CHAPTER 5

LOGIC

In the present chapter we introduce some notions of logic (propositional calculus and predicate calculus). Logic is the cornerstone of mathematical reasoning; it is widely used within computer science. In addition to formalizing reasoning rules, logic also highlights the distinction between formal manipulations of strings of symbols and their meanings or interpretations.

The notions of logic introduced are basic. They aid proofs of program correctness (termination, loop invariants, etc.), and also the design of programs in general and, in particular, programs written in languages such as PROLOG that are directly derived from the predicate calculus.

In this chapter, we define propositional and predicate calculus, their syntax and their semantics, and a proof system that is sound and complete for each. We prove in detail the completeness theorem for propositional logic. We illustrate predicate calculus by showing how Herbrand models characterize the satisfiability of Horn clauses; this is the basis of the semantics of languages such as PROLOG.

We recommend in the strongest possible terms the following handbook, which is delivered together with a software program (for Macintosh or PC) of exercises and computer aided learning:

Jon Barwise, John Etchemendy, *The Language of First-order Logic: Tarski's world*, 2nd edition, CSLI lecture notes n° 23, Stanford (1991).

We also recommend:

René Lalement, *Computation as Logic*, Prentice Hall, London (1993).

Anil Nerode, Richard Shore, *Logic for Applications*, Springer-Verlag, Berlin (1993).

Raymond Smullyan, *What is the Name of this Book?*, Prentice Hall, London (1978).

5.1 Remarks on mathematical reasoning

A *proposition* is an assertion which is either true or false, but not both: for instance '$2+2=5$' is a false proposition, '$p \Longrightarrow p$' is a true proposition. On the other hand, a *formula* states a property of an object or a relation between objects, and may take the value true or false after values are assigned to the objects; for instance, '$2+2=x$' takes the value true if we assign value 4 to x, and takes the value false for any other assignment to x, and '$p \Longrightarrow q$' may take values true or false according to the values assigned to p and q. A formula which is always true is called a *theorem*.

Let p and q be two propositions concerning the same objects. We say that p implies q, denoted by '$p \Longrightarrow q$', if, whenever p is true, q is also true: $p \Longrightarrow q$ is a theorem whose hypothesis is p and whose conclusion is q; the converse of $p \Longrightarrow q$ is $q \Longrightarrow p$, which is usually not a theorem (see Exercise 5.2).

EXAMPLE 5.1 Verify the truth of the following theorems (whose converses are false):

- $a = a'$ and $b = b' \quad \Longrightarrow \quad a + b = a' + b'$, where a, a', b, b' are integers,
- $A \cap B = C \quad \Longrightarrow \quad C \subseteq A$ and $C \subseteq B$,
- $A \cup B = C \quad \Longrightarrow \quad A \subseteq C$ and $B \subseteq C$.

5.1.1 Some useful facts

(a) Implication is transitive: $[(p \Longrightarrow q)$ and $(q \Longrightarrow r)] \Longrightarrow (p \Longrightarrow r)$. This transitivity is the basis of deductive arguments.
(b) The negation of proposition p is denoted by $\bar{p}$ or $\neg p$. An implication $p \Longrightarrow q$ and the *contrapositive implication* $\bar{q} \Longrightarrow \bar{p}$ or $(\neg q \Longrightarrow \neg p)$ are two different ways of stating the same theorem. This fact is the basis of *proofs by contradiction*, where in order to prove $p \Longrightarrow q$, we assume p and $\bar{q}$ and we deduce a contradiction.
(c) The following propositions are equivalent:
(i) $p \Longrightarrow q$ and $q \Longrightarrow p$,
(ii) $p \Longrightarrow q$ and $\bar{p} \Longrightarrow \bar{q}$,
(iii) $p \Longleftrightarrow q$,
(iv) $\bar{p} \Longleftrightarrow \bar{q}$.
For instance, in order to prove: $ab = 0 \Longleftrightarrow$ ($a = 0$ or $b = 0$) on $\mathbb{R}$, it suffices to prove: $ab \neq 0 \Longleftrightarrow$ ($a \neq 0$ and $b \neq 0$).
(d) *Modus ponens* rule: $[p$ and $(p \Longrightarrow q)] \Longrightarrow q$.

EXERCISE 5.1 Verify that the *modus ponens* rule is equivalent to the *modus tollens* rule

$$[\neg q \text{ and } (p \Longrightarrow q)] \Longrightarrow \neg p \,,$$

i.e. that we can prove the *modus tollens* rule from the *modus ponens* rule, and vice-versa. ♢

5.1.2 Some confusions to be avoided

(a) While $p \Longrightarrow q$ and the contrapositive implication $\neg q \Longrightarrow \neg p$ are indeed two different ways of asserting the same fact, the converse implication $q \Longrightarrow p$ usually asserts a totally different fact.
(b) If $\neg q \Longrightarrow \neg p$ is true, and q is true, p is not necessarily true. We refer to Section 5.1.1 (b) for the explanation of this fact.
(c) If $p \Longrightarrow q$ is false, this usually does not imply that the converse $q \Longrightarrow p$ is true. Compare this with the fact that $A \not\supseteq B$ usually does not imply that $A \subseteq B$.

Exercise 5.2
1. Let:

$$p = \text{'it rains'} ,$$
$$q = \text{'there are clouds'} .$$

Write the implication $p \Longrightarrow q$ together with its contrapositive, its converse and the contrapositive of its converse. Which implications are true?
2. We consider the formulas

$$p = \big(\forall x \in A\ ,\ \exists y \in B\ ,\ P(x,y)\big) \qquad \text{and}$$
$$q = \big(\exists y \in B\ ,\ \forall x \in A\ ,\ P(x,y)\big)\ ,$$

where:
- A is the set of men,
- B is the set of women and
- $P(x,y)$ means 'y loves x'.

What can be said of $p \Longrightarrow q$? Of its converse? ♢

5.1.3 Propositional calculus *versus* predicate calculus

The initial motivation of logic is the modelling of mathematical reasoning. This needs a clear distinction between syntax (the language, the formulas) and semantics (the interpretation of the language, the truth values true or false of formulas). The case of *propositional calculus* is the simplest case because the variables, i.e. propositions, can take only one of the two values true or false. First, we study propositional calculus in Section 5.2. However, propositional calculus is not able to model all mathematical reasonings: for instance, we cannot express in propositional calculus the existence of an object having a given property. *Predicate calculus* can express the properties of objects or the relations between objects, and can formalize mathematical reasoning. We study predicate calculus in Section 5.3.

5.2 Propositional calculus

In the remainder of this chapter, logic and its language will be the object of our study. Logic is omnipresent in mathematics as a tool for proofs, and as such is an element of the meta-language. In the present chapter we will therefore try to distinguish between the symbols of the logical language as an object of study and the symbols of the logical language as a tool of the meta-language. For instance, we will denote by $\supset$ the implication, considered as a formal symbol of the logical language, and by $\Longrightarrow$ the implication that is just a notation for the word 'implies' with its intuitive meaning.

One of the fundamental goals of logic is to write correct proofs. In order to reach that goal, the concept of consequence is essential: when can we safely assert, with no possibility of error, that a formula is consequence of a set of premises? In Section 5.2.2 we will define a notion of semantical consequence, in Section 5.2.3 establish a notion of provability or syntactical consequence and, finally, in Section 5.2.4, show that both notions coincide.

5.2.1 Syntax: formulas

Let $P = \{p, p', q, q', \ldots\}$ be a set of propositional symbols and let $\supset$, $\neg$ and left and right parentheses be symbols.

Definition 5.2 *A propositional formula is a string of symbols from* $P \cup \{\supset, \neg, (,)\}$ *defined by:*

1. *every propositional symbol of* P *is a formula,*
2. *if* F *is a formula then* $\neg F$ *is a formula,*
3. *if* F *and* F' *are two formulas then* $(F \supset F')$ *is a formula and*
4. *every formula is obtained by repeating a finite number of times the applications of steps* 1–3.

Z Note that Definition 5.2 is an example of an inductive definition of a set. We have already seen such definitions in Chapter 3, and will see others later (see Chapter 7).

Z Formulas are strings of symbols. They have no meaning whatsoever for the time being. The assignment of a meaning, i.e. a value 'true' or 'false' to a formula, constitutes the semantics of the formula and will be studied in Section 5.2.2.

Let σ be a mapping, called a *substitution*, from the set of propositional symbols P to the set of formulas. The formula $\sigma(F)$ obtained by substitution from formula F is defined by:

- if $F = p \in P$ then $\sigma(F) = \sigma(p)$,

- if $F = \neg F'$ then $\sigma(F) = \neg\sigma(F')$ and
- if $F = (F_1 \supset F_2)$ then $\sigma(F) = (\sigma(F_1) \supset \sigma(F_2))$.

EXAMPLE 5.3 Let σ be the substitution defined by $\sigma(p) = q$ and $\sigma(q) = (p \supset q)$. Then $\sigma(p \supset q) = (q \supset (p \supset q))$.

Definition 5.4 *A sequent is a pair $(\mathcal{F}, F)$ where $\mathcal{F}$ is a finite set of formulas and F is a formula.*

The intuition is that a sequent formalizes the notion of logical consequence: if all premises of the sequent are true, i.e. if all formulas of $\mathcal{F}$ are true, then its conclusion, formula F, is true.

5.2.2 Semantics: interpretation of formulas

Let F be a formula and let I be a mapping from the set of propositional symbols to the Boolean algebra $\mathbb{B} = \{1, 0\}$, equipped with its operations $+$, '.' denoting product, and $^-$ (see Definition 4.2). The Boolean constants 'true', 'false', here identified by 1 and 0, are sometimes also denoted by tt, ff or T, F.

We define the *truth value* $I(F)$ of formula F in I by:

- if $F = p \in P$ then $I(F) = I(p) \in \mathbb{B}$;
- if $F = \neg F'$ then $I(F) = \overline{I(F')}$;
- if $F = (F_1 \supset F_2)$ then $I(F) = \overline{I(F_1)} + I(F_2)$.

If $I(F) = 1$, we say that F is true in I. I is called an *interpretation*, and we also say that $I(F)$ is the interpretation of formula F.

EXERCISE 5.3 Let us define $F \wedge F' \stackrel{\text{def}}{=} \neg(F \supset \neg F')$ and $F \vee F' \stackrel{\text{def}}{=} \neg F \supset F'$.

1. Write the tables giving the truth values of $\wedge, \vee, \supset$. Deduce that

$$I(F \wedge F') = I(F).I(F') \quad \text{and} \quad I(F \vee F') = I(F) + I(F')\,.$$

2. Show that

$$\begin{aligned} I\big(F_n \supset (F_{n-1} \supset (\cdots (F_1 \supset F)\cdots))\big) &= \overline{I(F_n)} + \overline{I(F_{n-1})} + \cdots + \overline{I(F_1)} + I(F) \\ &= I\big(\neg F_n \vee (\neg F_{n-1} \vee (\cdots \vee (\neg F_1 \vee F)\cdots))\big). \end{aligned}$$

Deduce that
$I\big(F_n \supset (F_{n-1} \supset (\cdots (F_1 \supset F)\cdots))\big) = I\big((F_n \wedge (F_{n-1} \wedge (\cdots \wedge (F_2 \wedge F_1))\cdots)) \supset F\big)$. $\diamondsuit$

Definition 5.5 *Let F be a formula. We say that*

- *F is valid, or that F is a tautology, if for all I, $I(F) = 1$,*
- *F is satisfiable, if there exists an I such that $I(F) = 1$ and*
- *F is unsatisfiable, if for all I, $I(F) = 0$.*

EXAMPLE 5.6 $p \wedge \neg p$ is unsatisfiable; $p \vee \neg p$ is a tautology; $p \wedge (p \supset q)$ is satisfiable but is not valid.

EXERCISE 5.4 Verify that F is unsatisfiable if and only if $\neg F$ is valid. ♢

Definition 5.7 *A sequent $(\mathcal{F}, G)$ is true in I if*

$$(\forall F \in \mathcal{F}, I(F) = 1) \Longrightarrow I(G) = 1.$$

A sequent $(\mathcal{F}, G)$ is valid if it is true in I for each I, i.e. $(\forall F \in \mathcal{F}, I(F) = 1) \Longrightarrow I(G) = 1$. We write $\mathcal{F} \models G$ to denote the fact that sequent $(\mathcal{F}, G)$ is valid.

This definition formalizes the notion of *semantical consequence*.

EXERCISE 5.5
1. Show that formula G is true (resp. valid) in I if sequent $(\emptyset, G)$ is true (resp. valid) in I.
2. Are the following sequents valid?
   - $(\emptyset, (p \supset q))$,
   - $(\{p, (p \supset q)\}, q)$. ♢

Proposition 5.8 *If σ is a substitution and if $\mathcal{F} \models G$ is a valid sequent, then $\sigma(\mathcal{F}) \models \sigma(G)$ is a valid sequent.*

Proof. If I is an interpretation, we define the interpretation I_σ by $I_\sigma(p) = I(\sigma(p))$. We deduce that $I_\sigma(F) = I(\sigma(F))$.

Let I be an interpretation such that for all F' in $\sigma(\mathcal{F})$, $I(F') = 1$. We thus have for all F in $\mathcal{F}$, $I(\sigma(F)) = I_\sigma(F) = 1$. Because $\mathcal{F} \models G$, $I_\sigma(G) = I(\sigma(G)) = 1$. □

Proposition 5.9 $\{F_n, \ldots, F_1\} \models F$ *if and only if*

$$\emptyset \models F_n \supset (F_{n-1} \supset (\cdots (F_1 \supset F) \cdots)).$$

Proof. It suffices to show that $\mathcal{F} \cup \{F\} \models G$ if and only if $\mathcal{F} \models (F \supset G)$, and the result will follow by induction on n. We have $I(F \supset G) = 1$ if and only if $I(F) = 1 \Longrightarrow I(G) = 1$.

Let I be an interpretation. We have

$$(I(F') = 1 \text{ for all } F' \in \mathcal{F}) \text{ and } I(F) = 1 \Longrightarrow I(G) = 1$$

if and only if $\big(I(F') = 1 \text{ for all } F' \text{ in } \mathcal{F}\big) \Longrightarrow\ I(F \supset G) = 1$. □

We will often write $\mathcal{F}, F \models G$ instead of writing $\mathcal{F} \cup \{F\} \models G$.

