2

Preliminaries

In this chapter we briefly review the basic notions and facts from logic and complexity theory whose knowledge is assumed throughout the book. We shall always sketch important arguments, both from logic and from complexity theory, and so a determined reader can start with only a rough familiarity with the notions surveyed in the next two sections and pick the necessary material along the way.

For those readers who prefer to consult relevant textbooks we recommend the following books: The best introduction to logic are parts of Shoenfield (1967); for elements of structural complexity theory I recommend Balcalzár, Diáz, and Gabbarró (1988, 1990); for NP-completeness Garey and Johnson (1979); and for a Boolean complexity theory survey of lower bounds Boppana and Sipser (1990) or the comprehensive monograph Wegener (1987). A more advanced (but self-contained) text on logic of first order arithmetic theories is Hájek and Pudlák (1993).

2.1. Logic

We shall deal with first order and second order theories of arithmetic. The second order theories are, in fact, just two-sorted first order theories: One sort are numbers; the other are finite sets. This phrase means that the underlying logic is always the first order predicate calculus; in particular, no set-theoretic assumptions are a part of the underlying logic.

From basic theorems we shall use Gödel *completeness* and *incompleteness* theorems, Tarski's *undefinability of truth*, and, in arithmetic, constructions of *partial truth definitions*.

A prominent theory is *Peano arithmetic* (PA), in the *language of arithmetic* $L_{\mathrm{PA}} = \{0, 1, +, \cdot, <, =\}$ axiomatized by *Robinson's arithmetic* Q

1. $a + 1 \neq 0$

2. $a + 1 = b + 1 \to a = b$
3. $a + 0 = a$
4. $a + (b + 1) = (a + b) + 1$
5. $a \cdot 0 = 0$
6. $a \cdot (b + 1) = (a \cdot b) + a$
7. $a \neq 0 \to \exists x, x + 1 = a$

see Tarski, Mostowski, and Robinson (1953), and by the *induction scheme* IND

$$\big(\phi(0, \bar{a}) \wedge \forall x(\phi(x, \bar{a}) \to \phi(x + 1, \bar{a}))\big) \to \forall x \phi(x, \bar{a})$$

for every formula $\phi(x, \bar{a})$ in the language L_{PA}.

We shall use the letters $x, y, z, \ldots$ mostly for bounded variables; the letters $a, b, c, \ldots$ will be reserved for free variables (also called parameters). Free variables in axioms are assumed to be universally quantified; for example, the first axiom given is equivalent to the formula $\forall x, x + 1 \neq 0$.

There are other schemes that can equivalently replace the induction scheme, for example, *the least number principle* LNP scheme

$$\phi(b, \bar{a}) \to \exists x \forall y\big(\phi(x, \bar{a}) \wedge (y < x \to \neg\phi(y, \bar{a}))\big).$$

The *standard model* N of PA is the set of natural numbers with the symbols of L_{PA} interpreted with the usual meaning. A crucial fact about PA is that there are *nonstandard* models (models not isomorphic with N) of PA and indeed of the theory of N, $\mathrm{Th}(N)$. Natural numbers N are isomorphic to a unique initial substructure of any nonstandard model M and we shall usually simply assume that $N \subset M$.

A *cut* in a nonstandard model M is any nonempty $I \subseteq M$ satisfying

1. $a < b \wedge b \in I \to a \in I$, all $a, b \in M$
2. $a \in I \to a + 1 \in I$, all $a \in M$.

For example, N is a cut in every nonstandard model. Cuts in nonstandard models of PA closed under both addition and multiplication have special prominence as they are particular models of *bounded arithmetic* $I\Delta_0$: They satisfy induction for all *bounded arithmetic formulas* Δ_0, which are formulas in the language L_{PA} with all quantifiers bounded (Section 3.2 is devoted to bounded formulas).

Nonstandard models of PA and even of its proper subtheories are difficult to construct; it is a theorem of Tennenbaum (1959) that there are no countable recursive nonstandard models of PA (and, indeed, of a weak subtheory IE_1 with the induction just for bounded existential formulas, cf. Paris (1984). In particular, these results show that every nonstandard countable model of IE_1 has a nonstandard cut that is a model of whole PA; hence, in a sense, the model theory of bounded arithmetic is as complex as that of PA. Consult Hájek and Pudlák (1993), Kaye (1991), or Smorynski (1984) for the model theory of PA.

From proof theory we shall use theorems of Gentzen and Herbrand in various versions. The reader is advised to refer to Takeuti (1975) for Gentzen's sequent calculus.

We close this section with some remarks on notation. Logical connectives we shall use are the standard $\neg$, $\vee$, $\wedge$, $\rightarrow$, and $\equiv$ with the usual meaning – negation, disjunction, conjunction, implication, and equivalence – and the constants 1, 0 for *truth* and *falsity*.

The symbols $\subset$ and $\subseteq$ are used in the sense of *proper inclusion* and *inclusion*.

The symbols $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$ and $f(n) = \Theta(g(n))$ denote that eventually $f(n) \leq cg(n)$, $f(n) \geq cg(n)$, and $c_1 g(n) \leq f(n) \leq c_2 g(n)$ where c, c_1, and c_2 are positive constants, and $f(n) = o(g(n))$ means that $f(n)/g(n) \rightarrow 0$.

2.2. Complexity theory

I assume that the reader is acquainted with such notions as *Turing machine, oracle Turing machine*, and *time* and *space* complexity measures. We adopt the multitape version of Turing machines with a read-only input tape and with a finite but arbitrarily large alphabet.

