On the Limitations of Locally Robust
Positive Reductions

Lane A. Hemachandra and Sanjay Jain

UNIVERSITY OF
ROCHESTER
COMPUTER SCIENCE

On the Limitations of Locally Robust

Positive Reductions

Lane A. Hemachandra[1] and Sanjay Jain[2]

Department of Computer Science

University of Rochester

Rochester, New York 14627, USA

July 6, 1989 .

Abstract

Polynomial-time positive reductions, as introduced by Selman, are by definition globally robust — they are positive with respect to all oracles. This paper studies the extent to which the theory of positive reductions remains intact when their global robustness assumption is removed.

We note that two-sided locally robust positive reductions — reductions that are positive with respect to the oracle to which the reduction is made — are sufficient to retain all crucial properties of globally robust positive reductions. In contrast, we prove absolute and relativized results showing that one-sided local robustness fails to preserve fundamental properties of positive reductions, such as the downward closure of NP.

1 Introduction

In this paper we study the relative powers of different positive reducibilities. Informally a reduction is positive if converting some NO answers to YES does not cause a previously accepted string to be rejected.

Selman, in his seminal paper [Sel82b], defines and considers the properties of polynomial-time positive reductions. His positive reductions are by definition globally robust in the positivity.

An oracle machine, or a set of oracle machines, is said to robustly have a property P if it has property P for all oracles. Recent work on the power of robustness [Sch85, BI87, HH87, Ko87, Tar87] suggests that global robustness is a strong restriction. For example, it is known that if two nondeterministic machines N_1 and N_2 are robustly complementary — that is, complementary for every oracle — then for all oracles A, $L(N_1^A) \in P^{A \oplus NP}$ [HH87]. This, and the desire to broaden the domain of application of Selman's techniques, motivate us to relax the global robustness restriction.

Accordingly, we introduce three notions of locally robust polynomial-time positive reductions. We show that the Turing versions of these reducibilities differ. However, our ability to distinguish among the truth-table versions of these reducibilities depends on the structure of NP. In particular, we show that if $P=NP$ then these polynomial-time locally robust truth-table reducibilities are the same. However, if there exist uniformly $\log^*$-sparse tally sets in $NP-P$, then the reducibilities differ.

We study the extent to which the theory of positive reductions, as studied by Selman, remains intact for locally robust reductions. We prove results identifying the crucial properties of positive reductions required to obtain the results of [Sel82b]. One reason for introducing new reducibilities is that it is more likely that a set A reduces to B by locally positive reductions than by globally positive reductions. Our results thus enrich the domain in which Selman's techniques can be applied.

2 Notations

Let $\mathcal{N}$ denote the set of natural numbers. Σ is an alphabet set, usually $\{0,1\}$. A language is a subset of Σ^*. $\emptyset$ denotes the empty set. $M_0, M_1, \ldots$ denotes some standard enumeration of polynomial-time deterministic Turing machines. $N_1, N_2, \ldots$ denotes some standard enumeration of polynomial-time nondeterministic Turing machines. We assume that the running times of machine M_i (N_i) is bounded by deterministic (nondeterministic)

time $n^i + i$. P denotes the class of all languages accepted by some polynomial-time deterministic Turing machine [HU79]. NP denotes the class of languages accepted by some polynomial-time nondeterministic Turing machine and $coNP$ denotes the class of languages whose complement is in NP [HU79]. $L(M)$ denotes the language accepted by the machine M. E and NE denote respectively the class of languages accepted by exponential time deterministic and nondeterministic Turing machines; that is, $E = \bigcup_{c>0} DTIME[2^{cn}]$ and $NE = \bigcup_{c>0} NTIME[2^{cn}]$. $L(M^A)$ denotes the language accepted by the oracle machine M with the oracle A [HU79].

$A \leq_T B$ means there exists a machine M such that $A = L(M^B)$. $\leq_T^P$ denotes polynomial-time Turing reduction. $\leq_{tt}$ and $\leq_m$ similarly denote truth-table and many-one reductions. $P_r(A)$ denotes the class of languages r-reducible to A in polynomial time (see [BK88]). A tally language is a subset of 1^*. $\overline{A}$ denotes the complement of A, i.e., $\Sigma^* - A$. χ_A denotes the characteristic function of A. We sometimes denote a string x of length n by $x_1 x_2 \ldots x_n$, where x_i is the ith character of x. $|x|$ denotes the length of x. $A^{\leq n}$ denotes the set of strings in A with length at most n.

