 = \neg q(a)$$
$$A_4 = \neg p(b)$$

the constants a and b having been introduced when putting them into Skolem form.

The Herbrand universe HU, atoms HA and system HS are:

$$HU = \{a, b\}$$
$$HA = \{q(a), p(a), q(b), p(b)\}$$
$$HS = \{\neg p(a) \vee p(a), \neg p(a) \vee p(b), \neg p(b) \vee p(b)$$
$$\neg p(b) \vee p(a), p(a) \vee q(a), p(b) \vee q(b), \neg q(a), \neg p(b)\}$$
$$= \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8\}$$

We now draw a binary tree in which each level corresponds to a Herbrand atom and each path from the root to a node corresponds to an interpretation of all those Herbrand atoms through whose levels it passes. This is shown in Fig. 6, where the bold line corresponds to the interpretation:

$$i[q(a)] = \mathbf{F}$$

i[p(a)] = T
i[q(b)] = T
i[p(b)] = F

The development of the tree is stopped when the interpretation corresponding to a path contradicts one of the formulae of HS, and the node reached in this way is labelled with the formula that gives the contradiction.

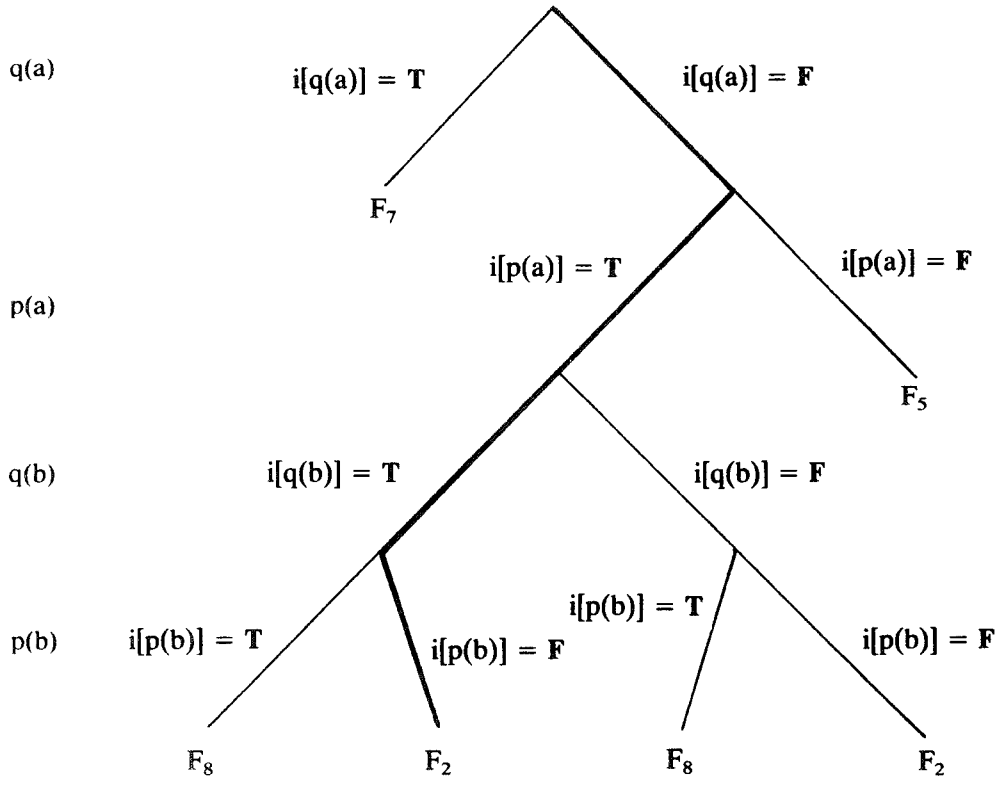

Fig. 6. Binary tree.

Three cases can arise:

1. All paths are ended in this way: this shows that there is no model of HS.
2. The tree is finite and one node is unlabelled: the interpretation corresponding to this path is a model of HS, which is therefore satisfiable.
3. The tree is infinite and by a well known result from graph theory known as König's lemma there is an infinite path to which there corresponds an interpretation of HS, which is therefore satisfiable.

