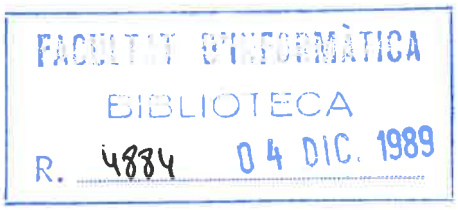
Strong and robustly strong polynomial time reducibilities to sparse sets

R. Gavaldà
J.L. Balcázar

Report LSI–88–15



STRONG AND ROBUSTLY STRONG POLYNOMIAL TIME REDUCIBILITIES TO SPARSE SETS *

Ricard Gavaldà and José L. Balcázar
Department of Software (Llenguatges i Sistemes Informàtics)
Universitat Politècnica de Catalunya (U.P.C.)
08028 Barcelona, Spain

Abstract. Reducibility defined by oracle strong nondeterministic machines is studied. Two definitions of relativized strength are presented and separated. The corresponding reduction classes of the sparse sets give two nonuniform analogs of the class $NP \cap coNP$. An oracle-restricted positive relativization of the probabilistic class ZPP is developed.

Resum. Estudiem reduïbilitats definides per màquines indeterministes fortes amb oracle. Proposem dues definicions de força relativitzada i les separem. Les classes de reducció corresponents a conjunts esparsos proporcionen classes no uniformes anàlogues a la classe $NP \cap coNP$. Desenvolupem amb aquestes definicions una relativització positiva de la classe ZPP.

1. Introduction

In the study of complexity classes defined by sequential models of computation, several classes have been identified by considering bounded amounts of the main resources. An important line of research has studied the advantages of having access to an "advice" of feasible size that can help the computations. This approach defines the nonuniform complexity classes, that can be related to some other models of computation.

Thus, Karp and Lipton (1980) show that the class P with advice, denoted as $P/poly$, can also be defined by considering polynomial time deterministic machines that can query sparse oracles. Pippenger (1979) shows that it coincides also with the class of sets that have small (i.e. polynomial) size acepting circuits. Similarly, Yap (1983) and Schöning (1984) study the class NP with advice, $NP/poly$, and show that it contains precisely the sets accepted by nondeterministic machines relativized to sparse oracles. It also turns out that such sets are the range of polynomial size circuits with many outputs (small generators).

* Part of this work was presented at Symposium on Mathematical Foundations of Computer Science, Carlsbad, Czechoslovakia, 1988

The purpose of this work is to find similar results for the class $NP \cap coNP$ with advice. In particular, we study what kind of sequential machine yields this class when relativized to sparse oracles, and characterize it in terms of boolean circuits.

Inspired by the work of Long (1982) on strong nondeterministic machines, we propose a model of strong generators, and study its relationship with $(NP \cap coNP)/poly$. These generators, however, define instead the class $(NP/poly) \cap (coNP/poly)$. As these two classes seem not to be exactly equivalent, we engage in a deeper study of their differences and relationships.

We then focus on the two concepts of "strength" for nondeterministic oracle machines used in the literature: that which depends of the oracle set and that which is "robust" against changes on the oracle set. References that use these definitions are, respectively, Long (1982) and Book, Long, and Selman (1985). We obtain two forms of reducibility, show that their zero degrees coincide, and then separate them in the class of the recursive sets. We then consider their relativizations to sparse sets, and we see that, due to an overlooked hypothesis, the very general theorem of Schöning (1984) can be applied only to one of these two reducibilities. (A complete restatement of this theorem can be found in the preliminaries.) Thus a characterization of $(NP \cap coNP)/poly$ in terms of sparse oracles is obtained.

These characterizations allow us to discuss the possibility that the two nonuniform versions of $NP \cap coNP$ with advice are equal. Recall from Yap (1983) that if $NP \subseteq coNP/poly$, then the polynomial time hierarchy collapses to its Σ_3 level. We conclude that if the two classes found are equal, we can prove a collapse to Σ_2 and so Yap's result is not optimal.

Finally, since sets in ZPP can be considered as well tractable, we have studied relativizations of this class, restricted in an analogous manner to a certain degree of robustness. The interest of the obtained class is justified by the fact shown here that the class yields an oracle-restricted positive relativization of the equality $P \stackrel{?}{=} ZPP$.