3 Polynomial-time positive reducibilities

In this section, we review Selman's notion of positive reducibility, which by definition is globally robust, and we introduce new notions of locally robust positive reducibility.

Positive reducibility was first studied for polynomial-time truth-table reductions in [LLS75]. Selman, in [Sel82b], extended the definition to Turing reductions. We first give the definition of globally positive reducibility due to Selman.[1]

Definition 3.1 *[Sel82b, Sel82a] A query machine M is globally positive if $(\forall x)(\forall A.B)[x \in L(M^B) \Rightarrow x \in L(M^{A \cup B})]$.*

Intuitively, a machine M is positive if converting some "no" answers to "yes" answers does not make the machine reject a previously accepted string. Moreover, this property holds for all oracles given to the machine (hence the term *globally* positive). Positive reducibility can now be defined using these globally positive machines.

Definition 3.2 *[Sel82b, Sel82a] $A \leq_{pos}^P C$ if $A \leq_T C$ by some polynomial-time, globally positive Turing machine M.*

---

[1] This reducibility is simply referred to as "positive" in [Sel82b]. However, we'll refer to it throughout this paper as "globally positive" in order to distinguish it from the locally positive reducibilities we define.

The conditions placed here on positive Turing reductions are analogous to those in the definition of positive truth-table reductions in [LLS75], which defined globally positive truth-table reductions.

Definition 3.3 *[LLS75] $A \leq_{ptt}^P C$ if $A \leq_T C$ by some polynomial-time, globally positive machine M, and there is a polynomial-time computable function $f : \{0,1\}^* \to \{c,0,1,\}^*$ such that M on input x makes queries only from the list $f(x)$ (here c acts as a separator of elements of the list). M above can be equivalently represented by a polynomial time evaluator e such that for all oracles C, $e(x, \chi_C(y_1), \chi_C(y_2) \ldots) = M^C(x)$, where $y_1, y_2, \ldots$ are the elements in the list $f(x)$.*[2]

Definition 3.4 *[LLS75] $A \leq_{pbtt}^P C$ if $A \leq_T C$ by some polynomial-time, globally positive machine M, and a polynomial-time computable function $f : \{0,1\}^* \to \{c,0,1,\}^*$ such that M on input x makes queries only from the list $f(x)$. Moreover the number of elements in the list $f(x)$ is bounded by some constant independent of x. M above can be equivalently represented by a polynomial time evaluator e such that for all oracles C, $e(x, \chi_C(y_1), \chi_C(y_2), \ldots) = M^C(x)$, where $y_1, y_2, \ldots$ are the elements in the list $f(x)$.*

The above definitions require global robustness; given *any* oracle A, $L(M^A)$ must never decrease when A is increased in *any* way. Note that all $\leq_m^P$ reductions are globally positive. However, global robustness is a strong restriction on Turing transducers. Machines exhibiting global robustness are known, in other contexts, to be weak [BI87, HH87, Tar87].

A more moderate definition of "positive" might require a reduction to be robust *only with respect to the particular set to which the reduction is being made*. We introduce three notions of locally robust positive reductions. In these definitions we require the machine to be robust only with respect to the oracle to which the reduction is made.

Definition 3.5 *A query machine M is locally right positive with respect to B if $(\forall x)(\forall A)[x \in L(M^B) \Rightarrow x \in L(M^{A \cup B})]$.*

Intuitively, M is locally right robust with respect to B if converting some "no" answers from the oracle B to "yes" answers does not make the machine reject a previously accepted string. Left robustness is just the other side of the above definition.

Definition 3.6 *A query machine M is locally left positive with respect to B if $(\forall x)(\forall A)[x \in L(M^{B-A}) \Rightarrow x \in L(M^B)]$ (or equivalently $(\forall x)(\forall A)[x \notin L(M^B) \Rightarrow x \notin L(M^{B-A})]$).*

---

[2]In [LLS75] the first argument of e is $\alpha(x)$, however without loss of generality we can take this to be x and let e do the (polynomial time) computations required to obtain α.

4

Definition 3.7 *A query machine M is locally right-left positive with respect to B if M is both right and left positive with respect to B.*

Locally robust reductions can now be defined with respect to reductions involving locally robust machines.