König's lemma states that a tree with a finite number of arcs per node and a finite number of nodes on each branch (i.e. a branch of finite length) contains only a finite number of nodes.

Figure 6 shows a finite tree with all the nodes labelled, giving the result that B is a consequence of B_1, B_2. Our next example, shown in Fig. 7, leads to an infinite tree:

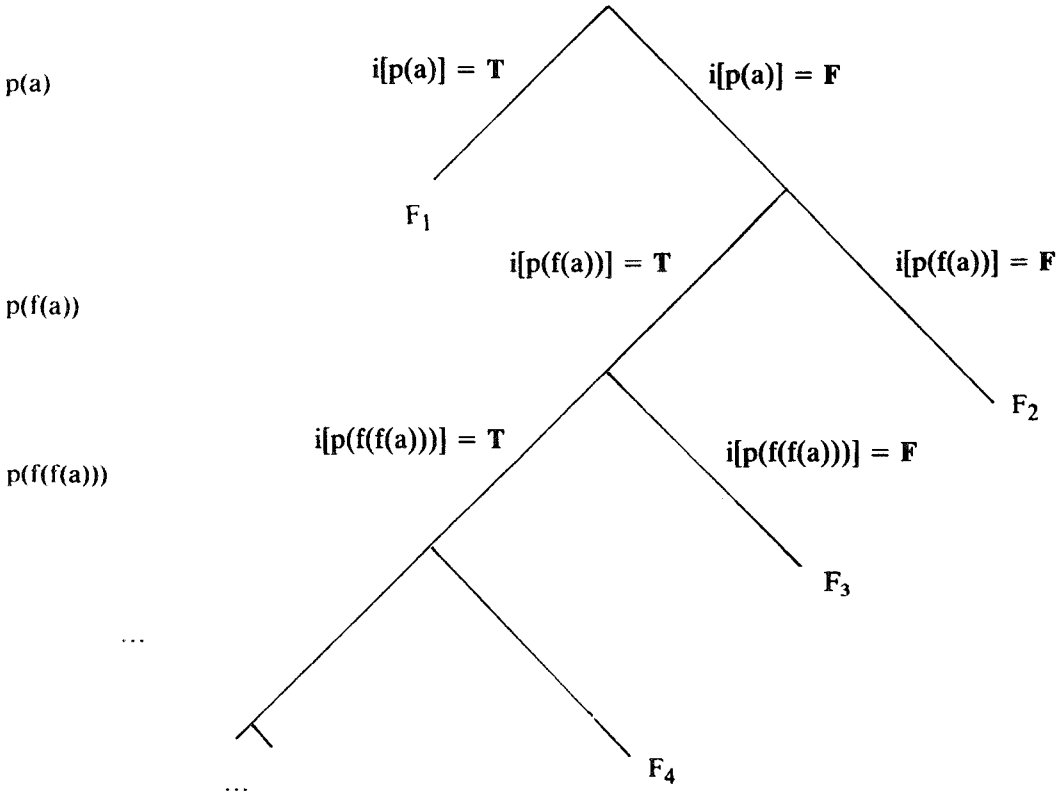

Fig. 7. Infinite tree.

$B = \forall x\ p(x)$, $B_1 = \forall x\ p(f(x))$

Is B a consequence of B_1?
Putting into clause form gives $\{\neg p(a), p(f(x))\}$ and hence:

$HU = \{a, f(a), f(f(a)), \ldots\}$
$HA = \{p(a), p(f(a)), p(f(f(a))), \ldots\}$
$HS = \{\neg p(a), p(f(a)), p(f(f(a))), \ldots\}$
$= \{F_1, F_2, F_3, \ldots\}$

The tree grows indefinitely, so B is not a consequence of B_1. It is to be noted, however, that the infinite path defines an interpretation i of HS that in turn gives an interpretation in HU such that $i[B] = F$, $i[B_1] = T$, which is thus a model of $\{B_1, \neg B\}$. Such a model is called a counter example for $B_1 \vdash\!\!\!/\ B$. This is a general result: when there are infinite paths, each one provides a counter example for the consequence that one is seeking to establish.

