***Theorem 3.1.16*** (Karchmer and Wigderson 1988). *For any Boolean function $f$*

$$\text{Depth}(f) = \text{CC}(f)$$

As little is known about the size of formulas as about the circuit-size.

***Theorem 3.1.17*** (Andreev 1987, Hastad 1993). *There is a polynomial-time language $Z$ such that*

$$L(Z_n) \geq n^{3-o(1)}$$

The language $Z$ from the theorem is a rather artificial one. Earlier Chrapchenko (1971) showed

$$L(\oplus(x_1, \ldots, x_n)) \geq n^2$$

## 3.2. Bounded arithmetic formulas

We shall consider several languages of arithmetic as underlying languages for various systems of bounded arithmetic, but there are two basic ones: the language of *Peano arithmetic* $L_{\text{PA}}$ defined in Section 2.1, and the language of the theory $S_2$, denoted simply $L$, which extends the language $L_{\text{PA}}$ by three new function symbols

$$\left\lfloor \frac{x}{2} \right\rfloor \qquad |x| \qquad x\#y$$

The intended values of $|x|$ and $x\#y$ are $\lceil \log_2(x+1) \rceil$ for $x > 0$ and $|0| = 0$, and $2^{|x|\cdot|y|}$, respectively. Note that $|x|$ is the length of the binary representation of $x$, if $x > 0$.

We shall consider the class of bounded formulas in the language $L_{PA}$ first. They were first defined by Smullyan (1961), who called sets defined by such formulas *constructive arithmetic sets*.

***Definition 3.2.1.***

1. *$E_0 = U_0$ is the class of quantifier free formulas.*
2. *Class $E_{i+1}$ is the class of formulas logically equivalent (i.e., in the predicate calculus) to a formula of the form*

$$\exists x_1 < t_1(\overline{a}) \ldots \exists x_k < t_k(\overline{a}) \phi(\overline{a}, \overline{x})$$

   *with the formula $\phi \in U_i$ and $t_i(\overline{a})$'s terms of the language $L_{\text{PA}}$*
3. *$U_{i+1}$ is the class of formulas logically equivalent to a formula of the form*

$$\forall x_1 < t_1(\overline{a}) \ldots \forall x_k < t_k(\overline{a}) \phi(\overline{a}, \overline{x})$$

   *with the formula $\phi \in E_i$.*

4. *Class* $\Delta_0$ *of* bounded arithmetic formulas *is the union of classes* $E_i$ *and* $U_i$

$$\Delta_0 = \bigcup_i E_i = \bigcup_i U_i$$

Note that both $E_i$ and $U_i$ are contained in both $E_{i+1}$ and $U_{i+1}$.

For $M$ a structure for language $L_{PA}$ symbols $E_i(M^\ell)$, $U_i(M^\ell)$, and $\Delta_0(M^\ell)$, respectively, denote the classes of subsets of $M^\ell$ definable by the $E_i$, $U_i$, and $\Delta_0$ formulas, respectively (we shall usually omit the superscript $\ell$ when it is obvious from the context). Already the class $E_1(M)$ can be quite nontrivial from the complexity-theoretic point of view, as according to Adleman and Manders (1977) the class $E_1(\boldsymbol{N})$ contains an NP-complete set

$$\{(a, b, c) \mid \exists x < c \exists y < c, ax^2 + by = c\}$$

There are several important characterizations of the class $\Delta_0(\boldsymbol{N})$. We start with the notion of *rudimentary sets* introduced by Smullyan (1961).

The intended structure for the language of rudimentary sets is the set of words over {0, 1} or, via dyadic coding, the set of natural numbers.

The language of rudimentary sets consists of

1. $\Lambda$: *the empty word*,
2. $\frown$: *the concatenation*,
3. 0, 1: *constants*,

and two special kind of quantifiers

4. $\exists x \subseteq_p y$ and $\forall x \subseteq_p y$: *the part-of quantifiers*,
5. $\exists |x| \leq |y|$ and $\forall |x| \leq |y|$: *the length-bounded quantifiers*.

The meaning of $x \subseteq_p y$ is that the word $x$ is a part of the word $y$

$$\exists z_1, z_2; z_1 \frown x \frown z_2 = y$$

and the meaning of $|x| \leq |y|$ is obvious: the length of $x$ is at most the length of $y$.

