2. NONUNIFORM COMPLEXITY MEASURES

Objekttyp: **Chapter**

Zeitschrift: **L'Enseignement Mathématique**

Band (Jahr): **28 (1982)**

Heft 1-2: **L'ENSEIGNEMENT MATHÉMATIQUE**

PDF erstellt am: **26.04.2024**

always": clearly there are sets with small circuits that are not even recursive. The very trivial nature of such "counter-examples" suggests, however, that a more careful investigation may still yield insight. Indeed, as we will show, if one considers not arbitrary sets but rather "well behaved ones" it is possible to achieve our goal. For example, we will show that if SAT has small circuits, then the Meyer-Stockmeyer [19] hierarchy collapses.

Thus, here is an example of a nonuniform upper bound that has uniform consequences. The proof, of course, will depend on the fact that SAT is not a "pathological" set, but is rather well behaved.

Our results also serve to rule out some plausible speculations about the complexity of problems in *NP*. For example, one might imagine that $P \neq NP$, but SAT is tractable in the following sense: for every l there is a very short program that runs in time n^2 and correctly treats all instances of length l. Theorem 5.2 shows that, if "very short" means "of length c log 1", then this speculation is false.

Finally, we mention that the proof techniques presented here were put to use by S. Mahaney in his proof that $P \neq NP$ implies the nonexistence of sparse *NP*-complete problems [11], and by S. A. Cook in his proof that

$$P \subseteq HARDWARE\ (log\ n) \Rightarrow P \subseteq DSPACE\ (log\ n\ log\ log\ n)$$

[5].

2. Nonuniform Complexity Measures

In this section we will define our basic notion of nonuniform complexity and relate it to circuit complexity.

Let S be a subset of $\{0, 1\}^*$. Let $h: N \to \{0, 1\}^*$ where N is the set of natural numbers. Define $S:h = \{x \mid h(|x|) \cdot x \in S\}$. Next, let V be any collection of subsets of $\{0, 1\}^*$ and let F be any collection of functions from N to N. The key definition is

$$V/F = \{S : h \mid S \,\varepsilon\, V \text{ and } h \,\varepsilon\, F\}$$

Intuitively, V/F is the collection of subsets of $\{0, 1\}^*$ that can be accepted by V with an amount of advice bounded by F. The idea behind this definition is foreshadowed in papers by Pippenger [14] and Plaisted [15].

We are mainly interested in *poly*, the collection of all polynomially-bounded functions, and *log*, the collection of all functions that are 0 (*log n*). Indeed, many of our results will concern the classes *P*/*poly* and *P*/*log*.

If f is a function, V/f is synonymous with $V/\{f\}$. Some preliminary facts are:

(1) for all V, $V/0 = V$;

(2) any subset of $\{0, 1\}^*$ is in $P/2^n$;

(3) if f is infinitely often nonzero, then P/f contains nonrecursive sets;

(4) if $g(n) < f(n) \leqslant 2^n$ (*i.o.*) then $P/f \subseteq P/g$.

The class *P/poly* can be characterized in terms of classic circuit complexity. An n-input m-gate *Boolean circuit* C is a function

$$C: \{n+1, \dots, n+m\} \to \{0, 1\}^4 \times \{1, \dots, n+m\}^2$$

satisfying: if $C(i) = \langle B, j, k \rangle$ then $j < i$ and $k < i$. The interpretation of C is that gate i uses the truth table B on inputs j and k to produce its output. If $1 \leqslant j \leqslant n$ then input j is simply the input variable x_j; otherwise, input j is the output of gate j. In the usual way we define what it means for a circuit C to *realize* the Boolean function f. Then let $\mathrm{L}(f)$ denote the minimum number of gates in a Boolean circuit realizing the Boolean function f. Next, as in the introduction, if S is a subset of $\{0, 1\}^*$, then $\mathrm{S}_n : \{0, 1\}^n \to \{0, 1\}$ is defined by

$$\mathrm{S}_n(x_1, x_2, \dots, x_n) = \begin{cases} 1, & \text{if } x_1 x_2 \dots x_n \in \mathrm{S} \\ 0, & \text{otherwise} \end{cases}$$

Finally, recall that a set S has *small circuits* if $\mathrm{L}(\mathrm{S}_n)$ is bounded by a polynomial in n.

The following simple theorem, which is given in [14], characterizes *P/poly*.

THEOREM 2.1. Let S be a subset of $\{0, 1\}^*$. Then the following are equivalent.

(1) S has small circuits.

(2) S is in *P/poly*.

Another way we can gain insight into our classes V/F is to use them to restate other known results. For example, the result in [2] that there are short universal traversal sequences for undirected graphs can be restated as

$$\textit{UGAP} \text{ is in } \textit{DSPACE}\,(\log n)/\textit{poly}.$$

Here *UGAP* is the undirected maze problem. As another example, we have Adleman's [1] result that R (the set of languages accepted in polynomial time by randomizing Turing machines) has small circuits, which can be restated as

$$R \text{ is a subset of } P/poly\,.$$

It may be interesting that both these results use the probabilistic method of Erdös to prove the existence of the required advice bits.

3. Summary of Main Results

We will discuss a variety of complexity classes. These include the basic time and space classes $DTIME\,(T\,(\mathrm{n}))$, $DSPACE\,(\mathrm{S}\,(\mathrm{n}))$ and $NSPACE\,(\mathrm{S}\,(\mathrm{n}))$ and the classes:

P = the set of languages accepted in deterministic polynomial time,

R = the set of languages accepted in polynomial time by randomizing Turing machines [1],

NP = the set of languages accepted in nondeterministic polynomial time,

$PSPACE$ = the set of languages accepted in polynomial space,

$$EXPTIME = \bigcup_{i>0} DTIME\,(2^{n^i})\,.$$

Also important is the polynomial-time hierarchy of Meyer and Stockmeyer [19]. For $i \geqslant 1$ we let $\sum_i^p$ (respectively $\prod_i^p$) denote those languages accepted in polynomial time by Turing machines that make i alternations starting from an existential (respectively universal) state. Note that $NP = \sum_1^p$ and co- $NP = \prod_1^p$. Finally, note that P, $PSPACE$ and $EXPTIME$ can be viewed as complexity classes associated with alternating Turing machines; specifically, $P = ASPACE\,(log\,n)$, $PSPACE = AP$ and $EXPTIME = APSPACE$ [3, 10].

Many of the following theorems take the form

$$L \subseteq S/F \Rightarrow L \subseteq S'$$

where L and S' are uniform complexity classes and V/F is a nonuniform complexity class. The proof usually consists of showing that