2. Preliminaries

We assume that the reader is familiar with basic concepts of complexity theory, such as the classes P, NP, $PSPACE$, and the polynomial time hierarchy, PH.

All our sets are defined over an alphabet Σ. Here Σ^n and $\Sigma^{\leq n}$ mean the sets of words of length exactly n and up to n, respectively. We denote with $\langle .,. \rangle$ any pairing function easy to compute and invert.

Besides the standard Turing machines with final, accepting states, we consider also machines having three kinds of computations: those that accept the input, those that reject the input, and those that are "undefined" in the sense that they stop without answer. For a machine M and an oracle set B, $M^B(x)$ denotes the set of possible computations of M with oracle B and input x, and $L(M, B)$ the set of accepted inputs.

We say that a set A is *self-reducible* if there is a polynomial time machine M such that M only queries its oracle about strings shorter than its input and $A = L(M, A)$. Recall that SAT, the set of satisfiable boolean formulas, is known to be NP-complete and self-reducible. We also make use of the concept of lowness, as defined in Schöning (1983). For any $n \geq 1$, define the class of low sets at level n as

$$L_n = \{A \in NP : \Sigma_n(A) \subseteq \Sigma_n\}$$

Our notation for nonuniform classes follows Karp and Lipton (1980). In particular, all our complexity classes are of the form $C/poly$, for a complexity class C. We say that a set A is in $C/poly$ if there is a polynomial p, an "advice function" $h : \mathbb{N} \to \Sigma^*$, and an "interpreter set" I in C such that for every n, $|h(n)| \leq p(n)$, and for every $x \in \Sigma^{\leq n}$

$$x \in A \text{ iff } \langle x, h(|x|)\rangle \in I$$

For any reasonable class C, Σ^n can be used instead of $\Sigma^{\leq n}$ in this definition.

Other notations are either adjacent to an appropriate reference, or are standard and can be found in textbooks like Garey and Johnson (1979), Hopcroft and Ullman (1979), or Balcázar, Díaz, and Gabarró (1988).

Two kinds of known results about these classes will be used here: the connection with classes relativized to sparse oracles and the equivalence with certain types of boolean circuits.

Generalizing previous results, Schöning (1984) presents a result which allows to relate, in an almost "mechanical" way, nonuniform "advice" classes to sparse relativizations. A somewhat informal argument, however, makes the result applicable only to complexity classes defined by oracle-independent conditions on the machines, as being polynomially clocked and the like. In order to apply this result to the machines we deal with here, an implicit hypothesis has to be remarked. Thus, we reformulate here this result. The proof is essentially the same, and will be omitted.

Let us start with some definitions. Let C be a relativizable complexity class; we assume that for each set in C there is a "type C" machine to witness it.

We say that C is *good* if and only if for every oracle machine N of type C and oracle machine M of type P (i.e. polynomial time clocked) with output, the machine that simulates M and then simulates N under the empty oracle on the output of M is also of type C.

We say that C is *oracle-resistant* if and only if for every oracle machine N of type C, the (non oracle) machine that on input $\langle x, y\rangle$ simulates N on input x using as oracle the set encoded by y is also of type C.

Informally, "goodness" means that the class is "closed under composition with P machines", which allows to perform some preprocessing on the input without exceeding

the computational power of the class; while "oracle-resistance" means that the fact that a machine behaves as a type C machine does not really depend of the oracle used, which allows to "get rid of" the oracle remaining again within the class.

1. Theorem (Schöning 1984). Let C be a good, oracle-resistant class such that type C machines make queries polynomially bounded on the length of the input. Then

$$C(\emptyset)/poly = \bigcup\{C(S) : S \text{ sparse}\}$$

The proof is as in Schöning (1984). It can be seen that the inclusion left to right only requires goodness, but that in the converse inclusion the additional property of oracle-resistance is needed as well. In fact, we will present below a class to which the theorem does not seem to apply, since it is good but not oracle-resistant.

The result is of course applicable to the classes in the polynomial time hierarchy, and to $PSPACE$.

2. Proposition. P, NP, $PSPACE$, and all the classes in the polynomial time hierarchy are good, oracle-resistant, and accepted by machines making polynomially bounded queries.