Exercises

Preprocessing Formulae

1. Put each of the following into prenex form, using first Method 1 and then Method 2:
   (a) $((\exists x\ p(x) \rightarrow (r(x) \vee \forall y\ p(y))) \wedge \forall x \exists y\ (r(y) \rightarrow p(x)))$

(b) $(((p_1(x_1) \rightarrow \exists\, x_2\, p_2(x_2)) \rightarrow \exists\, x_3\, p_3(x_3)) \rightarrow \exists\, x_4\, p_4(x_4))$

2. Put the following into clausal form:

$$\mathscr{A}_0 = \{\forall\, x\, (p(x) \wedge \exists\, y\, q(x,y)), (\forall\, x\, p(x) \rightarrow \exists\, y\, r(y))\}$$
$$\mathscr{A}_0 = \{\exists\, y\, (r(y) \wedge \forall\, x\, \forall\, z\, p(x,y,z)), \forall\, x_1\, \exists\, x_2\, r(x_1 x_2),$$
$$\forall\, x_1\, \exists\, x_2\, \forall\, x_3\, \exists\, x_4 (p(x_1, x_2, x_3) \rightarrow r(x_1, x_4))\}$$

Herbrand's Theorem

After making the appropriate transformations, use Herbrand's theorem to decide

1. if:

$$T = \forall\, x\, (r(x) \rightarrow p(x))$$

is a consequence of:

$$A_1 = \forall\, x\, (p(x) \rightarrow (q(x) \vee r(x)))$$
$$A_2 = \forall\, y\, (q(y) \rightarrow r(y))$$

2. if:

$$T = \forall\, x\, (p(x) \rightarrow r(x))$$

is a consequence of:

$$A_1 = \forall\, x\, (p(x) \rightarrow (q(x) \vee r(x)))$$
$$A_2 = \forall\, x\, (q(x) \rightarrow r(x))$$

3. if:

$$T = \forall\, z\, q(z)$$

is a consequence of:

$$A_1 = \forall\, x\, \exists\, y\, p(x,y)$$
$$A_2 = \forall x\, \forall\, y\, (p(x,y) \rightarrow q(x))$$

4. if:

$$T = \exists\, z\, q(z)$$

is a consequence of:

$$A_1 = \forall x\, (p(x) \rightarrow q(x))$$
$$A_2 = \exists\, x\, p(x)$$

Theorem-proving Algorithm (Section 5)

Let $\{A_1, A_2, \ldots, A_n\}$ be a set of clauses having no quantifiers and no function symbols and B a formula of the form:

$$\forall\, x_1\, \forall\, x_2 \ldots \forall\, x_n\, \exists\, y_1\, \exists\, y_2 \ldots \exists\, y_m\, F$$

where again F has neither quantifiers nor function symbols.

Show that when used to find if B is a consequence of $\{A_1, A_2, \ldots, A_n\}$, the algorithm will halt after a finite time.

If each of the formulae $A_1, A_2, \ldots, A_n$, and B are such that:

1. no function symbol appears,
2. the symbols $\rightarrow$, $\leftrightarrow$, $\neg$ do not appear,
3. the quantifiers are either all $\forall$ or all $\exists$.

show that the same result holds.

Semantic Trees

1. Use the method of semantic trees to find if the formula:

   $$\forall\, y\, \forall\, x\, p(x, y)$$

   is a consequence of:

   $$\forall\, x\, \forall\, y\, (p(x, y) \rightarrow p(y, x))$$
   $$\forall\, x\, \exists\, y\, p(x, y)$$

   The Herbrand atoms may be ordered so as to give a tree of reasonable size.
2. Use the method of semantic trees with the formulae of exercises 1–4 on Herbrand's Theorem (p. 103).

