15.3.4 Applications of Predicate Calculus

The predicate calculus is may be employed to formally state the system requirements of a proposed system. It may be used to conduct formal proof to verify the presence or absence of certain properties in a specification. It may also be employed to define piecewise defined functions such as $f(x, y)$ where $f(x, y)$ is defined by

$$\begin{array}{ll}
f(x,y) = x^2 - y^2 & \text{where}\, x \leq 0 \wedge y < 0; \\
f(x,y) = x^2 + y^2 & \text{where}\, x > 0 \wedge y < 0; \\
f(x,y) = x + y & \text{where}\, x \geq 0 \wedge y = 0; \\
f(x,y) = x - y & \text{where}\, x < 0 \wedge y = 0; \\
f(x,y) = x + y & \text{where}x \leq 0 \wedge y > 0; \\
f(x,y) = x^2 + y^2 & \text{where}\, x > 0 \wedge y > 0
\end{array}$$

The predicate calculus may be employed for program verification, and to show that a code fragment satisfies its specification. The statement that a program F is correct with respect to its precondition P and postcondition Q is written as $P\{F\}$ Q. The objective of program verification is to show that if the precondition is true before execution of the code fragment, then this implies that the postcondition is true after execution of the code fragment.

A program fragment a is *partially correct* for precondition P and postcondition Q if and only if whenever a is executed in any state in which P is satisfied and execution terminates, then the resulting state satisfies Q. Partial correctness is denoted by $P\{F\}Q$, and Hoare's Axiomatic Semantics is based on partial correctness. It requires proof that the postcondition is satisfied if the program terminates.

A program fragment a is *totally correct* for precondition P and postcondition Q, if and only if whenever a is executed in any state in which P is satisfied then the execution terminates and the resulting state satisfies Q. It is denoted by $\{P\}F\{Q\}$, and Dijkstra's calculus of weakest preconditions is based on total correctness [2, 4]. It is required to prove that if the precondition is satisfied then the program terminates and the postcondition is satisfied

15.3.5 Semantic Tableaux in Predicate Calculus

We discussed the use of semantic tableaux for determining the validity of arguments in propositional logic earlier in this chapter, and its approach is to negate the conclusion of an argument and to show that this results in inconsistency with the premises of the argument.

The use of semantic tableaux is similar with predicate logic, except that there are some additional rules to consider. As before, if all branches of a semantic tableau are closed, then the premises and the negation of the conclusion are mutually inconsistent, and all branches in the tableau are closed. From this, we deduce that the conclusion must be true.

The rules of semantic tableaux for propositional logic were presented in Table 15.12, and the additional rules specific to predicate logic are detailed in Table 15.14.

Example 15.9 (**Semantic Tableaux**) Show that the syllogism 'All Greeks are mortal; Socrates is a Greek; therefore Socrates is mortal' is a valid argument in predicate calculus.

Table 15.14 Extra rules of semantic tableaux (for predicate calculus)

Rule No.	Definition	Description
1.	$(\forall x)\ A(x)$ $A(t)$ where t is a term	Universal instantiation
2.	$(\exists x)\ A(x)$ $A(t)$ where t is a term that has not been used in the derivation so far	Rule of Existential instantiation. The term "t" is often a constant "a"
3.	$\neg(\forall x)\ A(x)$ $(\exists x)\ \neg A(x)$	
4.	$\neg(\exists x)\ A(x)$ $(\forall x)\neg A(x)$	

Solution
We expressed this argument previously as $(\forall x)(G(x) \rightarrow M(x))$; $G(s)$; $M(s)$. Therefore, we negate the conclusion (i.e. $\neg M(s)$), and try to construct a closed tableau.

$$
\begin{array}{ll}
(\forall x)(G(x) \rightarrow M(x)) & \\
G(s) & \\
\neg M(s). & \\
G(s) \rightarrow M(s) & \text{Universal Instantiation} \\
\quad \wedge & \\
\neg G(s) \quad M(s) & \\
\text{-----} \quad \text{--------} & \\
\textit{closed} \quad \textit{closed} &
\end{array}
$$

Therefore, as the tableau is closed we deduce that the negation of the conclusion is inconsistent with the premises, and that therefore the conclusion follows from the premises.

Example 15.10 (**Semantic Tableaux**) Determine whether the following argument is valid.

All lecturers are motivated
Anyone who is motivated and clever will teach well
Joanne is a clever lecturer
Therefore, Joanne will teach well.

Solution
We encode the argument as follows

$L(x)$ stands for 'x is a lecturer'
$M(x)$ stands for 'x is motivated'
$C(x)$ stands for 'x is clever'
$W(x)$ stands for 'x will teach well'

We therefore wish to show that

$$
\begin{aligned}
&(\forall x)(L(x) \rightarrow M(x)) \wedge (\forall x)((M(x) \wedge C(x)) \\
&\qquad \rightarrow W(x)) \wedge L(joanne) \wedge C(joanne) \models W(joanne)
\end{aligned}
$$

Therefore, we negate the conclusion (i.e. $\neg W(joanne)$) and try to construct a closed tableau.

1. $(\forall x)(L(x) \rightarrow M(x))$
2. $(\forall x)((M(x) \wedge C(x)) \rightarrow W(x))$
3. $L(joanne)$
4. $C(joanne)$
5. $\neg W(joanne)$
6. $L(joanne) \rightarrow M(joanne)$ Universal Instantiation (line 1)
7. $(M(joanne) \wedge C(joanne)) \rightarrow W(joanne)$ Universal Instantiation (line 2)

/ \

8. $\neg L(joanne)$ $M(joanne)$ From line 6

Closed

/ \

9. $\neg (M(joanne) \wedge C(joanne))$ $W(joanne)$ From line 7

Closed

/ \

10. $\neg M(joanne)$ $\neg C(joanne)$

-------------- -------------

Closed Closed

Therefore, since the tableau is closed we deduce that the argument is valid.

15.4 Summary

This chapter considered propositional and predicate calculus. Propositional logic is the study of propositions, and a proposition is a statement that is either true or false. A formula in propositional calculus may contain several variables, and the truth or falsity of the individual variables, and the meanings of the logical connectives determines the truth or falsity of the logical formula.

A rich set of connectives is employed in propositional calculus to combine propositions and to build up the well-formed formulae of the calculus. This includes the conjunction of two propositions ($A \wedge B$), the disjunction of two propositions ($A \vee B$), and the implication of two propositions ($A \Rightarrow B$). These connectives allow compound propositions to be formed, and the truth of the compound propositions is determined from the truth values of the constituent propositions and the rules associated with the logical connectives. The meaning of the logical connectives is given by truth tables.

Propositional calculus is both complete and consistent with all true propositions deducible in the calculus, and there is no formula A such that both A and $\neg A$ are deducible in the calculus.

An argument in propositional logic consists of a sequence of formulae that are the premises of the argument and a further formula that is the conclusion of the argument. One elementary way to see if the argument is valid is to produce a truth table to determine if the conclusion is true whenever all of the premises are true. Other ways are to use semantic tableaux or natural deduction.

Predicates are statements involving variables and these statements become propositions once the variables are assigned values. Predicate calculus allows expressions such as all members of the domain have a particular property to be expressed formally: e.g., $(\forall x)\mathrm{P}x$, or that there is at least one member that has a particular property: e.g., $(\exists x)\mathrm{P}x$.

Predicate calculus may be employed to specify the requirements for a proposed system and to give the definition of a piecewise defined function. Semantic tableaux may be used for determining the validity of arguments in propositional or predicate logic, and its approach is to negate the conclusion of an argument and to show that this results in inconsistency with the premises of the argument.