Finally, it should be noticed that nonuniform complexity classes frequently can be characterized by very natural computational models. In particular, $P/poly$ is the class of sets accepted by polynomial size boolean circuits; see Pippenger (1979). By using circuits with many outputs, Yap (1983) proposes to consider circuits as generators instead of as acceptors, considering the sets that are the range of a family of polynomially bounded circuits, and relates this class to $NP/poly$. Schöning (1984) completes the relationship by establishing a characterization of $NP/poly$. In section 6 below we present a "strong" version of these generators, and give a characterization in the framework of our discussion about the classes $(NP/poly) \cap (coNP/poly)$ and $(NP \cap coNP)/poly$.

3. Strong machines and robustly strong machines

In this section we present the strong nondeterministic machines, and two versions of the oracle strong nondeterministic machines. Strong machines were introduced by Long (1982), who studied extensively the strong nondeterministic Turing reducibility, based on previous work by Adleman and Manders (1977). Robustly strong machines have been used by Book, Long, and Selman (1985) to obtain positive relativizations of the equality $P \stackrel{?}{=} NP \cap coNP$, and later by Long and Selman (1986) to obtain oracle-restricted positive relativizations of the same equality, as discussed below. Hemachandra (1987) used another notion (robustly complementary machines) to study essentially the same idea.

3. Definition. A (non oracle) nondeterministic Turing machine is *strong* if and only if for every input x:

1. there is a defined computation in $M(x)$, and

2. there is an accepting computation in $M(x)$ if and only if there is no rejecting computation in $M(x)$.

Besides their theoretical importance to define reducibilities, strong machines are attractive since they are nondeterministic but not inconsistent: they do not give contradictory answers.

The following fact is clear: by switching accepting and rejecting states on a strong machine, a new strong machine is obtained which accepts the complement of the originally accepted language. This allows to characterize the class of sets accepted by strong machines in polynomial time.

4. Proposition (Long, 1982). A is accepted in polynomial time by a strong machine if and only if $A \in NP \cap coNP$.

Consider now strong polynomial time nondeterministic machines which have access to an oracle set. At least two interpretations of the word "strong" are possible; we present both, and define their corresponding polynomial time reducibilities.

5. Definition. M is *strong under oracle* B if and only if for every input x:

1. there is a defined computation in $M^B(x)$, and

2. there is an accepting computation in $M^B(x)$ if and only if there is no rejecting computation in $M^B(x)$.

The corresponding reducibility is denoted $\leq^{SN}$.

6. Definition. A set A is *SN-reducible* to a set B ($A \leq^{SN} B$) if and only if $A = L(M, B)$ for a polynomial time machine M which is strong under B.

The notation of Long (1982) for this reducibility is $\leq_T^{SN}$; other versions, more restricted, are defined as well, like $\leq_{tt}^{SN}$. Long shows that all of these reducibilities are different among them and from $\leq_T^P$ and $\leq_T^{NP}$, and that the power of $\leq_T^{SN}$ is intermediate between $\leq_T^P$ and $\leq_T^{NP}$.

An equivalent definition of $\leq^{SN}$ that we will ocasionally use is given by the following:

7. Proposition. For any A and B, $A \leq^{SN} B$ iff $A \in NP(B) \cap coNP(B)$.

As mentioned, a second interpretation of the concept of strong nondeterministic oracle machine has been used in the literature; it requires the machine to be strong no matter the oracle it uses. We call these machines "robustly strong", and define the corresponding reducibility.

8. Definition. M is *robustly strong* if and only if for every oracle A, M is strong under A.

9. Definition. A set A is *RS-reducible* to a set B ($A \leq^{RS} B$) if and only if $A = L(M, B)$ for a polynomial time machine M which is robustly strong.

It is clear that $\leq_T^P$ implies $\leq^{RS}$ and that $\leq^{RS}$ implies $\leq^{SN}$. The following theorem shows another basic relationship.

10. Theorem. For any two sets A and B:

(i) $A \leq^{SN} B$ and $B \in NP \cap coNP \Rightarrow A \in NP \cap coNP$.

(ii) $A \leq^{RS} B$ and $B \in NP \cap coNP \Rightarrow A \in NP \cap coNP$.

The proof for $\leq^{SN}$ can be found in Long (1982) and the other is similar. An interesting consequence is that the zero degree of both reducibilities is exactly $NP \cap coNP$.