EXERCISE 5.6 We can associate a formula $\phi((\mathcal{F}, G))$ with a sequent $S = (\mathcal{F}, G)$ in the following fashion:
- If $S = (\emptyset, G)$, then $\phi(S) = G$.
- If $S = (\{F\} \cup \mathcal{F}, G)$ and $\phi((\mathcal{F}, G)) = F'$, then $\phi(S) = (F \supset F')$.

Show that sequent $S = (\mathcal{F}, G)$ is true in I if and only if $\phi(S)$ is true in I. ♢

Proposition 5.10
1. *$\mathcal{F} \models F$ if and only if $\mathcal{F} \models \neg\neg F$; $\mathcal{F}, F \models G$ if and only if $\mathcal{F}, \neg\neg F \models G$.*
2. *If $\mathcal{F} \models \neg(F \supset F')$ then $\mathcal{F} \models F$ and $\mathcal{F} \models \neg F'$.*
3. *If $\mathcal{F}, (F \supset F') \models G$ then $\mathcal{F}, F' \models G$, $\mathcal{F}, \neg G \models \neg F'$, and $\mathcal{F}, \neg G \models F$.*
4. *If $\mathcal{F}, \neg(F \supset F') \models G$ then $\mathcal{F}, \neg G, F \models F'$.*

Proof.
1. It is straightforward that $I(F) = I(\neg\neg F)$.
2. Let I be such that for all G in $\mathcal{F}$, $I(G) = 1$. Then $I(F \supset F') = 0$ and thus $I(F) = 1$ and $I(F') = 0$. Hence, $I(\neg F') = 1$.
3. Let I be such that for all H in $\mathcal{F}$, $I(H) = 1$. If $I(F') = 1$, then $I(F \supset F') = 1$ and thus $I(G) = 1$. If $I(G) = 0$, then $I(F \supset F') = 0$, whence $I(F') = 0$ and $I(F) = 1$.
4. Let I be such that for all H in $\mathcal{F}$, $I(H) = 1$. If $I(G) = 0$ and $I(F) = 1$ then it must also be the case that $I(F') = 1$. Otherwise, $I(F \supset F') = 0$, $I(\neg(F \supset F')) = 1$, and $I(G) = 1$, a contradiction. □

5.2.3 Logical proofs

Definition 5.11 *A sequent $(\mathcal{F}, F)$ is said to be provable, denoted by $\mathcal{F} \vdash F$, if it is built from a finite number of the following rules:*
- *use of a hypothesis rule: $F \in \mathcal{F} \Longrightarrow \mathcal{F} \vdash F$,*
- *augmentation of the hypotheses: if $G \notin \mathcal{F}$ and $\mathcal{F} \vdash F$ then $\mathcal{F} \cup \{G\} \vdash F$,*
- *detachment rule (or modus ponens): if $\mathcal{F} \vdash (F \supset F')$ and if $\mathcal{F} \vdash F$ then $\mathcal{F} \vdash F'$,*
- *synthesis rule (or hypothesis withdrawal): if $\mathcal{F}, F \vdash F'$ then $\mathcal{F} \vdash (F \supset F')$,*
- *double negation rule: $\mathcal{F} \vdash F$ if and only if $\mathcal{F} \vdash \neg\neg F$,*
- *proof by contradiction rule: if $\mathcal{F}, F \vdash F'$ and $\mathcal{F}, F \vdash \neg F'$, then $\mathcal{F} \vdash \neg F$.*

This definition formalizes the notion of *logical consequence* for propositional calculus.

Z Provable sequents are characterized only by manipulations of strings of symbols, in contrast with valid sequents, which are characterized by their interpretation.

A *proof* of a provable sequent $\mathcal{F} \vdash F$ is a finite sequence of provable sequents $\mathcal{F}_i \vdash F_i$, the last of which is $\mathcal{F} \vdash F$, such that any sequent in this sequence is obtained by applying the above rules to preceding sequents in the sequence. The first sequent of a proof must thus be obtained by the use of a hypothesis.

EXAMPLE 5.12 We omit brackets for explicit sequents.

1.
$p \vdash p$ (use of a hypothesis)
$\emptyset \vdash (p \supset p)$ (synthesis)

2.
$p, q \vdash p$ (hypothesis)
$p \vdash (q \supset p)$ (synthesis)
$\emptyset \vdash \big(p \supset (q \supset p)\big)$ (synthesis)

3.
(i) $(p \supset q), \neg q, p \vdash \neg q$ (hypothesis)
(ii) $(p \supset q), \neg q, p \vdash p$ (hypothesis)
(iii) $(p \supset q), \neg q, p \vdash p \supset q$ (hypothesis)
(iv) $(p \supset q), \neg q, p \vdash q$ (*modus ponens* on (ii) and (iii))
(v) $(p \supset q), \neg q \vdash \neg p$ (contradiction on (i) and (iv))
(vi) $(p \supset q) \vdash (\neg q \supset \neg p)$ (synthesis)

4.
$p, \neg p, \neg q \vdash p$ (hypothesis)
$p, \neg p, \neg q \vdash \neg p$ (hypothesis)
$p, \neg p \vdash \neg\neg q$ (contradiction)
$p, \neg p \vdash q$ (double negation)
$p \vdash (\neg p \supset q)$ (synthesis)

5.
$p \vdash p$ (hypothesis)
$\emptyset \vdash (p \supset p)$ (synthesis)
$p \vdash (p \supset p)$ (augmentation)
$\emptyset \vdash p \supset (p \supset p)$ (synthesis)

Proposition 5.13 *$\mathcal{F}, F \vdash G$ if and only if $\mathcal{F}, \neg\neg F \vdash G$.*

Proof.

1.
$\mathcal{F}, F \vdash G$
$\mathcal{F} \vdash (F \supset G)$ (synthesis)
$\mathcal{F}, \neg\neg F \vdash (F \supset G)$ (augmentation)
$\mathcal{F}, \neg\neg F \vdash F$ (hypothesis + double negation)
$\mathcal{F}, \neg\neg F \vdash G$ (*modus ponens*)

2.
$\mathcal{F}, \neg\neg F \vdash G$
$\mathcal{F} \vdash (\neg\neg F \supset G)$ (synthesis)
$\mathcal{F}, F \vdash (\neg\neg F \supset G)$ (augmentation)
$\mathcal{F}, F \vdash \neg\neg F$ (hypothesis + double negation)
$\mathcal{F}, F \vdash G$ (*modus ponens*) □

Proposition 5.14 *$\mathcal{F}, F \vdash G$ if and only if $\mathcal{F}, \neg G \vdash \neg F$.*

Proof. We prove that if $\mathcal{F}, F \vdash G$, then $\mathcal{F}, \neg G \vdash \neg F$.

1. $\mathcal{F}, F \vdash G$
2. $\mathcal{F}, F, \neg G \vdash G$ (augmentation)
3. $\mathcal{F}, F, \neg G \vdash \neg G$ (hypothesis)
4. $\mathcal{F}, \neg G \vdash \neg F$ (contradiction)

Conversely, a similar proof shows that if $\mathcal{F}, \neg G \vdash \neg F$ then $\mathcal{F}, F \vdash G$. □

Proposition 5.15 *If σ is a substitution and $\mathcal{F} \vdash F$, then $\sigma(\mathcal{F}) \vdash \sigma(F)$.*

Proof. By induction. Let $(\mathcal{F}_i \vdash F_i)_{i=1,\ldots,n}$ be a proof of $\mathcal{F} \vdash F$. Then $\big(\sigma(\mathcal{F}_i) \vdash \sigma(F_i)\big)_{i=1,\ldots,n}$ is a proof of $\sigma(\mathcal{F}) \vdash \sigma(F)$.
(B) $\sigma(\mathcal{F}_1) \vdash \sigma(F_1)$ is a provable sequent since, because $(\mathcal{F}_1, F_1)$ is the first sequent of a proof, $F_1 \in \mathcal{F}_1$ and thus $\sigma(F_1) \in \sigma(\mathcal{F}_1)$.
(I) We assume that $\big(\sigma(\mathcal{F}_i) \vdash \sigma(F_i)\big)_{i=1,\ldots,k}$ is a proof. We show by structural induction over the form of the rules in Definition 5.11 that $(\sigma(\mathcal{F}_i) \vdash \sigma(F_i))_{i=1,\ldots,k+1}$ is a proof.

- If $\mathcal{F}_{k+1} \vdash F_{k+1}$ is obtained by the use of a hypothesis rule, $\sigma(\mathcal{F}_{k+1}) \vdash \sigma(F_{k+1})$ is also obtained by the use of a hypothesis.
- If $\mathcal{F}_{k+1} \vdash F_{k+1}$ is obtained by augmentation of the hypotheses from $\mathcal{F}_i \vdash F_i$ then $\mathcal{F}_{k+1} = \mathcal{F}_i \cup \{G\}$ and $F_{k+1} = F_i$, and thus $\sigma(\mathcal{F}_{k+1}) = \sigma(\mathcal{F}_i) \cup \{\sigma(G)\}$ and $\sigma(F_{k+1}) = \sigma(F_i)$.
  - If $\sigma(G) \notin \sigma(\mathcal{F}_i)$, then $\sigma(\mathcal{F}_{k+1}) \vdash \sigma(F_{k+1})$ is obtained by augmentation of the hypotheses.
  - If $\sigma(G) \in \sigma(\mathcal{F}_i)$, then we will have two identical sequents: deleting the second one will again yield a proof.
- If there exist $i, j \leq k$ such that $\mathcal{F}_i = \mathcal{F}_j = \mathcal{F}_{k+1}$, $F_j = (F_i \supset F_{k+1})$, i.e. $\mathcal{F}_{k+1} \vdash F_{k+1}$ is obtained by *modus ponens*, then

  $$\sigma(\mathcal{F}_i) = \sigma(\mathcal{F}_j) = \sigma(\mathcal{F}_{k+1}) \text{ and } \sigma(F_j) = \big(\sigma(F_i) \supset \sigma(F_{k+1})\big),$$

  and $\sigma(\mathcal{F}_{k+1}) \vdash \sigma(F_{k+1})$ is also obtained by *modus ponens*.
- We proceed in the same way for the other rules. □

Proposition 5.16

$$\{F_1, \ldots, F_n\} \vdash F \text{ if and only if } \emptyset \vdash (F_1 \supset (F_2 \cdots (F_n \supset F) \cdots)).$$

Proof. It suffices to show that $\mathcal{F}, F \vdash G$ if and only if $\mathcal{F} \vdash F \supset G$.

The 'only if' direction is true because of the synthesis rule. For the opposite direction, if $\mathcal{F} \vdash F \supset G$ then $\mathcal{F}, F \vdash F \supset G$ by augmentation of the hypotheses, $\mathcal{F}, F \vdash F$ by the use of a hypothesis rule and $\mathcal{F}, F \vdash G$ by *modus ponens*. □

5.2.4 Syntax and semantics

We will show that valid sequents and provable sequents coincide.

Theorem 5.17 (Soundness) *Every provable sequent is valid.*

Proof. By induction on the lengths of proofs. It suffices to show that each application of one of the rules given in Definition 5.11 generates only valid sequents from valid sequents. To this end, it suffices to verify that each rule of the form 'if $S_1, \ldots, S_n$, then S' of Definition 5.11 is valid, i.e. that if $S_1, \ldots, S_n$ are valid sequents, then S is also a valid sequent.

- If $\mathcal{F} \vdash G$ is obtained by use of a hypothesis, then $G \in \mathcal{F}$, and if for any F' in $\mathcal{F}$, if $I(F') = 1$, then $I(G) = 1$, and thus $\mathcal{F} \models G$.
- If $\mathcal{F} \cup \{F\} \vdash G$ is obtained by augmentation of the hypotheses, then $F \notin \mathcal{F}$ and $\mathcal{F} \vdash G$. By the induction $\mathcal{F} \models G$, and thus

$$\begin{aligned} (\forall F' \in \mathcal{F} \cup \{F\}, I(F') = 1) &\implies (\forall F' \in \mathcal{F}, I(F') = 1) \quad \text{and} \\ (\forall F' \in \mathcal{F}, I(F') = 1) &\implies I(G) = 1. \end{aligned}$$

Hence $\mathcal{F}, F \models G$.

- If $\mathcal{F} \vdash G$ is obtained by *modus ponens*, then $\mathcal{F} \vdash (F \supset G)$, $\mathcal{F} \vdash F$, and by the induction hypothesis $\mathcal{F} \models (F \supset G)$, $\mathcal{F} \models F$; then, $\big($for all $F' \in \mathcal{F}$, $I(F') = 1\big)$ implies $I(F \supset G) = 1$ and $I(F) = 1$, whence $I(G) = 1$, and thus $\mathcal{F} \models G$.
- If $\mathcal{F} \vdash G$ is obtained by synthesis, then $G = (F \supset F')$, $\mathcal{F}, F \vdash F'$, and by the induction hypothesis $\mathcal{F}, F \models F'$. If for all $H \in \mathcal{F}$, $I(H) = 1$, then
  - if $I(F) = 1$, $I(F') = 1$ and $I(F \supset F') = 1$ and
  - if $I(F) = 0$, $I(F \supset F') = 1$.

Hence $\mathcal{F} \models (F \supset F')$.

- If $\mathcal{F} \vdash G$ is obtained by double negation, then $\mathcal{F} \models G$ if and only if $\mathcal{F} \models \neg\neg G$ (because $I(G) = I(\neg\neg G)$).
- If $\mathcal{F} \vdash G$ is obtained by a proof by contradiction, then $G = \neg F$, and by the induction hypothesis $\mathcal{F}, F \models F'$ and $\mathcal{F}, F \models \neg F'$. If for all $H \in \mathcal{F}$, $I(H) = 1$, then $I(F) = 1$ cannot occur (otherwise we would have $I(F') = I(\neg F') = 1$, a contradiction). Hence, we must have $I(F) = 0$ and thus $\mathcal{F} \models G$. □

Theorem 5.18 (Completeness) *Every valid sequent is provable.*

Proof. Let us define the *weight* of a sequent as the sum of the number of $\neg$ symbols and twice the number of $\supset$ symbols occurring in this sequent. We argue by induction on the weight of a sequent.