Definition 3.8 *$A \leq^P_{rpos} B$ if $A \leq_T B$ by some polynomial-time machine that is locally right positive with respect to B.*

Definition 3.9 *$A \leq^P_{lpos} B$ if $A \leq_T B$ by some polynomial-time machine that is locally left positive with respect to B.*

Definition 3.10 *$A \leq^P_{rlpos} B$ if $A \leq_T B$ by some polynomial-time machine that is locally right-left positive with respect to B.*

$\leq^P_{rlptt}, \leq^P_{rptt}, \leq^P_{lptt}, \leq^P_{rlpbtt}, \leq^P_{rpbtt}$ and $\leq^P_{lpbtt}$ reductions can be defined similarly.

4 Relationships between different polynomial-time positive reducibilities

In this section, we compare the relative power of different polynomial-time positive reducibilities. Clearly:

Proposition 4.1 *$A \leq^P_s B \Rightarrow A \leq^P_{rls} B \Rightarrow [A \leq^P_{rs} B \bigwedge A \leq^P_{ls} B]$, where s is in $\{pos, ptt, pbtt\}$.*

We first consider the elementary properties of the reductions. The following proposition is easy to prove.

Proposition 4.2

1. *$A \leq^P_{pos} B$ and $B \leq^P_{pos} C \Rightarrow A \leq^P_{pos} C$.*

2. *$A \leq^P_{rpos} B$ and $B \leq^P_{rpos} C \Rightarrow A \leq^P_{rpos} C$.*

3. *$A \leq^P_{lpos} B$ and $B \leq^P_{lpos} C \Rightarrow A \leq^P_{lpos} C$.*

4. *$A \leq^P_{rlpos} B$ and $B \leq^P_{rlpos} C \Rightarrow A \leq^P_{rlpos} C$.*

Results similar to those of Proposition 4.2 can also be proved for bounded truth-table and truth-table reductions.

Proposition 4.3 $A \leq^P_{lpos} B \Rightarrow \overline{A} \leq^P_{rpos} \overline{B}$.

Proof: Let $A \leq^P_{lpos} B$ via M. Let M_1 be such that $M_1^C(x) = 1$ iff $M^{\overline{C}}(x) = 0$. Clearly $x \in L(M_1^{\overline{B}}) \Leftrightarrow x \notin L(M^B)$. Thus M_1 reduces $\overline{A}$ to $\overline{B}$. If $C \supseteq \overline{B}$ and $x \in \overline{A}$ (and thus $\overline{C} \subseteq B$ and $x \notin A$) then $x \notin L(M^{\overline{C}})$, since M is locally left positive, and thus $x \in L(M_1^C)$. So M_1 is locally right positive. ∎

A similar proof can be used for $\leq^P_{pos}, \leq^P_{rpos}, \leq^P_{rlpos}, \leq^P_{ptt}, \leq^P_{lptt}, \leq^P_{rptt}, \leq^P_{rlptt}$, yielding the following result.

Proposition 4.4

1) $A \leq^P_{rpos} B \Rightarrow \overline{A} \leq^P_{lpos} \overline{B}$.

2) $A \leq^P_{rlpos} B \Rightarrow \overline{A} \leq^P_{rlpos} \overline{B}$.

3) (implicit in [Sel82b]) $A \leq^P_{pos} B \Rightarrow \overline{A} \leq^P_{pos} \overline{B}$.

4) $A \leq^P_{lptt} B \Rightarrow \overline{A} \leq^P_{rptt} \overline{B}$.

5) $A \leq^P_{rptt} B \Rightarrow \overline{A} \leq^P_{lptt} \overline{B}$.

6) $A \leq^P_{rlptt} B \Rightarrow \overline{A} \leq^P_{rlptt} \overline{B}$.

7) ([LLS75], Proposition 3.1(v)) $A \leq^P_{ptt} B \Rightarrow \overline{A} \leq^P_{ptt} \overline{B}$.

We now consider the relative power of different locally robust positive reductions. Selman showed that globally robust positive Turing reductions are more powerful than globally robust positive truth-table reductions.

Theorem 4.5 [Sel82b] There exist recursive sets A and B such that $A \leq^P_{pos} B$ but $A \nleq^P_{tt} B$.