These two reducibilities, however, do not coincide in general: our main result in this section shows that $\leq^{RS}$ differs essentially from all the reducibilities of Long, above $NP \cap coNP$. We prove first that it differs from $\leq^{SN}$. Intuitively, the difference is due to the fact that plain strong machines may expect the adequate oracle and exploit its structure in order to be strong, while robustly strong machines must maintain their "coherence" by themselves, expecting no "a priori" particular property of the oracle.

11. Theorem. For every recursive set $A \notin NP \cap coNP$, there is a recursive set B such that $A \leq^{SN} B$ but $A \not\leq^{RS} B$.

Proof. Let A be fixed. We construct B such that, for all words x,

$$x \in A \Leftrightarrow \exists y(|y| = |x| \text{ and } \langle x, y, 1\rangle \in B) \Leftrightarrow \forall z(|z| = |x| \Rightarrow \langle x, z, 0\rangle \notin B) \qquad (i)$$

which guarantees that $A \leq^{SN} B$, using the natural procedure.

We construct B so that $A \not\leq^{RS} B$ by diagonalization over all the machines that could reduce A to B. At stage n, and using the initial segment B_{n-1} constructed so far, we search for the minimum word x that satisfies:

(1) it does not interfere with previous stages, and either

(2) the machine M_n with oracle B_{n-1} and input x is not strong, or

(3) $M_n(x)$ accepts and $x \notin A$, or

(4) $M_n(x)$ rejects and $x \in A$.

When found, x is used to extend B_{n-1} to B_n, in a way that preserves condition (i).

(5) find a word y, $|y| = |x|$, not queried by one computation found in (3) or (4), and

(6) add $\langle x, y, 1\rangle$ to B_{n-1} if (4) holds, and $\langle x, y, 0\rangle$ if (3) holds.

If case (2) appears then any word of length $|x|$ can be selected for y. It is easy to see that, by conditions (1) to (4), x witnesses that $A \not\leq^{RS} B$ via M_n.

We prove now that the witness x must exist at each stage. Otherwise, if conditions (2) to (4) are false for all x, M_n is correctly RS-reducing A to B_{n-1}. But, since B_{n-1} is finite, A must be in $NP \cap coNP$, which contradicts the hypothesis. ∎

We use a result from Long (1982) to show that $\leq^{RS}$ also differs from $\leq^{SN}_{tt}$.

12. Theorem. For every recursive set $A \notin NP \cap coNP$, there is a recursive set B such that $A \leq^{RS} B$ but $A \not\leq^{SN}_{tt} B$.

Proof. Given A, Long shows that there is a B such that $A \leq_T^P B$ and $A \not\leq_{tt}^{SN} B$. Since $\leq_T^P$ implies $\leq^{RS}$, the same B satisfies the statement of the theorem. ∎

All robustly strong machines that we can exhibit (and that really use its oracle) can be trivially made deterministic. This raises the question of whether robustly strong polynomial time reducibility differs from plain polynomial time reducibility, which has been addressed in Book, Long, and Selman (1985) to investigate positive relativizations of $P \stackrel{?}{=} NP \cap coNP$. The separation has been left there as an important open problem; moreover, if two additional technical conditions are imposed (confluence and maturity), then a positive relativization is obtained.

An important step forward has been obtained by Hemachandra (1987), who shows —theorem 5.3— that the reduction class of any recursive set A under $\leq^{RS}$ is always included in the class $P(A \oplus SAT)$. It follows that:

— if $\leq^{RS}$ equals $\leq_T^P$, then $P = NP \cap coNP$, and

— if $\leq^{RS}$ differs from $\leq_T^P$ anywhere in the recursive sets, then $P \neq NP$, since if $P = NP$ then $P(A \oplus SAT) = P(A)$ and therefore $\leq_T^P$ and $\leq^{RS}$ must coincide.

So, although robustly strong machines are not exactly a positive relativization of anything, proving equality or inequality with deterministic machines would have important consequences.

4. Nonuniform classes defined by strong machines

We characterize in this section the nonuniform classes corresponding to the reduction class of the sparse sets, using the reducibilities defined in the previous section. We start with the class corresponding to the robustly strong reducibility.