- If the weight of a sequent is zero, then this sequent can be written
$$\{p_1, \cdots, p_n\} \models p.$$

If $p \notin \{p_1, \cdots, p_n\}$, then the interpretation I defined by $I(p_i) = 1$, $I(p) = 0$ shows that this sequent is not valid. We thus have $p \in \{p_1, ..., p_n\}$, and the sequent is provable by use of a hypothesis rule.

• Let thus $\mathcal{F} \models F$ have weight $n + 1$.

(a) If $F = \neg\neg F'$ then $\mathcal{F} \models F'$ and, because that sequent has weight $n - 1$, then by the induction hypothesis $\mathcal{F} \vdash F'$ and thus $\mathcal{F} \vdash \neg\neg F'$ by double negation.

(b) If F is not of the form $\neg\neg F'$ and contains at least one symbol $\supset$ then $F = (F' \supset F'')$ or $F = \neg(F' \supset F'')$.

(b.1) If $F = (F' \supset F'')$ then $\mathcal{F} \models (F' \supset F'')$ implies $\mathcal{F}, F' \models F''$. The last sequent has weight $n - 1$, and so we obtain $\mathcal{F} \vdash (F' \supset F'')$ (synthesis rule).

(b.2) If $F = \neg(F' \supset F'')$, then as we have seen, $\mathcal{F} \models \neg(F' \supset F'')$ implies $\mathcal{F} \models F'$ and $\mathcal{F} \models \neg F''$. These sequents have weight $\leq n$, and hence $\mathcal{F} \vdash F'$ and $\mathcal{F} \vdash \neg F''$. We thus have:

$\mathcal{F}, (F' \supset F'') \vdash F'$	(augmentation)
$\mathcal{F}, (F' \supset F'') \vdash \neg F''$	(augmentation)
$\mathcal{F}, (F' \supset F'') \vdash (F' \supset F'')$	(hypothesis)
$\mathcal{F}, (F' \supset F'') \vdash F''$	(*modus ponens*)
$\mathcal{F} \vdash \neg(F' \supset F'')$	(contradiction)

(c) If F is not of the form $\neg\neg F'$ and contains no $\supset$ symbol then $F = r$ or $F = \neg r$. Since $\mathcal{F}, \neg\neg F \models G$ if and only if $\mathcal{F}, F \models G$, we may assume that the elements of $\mathcal{F}$ have one of the four following forms: $p, \neg p, (F_1 \supset F_2), \neg(F_1 \supset F_2)$.

(c.1) If $\mathcal{F}$ contains a formula $\neg(F_1 \supset F_2)$ then

$$\mathcal{F}', \neg(F_1 \supset F_2) \models F$$

implies $\mathcal{F}', \neg F, F_1 \models F_2$. We can apply the induction hypothesis: $\mathcal{F}', \neg F, F_1 \vdash F_2$. Hence:

$\mathcal{F}', \neg F \vdash (F_1 \supset F_2)$	(synthesis)
$\mathcal{F}', \neg F, \neg(F_1 \supset F_2) \vdash (F_1 \supset F_2)$	(augmentation)
$\mathcal{F}', \neg F, \neg(F_1 \supset F_2) \vdash \neg(F_1 \supset F_2)$	(hypothesis)
$\mathcal{F}', \neg(F_1 \supset F_2) \vdash F$	(contradiction + double negation)

(c.2) If $\mathcal{F}$ contains a formula $(F_1 \supset F_2)$ then $\mathcal{F}', (F_1 \supset F_2) \models F$ implies $\mathcal{F}', \neg F \models F_1$ (Proposition 5.10, 3) and $\mathcal{F}', F_2 \models F$ which have weight $\leq n$.

We can apply the induction hypothesis: $\mathcal{F}', \neg F \vdash F_1$, $\mathcal{F}', F_2 \vdash F$. Hence:

1.	$\mathcal{F}', \neg F \vdash F_1$	
2.	$\mathcal{F}', F_2 \vdash F$	
3.	$\mathcal{F}', (F_1 \supset F_2), \neg F \vdash \neg F$	(hypothesis)
4.	$\mathcal{F}' \vdash (F_2 \supset F)$	(synthesis on 2)
5.	$\mathcal{F}', (F_1 \supset F_2), \neg F \vdash (F_2 \supset F)$	(augmentation on 4)
6.	$\mathcal{F}', (F_1 \supset F_2), \neg F \vdash F_1$	(augmentation on 1)
7.	$\mathcal{F}', (F_1 \supset F_2), \neg F \vdash (F_1 \supset F_2)$	(hypothesis)
8.	$\mathcal{F}', (F_1 \supset F_2), \neg F \vdash F_2$	(*modus ponens* 6,7)
9.	$\mathcal{F}', (F_1 \supset F_2), \neg F \vdash F$	(*modus ponens* 5,8)
10.	$\mathcal{F}', (F_1 \supset F_2) \vdash F$	(contradiction on 3, 9, + double negation)

(c.3) The problem is thus reduced to the case in which $\mathcal{F}$ contains only formulas of the form p or $\neg p$. Let us write $\mathcal{F}$ in the form $\mathcal{F}^+ \cup \mathcal{F}^-$ with $\mathcal{F}^+$ equal to the set of formulas of $\mathcal{F}$ of the form 'p', and $\mathcal{F}^-$ the set of formulas of the form '$\neg p$'. Let P^+ be the set of all propositional symbols occurring in $\mathcal{F}^+$ and let P^- be the set of all propositional symbols occurring in $\mathcal{F}^-$.

(c.3.1) If $P^+ \cap P^- \neq \emptyset$ then $\mathcal{F} = \mathcal{F}', p, \neg p$; we deduce

$$\left.\begin{array}{l} \mathcal{F}, \neg F \vdash p \\ \mathcal{F}, \neg F \vdash \neg p \end{array}\right\} \quad \text{and thus} \quad \mathcal{F} \vdash F\,.$$

(c.3.2) We assume that $P^+ \cap P^- = \emptyset$, and we let r be the propositional symbol occurring in F.

(c.3.2.1) If $r \notin P^+ \cup P^-$, we could find an interpretation I true on $\mathcal{F}$ and false on F which is impossible.

(c.3.2.2) If $r \in P^+$, then any interpretation true on $\mathcal{F}$ verifies $\mathrm{I}(r) = 1$. As we then have $I(F) = 1$, then $F = r$ by necessity and $\mathcal{F} \vdash r$ by use of a hypothesis.

(c.3.2.3) If $r \in P^-$, then any interpretation true on $\mathcal{F}$ verifies $I(r) = 0$. Thus $F = \neg r$ and $\mathcal{F} \vdash \neg r$ by use of the hypothesis. □

Grouping together Theorem 5.17 and Theorem 5.18, we deduce the following corollary.

Corollary 5.19 *$\mathcal{F} \models F$ if and only if $\mathcal{F} \vdash F$.*

5.2.5 Additional logical connectors

In propositional logic we can also use the connectors $\wedge$ (and) and $\vee$ (or).

Formulas are then defined by the additional rule: if F and F' are formulas then $(F \wedge F')$ and $(F \vee F')$ are formulas.

The interpretation of these formulas is defined by adding, see Exercise 5.3,
$I(F \wedge F') = I(F)\ .\ I(F')$,
$I(F \vee F') = I(F) + I(F')$,
so that
$I(F \wedge F') = I(\neg(F \supset \neg F'))$,
$I(F \vee F') = I(\neg F \supset F')$.
Similarly, the definitions of provable sequents are extended by adding the rules:

- if $\mathcal{F} \vdash F$ and $\mathcal{F} \vdash F'$ then $\mathcal{F} \vdash (F \wedge F')$,
- if $\mathcal{F} \vdash (F \wedge F')$ then $\mathcal{F} \vdash F$,
- if $\mathcal{F} \vdash (F \wedge F')$ then $\mathcal{F} \vdash F'$,
- if $\mathcal{F}, G \vdash F$ and $\mathcal{F}, \neg G \vdash F'$ then $\mathcal{F} \vdash (F \vee F')$,
- if $\mathcal{F}, F \vdash G$ and $\mathcal{F}, F' \vdash G$ then $\mathcal{F}, (F \vee F') \vdash G$.

We deduce the following proposition.

Proposition 5.20
1. $(F \wedge F') \vdash \neg(F \supset \neg F')$,
2. $\neg(F \supset \neg F') \vdash (F \wedge F')$,
3. $(F \vee F') \vdash (\neg F \supset F')$,
4. $(\neg F \supset F') \vdash (F \vee F')$.

Proof. We prove in detail 2 and 3; 1 and 4 are simpler, and we just sketch their proofs.

1.
$(F \wedge F')\,,\,(F \supset \neg F') \ \vdash\ F'$ (third rule for $\wedge$)
$(F \wedge F')\,,\,(F \supset \neg F') \ \vdash\ F$ (second rule for $\wedge$)
$(F \wedge F')\,,\,(F \supset \neg F') \ \vdash\ (F \supset \neg F')$ (hypothesis)
$(F \wedge F')\,,\,(F \supset \neg F') \ \vdash\ \neg F'$ (*modus ponens*)
$(F \wedge F') \ \vdash\ \neg(F \supset \neg F')$ (contradiction)

2.
$\neg(F \supset \neg F')\,,\,\neg F' \ \vdash\ (F \supset \neg F')$ (augmentation of $p \vdash (q \supset p)$, see Example 5.12, 2)
$\neg(F \supset \neg F')\,,\,\neg F' \ \vdash\ \neg(F \supset \neg F')$ (hypothesis)
$\neg(F \supset \neg F') \ \vdash\ F'$ (contradiction)

$\neg(F \supset \neg F')\,,\ \neg F \ \vdash\ \neg(F \supset \neg F')$ (hypothesis)
$\neg(F \supset \neg F')\,,\ \neg F\,,\ F\,,\ F' \ \vdash\ \neg F$ (hypothesis)
$\neg(F \supset \neg F')\,,\ \neg F\,,\ F\,,\ F' \ \vdash\ F$ (hypothesis)
$\neg(F \supset \neg F')\,,\ \neg F\,,\ F \ \vdash\ \neg F'$ (contradiction)
$\neg(F \supset \neg F')\,,\ \neg F \ \vdash\ (F \supset \neg F')$ (synthesis)
$\neg(F \supset \neg F') \ \vdash\ F$ (contradiction + double negation)
$\neg(F \supset \neg F') \ \vdash\ (F \wedge F')$ (first rule for $\wedge$)

3. $F' \ \vdash\ (\neg F \supset F')$ (Example 5.12, 2)
$F\,,\ \neg F\,,\ \neg F' \ \vdash\ F$ (hypothesis)
$F\,,\ \neg F\,,\ \neg F' \ \vdash\ \neg F$ (hypothesis)
$F\,,\ \neg F \ \vdash\ \neg\neg F'$ (contradiction)
$F\,,\ \neg F \ \vdash\ F'$ (double negation)
$F \ \vdash\ (\neg F \supset F')$ (synthesis)
$(F \vee F') \ \vdash\ (\neg F \supset F')$ (second rule for $\vee$)

4. $(\neg F \supset F')\,,\ \neg F \ \vdash\ F'$ (twice hypothesis + *modus ponens*)
$(\neg F \supset F')\,,\ F \ \vdash\ F$ (hypothesis)
$(\neg F \supset F') \ \vdash\ F \vee F'$ (first rule for $\vee$) $\square$

We now define the transformation η which suppresses the symbols $\wedge$ and $\vee$ from a formula:

$$\begin{aligned}
&\eta(p) = p,\\
&\eta(\neg F) = \neg\eta(F),\\
&\eta(F \supset F') = (\eta(F) \supset \eta(F')),\\
&\eta(F \wedge F') = \neg(\eta(F) \supset \neg\eta(F')),\\
&\eta(F \vee F') = (\neg\eta(F) \supset \eta(F')).
\end{aligned}$$

It is easy to see that $\mathcal{F} \models F$ if and only if $\eta(\mathcal{F}) \models \eta(F)$ and, using the preceding property, we show that if $\eta(\mathcal{F}) \vdash \eta(F)$ is a provable sequent which can be proved without the rules concerning $\wedge$ and $\vee$, then $\mathcal{F} \vdash F$ is a provable sequent that can be proved using these rules.

Similarly, we can introduce the equivalence symbol $\equiv$ whose interpretation is given by:

$$I(F \equiv F') = 1 \ \text{ if and only if } \ I(F) = I(F')\,.$$

So that $I(F \equiv F') = I\big((F \supset F') \wedge (F' \supset F)\big)$.

EXERCISE 5.7 We consider the set of formulas as an algebra equipped with the binary operations $\wedge, \vee, \supset$ and with the unary operation $\neg$. Is the equivalence relation $\Longleftrightarrow$ defined by $F \Longleftrightarrow F'$ if and only if $I(F \equiv F') = 1$ a congruence? $\diamondsuit$

The proof rules associated with the equivalence symbol $\equiv$ are:

- if $\mathcal{F} \vdash (F \equiv F')$ then $\mathcal{F} \vdash (F \supset F')$ and $\mathcal{F} \vdash (F' \supset F)$,
- if $\mathcal{F} \vdash (F \supset F')$ and $\mathcal{F} \vdash (F' \supset F)$ then $\mathcal{F} \vdash (F \equiv F')$,

or in other words:

$$\mathcal{F} \vdash (F \equiv F') \qquad \text{if and only if} \qquad \mathcal{F} \vdash \big((F \supset F') \wedge (F' \supset F)\big)\,.$$

The 'meta-logical' use of symbol $\Longleftrightarrow$ can then be formalized by: $F \Longleftrightarrow F'$ if and only if $(F \equiv F')$ is a valid formula, or in other words if and only if $\vdash (F \equiv F')$.