***Definition 3.2.2*** (Smullyan 1961).

1. *The class of* rudimentary sets *RUD is the class of subsets of* $\boldsymbol{N}^\ell$ *definable in the language of rudimentary sets with all quantifiers either* part-of *or* length-bounded.
2. *The class of* strictly rudimentary sets *SRUD is the class of subsets of* $\boldsymbol{N}^\ell$ *definable in the language of rudimentary sets with all quantifiers of the* part-of *type.*
3. *The class of* positive rudimentary sets $RUD^+$ *is the class of subsets of* $\boldsymbol{N}^\ell$ *definable in the language of rudimentary sets with all quantifiers are either* part-of *or* length-bounded, *and in which all quantifiers* $\exists |x| \leq |y|$ *appear positively and all quantifiers* $\forall |x| \leq |y|$ *appear negatively.*

4. *The class of* strongly rudimentary sets *strRUD is the class of sets that are positive rudimentary and whose complements are also positive rudimentary.*

*Note that terms are allowed to appear in the quantifiers.*

5. *A function* $f : \boldsymbol{N}^{\ell} \mapsto \boldsymbol{N}$ *is* rudimentary *if its graph is a rudimentary set and the function is majorized by a polynomial.*

***Theorem 3.2.3*** (Bennett 1962).

$$RUD = \Delta_0(\boldsymbol{N})$$

*Proof (sketch).* Clearly there are only two claims to be established:

***Claim 1.*** *The graphs of addition and the multiplication are in RUD.*

***Claim 2.*** *The graph of the operation of concatenation is in* $\Delta_0(\boldsymbol{N})$.

The idea of the proof of Claim 1 is in *Bennett*'s lemma saying that any function defined by bounded recursion on notation is rudimentary (see Lemma 3.2.4). We shall see a bit stronger argument of the same type in Theorem 3.2.8.

For Claim 2 note that

$$x \frown y = z \quad \text{iff} \quad \exists w < z,\ y < w \wedge x \cdot w + y = z \wedge \text{“ } w \text{ is a power of two”}$$

where the last condition is expressed by

$$\forall 1 < u,\ v < w\ \exists t \leq u,\ u \cdot v = w \rightarrow 2 \cdot t = u$$

Q.E.D.

***Lemma 3.2.4*** (Bennett 1962). *Assume that a function* $f$ *is defined from two rudimentary functions* $g$ *and* $h$ *by* bounded recursion on the notation

1. $f(\overline{0}, \overline{y}) = h(\overline{y})$
2. $f(x_1^{\frown}\epsilon_1, \ldots, x_n^{\frown}\epsilon_n, \overline{y}) = g(\overline{x}, \overline{\epsilon}, \overline{y}, f(\overline{x}, \overline{y}))$ *for all* $\overline{\epsilon} \in \{0, 1\}^n$

*and satisfies the condition*

3. $|f(\overline{x}, \overline{y})| \leq O\left(\sqrt{\sum_i |x_i| + \sum_j |y_j|}\right)$.

*Then the function* $f$ *is rudimentary too.*

***Theorem 3.2.5*** (Wrathall 1978).

$$\mathrm{Lin}H = RUD$$

*Proof (sketch).* Using the natural coding of computations of machines by 0–1 strings one verifies that $\Sigma_0^{\mathrm{lin}} \subseteq$ RUD, from which LinH $\subseteq$ RUD follows immediately.

The opposite inclusion is obvious. Q.E.D.

The possibility of coding in $\Delta_0(N)$ merits further discussion. We shall now mention two results and return to this topic again in Section 5.4.

***Theorem 3.2.6*** (Bennett 1962). *The graph of exponentiation*

$$\{(x, y, z) \mid x^y = z\}$$

*is rudimentary.*

***Theorem 3.2.7*** (Wrathall 1978). *All context-free languages are rudimentary and hence in* $\Delta_0(N)$.

The last theorem finds a root in an important theorem of Nepomnjascij (1970), generalizing Lemma 3.2.4.

The term TimeSpace($f(n)$, $g(n)$) denotes the class of languages recognized by a Turing machine working simultaneously in time $f(n)$ and space $g(n)$.

***Theorem 3.2.8*** (Nepomnjascij 1970). *Let* $c > 0$ *and* $1 > \epsilon > 0$ *be two constants. Then*

$$\text{TimeSpace}(n^c, n^\epsilon) \subseteq \Delta_0(N)$$

*Proof (sketch).* We shall give an idea of the proof. By induction on $k$ prove that

$$\text{TimeSpace}(n^{k\cdot(1-\epsilon)}, n^\epsilon) \subseteq \Delta_0(N)$$

If $k = 1$ then the sequence consisting of the instantaneous descriptions of a TimeSpace($n^{k\cdot(1-\epsilon)}$, $n^\epsilon$) computation has size $O(n)$, and hence its code is bounded by a polynomial in input $x$, $|x| = n$.

Assume we have

$$\text{TimeSpace}(n^{k\cdot(1-\epsilon)}, n^\epsilon) \subseteq \Delta_0(N)$$

and let

$$L \in \text{TimeSpace}(n^{(k+1)\cdot(1-\epsilon)}, n^\epsilon)$$

Then $x \in L$ if and only if there exists a sequence $w = (w_0, \ldots, w_r)$ such that $w_0 = x$, each $w_{i+1}$ is an instantaneous description obtained from the instantaneous description $w_i$ by a TimeSpace($n^{k\cdot(1-\epsilon)}$, $n^\epsilon$) computation, $w_r$ is a halting accepting position, and $r \leq n^{1-\epsilon}$.