13. Theorem. $\{A : \exists S(S \text{ sparse and } A \leq^{RS} S)\} = (NP \cap coNP)/poly$

Proof. It is enough to prove that this class satisfies the hypothesis required to apply theorem 1. For goodness, note that performing a deterministic computation phase (that can alter the input) before running a robustly strong machine keeps it robustly strong. For oracle resistance, note that a robustly strong machine that is given a finite oracle as part of the input must be strong on that input, since it is strong on *any* oracle and input. ∎

However, when we want to characterize in a similar manner the class

$$\{A : \exists S(S \text{ sparse and } A \leq^{SN} S)\}$$

some problems arise. Indeed, it is not difficult to see that if there is an oracle B and an input x under which a given machine M is not strong, then the machine that is given as input a pair formed by x and an initial segment of B is not strong (in the non-oracle sense). Thus, this class is not oracle-resistant and Schöning's theorem does not apply.

Anyway, a characterization in the style of the advice classes (although not properly an advice class) is obtained in the remaining of this section, preceded by a result that

amounts, roughly speaking, to "factor out" sparse oracles from complementary machines.

14. Theorem.

$$\{A : \exists S(S \text{ sparse and } A \leq^{SN} S)\} = \bigcup\{NP(S) : S \text{ sparse }\} \cap \bigcup\{coNP(S) : S \text{ sparse }\}$$

Proof. Inclusion left to right is immediate from proposition 7. For the converse, form the join of the two sparse oracles involved and design a machine that chooses nondeterministically among the two machines involved, using for each one the corresponding part of the oracle. This new machine is strong with this oracle, since one and only one of the two machines can accept the input. ∎

The nonuniform-like class characterization is as follows:

15. Theorem. $\{A : \exists S(S \text{ sparse and } A \leq^{SN} S)\} = (NP/poly) \cap (coNP/poly)$

Proof. Follows from the previous result by two applications of theorem 1, since both NP and $coNP$ belong to the polynomial time hierarchy. ∎

To end this section, we note the interesting (although not surprising) fact that tally sets can be substituted for the sparse sets in all the characterizations given.

16. Proposition.

(i) $A \leq^{SN} S$ for some sparse S iff $A \leq^{SN} T$ for some tally T.

(ii) $A \leq^{RS} S$ for some sparse S iff $A \leq^{RS} T$ for some tally T.

The proof follows standard techniques; see Hartmanis (1983).

5. Comparison of the two classes obtained

We study now some relationships between the classes obtained in the previous section. More explicitely, we obtain some consequences of the following

17. Hypothesis for this section. $(NP \cap coNP)/poly = (NP/poly) \cap (coNP/poly)$.

In Yap (1983) it is shown that if $NP \subseteq coNP/poly$, then $PH = \Sigma_3$. We show that if the hypothesis holds, this result can be improved to a collapse to Σ_2.

Some comments are in order before stating the new result. Note first that, in the hypothesis, inclusion from left to right is trivial, and both classes are equal to $NP/poly$ if $NP = coNP$. So proving the hypothesis false is at least as hard as proving $NP \neq coNP$.

Let B a set in $(NP \cap coNP)/poly$ with an interpreter set $I \in NP \cap coNP$. We can then define the set of "good advices" for B:

$$GA(B) = \{\langle 0^n, h\rangle : \forall x \in \Sigma^{\leq n}(x \in B \Leftrightarrow \langle x, h\rangle \in I)\}$$

As a correct advice function exists, there is at least one word $\langle 0^n, h\rangle$ in $GA(B)$ for every n. This set can be shown to be in $\Pi_1(B)$, and in some cases we can even drop the oracle B.

18. Lemma. For any set $B \in (NP \cap coNP)/poly$ that is self-reducible, $GA(B) \in \Pi_1$.

Proof. If I is the interpreter for B, the fact that a word $\langle 0^n, h\rangle$ is a good advice for B up to length n can be expressed:

$$\forall x \in \Sigma^{\leq n}(\langle x, h\rangle \in I \Leftrightarrow x \in B)$$

Let M be the deterministic, polynomial-time machine that self-reduces B. We then can check $x \in B$ by running M and answering the oracle queries with the help of h, so inductively if h is "good" with the (shorter) queried strings, it will be with x. Thus this predicate can be written as:

$$\forall x \in \Sigma^{\leq n}(\langle x, h\rangle \in I \Leftrightarrow x \in L(M, \{y : \langle y, h\rangle \in I\})$$

This is a $\Pi_1(I)$ predicate for $GA(B)$. Since $\Pi_1(NP \cap coNP) = \Pi_1$ (see Balcázar, Díaz and Gabarró (1988) for a proof), we have $GA(B) \in \Pi_1$. ∎

Another technical result shows that sets in these conditions are low.