The operations $\vee$ and $\wedge$ enjoy associativity, commutativity, and distributivity properties similar to those of Boolean algebras:

1. Distributivity of $\wedge$ over $\vee$: $F \wedge (G \vee H) \iff (F \wedge G) \vee (F \wedge H)$.
2. Distributivity of $\vee$ over $\wedge$: $F \vee (G \wedge H) \iff (F \vee G) \wedge (F \vee H)$.
3. Associativity of $\wedge$: $F \wedge (G \wedge H) \iff (F \wedge G) \wedge H$.
4. Associativity of $\vee$: $F \vee (G \vee H) \iff (F \vee G) \vee H$.
5. Commutativity of $\wedge$: $F \wedge G \iff G \wedge F$.
6. Commutativity of $\vee$: $F \vee G \iff G \vee F$.

The associativity properties allow us to omit parentheses.

The following equivalences are quite useful:

$$\begin{aligned}
F \supset G &\iff \neg F \vee G\,,\\
F \supset G &\iff \neg G \supset \neg F\,,\\
\neg(F \supset G) &\iff F \wedge \neg G\,,\\
F \equiv G &\iff (F \supset G) \wedge (G \supset F)\,,\\
F \equiv G &\iff (F \wedge G) \vee (\neg G \wedge \neg F)\,.
\end{aligned}$$

EXERCISE 5.8 A logician tells his son: 'if you don't eat porridge, you won't watch television'; the son eats porridge, and is sent straight to bed. What was the error which caused him to expect watching television after dinner? $\diamondsuit$

EXERCISE 5.9

1. A logician, who is assumed to always tell the truth, is interviewed about his feelings, and says both the following statements:
   (a) I love Mary or I love Anne.
   (b) If I love Mary, then I love Anne.

What can you conclude: does he love Mary, Anne or both?

2. Assume the same logician had answered the question: 'Is it true that if you love Mary, then you love Anne?' by both of the following statements:
   (a) If it's true, then I love Mary.
   (b) If I love Mary, then it's true.

What would you conclude? ♢

5.2.6 Deductive systems

In order to define provable sequents we defined manipulation rules for strings of symbols. There are other systems of rules that can obtain the same result.

First, we say that formula F is *provable* if and only if $\emptyset \vdash F$ is a provable sequent. By the preceding theorems, a formula is provable if and only if it is valid.

We will now show an example of another way of proving formulas containing only propositional symbols and the symbols $\supset$ and $\neg$. The formulas that we can prove with rules will be called 'logical theorems' (to distinguish them from provable formulas).

Let p, q, r be three arbitrary propositional symbols.

(i) The following three formulas, called *axioms*, are logical theorems:
- $(p \supset (q \supset p))$,
- $((p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r)))$,
- $((\neg p \supset \neg q) \supset ((\neg p \supset q) \supset p))$.

(ii) If σ is a substitution and if F is a logical theorem, then $\sigma(F)$ is a logical theorem.

(iii) If F and $(F \supset F')$ are logical theorems, then F' is a logical theorem.

(ii) and (iii) are *rules of inference*; (iii) is called the *modus ponens* rule.

A *deduction* of formula F from the set of formulas $\mathcal{F}$ is a finite sequence of formulas $F_1, F_2, \ldots, F_n$ such that

- F_n is identical to F
- for each $i \leq n$,
  - either F_i is one of the axioms (i),
  - or $F_i \in \mathcal{F}$,
  - or F_i can be deduced from the preceding F_js by an application of one of the rules (ii) or (iii).

Formula F is a logical theorem if and only if there is a deduction of F from the empty set of formulas $\mathcal{F} = \emptyset$.

It is easy to see that every logical theorem is a valid formula (and thus a provable formula); this shows the soundness of the system for deducing logical

theorems. Conversely, we can show the completeness of the system, i.e. that every valid formula is a logical theorem. The completeness is harder to prove.

Using other systems of rules, we might define other sets of provable formulas which may or may not coincide with valid formulas.

Deductive systems are systems of rules which enable us to define sets of formulas **included** in the set of valid formulas.

5.3 First order predicate calculus

A 'predicate' is an assertion about objects that may be true or false according to the objects to which it is applied. For instance, 'to be an even number' is true when applied to '2' and false when applied to '3'. A predicate can also be applied to several objects, for instance 'to be less than'. This assertion is true for the pair (2,3) and false for the pair (3,2).

Predicate calculus enables us to build complex statements from predicates. For instance, 'every prime number strictly greater than 2 is odd', which will be formally written as

$$\forall x \quad \big(\text{Prime}(x) \wedge x > 2\big) \Longrightarrow \text{odd } (x) \ .$$

Such complex statements may also be true or false.

5.3.1 Syntax: first order formulas

Let $\mathcal{G}$ be a set of *function symbols* . With each symbol f of $\mathcal{G}$ is associated an arity (or rank) $\rho(f) \in \mathbb{N}$. If $\rho(f) = 0$, then f is called a constant. Let $C = \{a, b, \ldots, a', b', \ldots, a_1, b_1, \ldots\} \subset \mathcal{G}$ be the set of constants.

Let $\mathcal{R}$ be a set of *relational symbols* . With each symbol R of $\mathcal{R}$ is associated an arity (or rank) $\rho(R) \in \mathbb{N}$. If $\rho(R) = 0$, then R is also called a propositional symbol.

Let $X = \{x, y, \ldots, x', y', \ldots, x_0, y_0, x_1, y_1, \ldots\}$ be a set of variables.

Define the language $\mathcal{L} = \mathcal{R} \cup \mathcal{G}$. We also consider the symbols $\supset, \neg, \wedge, \vee$ of propositional logic, and two symbols $\forall$ and $\exists$, called universal and existential quantifiers, together with both parentheses and the comma.

Recall that the set T of *terms built on* $\mathcal{G} \cup X$ is inductively defined by:

(B) $C \cup X \subseteq T$,

(I) for any n-ary f in $\mathcal{G}$, and for any $t_1, \ldots, t_n$ in T, $f\big(t_1, \ldots, t_n\big) \in T$.

A *ground term* is a variable-free term, i.e. a term built on $\mathcal{G}$.

First order formulas on $\mathcal{L}$ are inductively defined by:

- If R is an arity n relational symbol, and if $t_1, \ldots, t_n \in T$, then $R(t_1, \ldots, t_n)$ is a formula, called an *atomic formula*.

- If F and F' are formulas, then $\neg F$, $(F \supset F')$, $(F \wedge F')$, and $(F \vee F')$ are formulas.
- If F is a formula and x is a variable, then $\forall x F$ and $\exists x F$ are formulas.

Example 5.21
1. $F = (\forall x \exists y R(x, y) \supset \exists x R'(x, y, a))$.
2. Because $\mathcal{R}$ may contain arity 0 relational symbols, propositional calculus is a 'subcalculus' of predicate calculus. Every propositional formula is thus a first order formula because, on the one hand, propositional symbols are arity 0 relational symbols and, on the other hand, all other symbols of propositional calculus are also symbols of predicate calculus.

Definition 5.22 *An occurrence of a variable x in a formula F is a pair (x, n) such that the nth symbol of F is x and the $(n-1)$th symbol is neither $\forall$ nor $\exists$.*

Example 5.23 $(x, 8)$ and $(x, 17)$ are the two occurrences of x in the above formula F, $(x, 7)$ and $(x, 14)$ are not occurrences: $(x, 7)$ because the 7th symbol of F is not an x, and $(x, 14)$ because the 14th symbol of F, which indeed is an x, is quantified by $\exists$.

Let F be a formula. The set $SF(F)$ of the subformulas of F is the set of pairs (n, F') with $n \in \mathbb{N}$ and where

- F' is a consecutive sequence of symbols from F which is itself a formula,
- n is the occurrence of the first symbol of F' in F.

Example 5.24 The subformulas of

$$(\forall x \exists y R(x, y) \supset \exists x R'(x, y, a))$$

are $(1, F)$, $(2, \forall x \exists y R(x, y))$, $(4, \exists y R(x, y))$, $(6, R(x, y))$, $(13, \exists x R'(x, y, a))$, and $(15, R'(x, y, a))$.

Formulas can be represented by trees; for instance the formula

$$(\forall x \exists y R(x, y) \supset \exists x R'(x, y, a))$$

is represented by the tree t depicted in Figure 5.1.

With each node of t labelled by a relational symbol, a quantifier, or one of the symbols $\supset, \neg, \wedge, \vee$, is associated a subtree t' of t; each subtree t' represents a subformula of F. The subformulas of $(\forall x \exists y R(x, y) \supset \exists x R'(x, y, a))$ are depicted in Figure 5.2.

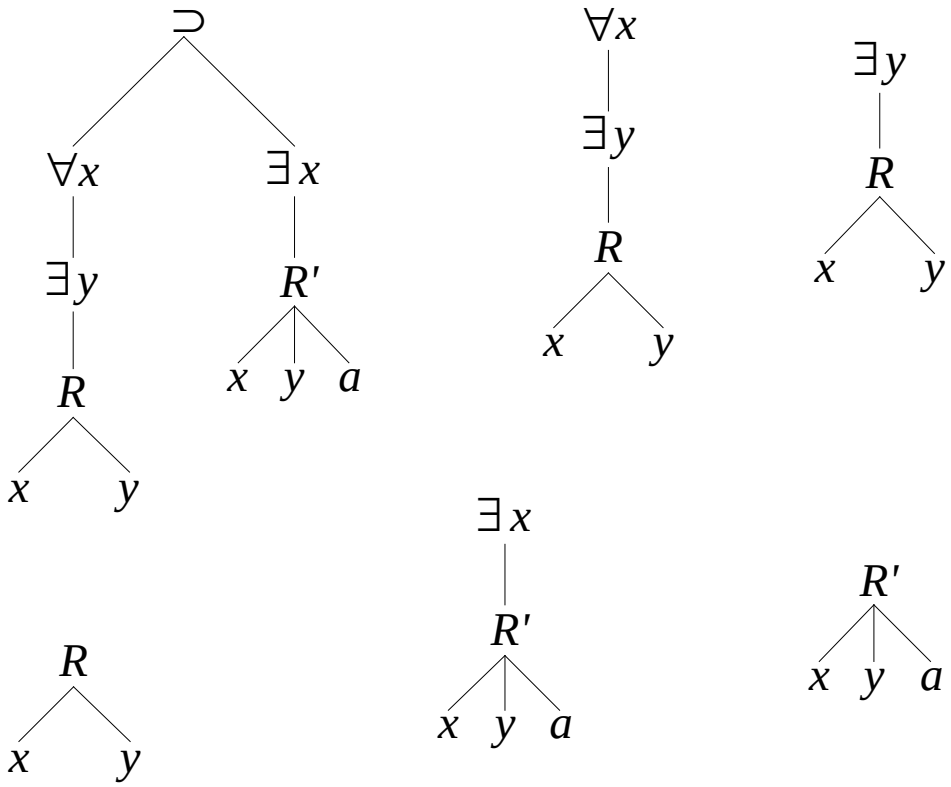

Figure 5.1

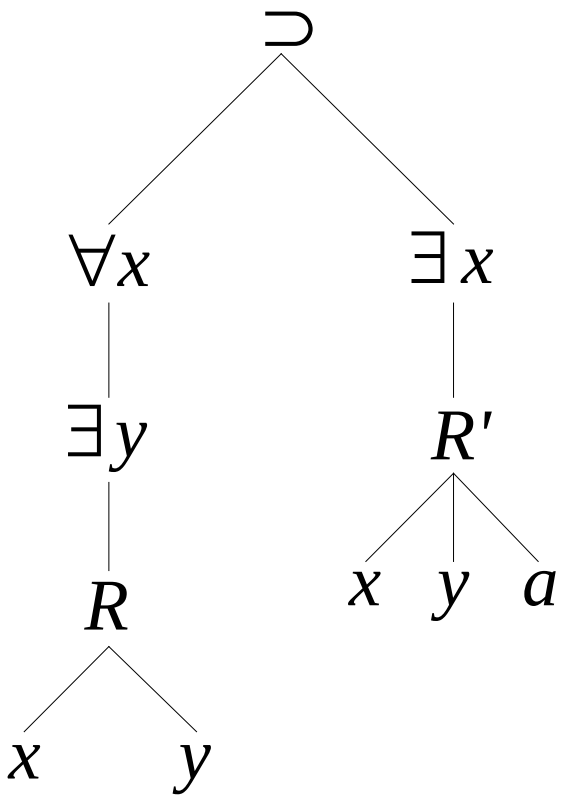

Figure 5.2

An occurrence (x, n) of x in F occurs in subformula (p, F') of F if and only if $p \leq n \leq p + |F'|$, where $|F'|$ denotes the number of symbols of F'.

EXAMPLE 5.25 $(x, 8)$ occurs in $(1, F)$, in $(2, \forall x \exists y R(x, y))$, in $(4, \exists y R(x, y))$ and in $(6, R(x, y))$.

$(y, 19)$ occurs in subformulas $(1, F)$, $(13, \exists x R'(x, y, a))$, and $(15, R'(x, y, a))$.

Definition 5.26 *An occurrence (x, n) of variable x in formula F is said to be bound if it occurs in a subformula (p, QxF'), where $Q \in \{\forall, \exists\}$. Otherwise it is said to be free.*

Variable x is said to be free in formula F if it has at least one free occurrence.

EXAMPLE 5.27 In Example 5.21, occurrences $(x, 8)$, $(x, 17)$ and $(y, 10)$ are bound; occurrence $(y, 19)$ is free.

EXERCISE 5.10 What are the free variables and the free occurrences of variables in the following formulas:

- $\exists x \big(\text{logician}(x) \wedge \text{astute}(x)\big)$,
- $\big(\exists x \, \text{logician}(x)\big) \wedge \text{astute}(x)$. ♢

Let $f(F)$ be the set of free variables of F.

Proposition 5.28

- $f(R(t_1,\ldots,t_n)) = \{u_i \,/\, u_i \in X$ *and* u_i *occurs in* $R(t_1,\ldots,t_n)\}$,
- $f(\neg F) = f(F)$,
- $f(F \supset F') = f(F \wedge F') = f(F \vee F') = f(F) \cup f(F')$ *and*
- $f(\forall x F) = f(\exists x F) = f(F) \backslash \{x\}$.