The basic relations between classes of languages Time(f) and Space(f) recognized by a deterministic Turing machine in time (respectively space) bounded by $f(n)$, n the length of the input, and their nondeterministic versions NTime(f) and NSpace(f) are

1. $\text{Time}(f(n)) \subseteq \text{NTime}(f(n)) \subseteq \text{Space}(f(n))$
2. $\text{Space}(f(n)) \subseteq \bigcup_c \text{Time}(c^{f(n)})$
3. (Hartmanis and Stearns 1965) $\text{Time}(f(n)) = \text{Time}(c \cdot f(n))$ and $\text{Space}(f(n)) = \text{Space}(c \cdot f(n))$ whenever $n = o(f(n))$ and $n \leq f(n)$
4. (Hartmanis and Stearns 1965, Hartmanis, Lewis, and Stearns 1965)
$$\text{Space}(f) \subset \text{Space}(g)$$
and
$$\text{Time}(f) \subset \text{Time}(g \log(g))$$
whenever $f = o(g(n))$.
5. (Savitch 1970)
$$\text{NSpace}(f) \subseteq \text{Space}(f^2)$$
whenever $f(n)$ is itself computable in space $f(n)$
6. (Szelepcsényi 1987, Immerman 1988)
$$\text{NSpace}(f) = \text{coNSpace}(f)$$
for $f(n) \geq \log(n)$ and f itself computable in nondeterministic space $f(n)$.

7. (Hopcroft, Paul, and Valiant 1975) For $n \leq f(n)$

$$\mathrm{Time}(f(n)) \subseteq \mathrm{Space}\left(\frac{f(n)}{\log f(n)}\right)$$

Particular bounds to time or space define the usual complexity classes

$$\begin{aligned}
\mathrm{LinTime} &= \bigcup_c \mathrm{Time}(cn) \\
\mathrm{P} &= \bigcup_c \mathrm{Time}(n^c) \\
\mathrm{NP} &= \bigcup_c \mathrm{NTime}(n^c) \\
\mathrm{L} &= \mathrm{Space}(\log(n)) \\
\mathrm{PSpace} &= \bigcup_c \mathrm{Space}(n^c) \\
\mathrm{LinSpace} &= \mathrm{Space}(n) \\
\mathrm{E} &= \bigcup_c \mathrm{Time}(c^n) \\
\mathrm{EXP} &= \bigcup_c \mathrm{Time}(2^{n^c})
\end{aligned}$$

Oracle computations allow one to define hierarchies of languages, the most important of which are the *linear time hierarchy* LinH of Wrathall (1978)

$$\Sigma_0^{\mathrm{lin}} = \mathrm{LinTime} \text{ and } \Sigma_{i+1}^{\mathrm{lin}} = \mathrm{NLinTime}^{\Sigma_i^{\mathrm{lin}}}$$

and the *polynomial time hierarchy* PH of Stockmeyer (1977)

$$\Sigma_0^p = \mathrm{P} \quad \text{and} \quad \Sigma_{i+1}^p = \mathrm{NP}^{\Sigma_i^p}$$

The class of complements of languages from class X is denoted co$X$, and special classes of this form coΣ_i^{lin} and coΣ_i^p are denoted Π_i^{lin} and Π_i^p, respectively.

The class $\Box_{i+1}^p$ is the class of functions computable by a polynomial-time machine with access to an oracle from the class Σ_i^p.

Some important facts about these classes include the following: $\Sigma_i^{\mathrm{lin}} \subset \Sigma_i^p$ (and generally more resource in the "same" computational class properly increases the class; see Žák 1983 for a general diagonalization technique), and LinH contains L and is, in fact, equal to the class of *rudimentary predicates* as defined by Smullyan (1961) (cf. Wrathall 1978). It is also known that LinH also equals the class of predicates definable by Δ_0-formulas; we shall prove that in Section 3.2. Also note that either LinH $\neq$ PH or LinH does not collapse (i.e., LinH $\neq \Sigma_i^{\mathrm{lin}}$ for all i).

The notion of NP-completeness, Cook's theorem, and the *P versus NP problem* are central to complexity theory, as well as to the connections with logic, and in Section 3.1 we shall review more basics, in particular some facts from circuit complexity.

Many interesting problems and notions arise in connection with *counting functions*. For $R(x, y)$ a binary predicate with the property that for every x there are only finitely many y's satisfying $R(x, y)$ defines the function

$$\#R(x) := \text{ the number of } y\text{'s such that } R(x, y)$$

Class #P consists of all functions $\#R(x)$ with the polynomial time computable relation $R(x, y)$ and satisfying the preceding finiteness property in a stronger form (cf. Valiant 1979):

$$R(x, y) \rightarrow |y| \leq |x|^{O(1)}$$

An important result of Toda (1989) is that every language in PH is polynomial–time reducible to a function in #P.

Nonuniform versions of the preceding classes are defined with the help of *advice functions*. *Polynomially bounded advice* is a function $f : N \rightarrow \{0, 1\}^*$ such that:

$$|f(n)| = n^{O(1)}$$

The class P/poly, a nonuniform version of P, is the class of all sets A such that there are a set $B \in \mathrm{P}$ and a polynomially bounded advice function f for which it holds

$$x \in A \text{ iff } (x, f(|x|)) \in B$$

The classes NP/poly, L/poly, and so on, are defined analogously (see the paragraph after Theorem 3.1.4).