19. Lemma. If $B \in NP$ and $GA(B) \in coNP$, then $B \in L_2$.

Proof. Let A be any set in $\Sigma_2(B)$. We will prove $A \in \Sigma_2$. We know that for a suitable polynomial-time machine M, it holds for all x

$$x \in A \Leftrightarrow \exists y_1\ \forall y_2\ \langle x, y_1, y_2\rangle \in L(M, B)$$

But, as there are good advices at every length, this is equivalent to

$$x \in A \Leftrightarrow \exists h\ (\langle 0^{|x|}, h\rangle \in GA(B) \wedge \exists y_1\ \forall y_2\ \langle x, y_1, y_2, h\rangle \in L(M', I))$$

where M' behaves as M substituting all queries to B about strings z by queries to I about $\langle z, h\rangle$. Now we have a predicate for A of the form $\exists((a) \wedge (b))$, where (a) is in $coNP$ by lemma 18 and (b) is in Σ_2 since $I \in NP \cap coNP$. So, the whole predicate is in Σ_2. ∎

A similar result was obtained by Kämper (1987), who proved that all sets in NP and $(NP \cap coNP)/poly$ (but not necessarily self-reducible) are in fact in L_3. Related material is stated (without a proof) in Abadi, Feigenbaum, and Kilian (1987).

Now it is easy to show the announced improvement of Yap's collapse.

20. Theorem. If hypothesis 17 is true and $NP \subseteq coNP/poly$, then $PH = \Sigma_2$.

Proof. It will suffice to show that $SAT \in L_2$, since $\Sigma_2(SAT) = \Sigma_3$. Assume that SAT is in $coNP/poly$. Then it is in $(NP/poly) \cap (coNP/poly)$ and so in $(NP \cap coNP)/poly$ by hypothesis 17, but as it is self-reducible, it is in L_2 by the previous lemmas. ∎

It is interesting to observe that, for the proof, a statement weaker than hypothesis 17 is enough, namely that $(NP/poly) \cap (coNP/poly)$ and $(NP \cap coNP)/poly$ coincide in NP.

The proof of theorem 20 is easily seen to relativize. Kadin (1988) asks whether Yap's result is optimal, that is, if there is any relativized world where the collapse of PH cannot be improved any more. It follows from theorem 20 that a proof of optimality implies a separation of our two nonuniform counterparts of $NP \cap coNP$, and moreover the existence of a witness in NP for this separation. In turn, this would improve our theorem 11 by producing a set A in NP and a *sparse* set S such that $A \leq^{SN} S$ and $A \not\leq^{RS} S$.

It is not known whether assuming hypothesis 17 has any consequences on its own (i.e. not simply improvements of previous results). No easy proofs seem at hand, mainly because little structural properties of $NP \cap coNP$ are known, such as having complete sets or being self-reducible.

6. Strong generators

As previously indicated, we present here a nonuniform model corresponding to the SN-reducibility to sparse oracles. It is similar to the small generators described by Yap (1983) and Schöning (1984).

21. Definition. A set A has *polynomial size strong generators* if and only if there is a polynomial p such that for every n, a circuit C_n and an integer e_n exist for which

1) C_n has at most $p(n)$ gates.

2) C_n has e_n inputs.

3) C_n has $n+1$ outputs, plus one additional output "domain indicator". Only "valid outputs" are considered, and these are those that appear under an input for which the domain indicator evaluates to 1.

4) For every $x \in \Sigma^n$

$$x \in A \Leftrightarrow \exists y(|y| = e_n \text{ and } C_n(y) = 1x) \Leftrightarrow \forall z(|z| = e_n \Rightarrow C_n(z) \neq 0x)$$

Notice that from 1 and 2 it follows that $e_n \leq p(n)$.

Those circuits are very similar to the generators studied in Yap (1983) and Schöning (1984), the difference being that a "strength" condition has been added. Indeed, condition 4 is analogous to the condition imposed to the nondeterministic machines in order to consider them strong. Intuitively, if C_n is a family of strong generators for A then for each n every word of length n appears as output of C_n if the appropriate input is chosen, and moreover C_n correctly indicates whether this output word is in A or in $\overline{A}$. The reader is advised to compare this model with the generators of Yap (1983) and Schöning (1984).