Proof. Simple: by structural induction on F (see Proposition 3.11). □

5.3.2 Semantics : Interpretation of formulas

Let $\mathcal{R}$ be a set of relational symbols and $\mathcal{G}$ a set of function symbols. Let $\mathcal{L}$ be the language $\mathcal{L} = \mathcal{R} \cup \mathcal{G}$. The language $\mathcal{L}$ will have many possible interpretations, each tailored for some domain of discourse. In order to interpret the language $\mathcal{L}$ we must specify the domain of discourse, together with the intended meanings of the predicate and function symbols; this is done by defining an $\mathcal{L}$-structure.

Definition 5.29 *An $\mathcal{L}$-structure is a triple $S = \langle E, \gamma, h\rangle$, where*

- *E is a non-empty set,*
- *γ is a mapping associating with each $R \in \mathcal{R}$ a subset $\gamma(R)$, also denoted by R_S, of $E^{\rho(R)}$ and*
- *h is a mapping associating with each $f \in \mathcal{G}$ a function $h(f) = f_S$ from $E^{\rho(R)}$ to E. (With each constant $a \in C$, h associates an element $h(a) = a_S$ of E.)*

Note that, here, E^0 has, by definition, a single element (for the same reason as $n^0 = 1$!). Hence, $\mathcal{P}(E^0)$ has only two subsets $\emptyset$ and E^0 and may be identified with the Boolean algebra with two elements.

A valuation v is a mapping from the set of variables to E. Two valuations v and v' are congruent on a subset Y of X, which is denoted by $v \underset{Y}{=} v'$, if: for all $x \in Y$, $v(x) = v'(x)$.

Definition 5.30

(i) *If t is a term and v a valuation, we define $v^*(t) \in E$ by:*
- *If $t = a \in C$, then $v^*(t) = a_S$.*
- *If $t = u \in X$, then $v^*(t) = v(u)$.*
- *If $t = f(t_1,\ldots,t_n)$, then $v^*(t) = f_S(v^*(t_1),\ldots,v^*(t_n))$.*

(ii) *If F is a formula and v a valuation, F can be assigned a unique truth value $\bar{v}(F) \in \mathbb{B}$ defined by:*
- *If $F = R(t_1,\ldots,t_n)$, then $\bar{v}(F) = 1$ if and only if $\big(v^*(t_1),\ldots,v^*(t_n)\big) \in R_S$. Note that if R has arity 0, then*

$$\bar{v}(F) = \begin{cases} 1 & \text{if } R_S \neq \emptyset, \\ 0 & \text{if } R_S = \emptyset. \end{cases}$$

- $\bar{v}(\neg F) = \overline{\bar{v}(F)}$.
- $\bar{v}(F \supset F') = 1$ *if and only if* $\bar{v}(F) \leq \bar{v}(F')$.
- $\bar{v}(F \wedge F') = 1$ *if and only if* $\bar{v}(F) = 1$ *and* $\bar{v}(F') = 1$.
- $\bar{v}(F \vee F') = 1$ *if and only if* $\bar{v}(F) = 1$ *or* $\bar{v}(F') = 1$.
- $\bar{v}(\forall x F) = 1$ *if and only if for all* v' *such that* $v' \underset{X-\{x\}}{=} v$, *we have*
$$\bar{v}'(F) = 1.$$
- $\bar{v}(\exists x F) = 1$ *if and only if there exists* v' *such that*
$$v' \underset{X-\{x\}}{=} v \text{ and } \bar{v}'(F) = 1.$$

(iii) *Two formulas* F *and* F' *are equivalent if, for any* $\mathcal{L}$*-structure* S *and for any valuation* v, $\bar{v}(F) = \bar{v}(F')$. *We write* $F \approx F'$.

This semantics agrees with the semantics already given for propositional calculus. We have seen in Example 5.21 that any formula F of propositional calculus is a formula of predicate calculus. If S is an $\mathcal{L}$-structure, the restriction I of S to propositional symbols is a mapping from these propositional symbols to the Boolean algebra, and is thus an interpretation in the sense of propositional calculus; and we indeed have for any propositional formula F that $I(F) = \bar{v}(F)$ for any valuation v with values in S. In short, interpretations I that we have considered for propositional logic can be obtained as restrictions of $\mathcal{L}$-structures.

Proposition 5.31 $\forall x F \approx \neg \exists x \neg F$.

Proof.

$$\begin{aligned}
\bar{v}(\forall x F) = 0 &\iff \text{there is a } v' \text{ such that } v' \underset{X-\{x\}}{=} v \text{ and } \bar{v}'(F) = 0 \\
&\iff \text{there is a } v' \text{ such that } v' \underset{X-\{x\}}{=} v \text{ and } \bar{v}'(\neg F) = 1 \\
&\iff \bar{v}(\exists x \neg F) = 1 \\
&\iff \bar{v}(\neg \exists x \neg F) = 0 \ .
\end{aligned}$$

□

Proposition 5.32 *Let* Y *be the set of variables having a free occurrence in* F. *If* $v \underset{Y}{=} v'$, *then* $\bar{v}(F) = \bar{v}'(F)$.

Proof. For this proof we use the fact that if $Y' \subseteq Y$ then $v \underset{Y}{=} v' \Longrightarrow v \underset{Y'}{=} v'$.

The proof is by induction on the structure of F.

- Basis. If $F = R(t_1, \ldots, t_n)$ then $f(F) = \{u_i \,/\, u_i \in X$, and u_i occurs in some $t_j\}$, and if $v \underset{f(F)}{=} v'$ then $\bar{v}(F) = \bar{v}'(F)$.
- Inductive step.
  - If $F = (F_1 \,\square\, F_2)$ with $\square \in \{\supset, \wedge, \vee\}$, we use the induction hypothesis: because $f(F_i) \subseteq f(F)$, we have that $v \underset{f(F)}{=} v' \implies v \underset{f(F_i)}{=} v'$.

– If $F = \exists x F'$, then $f(F) = f(F') - x$. Let $v_1 \underset{f(F)}{=} v_2$. If $\bar{v}_1(\exists x F') = 1$, there exists $v_1' \underset{X-\{x\}}{=} v_1$ such that $\bar{v}_1'(F') = 1$. Let v_2' be defined by

$$v_2'(y) = \begin{cases} v_2(y) & \text{if } y \neq x, \\ v_1'(x) & \text{otherwise.} \end{cases}$$

We have $v_2' \underset{X-\{x\}}{=} v_2$. As $f(F) \subseteq X - x$,

$$v_2' \underset{f(F)}{=} v_2 \underset{f(F)}{=} v_1 \underset{f(F)}{=} v_1' ,$$

and as $v_2'(x) = v_1'(x)$, $v_2' \underset{f(F)\cup\{x\}}{=} v_1'$. Because $f(F') \subseteq f(F) \cup \{x\}$, we have that $\bar{v}_2'(F') = \bar{v}_1'(F') = 1$, and hence $\bar{v}_2(\exists x F') = 1$.

– If $F = \forall x F'$, then $f(F) = f(F') - x = f(\neg\exists x \neg F')$, and hence

$$\begin{aligned} v_1 \underset{f(F)}{=} v_2 &\Longrightarrow \bar{v}_1(\neg\exists x \neg F') = \bar{v}_2(\neg\exists x \neg F') \\ &\Longrightarrow \bar{v}_1(\forall x F) = \bar{v}_2(\forall x F). \end{aligned}$$

□

REMARK 5.33 If a formula F contains no occurrence of a free variable (in which case it is said to be a *closed* or *ground formula* or a *sentence*), then its truth value in S does not depend on the valuation. Indeed, for any v, v', we have $v \underset{\emptyset}{=} v'$, and hence for any v, v', $\bar{v}(F) = \bar{v}'(F)$. This occurs if F is a propositional logic formula.

EXERCISE 5.11 In Aristotle's syllogisms, assertions about properties P and Q of individuals frequently occur in the following forms:
(i) *All Ps are Qs.*
(ii) *Some Ps are Qs.*
(iii) *No P is a Q.*
(iv) *Some Ps are not Qs.*

Translate these assertions into predicate calculus formulas by introducing the predicates $P(x)$ and $Q(x)$. ♢

EXERCISE 5.12 Show that if x is not free in F, then

$$\bar{v}(\forall x F) = \bar{v}(\exists x F) = \bar{v}(F).$$ ♢

Definition 5.34 *A formula F is said to be*

- *satisfiable in S if there exists a valuation v such that $\bar{v}(F) = 1$,*
- *satisfiable if there exist a structure S and a valuation v such that $\bar{v}(F) = 1$,*
- *valid in S if for all v, $\bar{v}(F) = 1$ and*
- *universally valid if it is valid in all $\mathcal{L}$-structures.*

Example 5.35
1. $\big(\,(\neg\exists x P(x)) \Longleftrightarrow \forall x(\neg P(x))\,\big)$ is universally valid.

If x and y are real numbers, and S is the structure associated with $\mathbb{R}$, then $x \leq x + y$ is satisfiable in S but it is not valid in S.

Let $F = R(x,z) \wedge Q(x,y,z)$. Consider the structure $S = (\mathbb{N}, R_{\mathbb{N}}, Q_{\mathbb{N}})$, defined by $R_{\mathbb{N}} = \{(n,m)\,/\,n < m\}$ and $Q_{\mathbb{N}} = \{(n,m,p)\,/\,n + m = p\}$. F is satisfiable in S (let, for instance, $v(x) = v(y) = 1$, $v(z) = 2$), but F is not valid in S (let for instance $v(x) = v(z) = 1$, $v(y) = 0$).
2. (Examples are given with the PROLOG terminology.) Let $\mathcal{R} = \{\text{male}, \text{female}\}$ be a set of two unary predicates. Then

$$A = \big(\,(\neg\ \text{male}(x)) \Longrightarrow\ \text{female}(x)\,\big)$$

is satisfiable but not valid, whilst

$$C = \Big(\big(\,(\neg\text{male}(x)) \Longrightarrow\ \text{female}(x)\,\big)\ \vee\ \big(\,\neg\text{male}(x) \wedge \neg\text{female}(x)\,\big)\Big)$$

is valid.

Exercise 5.13 Let $S = \langle E, \{R, =\}\rangle$ be a set equipped with a relation R and a predicate $=$ that we assume to be interpreted as equality. Write a formula that is valid in S if and only if R is a (total) ordering. ♢

Exercise 5.14
1. Is $\exists y \forall x\, r(x,y) \approx \forall x \exists y\, r(x,y)$ valid for a binary predicate r? Same question for $(\exists y p(y)) \wedge (\exists y q(y)) \approx (\exists y (p(y) \wedge q(y)))$, with p and q unary predicates. Give a proof if the answer is yes, a counterexample if the answer is no.
2. Show that

$$\exists y \forall x \big(p(x) \wedge q(y)\big) \approx \forall x \exists y \big(p(x) \wedge q(y)\big)\ ,$$

for unary predicates p and q. ♢

As for propositional logic we define sequents.

Definition 5.36 *A sequent $(\mathcal{F}, F)$ is valid in S, denoted by $\mathcal{F} \underset{S}{\models} F$, if*

$$\textit{for any } v, \quad \Big(\big(\textit{ for all } G \textit{ in } \mathcal{F},\ \bar{v}(G) = 1\big) \Longrightarrow (\bar{v}(F) = 1)\Big).$$

A sequent is universally valid, denoted by $\mathcal{F} \models F$, if it is valid in all Ss.

Proposition 5.37

$$\{F_1, \ldots, F_n\} \underset{S}{\models} F \textit{ if and only if } \emptyset \underset{S}{\models} (F_1 \supset (F_2 \supset \cdots (F_n \supset F) \cdots)).$$

Proposition 5.38 *If $\mathcal{F} \underset{S}{\models} F$ and if x is not free in any formula of $\mathcal{F}$, then $\mathcal{F} \underset{S}{\models} \forall x F$.*

Proof. Let v be a valuation such that for any $G \in \mathcal{F}$ $\bar{v}(G) = 1$. Let v' be any valuation such that $v' \underset{X-\{x\}}{=} v$. Since x never has a free occurrence in $\mathcal{F}$, for any $G \in \mathcal{F}$ we also have $\bar{v}'(G) = 1$ and thus $\bar{v}'(F) = 1$. Since this is true for any $v' \underset{X-\{x\}}{=} v$, we have that $\bar{v}(\forall x F) = 1$. □

Let F be a formula and let x be a variable. Let t be a term. Let $F[x := t]$ be the formula where all *free* occurrences of x have been replaced by t. $F[x := t]$ is said to be an *instance* of F; if $F[x := t]$ is a formula without variables, then it is said to be a *ground instance* of F. If x has no free occurrence in F, then $F[x := t] = F$.

Let u be a term. We will say that u is *substitutable for x in F* if u is a ground term or if u is such that any occurrence of a variable in u is free in $F[x := u]$.

EXAMPLE 5.39 Let $F = \big(\forall y R(x, y, z)\big) \vee \big(\forall z R'(z)\big)$, where R, R' are relational symbols. u is substitutable for x in F if and only if y does not occur in u. For example y is not substitutable for x in F because $F[x := y] = \big(\forall y R(y, y, z)\big) \vee \big(\forall z R'(z)\big)$ and occurrence $(y, 5)$ becomes bound. Similarly let f be a function symbol, $f(y, z)$ is not substitutable for x in F, but $f(x, z)$ is substitutable for x in F.

From now on, when we write $F[x := u]$, we will implicitly assume that u is substitutable for x in F.

Proposition 5.40 *Let x be a variable and let u be a term substitutable for x in F. Let v be a valuation. Let v' be defined by:*

$$v'(y) = \begin{cases} v(y) & \text{if } y \neq x, \\ v^*(u) & \text{if } y = x. \end{cases}$$

Then $\bar{v}'(F) = \bar{v}(F[x := u])$.

Proof. By induction on the construction of F. □

Proposition 5.41

1. $\mathcal{F} \underset{S}{\models} \forall x F \implies \mathcal{F} \underset{S}{\models} F[x := u]$;
2. $\mathcal{F} \underset{S}{\models} F[x := u] \implies \mathcal{F} \underset{S}{\models} \exists x F$.

Proof. Let v be such that for any $G \in \mathcal{F}$, $\bar{v}(G) = 1$. Let v' be constructed as previously in Proposition 5.40; we have $v' \underset{X-\{x\}}{=} v$ and $\bar{v}'(F) = \bar{v}(F[x := u])$.

