Chapter 3
Propositional Logic: Deductive Systems

The concept of deducing theorems from a set of axioms and rules of inference is very old and is familiar to every high-school student who has studied Euclidean geometry. Modern mathematics is expressed in a style of reasoning that is not far removed from the reasoning used by Greek mathematicians. This style can be characterized as 'formalized informal reasoning', meaning that while the proofs are expressed in natural language rather than in a formal system, there are conventions among mathematicians as to the forms of reasoning that are allowed. The deductive systems studied in this chapter were developed in an attempt to formalize mathematical reasoning.

We present two deductive systems for propositional logic. The second one $\mathscr{H}$ will be familiar because it is a formalization of step-by-step proofs in mathematics: It contains a set of three axioms and one rule of inference; proofs are constructed as a sequence of formulas, each of which is either an axiom (or a formula that has been previously proved) or a derivation of a formula from previous formulas in the sequence using the rule of inference. The system $\mathscr{G}$ will be less familiar because it has one axiom and many rules of inference, but we present it first because it is almost trivial to prove the soundness and completeness of $\mathscr{G}$ from its relationship with semantic tableaux. The proof of the soundness and completeness of $\mathscr{H}$ is then relatively easy to show by using $\mathscr{G}$. The chapter concludes with three short sections: the definition of an important property called *consistency*, a generalization to infinite sets of formulas, and a survey of other deductive systems for propositional logic.

3.1 Why Deductive Proofs?

Let $U = \{A_1, \ldots, A_n\}$. Theorem 2.50 showed that $U \models A$ if and only if $\models A_1 \wedge \cdots \wedge A_n \rightarrow A$. Therefore, if U is a set of axioms, we can use the completeness of the method of semantic tableaux to determine if A follows from U (see Sect. 2.5.4 for precise definitions). Why would we want to go through the trouble of searching for a mathematical proof when we can easily compute if a formula is valid?

M. Ben-Ari, *Mathematical Logic for Computer Science*,
DOI 10.1007/978-1-4471-4129-7_3, © Springer-Verlag London 2012

There are several problems with a purely semantical approach:

- The set of axioms may be infinite. For example, the axiom of induction in arithmetic is really an infinite set of axioms, one for each property to be proved. For semantic tableaux in propositional logic, the only formulas that appear in the tableaux are subformulas of the formula being checked or their negations, and there are only a finite number of such formulas.
- Very few logics have decision procedures like propositional logic.
- A decision procedure may not give insight into the relationship between the axioms and the theorem. For example, in proofs of theorems about prime numbers, we would want to know exactly where primality is used (Velleman, 2006, Sect. 3.7). This understanding can also help us propose other formulas that might be theorems.
- A decision procedure produces a 'yes/no' answer, so it is difficult to recognize intermediate results (lemmas). Clearly, the millions of mathematical theorems in existence could not have been inferred directly from axioms.

Definition 3.1 A *deductive system* is a set of formulas called *axioms* and a set of *rules of inference*. A *proof* in a deductive system is a sequence of formulas $S = \{A_1, \ldots, A_n\}$ such that each formula A_i is either an axiom or it can be inferred from previous formulas of the sequence $A_{j_1}, \ldots, A_{j_k}$, where $j_1 < \cdots < j_k < i$, using a rule of inference. For A_n, the last formula in the sequence, we say that A_n is a *theorem*, the sequence S is a *proof* of A_n, and A_n is *provable*, denoted $\vdash A_n$. If $\vdash A$, then A may be used like an axiom in a subsequent proof. ∎

The deductive approach can overcome the problems described above:

- There may be an infinite number of axioms, but only a finite number will appear in any proof.
- Although a proof is not a decision procedure, it can be mechanically *checked*; that is, given a sequence of formulas, an syntax-based algorithm can easily check whether the sequence is a proof as defined above.
- The proof of a formula clearly shows which axioms, theorems and rules are used and for what purposes.
- Once a theorem has been proved, it can be used in proofs like an axiom.