The length of any such $w$ is again $O(n)$ and the conditions defining it are $\Delta_0$-definable by the induction assumption. Q.E.D.

***Corollary 3.2.9.***

$$L \subseteq \Delta_0(N)$$

The main problem about $\Delta_0(\boldsymbol{N})$ is whether the hierarchy collapses, which is the same as whether LinH collapses: that is, whether

$$\Delta_0(\boldsymbol{N}) = E_i(\boldsymbol{N})$$

for some $i$.

The only partial result is the following *weak hierarchy theorem* of Wilkie and Woods.

***Theorem 3.2.10*** (Wilkie 1980, Woods 1986). *Denote by $V_k(\boldsymbol{N})$ the class of subsets of $\boldsymbol{N}$ definable by a $\Delta_0$-formula $\phi(x)$ with at most $k$ quantifiers bounded by $\leq x$.*

*Then for all $k$*

$$V_k(\boldsymbol{N}) \subset V_{k+1}(\boldsymbol{N})$$

The rest of this section is devoted to bounded formulas in the language $L$.

***Definition 3.2.11*** (Buss 1986).

1. *The class $\Sigma_0^b = \Pi_0^b$ of* sharply bounded *formulas consists of formulas in which all quantifiers have the form*

$$\exists x < |t| \quad or \quad \forall x < |t|$$

*That is, the quantifiers are bounded by the length of a term.*

2. *For $0 \leq i$ the classes $\Sigma_{i+1}^b$ and $\Pi_{i+1}^b$ are the smallest classes satisfying*
   (a) $\Sigma_i^b \cup \Pi_i^b \subseteq \Sigma_{i+1}^b \cap \Pi_{i+1}^b$
   (b) *both $\Sigma_{i+1}^b$ and $\Pi_{i+1}^b$ are closed under sharply bounded quantification, disjunction $\vee$, and conjunction $\wedge$*
   (c) *$\Sigma_{i+1}^b$ is closed under bounded existential quantification*
   (d) *$\Pi_{i+1}^b$ is closed under bounded universal quantification*
   (e) *the negation of a $\Sigma_{i+1}^b$-formula is $\Pi_{i+1}^b$, and the negation of a $\Pi_{i+1}^b$-formula is $\Sigma_{i+1}^b$.*
3. *The class $\Sigma_\infty^b$ of bounded $L$-formulas is the union $\bigcup_i \Sigma_i^b = \bigcup_i \Pi_i^b$.*
4. *A $\Sigma_i^b$-formula is $\Delta_i^b$ (respectively $\Delta_i^b$ in a theory $T$) iff it is equivalent to a $\Pi_i^b$-formula in predicate logic (respectively in $T$).*

In words: The complexity of bounded formulas in language $L$ is defined by counting the number of alternations of bounded quantifiers, ignoring the sharply bounded ones, analogously to the definition of levels of the *arithmetical hierarchy* where one counts the number of alternations of quantifiers, ignoring the bounded ones.

***Theorem 3.2.12.*** *The subsets of N defined by $\Sigma^b_\infty$-formulas are exactly the sets from the polynomial-time hierarchy PH.*

*In fact, for $i \geq 1$ the $\Sigma^b_i$-formulas exactly define the $\Sigma^p_i$-predicates.*

*Proof (sketch).* The only difference from Lemma 3.2.4 and Theorem 3.2.5 is that now we need to code computations of length $n^{O(1)}$, $n = |x|$. If $|y| \leq n^{O(1)}$ then $y \leq x\#\ldots\#x$, which is a term of $L$; hence such $y$'s can appear in bounded quantifiers. Q.E.D.

We should note that Bennett (1962) also considered a class of the *extended rudimentary* sets, which are defined similarly to the rudimentary sets except that the language is augmented by a function of the growth rate of the function #. It is then a straightforward extension of Theorem 3.2.5 that the extended rudimentary sets are exactly those from the polynomial time hierarchy PH.

## 3.3. Bibliographical and other remarks

For the history of results and ideas from Section 3.1 the reader should consult Boppana and Sipser (1990), Wegener (1987), and Sipser (1992). Important topics omitted are NP-*completeness*, for which Garey and Johnson (1979) is a good source, and the completeness results for other classes, in particular, the completeness of *directed st-connectivity* for class NL and the completeness of *undirected st-connectivity* for class L/poly (Aleliunas et al. 1979). Karchmer and Wigderson (1988) and Raz and Wigderson (1990) study the depth of monotone circuits for connectivity and matching.

Other interesting facts, but not used later in the book, concern branching programs: Barrington (1989) characterized Boolean functions with polynomial size formulas as those computed by width 5, polynomial-size branching programs, and a relation of space bounded Turing computations to size of branching programs: L/poly = BP (cf. Wegener 1987).

Very important but unfortunately unpublished is Bennett's Ph.D. thesis (Bennett 1962), containing either explicitly or implicitly most later definability results such as Cobham (1965) and Nepomnjascij (1970). Paris and Wilkie (1981b) study rudimentary sets explicitly.