1. If $\mathcal{F} \underset{S}{\models} \forall x F$ then $\bar{v}(\forall x F) = 1$, and $\bar{v}'(F) = 1$; thus $\bar{v}(F[x := u]) = 1$ which proves that $\mathcal{F} \underset{S}{\models} F[x := u]$.
2. If $\mathcal{F} \underset{S}{\models} F[x := u]$ then $\bar{v}'(F) = 1$, and hence $\bar{v}(\exists x F) = 1$. □

5.3.3 Some particular formulas

We now give some identities of predicate calculus that are quite useful. In the present section we abbreviate $F \approx G$ by $F \iff G$ to comply with the usual notations when logic is used as meta-language and to increase readability; we will also use the notation $F \Longrightarrow G$ to denote that $F \supset G$ is universally valid, i.e. that $\emptyset \models F \supset G$.

Proposition 5.42

(i) $\quad \forall x\,(p(x) \wedge q(x)) \iff \forall x\, p(x) \wedge \forall x\, q(x)\ .$

(ii) $\quad \exists x\,(p(x) \wedge q(x)) \Longrightarrow \exists x\, p(x) \wedge \exists x\, q(x)\ .$

By duality between $\forall$ and $\exists$ we also have

(iii) $\quad \exists x\,(p(x) \vee q(x)) \iff \exists x\, p(x) \vee \exists x\, q(x)\ .$

(iv) $\quad \forall x\, p(x) \vee \forall x\, q(x) \Longrightarrow \forall x\,(p(x) \vee q(x))\ .$

Z The converses of rules (ii) and (iv), namely,

$$\exists x\, p(x) \wedge \exists x\, q(x) \implies \exists x\,(p(x) \wedge q(x))$$

and

$$\forall x\,(p(x) \vee q(x)) \implies \forall x\, p(x) \vee \forall x\, q(x)\ ,$$

are false (see Exercise 5.14).

Lastly, the following rules, written with the same conventions as above, are useful for putting formulas in prenex form, i.e. with all quantifiers at the beginning of the formula.

Proposition 5.43 *Let $* \in \{\vee, \wedge\}$, let F be a formula, let x be a variable and let G be a formula in which x has no free occurrence. We have:*

(i)	$\neg\forall x\, F \iff \exists x \neg F$	$\neg\exists x\, F \iff \forall x \neg F$
(ii)	$(\forall x\, F) * G \iff \forall x\,(F * G)$	$(\exists x\, F) * G \iff \exists x\,(F * G)$
(iii)	$G * (\forall x\, F) \iff \forall x\,(G * F)$	$G * (\exists x\, F) \iff \exists x\,(G * F)$
(iv)	$(\forall x\, F) \supset G \iff \exists x\,(F \supset G)$	$(\exists x\, F) \supset G \iff \forall x\,(F \supset G)$
(v)	$G \supset (\forall x\, F) \iff \forall x\,(G \supset F)$	$G \supset (\exists x\, F) \iff \exists x\,(G \supset F)$

The proofs of the two preceding propositions are straightforward.

5.3.4 Lexical variants

Let F be a formula. A lexical variant of F is a formula deduced from F by renaming some bound variables. Formally:

Definition 5.44 *F' is a lexical variant of F if one of the following holds:*

- *$F = F'$.*
- *For $Q \in \{\forall, \exists\}$, $F = QxG$ and $F' = QyG'[x := y]$, where G' is a lexical variant of G and y is not free in G'.*
- *$F = \neg G$, $F' = \neg G'$, and G' is a lexical variant of G.*
- *For $\square \in \{\supset, \wedge, \vee\}$, $F = (F_1 \square F_2)$, $F' = (F'_1 \square F'_2)$, and F'_i is a lexical variant of F_i.*

EXAMPLE 5.45 $\forall z P(z, y)$ is a lexical variant of $\forall x P(x, y)$, but neither $\forall y P(y, y)$ nor $\forall z P(z, x)$ is.

Proposition 5.46 *If F is a lexical variant of F' then $f(F) = f(F')$, and for any $\mathcal{L}$-structure S and any valuation v in S, $\bar{v}(F) = \bar{v}(F')$.*

Proof. The first assertion is easy to prove. For the second one, it suffices to show that if for all v, $\bar{v}(F) = \bar{v}(F')$, then for all v, $\bar{v}(\exists x F) = \bar{v}(\exists y F'[x := y])$ if y is not free in F'.

Let v be a valuation, and let V be the set of valuations v' such that $v' \underset{X-\{x\}}{=} v$ and $v'(x) = v'(y)$.

Because y is not a free variable of F', y is not a free variable of F either, and: there exists v' such that $v' \underset{X-\{x\}}{=} v$ and $\bar{v}'(F) = 1$ if and only if there exists v' such that $v' \in V$ and $\bar{v}'(F) = 1$.

Similarly, x is not a free variable of $F'[x := y]$, and thus: there exists v' such that $v' \underset{X-\{x\}}{=} v$ and $\bar{v}'(F'[x := y]) = 1$ if and only if there exists v' such that $v' \in V$ and $\bar{v}'(F'[x := y])$.

Hence $\bar{v}(\exists x F) = \bar{v}(\exists y F'[x := y])$ is equivalent to: there exists $v' \in V$ such that $\bar{v}'(F) = 1$ if and only if there exists v' such that $v' \in V$ and $\bar{v}'(F'[x := y]) = 1$. Finally, we observe that this last equivalence is true. □

5.3.5 Prenex formulas

Definition 5.47 *A formula F is said to be prenex if it is in the form*

$$Q_1 x_1 Q_2 x_2 \ldots Q_n x_n F' \ ,$$

where the Q_is are quantifiers for $i = 1, 2, \ldots, n$, and where F' is a formula without quantifier.

Theorem 5.48 *Any formula F is equivalent to a prenex formula G.*

Proof. By structural induction on F we effectively build a prenex formula G equivalent to F. The inductive assumption is that there exists a prenex formula G equivalent to F.

- Basis. If F is in the form $R(t_1, \ldots, t_n)$, it is clear because F is prenex.
- Inductive step.
  - If F is in the form $\forall x F'$ (resp. $\exists x F'$), with $F' \approx F''$, where F'' is prenex, then $F \approx \forall x F''$ (resp. $F \approx \exists x F''$), which is prenex.
  - If F is in the form $\neg F'$, with $F' \approx F''$, F'' prenex and

$$F'' = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n G,$$

then, by Proposition 5.43 (i),

$$F' \approx Q'_1 x_1 Q'_2 x_2 \ldots Q'_n x_n \neg G,$$

with $Q'_i = \forall$ if $Q_i = \exists$ and $Q'_i = \exists$ if $Q_i = \forall$.

  - If F is in the form $F_1 * F_2$, with $* \in \{\vee, \wedge\}$, we put F_1 and F_2 in prenex form, and we apply Proposition 5.43 (ii) and (iii) in order to 'pull' the quantifiers to the beginning of the formula. We must, however, proceed with care; if, for instance, $F = F_1 \wedge F_2 \approx (\forall x F'_1) \wedge F'_2$, with x free in F'_2, we must first rename variable x in F_1 and replace x by a new variable z occurring neither in F_1 nor in F'_2.
  - If F is in the form $F_1 \supset F_2$, we put F_1 and F_2 in prenex form, and we apply Proposition 5.43 (iv) and (v), possibly with renamings, to 'pull' quantifiers to the beginning of the formula. □

Exercise 5.15 Find a prenex formula equivalent to

$$\exists x P(x) \wedge \forall x \big(\exists y Q(y) \supset R(x)\big).$$ ♢

Exercise 5.16 We assume that:

(a) Programmers write programs for all those who do not write programs for themselves.

(b) No programmer writes programs for someone who writes programs for him(her)self.

We then notice the paradox: if a programmer writes a program for him(her)self, he/she violates rule (b); if a programmer does not write programs for him(her)self, he/she violates rule (a) because he/she does not write programs for someone who does not write programs for him(her)self. How do you explain this paradox? (Hint: express the requirements (a) and (b) by formulas F and G of predicate calculus, and show that $F \wedge G$ implies that there is no programmer.) ♢

5.4 Herbrand's theorem and consequences

5.4.1 Theories and Models

Definition 5.49 *A provable sequent is a sequent obtained by the rules for propositional logic, extended by:*

- *If $\mathcal{F} \vdash \forall x F$ then: $\mathcal{F} \vdash F[x := t]$ (instantiation rule).*
- *If $\mathcal{F} \vdash F$ and if x is not free in $\mathcal{F}$, then: $\mathcal{F} \vdash \forall x F$ (universal generalization rule).*
- *$\mathcal{F} \vdash \exists x F$ if and only if $\mathcal{F} \vdash \neg\forall x \neg F$ (definition of $\exists$).*

Z The universal generalization rule does not apply if x is free in $\mathcal{F}$. For instance, $p(x) \vdash p(x)$ is provable, but $p(x) \vdash \forall x p(x)$ is not provable. This rule is the formalization of the following reasoning: 'if a property is true for an *arbitrary* object x then it is true for any x'; x is arbitrary means that no hypothesis is made about x and that lack of knowledge about x is formally expressed by the fact that x does not occur free in $\mathcal{F}$.

Example 5.50

- If $\mathcal{F}, F \vdash G$ and if x is not free in $\mathcal{F}$ and G then $\mathcal{F}, \exists x F \vdash G$. Indeed, we have:

1.	$\mathcal{F}, F \vdash G$	
2.	$\mathcal{F}, \neg G \vdash \neg F$	(by Proposition 5.14)
3.	$\mathcal{F}, \neg G \vdash \forall x \neg F$	(generalization)
4.	$\mathcal{F}, \neg\forall x \neg F \vdash G$	(by Proposition 5.14)
5.	$\mathcal{F}, \exists x F \vdash G$	(definition of $\exists$)

- $\exists x \forall y F \vdash \forall y \exists x F$. Indeed, we have:

1.	$\forall y F\,, \forall x \neg F \vdash \forall y F$	(hypothesis)
2.	$\forall y F\,, \forall x \neg F \vdash F[y := y]$	(instantiation)
3.	$\forall y F\,, \forall x \neg F \vdash \forall x \neg F$	(hypothesis)
4.	$\forall y F\,, \forall x \neg F \vdash \neg F[x := x]$	(instantiation)
5.	$\forall y F \vdash \neg(\forall x \neg F)$	(contradiction on 2, 4 because $F[y := y] = F[x := x] = F$)
6.	$\forall y F \vdash \exists x F$	(definition of $\exists$)
7.	$\forall y F \vdash \forall y \exists x F$	(universal generalization)
8.	$\exists x \forall y F \vdash \forall y \exists x F$	

We detail steps 7 and 8 of the proof: because y is not free in $\forall y F$, universal generalization applied to 6 gives $\forall y F \vdash \forall y \exists x F$, and because x is not free in $\forall y \exists x F$, we have, by applying to 7 the sequent obtained in the first part of the present example, $\exists x \forall y F \vdash \forall y \exists x F$.

Theorem 5.51 (Soundness) *If a sequent is provable then it is universally valid.*

EXERCISE 5.17 Prove this theorem. ♢

Theorem 5.52 (Completeness) *If a sequent is universally valid then it is provable.*

We will not give the proof of this theorem; rather, we will provide some ideas behind the proof.

Definition 5.53 *A theory is a set T of formulas such that for any finite subset $\mathcal{F}$ of T, if $\mathcal{F} \vdash F$ then $F \in T$.*

A theory T is contradictory if there exists a formula F such that $F \in T$ and $\neg F \in T$.

EXAMPLE 5.54 Let $\mathcal{F}$ be a finite set of formulas. The set

$$Th(\mathcal{F}) = \{F \ / \ \mathcal{F} \vdash F\}$$

is a theory.

Proposition 5.55 *A theory T is contradictory if and only if it contains all formulas.*

Proof. Let G be a formula.

$$\left.\begin{array}{l} F, \neg F, \neg G \vdash F \\ F, \neg F, \neg G \vdash \neg F \end{array}\right\} \quad \text{hence} \quad F, \neg F \vdash G \ .$$

Thus $G \in T$. □

Proposition 5.56 *$\mathcal{F} \vdash F$ if and only if $Th(\mathcal{F} \cup \{\neg F\})$ is contradictory.*

Proof. If $\mathcal{F} \vdash F$, then:

$$\left.\begin{array}{ll} & \mathcal{F}, \neg F \vdash F \\ \text{and} & \mathcal{F}, \neg F \vdash \neg F \end{array}\right\} \Longrightarrow \text{ the theory } \mathcal{F} \cup \{\neg F\} \text{ is contradictory .}$$

If $Th(\mathcal{F} \cup \{\neg F\})$ is contradictory, $\mathcal{F}, \neg F \vdash F$ and $\mathcal{F}, \neg F \vdash \neg F$, and hence $\mathcal{F} \vdash F$. □

An $\mathcal{L}$-structure S is a *model* of a (finite or infinite) set $\mathcal{G}$ of formulas if for any v and for any F in $\mathcal{G}$, we have $\bar{v}(F) = 1$.

We denote by $\emptyset \underset{S}{\models} \mathcal{G}$ the fact that S is a model of $\mathcal{G}$.

A set $\mathcal{G}$ of formulas is *satisfiable* if *there exists* an $\mathcal{L}$-structure S and *there exists* a valuation v such that for any F in $\mathcal{G}$, we have $\bar{v}(F) = 1$. A set $\mathcal{G}$ of formulas has a *model* if *there exists* an $\mathcal{L}$-structure S such that *for any* valuation v and for any F in $\mathcal{G}$, we have $\bar{v}(F) = 1$. A set $\mathcal{G}$ of formulas which does not have a model may thus be satisfiable: for example, the set $\mathcal{F}$ of formulas defined in Remark 5.58 does not have a model, but it is satisfiable.

If a theory has a model, then it is not contradictory. The converse is one of the fundamental theorems of logic. The proof of this theorem is quite long, so we will not give it; we state the theorem.