Deductive proofs are not generated by decision procedures because the formulas that appear in a proof are not limited to subformulas of the theorem and because there is no algorithm telling us how to generate the next formula in the sequence forming a proof. Nevertheless, algorithms and heuristics can be used to build software systems called *automatic theorem provers* which search for proofs. In Chap. 4, we will study a deductive system that has been successfully used in automatic theorem provers. Another promising approach is to use a *proof assistant* which performs administrative tasks such as proof checking, bookkeeping and cataloging previously proved theorems, but a person guides the search by suggesting lemmas that are likely to lead to a proof.

α	α_1	α_2	β	β_1	β_2
$\neg\neg A$	A				
$\neg(A_1 \wedge A_2)$	$\neg A_1$	$\neg A_2$	$B_1 \wedge B_2$	B_1	B_2
$A_1 \vee A_2$	A_1	A_2	$\neg(B_1 \vee B_2)$	$\neg B_1$	$\neg B_2$
$A_1 \rightarrow A_2$	$\neg A_1$	A_2	$\neg(B_1 \rightarrow B_2)$	B_1	$\neg B_2$
$A_1 \uparrow A_2$	$\neg A_1$	$\neg A_2$	$\neg(B_1 \uparrow B_2)$	B_1	B_2
$\neg(A_1 \downarrow A_2)$	A_1	A_2	$B_1 \downarrow B_2$	$\neg B_1$	$\neg B_2$
$\neg(A_1 \leftrightarrow A_2)$	$\neg(A_1 \rightarrow A_2)$	$\neg(A_2 \rightarrow A_1)$	$B_1 \leftrightarrow B_2$	$B_1 \rightarrow B_2$	$B_2 \rightarrow B_1$
$A_1 \oplus A_2$	$\neg(A_1 \rightarrow A_2)$	$\neg(A_2 \rightarrow A_1)$	$\neg(B_1 \oplus B_2)$	$B_1 \rightarrow B_2$	$B_2 \rightarrow B_1$

Fig. 3.1 Classification of α- and β-formulas

3.2 Gentzen System $\mathscr{G}$

The first deductive system that we study is based on a system proposed by Gerhard Gentzen in the 1930s. The system itself will seem unfamiliar because it has one type of axiom and many rules of inference, unlike familiar mathematical theories which have multiple axioms and only a few rules of inference. Furthermore, deductions in the system can be naturally represented as trees rather in the linear format characteristic of mathematical proofs. However, it is this property that makes it easy to relate Gentzen systems to semantic tableaux.

Definition 3.2 (Gentzen system $\mathscr{G}$) An *axiom* of $\mathscr{G}$ is a set of literals U containing a complementary pair. *Rule of inference* are used to infer a set of formulas U from one or two other sets of formulas U_1 and U_2; there are two types of rules, defined with reference to Fig. 3.1:

- Let $\{\alpha_1, \alpha_2\} \subseteq U_1$ and let $U_1' = U_1 - \{\alpha_1, \alpha_2\}$. Then $U = U_1' \cup \{\alpha\}$ can be inferred.
- Let $\{\beta_1\} \subseteq U_1$, $\{\beta_2\} \subseteq U_2$ and let $U_1' = U_1 - \{\beta_1\}$, $U_2' = U_2 - \{\beta_2\}$. Then $U = U_1' \cup U_2' \cup \{\beta\}$ can be inferred.

The set or sets of formulas U_1, U_2 are the *premises* and set of formulas U that is inferred is the *conclusion*. A set of formulas U that is an axiom or a conclusion is said to be *proved*, denoted $\vdash U$. The following notation is used for rules of inference:

$$\frac{\vdash U_1' \cup \{\alpha_1, \alpha_2\}}{\vdash U_1' \cup \{\alpha\}} \qquad \frac{\vdash U_1' \cup \{\beta_1\} \qquad\qquad \vdash U_2' \cup \{\beta_2\}}{\vdash U_1' \cup U_2' \cup \{\beta\}}.$$

Braces can be omitted with the understanding that a sequence of formulas is to be interpreted as a set (with no duplicates). ■

Example 3.3 The following set of formulas is an axiom because it contains the complementary pair $\{r, \neg r\}$:

$$\vdash p \wedge q, q, r, \neg r, q \vee \neg r.$$