Our main result in this section is a characterization of the reduction class of the sparse sets under the SN-reducibility.

22. Theorem. A set A has polynomial size strong generators if and only if $A \leq^{SN} S$ for some S sparse.

Proof. Given A with polynomial size strong generators, it is an easy task to obtain standard generators for both A and $\overline{A}$: it suffices to give undefined output for the inputs that are not in the desired set. By the results in Yap (1983), both sets are in $NP/poly$, and therefore in $NP(S)$, resp. $NP(S')$, for some sparse set S, resp. S', by theorem 1. It only remains to apply theorem 15 to obtain the implication left to right.

For the converse, follow backwards the same argument, using the converse of Yap's result —which appears in Schöning (1984)—, to obtain generators for both A and $\overline{A}$, and combine the generators into a family of strong generators. ∎

Generators can be viewed also as acceptors with nondeterministic gates in a standard manner. It is easy to see that this view can be adapted as well to the strong generator model. In this case, for every word there is an extension of values of the nondeterministic gates which yields a valid output, and this output always correctly determines membership to A of the input word.

7. An oracle-restricted positive relativization of ZPP

Positive relativizations have appeared in several previous papers; see Book, Long, and Selman (1984) and the references there. A positive relativization of a pair of classes C and D is a restricted way of relativizing both classes, such that the restriction is meaningless for the unrelativized case, and such that the unrelativized classes coincide if and only if their restrictions coincide in every relativization.

A different sort of positive relativization has been developed in Long and Selman (1986) and in Balcázar, Book, and Schöning (1986). In it, an additional condition is imposed to the class of oracles, sometimes substituting the restriction on the oracle machines. As examples of this kind of oracle-restricted positive relativizations, we state some results from these references.

23. Theorem. The polynomial time hierarchy equals $PSPACE$ if and only if for every sparse set S, the polynomial time hierarchy relative to S equals $PSPACE(S)$; and the polynomial time hierarchy differs from $PSPACE$ if and only if for every sparse set S, the polynomial time hierarchy relative to S differs from $PSPACE(S)$.

24. Theorem. $P = NP$ if and only if for every tally set T, $P(T) = NP(T)$.

25. Theorem. $P = NP \cap coNP$ if and only if for every tally set T, $P(T)$ equals the class of sets $L(M, T)$ where M is robustly strong.

By restricting probabilistic machines so that certain functions describing machine's behavior are computable by certain probabilistic models, Russo (1985) has obtained (in joint work with S. Zachos) positive relativizations of ZPP and other probabilistic classes, defined by Gill (1977). Here we present a different view, by exhibiting an oracle-restricted positive relativization of ZPP. It is based on the following definition.

26. Definition. A nondeterministic machine M is *robustly* ZPP if and only if for every oracle set A and input x, either more than half the computations accept and no computation rejects, or more than half the computations reject and no computation accepts.

The main result of this section is as follows.

27. Theorem. $P = ZPP$ if and only if for every tally set T, $P(T)$ equals the class of sets $L(M,T)$ where M is robustly ZPP.

Proof. The only nontrivial part of the proof is to show that if $P = ZPP$ then robustly ZPP machines can be simulated in deterministic polynomial time. The idea is that the condition of robust-ZPP-ness implies that a machine which is given part of the oracle as input is also a ZPP machine. Then a $P(T)$ machine can be constructed that first scans the oracle to construct a table recording the accesible part of it, then simulates the ZPP machine that incorporates the oracle as part of the input. ∎

For similar proofs see Long and Selman (1986). The result is also true for a slightly more general (but very technical) class of oracles: those with "self-producible circuits". These sets are characterized in terms of tally sets in Balcázar and Book (1986), and the generalization of theorem 27 to them is immediate from this characterization.

8. Conclusions

Motivated by the question of finding a nonuniform analog of the class $NP \cap coNP$, two versions of strong nondeterministic reducibilities have been defined and compared, and differences have been found at arbitrary height within the class of recursive sets. The corresponding reduction classes of the sparse sets have been characterized in different manners. Strangely enough, the class corresponding more naturally to a nonuniform model of computation is the one that is not properly a nonuniform class in the sense of Karp and Lipton (1980), and is also the one for which the very general technique of Schöning (1984) for dealing with this kind of classes fails.