Theorem 5.57 *If $\mathcal{F}$ consists of closed formulas and $Th(\mathcal{F})$ is not contradictory, then $\mathcal{F}$ has a model.*

REMARK 5.58 Theorem 5.57 is false if non-closed formulas are allowed. Let $\mathcal{F} = \{\exists x p(x)\,,\, \neg p(x)\}$; $\mathcal{F}$ is satisfiable: let $S = \langle E, \gamma\rangle$ with $E = \{0, 1\}$, and $\gamma(p) = p_S$ defined by $p_S(0) = 0$, $p_S(1) = 1$, the valuation $v(x) = 0$ is such that $\bar{v}(F) = 1$ for any F in $\mathcal{F}$. $\mathcal{F}$ does not have a model: if a structure S' is such that for any valuation v, $\bar{v}\big(\neg p(x)\big) = 1$, then, for any valuation v, $\bar{v}\big(\exists x p(x)\big) = 0$. Nevertheless $Th(\mathcal{F})$ is not contradictory: otherwise, by Proposition 5.56, we would conclude that $\exists x p(x) \vdash p(x)$, which is false because the sequent $\{\exists x p(x)\,,\, p(x)\}$ is not universally valid, e.g. $\exists x p(x) \underset{S}{\not\models} p(x)$.

Let us deduce the completeness theorem from Theorem 5.57. We prove that if $\mathcal{F} \models F$ then $\mathcal{F} \vdash F$.

1. First, we consider closed formulas: we will assume that $\mathcal{F}$ and F consist of closed formulas, that $\mathcal{F} \models F$, and that the sequent $(\mathcal{F}, F)$ is not provable. Then $\mathcal{F} \cup \{\neg F\}$ is not contradictory and thus has a model S by Theorem 5.57. Any valuation v thus verifies for any $F_i \in \mathcal{F}$, $\bar{v}(F_i) = 1$ and $\bar{v}(\neg F) = 1$. But, because $\mathcal{F} \models F$, the sequent $(\mathcal{F}, F)$ is valid in S and thus any valuation v verifying $\bar{v}(F_i) = 1$ for any F_i in $\mathcal{F}$ also verifies $\bar{v}(F) = 1$, a contradiction.
2. Now, we will assume that $\mathcal{F} \models F$, and that $\mathcal{F} = \{F_1, \ldots, F_n\}$ and F consist of (not necessarily closed) formulas. $\mathcal{F} \models F$ and Proposition 5.37 imply that $\emptyset \models (F_1 \supset (F_2 \supset \cdots (F_n \supset F)\cdots))$. Let $x_1, \ldots, x_k$ be the free variables of $(F_1 \supset (F_2 \supset \cdots (F_n \supset F)\cdots))$; by Proposition 5.38, $\emptyset \models \forall x_1 \cdots \forall x_k (F_1 \supset (F_2 \supset \cdots (F_n \supset F)\cdots))$.

$\big(\emptyset\,,\, \forall x_1 \cdots \forall x_k (F_1 \supset (F_2 \supset \cdots (F_n \supset F)\cdots))\big)$ thus is a universally valid sequent consisting of closed formulas, hence it is provable by case 1, and $\emptyset \vdash \forall x_1 \cdots \forall x_k (F_1 \supset (F_2 \supset \cdots (F_n \supset F)\cdots))$. By the instantiation rule, $\emptyset \vdash (F_1 \supset (F_2 \supset \cdots (F_n \supset F)\cdots))$. As in Proposition 5.16, we deduce that $\{F_1, \ldots, F_n\} \vdash F$, hence the sequent $(\mathcal{F}, F)$ is provable. □

EXERCISE 5.18 Bernard and Christopher are members of the Alpine Club. Any member of the Alpine Club is either a skier, or an alpinist or both. Alpinists do not like rain, and skiers like snow. Christopher likes all that Bernard does not like, and does not like all that Bernard likes (i.e. there are things that Bernard likes and that Christopher does not like).

1. Express the requirements of the Alpine Club by a set $\mathcal{F}$ of formulas of predicate calculus.
2. Can you find a model of $\mathcal{F}$?
3. Can you prove that there is a member of the Alpine Club who is an alpinist and not a skier (or vice versa)? ♢

5.4.2 Herbrand's models

Let $\mathcal{G}$ be a set of functions whose set C of constants is non-empty. Let $\mathcal{L}$ be the language $\mathcal{L} = \mathcal{R} \cup \mathcal{G}$. The *Herbrand universe* of $\mathcal{L}$ is the set of ground (i.e. variable-free) terms built over $\mathcal{G}$. The Herbrand universe is denoted by U_H, which is inductively defined by:

(B) $C \subseteq U_H$,
(I) for any n-ary f in $\mathcal{G}$, and for any $t_1, \ldots, t_n$ in U_H, $f\big(t_1, \ldots, t_n\big) \in U_H$.

Because C is non-empty, U_H is non-empty. (Indeed U_H is empty if and only if C is empty.)

An $\mathcal{L}$-structure $S = \langle E, \gamma, h\rangle$ is a *Herbrand structure* if:

- $E = U_H$, and
- h associates with each $f \in \mathcal{G}$ the function f_S from $U_H^{\rho(f)}$ to U_H defined by $f_S(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$, see Definition 3.14. (This implies that with each constant $a \in C$, h associates the element a of U_H.)

The definition of an $\mathcal{L}$-structure (Definition 5.29) requires the domain of the structure to be non-empty. Thus U_H must be non-empty. This is why we require C to be non-empty.

The *Herbrand basis* of $\mathcal{L}$ is the set B_H of ground (i.e. variable-free) atomic formulas, i.e. formulas of the form $R(t_1, \ldots, t_n)$ with $R \in \mathcal{R}$ and $t_1, \ldots, t_n$ in U_H. For a given language $\mathcal{L}$, there is a single Herbrand universe, but on that Herbrand universe numerous Herbrand structures can be defined; a Herbrand structure H is defined by a *Herbrand interpretation* which is a subset I of the Herbrand basis B_H; I specifies the atomic formulas which are true in H. Formally, I defines $H = \langle U_H, \gamma, h\rangle$ means that for $t_1, \ldots, t_n$ in U_H and $R \in \mathcal{R}$, $(t_1, \ldots, t_n)$ is in R_H if and only if $R(t_1, \ldots, t_n) \in I$. A Herbrand structure will thus be denoted by $H = \langle U_H, I, h\rangle$, or by $H = \langle U_H, I\rangle$ or simply by I, since U_H and h are uniquely specified by the language.

EXAMPLE 5.59
1. Let $\mathcal{L}$ be the language $\mathcal{L} = \{a, p, q\}$ where a is a constant symbol and p, q are nullary relational symbols. Then the Herbrand universe is $\{a\}$, the Herbrand basis is the set $B_H = \{p, q\}$ and there are exactly four Herbrand structures specified by $I_0 = \emptyset$, $I_1 = \{p\}$, $I_2 = \{q\}$, $I_3 = \{p, q\}$.
2. Let $\mathcal{L}$ be the language $\mathcal{L} = \{a, f, p, q\}$ where p, q are unary relational symbols, a is a constant symbol and f is a unary function symbol. Then the Herbrand universe is $U_H = \{a\,,\, f(a)\,,\, f^2(a)\,,\, \ldots\,,\, f^n(a)\,, \ldots\} = \{f^n(a) \,/\, n \in \mathbb{N}\}$, the Herbrand basis is the set $B_H = \{p(t), q(t) \,/\, t \in U_H\}$ and the Herbrand structures are specified by subsets I of B_H; for instance $I_0 = \emptyset$, $I_1 = B_H$, $I_2 = \{p(a), p(f(a))\}$, $I_3 = \{p(t) \,/\, t \in U_H\}$, etc., specify Herbrand structures H_0, H_1, H_2, H_3, etc.

Proposition 5.60 *For any $\mathcal{L}$-structure $A = \langle E, \gamma, h\rangle$ there is a unique Herbrand structure H and a unique mapping $h^*\colon U_H \longrightarrow E$ such that*
(i) $h^*\big(f(t_1, \ldots, t_n)\big) = f_A(h^*(t_1), \ldots, h^*(t_n))$ *and*
(ii) *for $t_1, \ldots, t_n$ in U_H, $(t_1, \ldots, t_n) \in R_H \Longleftrightarrow (h^*(t_1), \ldots, h^*(t_n)) \in R_A$.*

Proof. The uniqueness of the mapping $h^*\colon U_H \longrightarrow E$ follows from (i) and Proposition 3.15; (ii) implies that the Herbrand structure on U_H must be defined by $I = \{R(t_1, \ldots, t_n) \in B_H \,/\, (h^*(t_1), \ldots, h^*(t_n)) \in R_A\}$. □

Definition 5.61 *Let $\mathcal{F}$ be a set of formulas of the language $\mathcal{L}$ and let H be a Herbrand structure for $\mathcal{L}$. H is said to be a Herbrand model of $\mathcal{F}$ if and only if H is a model of $\mathcal{F}$.*

EXAMPLE 5.62 Let $\mathcal{L} = \{a, f, p, q\}$, let $\mathcal{F} = \{p(a)\,,\, \forall x\big(p(x) \supset p(f(x))\big)\}$. I_1 and I_3 in Example 5.59, 2, define Herbrand models of $\mathcal{F}$; I_0 and I_2 in Example 5.59, 2, do not define Herbrand models of $\mathcal{F}$; any $I = I_3 \cup \{q(f^k(a)) \,/\, k \in K \subset \mathbb{N}\}$ also defines a Herbrand model of $\mathcal{F}$.

5.4.3 Herbrand's theorem

Definition 5.63 *A prenex formula is said to be universal if and only if it has only universal quantifiers.*

Theorem 5.64 (Herbrand's theorem) *Let $\mathcal{L}$ be a language with a non-empty set C of constants, and let $\mathcal{F}$ be a set of closed universal formulas, then $\mathcal{F}$ has a model if and only if $\mathcal{F}$ has a Herbrand model.*

Proof. The 'if' direction is clear. For the 'only if' direction, assume that $\mathcal{F}$ has a model $S = \langle E, \gamma, h\rangle$ and construct a Herbrand model for $\mathcal{F}$. Let H be

the Herbrand structure defined by the following set I of atomic formulas in the Herbrand basis:

$$I = \{F \in B_H \,/\, \emptyset \underset{S}{\models} F\} \,.$$

We will prove that H is a Herbrand model of $\mathcal{F}$. Because C is non-empty, U_H is non-empty, and with any valuation $v_H \colon X \longrightarrow U_H$ we can associate a unique valuation $v = h^* \circ v_H$,

$$v \colon X \xrightarrow{v_H} U_H \xrightarrow{h^*} E \,,$$

where h^* is defined in Proposition 5.60. By structural induction on the formulas, it can be shown that for any quantifier-free formula G, $\bar{v}_H(G) = \bar{v}(G)$. The base case follows from Proposition 5.60 and the inductive step is straightforward. We must prove that for any $F = \forall x_1 \cdots \forall x_n G$ in $\mathcal{F}$, where G is a quantifier-free formula, $\emptyset \underset{H}{\models} F$ holds, i.e. for any valuation $v_H \colon \{x_1, \ldots, x_n\} \longrightarrow U_H$, $\bar{v}_H(G) = 1$. If v_H is a valuation, $v = h^* \circ v_H$ is a valuation into E, and since $\emptyset \underset{S}{\models} F$, $\bar{v}(G) = 1$. Hence, $\bar{v}_H(G) = 1$. □

Example 5.65 Herbrand's theorem does not hold if $\mathcal{F}$ is not a set of universal formulas. Let $\mathcal{L} = \{a, R\}$ with a a constant, R a unary relational symbol, and let $\mathcal{F} = \{R(a) \ , \ \exists x \neg R(x)\}$. $\mathcal{F}$ has a model but $\mathcal{F}$ has no Herbrand model. The structure S defined by $E = \{0, 1\}$ with $a_S = 0$ and $0 \in R_S$, $1 \notin R_S$ is a model of $\mathcal{F}$.

$\mathcal{F}$ has no Herbrand model. There are exactly two Herbrand structures on the Herbrand universe $U_H = \{a\}$, defined by, respectively, $I_0 = \emptyset$ (i.e. $R_{I_0} = \emptyset$ is always false) and $I_1 = \{R(a)\}$ (i.e. $R_{I_1} = \{a\}$ is always true), neither of which is a model of $\mathcal{F}$.

Remark 5.66 Herbrand's theorem holds if $\mathcal{F}$ is a set of formulas without quantifiers. Indeed, if $F(x)$ is not closed, S is a model of $F(x)$ if and only if S is a model of $\forall x F(x)$.

In fact, Theorem 5.64 is a weakened form of Herbrand's theorem which asserts the following more general result.

Theorem 5.67 *Let $\mathcal{F}$ be a set of closed universal formulas, either*

- *$\mathcal{F}$ has a Herbrand model or*
- *$\mathcal{F}$ does not have a model and, moreover, there are finitely many ground instances of $\mathcal{F}$ whose conjunction is unsatisfiable.*

The proof of Theorem 5.67 will not be given here.

Herbrand's theorem has many useful consequences in logic programming and proof theory including:

- A satisfiable set $\mathcal{F}$ of universal formulas has a Herbrand model and so has a model which is finite or countable.
- If the set $\mathcal{F}$ of universal formulas is unsatisfiable, then Theorem 5.67 directly exhibits a finite set of unsatisfiable ground instances. Thus Theorem 5.67 gives a method for effectively producing either a Herbrand model for $\mathcal{F}$ or a particular finite counter-example to the existence of any model of $\mathcal{F}$.
- Herbrand's theorem implies the completeness of the resolution method; the resolution method is based on the following idea: the formula $F = \exists x G(x)$, where G is quantifier-free, is a consequence of the set of universal formulas $\mathcal{F}$ if and only if $\mathcal{F} \cup \{\neg F\}$ is unsatisfiable. $\mathcal{F} \cup \{\neg F\}$ is a set of universal formulas that can be proved to be unsatisfiable by exhibiting a finite set of unsatisfiable ground instances. It can be shown that exhibiting the unsatisfiable ground instances also gives valuations $v(x) = t$ such that $\mathcal{F} \vdash G[x := t]$, which are called answer substitutions.
- Herbrand's theorem can be used to prove Theorem 5.52.