Also, it is easy to see as a corollary of previous work on disjunctive reductions that (i) there exist recursive sets A and B such that $A \leq^P_{ptt} B$ but $A \nleq^P_{btt} B$, and (ii) there exist recursive sets A and B such that $A \leq^P_{pbtt} B$ but $A \nleq^P_m B$ [LLS75].

Though locally robust positive reductions are in general more flexible than globally robust positive reductions, the following theorem shows that local robustness does not add extra power for the special case of positive *bounded* truth-table reductions.

Theorem 4.6 For all $A, P_{pbtt}(A) = P_{rpbtt}(A) = P_{lpbtt}(A)$.

Proof: Let $B \leq^P_{rpbtt} A$ via M. Let $f(x)$ be the polynomial time computable list such that M, on input x, makes queries only from the list $f(x)$. Let e be the evaluator equivalent to

M (as in the definition of $\leq^P_{pbtt}$ reduction). Recall that this means that the size of list $f(x)$ is bounded by some constant c and e is positive with respect to A. Thus if $f(x) = x_1, x_2, \ldots x_c$, $\chi_A(x_i) = b_i$ and $e(x, b_1, \ldots b_c) = 1$, then converting some of b_i from 0 to 1 does not make e evaluate to 0. To make a globally robust reduction from B to A we need to convert this e to e' that is positive with respect to all oracles. We do this by converting some evaluation of e from 1 to 0.

Let $e'(x, b_1, \ldots, b_c) = 1$ iff $(\forall d_1, \ldots, d_c, b_j = 1 \Rightarrow d_j = 1) [e(x, d_1, \ldots, d_c) = 1]$ (Example: if $c = 2, e(x, 0, 0) = 1$, $e(x, 0, 1) = 1$, $e(x, 1, 0) = 0$ and $e(x, 1, 1) = 1$ then we replace e by e', where $e'(x, 0, 0) = 0$, $e'(0, 1) = 1$, $e'(x, 1, 0) = 0$ and $e'(x, 1, 1) = 1$). This makes e' globally positive, and does not effect the reduction from B to A (since e was right positive with respect to A). Thus, f and e' form a positive bounded truth-table reduction from B to A. A similar proof can be used to show that $P_{lpbtt}(A) = P_{pbtt}(A)$. ∎

For **unbounded** truth-table reductions, the distinction between different positive reducibilities depends on the structure of NP, as shown by the following two theorems.

Theorem 4.7 *If* P=NP, *then for all* $A, P_{ptt}(A) = P_{rlptt}(A) = P_{rptt}(A) = P_{lptt}(A)$.

Theorem 4.8 *Let* $g(0) = 1$, $g(n + 1) = 2^{g(n)}$, $n > 0$. *If there exist tally sets in* $\bigcup_{c>0} NTIME[g^c(n)] - \bigcup_{c>0} DTIME[g^c(n)]$ *then there is a recursive set* A *such that* $P_{rptt}(A) - P_{lptt}(A) \neq \emptyset$ *and* $P_{lptt}(A) - P_{rptt}(A) \neq \emptyset$.

Proof (of Theorem 4.7): We prove that $P_{rptt}(A) = P_{ptt}(A)$. Proof for $P_{lptt}(A) = P_{ptt}(A)$ is similar. $P_{rlptt}(A) = P_{ptt}(A)$ follows from Proposition 4.1.

Let $B \leq^P_{rptt} A$ via M. Let $f(x)$ be the polynomial time computable list such that M, on input x, makes queries only from the list $f(x)$. Let e be the evaluator equivalent to M (as in the definition of $\leq^P_{ptt}$ reduction). We now proceed as in Theorem 4.6. Let $e'(x, b_1, b_2, \ldots, b_{p(n)}) = 1$ iff $[(\forall d_1, \ldots, d_{p(n)}, b_j = 1 \Rightarrow d_j = 1)[e(x, d_1, \ldots, d_{p(n)}) = 1]]$. Note that e' can be calculated in polynomial time if $NP=P$. Clearly, f and e' witness that $B \leq^P_{ptt} A$. ∎

Proof (of Theorem 4.8): We only prove that $(\exists A)[P_{rptt}(A) - P_{lptt}(A) \neq \emptyset]$. The proof can be easily modified to show that $[(\exists A)[P_{rptt}(A) - P_{lptt}(A) \neq \emptyset \wedge P_{lptt}(A) - P_{rptt}(A) \neq \emptyset]]$.