More precisely, the reduction classes of sparse sets under both reducibilities turn out to be, respectively, $(NP/poly) \cap (coNP/poly)$ and $(NP \cap coNP)/poly$. There seems to be more than a syntactic difference between these classes. It is not difficult to see that the second class is included in the first, and that they are equal if $NP = coNP$; it is shown that, assuming equality, a result by Yap can be improved, by collapsing the polynomial time hierarchy to Σ_2 instead of Σ_3.

As shown by Long and Selman (1986), the reduction class of the tally sets under the robustly strong machines yields an oracle-restricted positive relativization of the equality $P \stackrel{?}{=} NP \cap coNP$. Motivated by this fact, we have shown that a natural analogous condition yields an oracle-restricted positive relativization of the equality $P \stackrel{?}{=} ZPP$.

We consider that the main question left regards the robust reduction class to sparse sets, and can be expressed as follows: What is the reason that such a natural condition does

not yield a natural nonuniform model? Naturalness is argued in the following grounds: it is the immediate consequence of the nonuniform notation, and gives rise to positive relativizations. The answer is not known yet, but surely more insight has been gained from the comparison between both reducibilities.

9. References

M. Abadi, J. Feigenbaum, J. Kilian: "On hiding information from an oracle". 19th ACM STOC (1987), 195–203.

L. Adleman, K. Manders: "Reducibility, randomness, and intractability". 9th ACM STOC (1977), 151–163.

J.L. Balcázar, R.V. Book: "Sets with small generalized Kolmogorov complexity". *Acta Informatica* 23 (1986), 679–688.

J.L. Balcázar, R.V. Book, U. Schöning: "The polynomial time hierarchy and sparse oracles". *J. ACM* 33 (1986), 603–617.

J.L. Balcázar, J. Díaz, J. Gabarró: *Structural Complexity I.* Springer-Verlag 1988.

R. Book, T. Long, A. Selman: "Qualitative relativizations of complexity classes". *J. Comp. Sys. Sci.* 30 (1985), 395–413.

R. Book, T. Long, A. Selman: "Quantitative relativizations of complexity classes". *SIAM J. Comp.* 13 (1984), 461–487.

M. Garey, D. Johnson: *Computers and intractability: a guide to the theory of NP-completeness.* Freeman 1979.

J. Gill: "Computational complexity of probabilistic Turing machines". *SIAM J. Comp.* 6 (1977), 675–695.

J. Hartmanis: "On sparse sets in $NP - P$". *Inf. Proc. Lett.* 16 (1983), 55–60.

L. Hemachandra: "Counting in Structural Complexity Theory". Ph.D. dissertation, Cornell University, Tech. Rep. 87-830 (1987).

J. Hopcroft, J. Ullman: *Introduction to automata theory, languages, and computation.* Addison-Wesley 1979.

R. Karp, R. Lipton: "Some connections between nonuniform and uniform complexity classes". 12th ACM STOC (1980), 302–309.

J. Kadin: "Restricted Turing reducibilities and the structure of the polynomial time hierarchy". Ph.D. dissertation, Cornell University (1988).

J. Kämper: "Non-uniform proof systems: a new framework to describe non-uniform and probabilistic complexity classes". Univ. Oldenburg Tech. Rep. 3/87 (1987).

T. Long: "Strong nondeterministic polynomial time reducibilities". *Theor. Comp. Sci.* 21 (1982), 1-25.

T. Long, A. Selman: "Relativizing complexity classes with sparse oracles". *J. ACM* 33 (1986), 618–628.

N. Pippenger: "On simultaneous resource bounds". 20th IEEE FOCS (1979), 307–311.

D. Russo: "Structural properties of complexity classes". Ph.D. dissertation, Univ. California Santa Barbara (1985).

U. Schöning: "A low and a high hierarchy within NP^+". *J. Comp. Sys. Sci.* 27 (1983), 14–28.

U. Schöning: "On small generators". *Theor. Comp. Sci.* 34 (1984), 337–341.

L. Stockmeyer: "The polynomial time hierarchy". *Theor. Comp. Sci.* 3 (1977), 1–22.

C. Yap: "Some consequences of nonuniform conditions on uniform classes". *Theor. Comp. Sci.* 27 (1983), 287–300.