EXERCISE 5.19 Find all Herbrand models of

$$\mathcal{F} = \{edge(a,b)\ ,\ edge(b,c)\ ,\ \forall x \forall y \big(edge(x,y) \supset path(x,y)\big)\ ,$$
$$\forall x \forall y \big((edge(x,z) \wedge path(z,y)) \supset path(x,y)\big)\}\ ,$$

where the language $\mathcal{L}$ consists of the constants a, b, c and the binary relational symbols *edge* and *path*. With the PROLOG notations, $\mathcal{F}$ would be denoted by:

$$\begin{aligned}
r_1 &: && \Longrightarrow edge(a,b)\\
r_2 &: && \Longrightarrow edge(b,c)\\
r_3 &: && edge(X,Y) \Longrightarrow path(X,Y)\\
r_4 &: && edge(X,Z), path(Z,Y) \Longrightarrow path(X,Y)
\end{aligned}$$

where universal quantifications are omitted and the comma denotes $\wedge$. ♢

5.4.4 Skolemization

We have seen in Example 5.65 that Herbrand's theorem does not hold for non-universal formulas. This can be remedied by constructing, for each formula F, a universal formula F' which is equisatisfiable with F: i.e. F is satisfiable if and only if F' is satisfiable. (Note: F' will not be equivalent to F, see Exercise 5.21.) Each formula

$$F = \forall x_1 \cdots \forall x_n \exists y_1 \cdots \exists y_p G(x_1, \ldots, x_n, y_1, \ldots, y_p)$$

will be replaced by

$$F' = \forall x_1 \cdots \forall x_n G(x_1, \ldots, x_n, f_1(x_1, \ldots, x_n), \ldots, f_p(x_1, \ldots, x_n)) \,,$$

where $f_1, \ldots, f_p$ are new function symbols, called *Skolem functions*. F' is called a *Skolemization* of F.

Theorem 5.68 *Let F be a closed formula in a language $\mathcal{L}$; there exists a universal formula F' in a language $\mathcal{L}' = \mathcal{L} \cup \{f_1, \ldots, f_p\}$, where $f_1, \ldots, f_p$ are new function symbols, such that F is satisfiable if and only if F' is satisfiable.*

Proof. By Theorem 5.48 we may assume that F is in prenex form; assume

$$F = \forall x_1 ... \forall x_{n_1} \exists y_1 \forall x_{n_1+1} ... \forall x_{n_2} \exists y_2 \cdots \forall x_{n_{p-1}+1} ... \forall x_{n_p} \exists y_p \forall x_{n_p+1} ... \forall x_n G \,,$$

where G is a formula without quantifiers. We add p new function symbols $f_1, \ldots, f_p$ to $\mathcal{L}$; for $i = 1, \ldots, p$, each f_i is of arity n_i and depends on the x_js such that $\forall x_j$ occurs before $\exists y_i$ in F. F' is the formula

$$F' = \forall x_1 \cdots \forall x_n G[y_1 := f_1(x_1, \ldots, x_{n_1})] \cdots [y_p := f_p(x_1, \ldots, x_{n_p})].$$

F is satisfiable if and only if F' is satisfiable. By induction on p it suffices to prove Lemma 5.69. □

Lemma 5.69 *$F = \forall x_1 \cdots \forall x_n \exists y G$ is satisfiable if and only if $F' = \forall x_1 \cdots \forall x_n G[y := f(x_1, \ldots, x_n)]$ is satisfiable, where f is a new function symbol.*

Exercise 5.20 Prove Lemma 5.69. ♢

Remark 5.70 F' will not be equivalent to F. See Exercise 5.21.

Exercise 5.21 Find Skolemizations of $F = (\forall x R(x)) \vee (\exists y R'(y))$. ♢

Exercise 5.22 Find Skolemizations of $F = (\forall x \exists y R(x, y)) \vee \neg(\exists x \forall y R'(x, y))$. ♢

When we are interested in the existence of a model for a formula, Skolemization enables us to suppress all existential quantifiers. By Remark 5.66, models of $F(x_1, \ldots, x_n)$ and models of $\forall x_1 \cdots \forall x_n F(x_1, \ldots, x_n)$ coincide. We can thus assume that all variables are universally quantified and omit the universal quantifiers in the denotation of the formula: this is the usual notation for PROLOG.

5.4.5 Horn clauses

Horn clauses are very useful examples of universal formulas. PROLOG and most logic programming languages are based on Horn clauses.

Definition 5.71
(i) *Literals are atomic formulas or their negations, i.e. formulas of the form* $L = R(t_1, \ldots, t_n)$ *(positive literal), or of the form* $L = \neg R(t_1, \ldots, t_n)$ *(negative literal).*
(ii) *A Horn clause is a universal formula of the form* $\forall x_1 \cdots \forall x_p(L_1 \vee \cdots \vee L_n)$, *where the* L_i*s are literals, and at most one of them is positive.*
(iii) *A program clause or definite clause is a Horn clause with exactly one positive literal.*

Thus, a Horn clause can be of one of the following three forms:
(i) $\forall x_1 \cdots \forall x_p A$ (positive clause or *fact*).
(ii) $\forall x_1 \cdots \forall x_p(\neg A_1 \vee \cdots \vee \neg A_n \vee A)$, where A and the A_is are atomic formulas.
(iii) $\forall x_1 \cdots \forall x_p(\neg A_1 \vee \cdots \vee \neg A_n)$, where the A_is are atomic formulas (negative clause or *goal*).

Example 5.72 A PROLOG program consists of Horn clauses of the form (i) or (ii), which are usually written in the form (omitting the universal quantifiers and substituting a comma for $\wedge$):

$$\text{(i)} \qquad \Longrightarrow A\ ,$$
$$\text{(ii)} \qquad A_1, \ldots, A_n \Longrightarrow A\ .$$

Example 5.73 The set $\mathcal{F}$ of formulas of Exercise 5.19 is a set of program clauses. $\mathcal{F}$ can be written as $\mathcal{F} = \{edge(a,b)\ ,\ edge(b,c)\ ,\ \forall x \forall y\big(\neg edge(x,y) \vee path(x,y)\big)\ ,\ \forall x \forall y\big(\neg edge(x,z) \vee \neg path(z,y) \vee path(x,y)\big)\}$, With the PROLOG notations, $\mathcal{F}$ is denoted by:

$$r_1 : \qquad \Longrightarrow edge(a,b)\ ,$$
$$r_2 : \qquad \Longrightarrow edge(b,c)\ ,$$
$$r_3 : \qquad edge(X,Y) \Longrightarrow path(X,Y)\ ,$$
$$r_4 : \qquad edge(X,Z), path(Z,Y) \Longrightarrow path(X,Y)\ .$$

Theorem 5.74 *A set P of program clauses has a least Herbrand model $M = \langle U_H, I_M \rangle$ (i.e. a Herbrand model such that I_M is contained in any other Herbrand model I_H).*

Proof. Let $P = \{C_i \,/\, i \in J\}$, where each C_i is either of the form (i) or (ii). We will prove that the intersection of all Herbrand models of P is itself a Herbrand model of P. Let M be defined by

$$I_M = \bigcap \{I_H \;\; / \;\; \emptyset \underset{I_H}{\models} C_i \;, \;\; \text{for all } C_i \in P\} \,,$$

i.e. I_M is the intersection of all the Herbrand models of P; then $\emptyset \underset{M}{\models} C_i$ for all $C_i \in P$. We verify that all the clauses C_i of P are valid in M.

(i) If C_i is of the form $\forall x_1 \cdots \forall x_p A$, then all ground instances of A are true in all Herbrand models, hence they belong to all I_Hs, and also to I_M, and thus they are true in M.
(ii) If C_i is of the form $\forall x_1 \cdots \forall x_p(\neg A_1 \vee \cdots \vee \neg A_n \vee A)$, let $C_i' = (\neg A_1 \vee \cdots \vee \neg A_n \vee A)$, let v be a valuation $v\colon \{x_1, \ldots, x_p\} \longrightarrow U_H$ and let, for $B \in \{A, A_1, \ldots, A_n\}$, $v^*(B) = B[x_1 := v(x_1)] \cdots [x_p := v(x_p)]$ be the ground atom obtained by substituting $v(x_i)$ for x_i in B; then

- either there exists an A_j such that $v^*(A_j) \notin I_M$, and then $\bar{v}(\neg A_j) = 1$ and $\bar{v}(C_i') = 1$.
- or for any A_j, $v^*(A_j) \in I_M$, and then, for any I_H defining a Herbrand model H of P: $v^*(A_j) \in I_H$, and because H is a Herbrand model of P, we also have that $v^*(A) \in I_H$, hence $v^*(A) \in I_M$, and thus $\bar{v}(A) = 1$ and $\bar{v}(C_i') = 1$.

Hence, for any valuation $v\colon \{x_1, \ldots, x_p\} \longrightarrow U_H$, $\bar{v}(C_i') = 1$, and thus $\emptyset \underset{M}{\models} C_i$. □

Example 5.75 The least Herbrand model of the program P of Example 5.73 is defined by

$$I_M = \{edge(a,b) \,,\, edge(b,c) \,,\, path(a,b) \,,\, path(b,c), path(a,c)\} \,.$$

Remark 5.76
1. Any set P of program clauses also has a greatest Herbrand model M', which is defined by the whole Herbrand basis B_H. See also Exercise 5.28 (2).
2. A set $\mathcal{F}$ of universal formulas which are not Horn clauses may have several minimal incomparable Herbrand models and as a result no least Herbrand model. For example, let $\mathcal{F} = \{\forall x\big(p(x) \vee q(x)\big)\}$, where $\mathcal{L} = \{a, p, q\}$, a a constant symbol and p, q unary relational symbols. Then $U_H = \{a\}$, $B_H = \{p(a) \,,\, q(a)\}$, and

$\mathcal{F}$ has three Herbrand models, respectively defined by the subsets $I_1 = \{p(a)\}$, $I_2 = \{q(a)\}$ and $I_3 = B_H$; both models I_1 and I_2 are minimal (none is included in the other one) and their intersection is the Herbrand interpretation defined by $I_H = \emptyset$, which is *not* a model of $\mathcal{F}$.

EXERCISE 5.23 A set of formulas which are not universal may have minimal Herbrand models. Find the minimal Herbrand models of the set of formulas $\mathcal{F}$ of Exercise 5.18. ♢

EXERCISE 5.24 Find the least Herbrand model of the set of program clauses $P = \{\forall x \forall y \big(\neg edge(x,y) \vee path(x,y)\big)\ ,\ \forall x \forall y \big(\neg edge(x,z) \vee \neg path(z,y) \vee path(x,y)\big)\}$. ♢

EXERCISE 5.25 Find the least Herbrand model of the set of program clauses $P = \{i(a)\ ,\ \forall x \big(i(s(x)) \vee \neg i(x)\big)\}$, where $\mathcal{L} = \{a, s, i\}$, a a constant symbol, s a unary function symbol and i is a unary relational symbol. ♢

EXERCISE 5.26 Let P be a set of program clauses; P is a set of formulas, hence (see Definition 5.53 and Example 5.54) $Th(P) = \{F \ / \ P \vdash F\}$ is a theory.

Show that $\{A \in B_H \,/\, A \in Th(P)\}$ defines the least Herbrand model of the set of program clauses P. ♢

EXERCISE 5.27 Does any set $\mathcal{F}$ of Horn clauses have a least Herbrand model? ♢

There is a constructive proof of the existence of the least Herbrand model of a set of program clauses, which is most useful in logic programming, and which is given in the following exercise.

EXERCISE 5.28 Recall that a complete lattice is a lattice where every subset has a least upper bound and a greatest lower bound. If f is a monotone mapping from a complete lattice to itself, then we can prove as in Theorem 2.39 that f has a least fixed point defined by $e = \inf\{x \in E \,/\, f(x) \leq x\}$.

Let P be a set of program clauses. Let $\mathcal{P}(B_H)$ be the set of subsets of the Herbrand basis B_H. Then $\mathcal{P}(B_H)$ when equipped with inclusion is a complete lattice. The least element of $\mathcal{P}(B_H)$ is $\emptyset$, its greatest element is B_H, $\sup_i K_i = \cup_i K_i$, $\inf_i K_i = \cap_i K_i$.

The immediate consequence operator $T_P : \mathcal{P}(B_H) \longrightarrow \mathcal{P}(B_H)$ is defined by: $T_P(I) = \{A \in B_H |$ there exists $r = (B_1, \ldots, B_n \Longrightarrow B) \in P$, there exists a valuation $s : X \longrightarrow U_H$ such that, for $i = 1, \ldots, n$, $s^*(B_i) = A_i \in I$, $s^*(B) = A\}$. ($s^*(B) = B[x_1 := s(x_1)] \cdots [x_p := s(x_p)]$ (resp. $s^*(B_i) = B_i[x_1 := s(x_1)] \cdots [x_p := s(x_p)]$) denotes the ground atom obtained by substituting the term $s(x_k)$ for x_k in B (resp. B_i), for any variable $x_k \in X$.)

In other words, $T_P(I)$ is the set of atomic formulas A, such that $A_1, \ldots, A_n \Longrightarrow A$ is a ground instance of a clause r of P and, moreover, $A_1, \ldots, A_n$ are in I.

1. Show that T_P is monotone.
2. Let I be a Herbrand interpretation; show that I is a model of P if and only if $T_P(I) \subset I$.
3. Show that the least fixpoint of T_P is the least Herbrand model of P.
4. Show that T_P is continuous (i.e. for any increasing sequence $K_1 \subset K_2 \subset \cdots \subset K_n \subset \cdots$ of $\mathcal{P}(B_H)$, $\sup_i T_P(K_i) = T_P(\sup_i K_i)$).
5. Show that the least Herbrand model of P is defined by the basis

$$I_M = \sup(\{T_P^n(\emptyset) \,/\, n \in \mathbb{N}\}).$$

6. Show that, for any $n \in \mathbb{N}$, $T_P^n(B_H)$ is a Herbrand model of P.

Let $K = \inf(\{T_P^n(B_H) \,/\, n \in \mathbb{N}\})$.

7. Is K a model of P?

8. Is K a fixpoint of P? What can you say about the greatest fixpoint of P? $\diamondsuit$