Let N be a polynomial-time nondeterministic machine accepting a tally language $L \subseteq \{1^{g(k)} : k \in \mathcal{N}\}$ which is not in P (the existence of such a machine follows from the

assumption that there exist tally sets in $\bigcup_{c>0} NTIME[g^c(n)] - \bigcup_{c>0} DTIME[g^c(n)]$, by the techniques of [HIS85][3]).

W.l.o.g., let all certificates of $x \in L$ be of length $|x|^j + j$ and w.l.o.g., $0^{|x|^j+j}$ is never such a certificate. Let r be the polynomial-time predicate associated with N and L, i.e., $r(x,y) = 1$ iff y is a certificate for x. Let $e(x, y_1, \ldots y_{n^j+j}) = 1 - r(x,y)$, where $y = y_1 \ldots y_{n^j+j}$. Let $plus(a, j)$ be the string j greater than a in standard lexicographical order; e.g., $plus(1010, 3) = 1101$. Let c be the separation character from the definition of f (see Definitions 3.3 and 3.4). Let $f(x) = plus(x, 1) \, c \, plus(x, 2) \, c \ldots c \, plus(x, |x|^j + j)$. Clearly, functions f and e are computable in polynomial time.

A will be defined so that e is locally right positive. Also all strings not of the form $x, plus(x, 1), plus(x, 2), \ldots, plus(x, |x|^j + j)$, where $x \in \{1^{g(k)} : k \in \mathcal{N}\}$, are not in A. $\chi_A(plus(x, 1)) \ldots \chi_A(plus(x, |x|^j + j))$ will be $0^{|x|^j+j}$ if $x \notin L$, and otherwise will be a certificate of the fact that $x \in L$. Let R_i be the requirement that $M_i : \overline{L} \not\leq^P_{lptt} A$, that is, M_i does not $\leq^P_{lptt}$ reduce $\overline{L}$ to A. Below A_s denotes the strings of A determined before stage s. Go to stage 0.

Stage s
1. Let x be the least element in $\{1^{g(k)} : k \in \mathcal{N}\}$ not considered until this stage.
2. Let i be the least requirement not satisfied until now.
3. Let e_i, f_i be the evaluator and set calculator (as in the definition of positive truth-table reducibility) for the truth-table reducer M_i
4. If $x \notin L$ then let $A_{s+1} = A_s$.
5. Else If $(\exists z)[z$ is a certificate for x and $e_i(x, \chi_{D(z)}(q_1), \chi_{D(z)}(q_2), \ldots) = 1]$,
   where $f_i(x) = q_1 c q_2 \ldots$ and $D(z) = A_s \bigcup \{plus(x, i) : z_i = 1\}$,
   then let y be least such certificate z. Set $A_{s+1} = A_s \bigcup \{plus(x, i) : y_i = 1\}$.
   (Note that here R_i is satisfied)
6. Else
   Let y be the least certificate for x.
   Let $A_{s+1} = A_s \bigcup \{plus(x, i) : y_i = 1\}$.
end stage s.

It is clear that $\overline{L} \leq^P_{rptt} A$ via the functions f and e. This is because when $x \in \overline{L}$, then $e(x, y) = 1$ for *any* length $|x|^i + i$ string y; so even if A has some strings added—and the

---

[3]Though recent work by Allender [All] has corrected parts of [HIS85], the techniques of [HIS85] as used here are correct.

"address" $\chi_A(plus(x, 1)) \cdots \chi_A(plus(x, |x|^i + i)) = 0^{|x|^i + i}$ thus has some zeros corrupted to ones—$e(x,$ corrupted address) will nonetheless accept.

Now consider the following cases:

case 1: All requirements are satisfied.

Clearly, $\overline{L} \not\leq^P_{lptt} A$.

case 2: R_i is the least requirement not satisfied.

In this case we show that $L \in P$. Let n be so large that $2^{n/10} > n^i + i$, and all the smaller requirements have been satisfied before stage s, $n > g(s)$. Clearly, when $m \in \{g(k) : k \in \mathcal{N}\}$, then $A^{\leq m-1}$ can be determined in time polynomial in m (by just going through all possible certificates). Now for $x \in \{1^{g(k)} : k \in \mathcal{N}\}, |x| > n$. we have $x \notin L \Rightarrow e_i(x, \chi_{A^{\leq |x|-1}}(q_1), \chi_{A^{\leq |x|-1}}(q_2), \ldots) = 1$ (since $A^{\leq |x|-1} = A^{\leq |x|^i + j}$ due to step 4 of the construction, and M_i reduces $\overline{L}$ to A). And similarly, we have $x \in L \Rightarrow e_i(x, \chi_{A^{\leq |x|-1}}(q_1), \chi_{A^{\leq |x|-1}}(q_2), \ldots) = 0$ (since $A^{\leq |x|-1} \subseteq A^{\leq |x|^i + i}$ and $\overline{L} \leq^P_{lptt} A$ via M_i). This gives us a polynomial-time decision procedure for L contrary to the assumption.