The Davis–Putnam Method

Given a clause:

$$At_1 \vee At_2 \vee \ldots \vee At_n \vee \neg At'_1 \vee \neg At'_2 \ldots \vee \neg At'_m$$

the atoms At_i are called the *positive literals* of the clause and $\neg At'_1$, the *negative literals*; and if l is any literal, the complementary literal l^c is defined by:

$$l^c = \neg At \text{ if } l = At$$
$$= At \text{ if } l = \neg At$$

The empty clause is denoted by $\square$, and by convention is taken to be unsatisfiable. This is not arbitrary: it can be justified by the argument that since a clause is easier to satisfy the greater the number of literals it has, it is reasonable to assume that a clause with no literals is impossible to satisfy. The empty set is denoted by $\emptyset$ and, in contrast to $\square$, is satisfiable because any interpretation whatever provides a model.

The Davis–Putnam procedure concerns clauses without variables; it consists in applying successively the following rules until no further application is possible; by convention, when more than one rule can be applied the first in the order is chosen.

Rule 1

Remove all tautologies, i.e. clauses containing both a literal and its complement.

Rule 2

If a clause has only a single literal l, remove all the clauses containing l and remove from the other clauses all occurrences of l^c.

Rule 3

If a literal l appears in some clauses but l^c does not appear in any clause, remove all the clauses containing l.

Rule 4

If all the literals of a clause C appear also in another clause C′, remove C′.

Rule 5

If both a literal l and its complement l^c appear somewhere in the set of clauses, replace the set by two others as follows:

one by removing all clauses containing l, and all occurrences of l^c
one by removing all clauses containing l^c, and all occurrences of l.

Example 1

We apply the procedure to the set of clauses:

$\{H \vee \neg H \vee G \vee A,\ G,\ \neg G \vee D \vee \neg E,\ \neg G \vee D \vee E,\ \neg G \vee A \vee \neg B,$
$\neg A \vee B,\ \neg A \vee B \vee C,\ A \vee \neg B \vee \neg C\}$

Rule 1:

$\{G,\ \neg G \vee D \vee \neg E,\ \neg G \vee D \vee E,\ \neg G \vee A \vee \neg B,\ \neg A \vee B,$
$\neg A \vee B \vee C,\ A \vee \neg B \vee \neg C\}$

Rule 2:

$\{D \vee \neg E,\ D \vee E,\ A \vee \neg B,\ \neg A \vee B,\ \neg A \vee B \vee C,\ A \vee \neg B \vee \neg C\}$

Rule 3:

$\{A \vee \neg B,\ \neg A \vee B,\ \neg A \vee B \vee C,\ A \vee \neg B \vee \neg C\}$

Rule 4:

$\{A \vee \neg B,\ \neg A \vee B,\ A \vee \neg B \vee \neg C\}$

Rule 3:

$\{A \vee \neg B,\ \neg A \vee B\}$

Rule 5:

$\{B\},\ \{\neg B\}$

Rule 2:

$\emptyset$, $\emptyset$

Note that when we remove all occurrences of a literal l from a clause that consists solely of l we get the empty clause $\square$: this is not the same as deleting the clause.

Example 2

Applying Rule 2 with $l = \neg B$ to the set:

$$\{A \vee B , \neg A \vee B , \neg B\}$$

gives:

$$\{A, \neg A\}$$

and a second application with $l = A$ gives:

$\{\square\}$

Now:

1. Apply the procedure to these sets:

$$S_1 = \{A \vee \neg B \vee \neg C , \neg A \vee \neg B \vee C , A \vee B \vee \neg C\}$$
$$S_2 = \{A \vee \neg B \vee \neg C \vee E, A \vee \neg B \vee C \vee \neg C \vee \neg E,$$
$$A \vee \neg E, B \vee \neg A \vee \neg E, A \vee B \vee \neg C\}$$

2. Prove that a set of clauses is satisfiable if and only if one of the sets derived by the Davis–Putnam procedure is satisfiable.
3. Apply the result from 2 to the sets of Examples 1 and 2 and to S_1, S_2 of 1; use another method to reach the same conclusions.