Thus all requirements are satisfied. ∎

Note that the above proof can also be used to distinguish between $\leq^P_{rptt}$ and $\leq^P_{lpos}$ reductions, under the same assumption.

We now consider the relationship between various positive Turing reducibilities.

Theorem 4.9 $(\exists A)[P_{rpos}(A) - P_{lpos}(A) \neq \emptyset \bigwedge P_{lpos}(A) - P_{rpos}(A) \neq \emptyset]$.

Corollary 4.10 $(\exists A)[P_{rpos}(A) - P_{rlpos} \neq \emptyset \bigwedge P_{lpos}(A) - P_{rlpos}(A) \neq \emptyset]$.

Proof: Let $g(0) = 1, g(n+1) = 2^{g(n)}$. Consider the following languages:

$L_A = \{1^n : n = g(k) \text{ for some even } k \bigwedge 1^n b_0 b_1 b_2 \ldots b_{n-1} \in A \text{ where } b_j = \chi_A(0^{n+j})\}$.

$L'_A = \{1^n : n = g(k) \text{ for some odd } k \bigwedge 1^n b_0 b_1 b_2 \ldots b_{n-1} \in A \text{ where } b_j = \chi_A(0^{n+j})\}$.

To ensure that $L_A \leq^P_{rpos} A$ it suffices to construct A so that for all n of form $g(2k)$, for $b_j = \chi_A(0^{n+j})$, we have $[[1^n b_0 b_1 \ldots b_{n-1} \in A] \bigwedge [b_j = 1 \Rightarrow d_j = 1]] \Rightarrow [1^n d_0 d_1 \ldots d_{n-1} \in A]$. Thus an oracle machine M^B which accepts 1^n iff $n = g(k)$ for some even k and $1^n a_0 a_1 \ldots a_{n-1} \in B$, where $a_i = \chi_B(0^{n+i})$ witnesses that $L_A \leq^P_{rpos} A$.

Similarly, $L'_A \leq^P_{lpos} A$ is ensured if for all n of form $g(2k+1)$, for $b_j = \chi_A(0^{n+j})$, $[[1^n b_0 b_1 \ldots b_{n-1} \notin A] \bigwedge [b_j = 0 \Rightarrow d_j = 0]] \Rightarrow [1^n d_0 d_1 \ldots d_{n-1} \notin A]$.

We now construct A in stages. A will satisfy the conditions above so that $L_A \leq^P_{rpos} A$ and $L'_A \leq^P_{lpos} A$.

At stage s we decide the membership in A of strings of length $g(s), \ldots, g(s+1) - 1$. We always assume that strings not of the form $0^{g(k)+i}, i < g(k)$ or $1^{g(k)}\{0,1\}^{g(k)}$, are not in A (without explicitly mentioning it below).

Let R_{2i} be the requirement that $M_i : L_A \not\leq^P_{lpos} A$, that is, M_i does not $\leq^P_{lpos}$ reduce L_A to A. Let R_{2i+1} be the requirement that $M_i : L'_A \not\leq^P_{rpos} A$, that is, M_i does not $\leq^P_{rpos}$ reduce L'_A to A. Note that if all the requirements are satisfied then $L_A \not\leq^P_{lpos} A$ and $L'_A \not\leq^P_{rpos} A$. Below A_s denotes the strings of A determined before stage s. Go to stage 0.

Stage $2s$

1. Let R_{2i} be the least unsatisfied even requirement.
2. Let $n = g(2s)$.
3. If $2^{n/10} \leq n^i + i$ then exclude from A all strings of length l,
   $$g(2s) \leq l < g(2s+1).$$
4. Else If

   $M_i^A(1^n)$ rejects when all new questions x (i.e., those not decided in A_{2s}) are answered as :

   > If x is of form $1^n\{0,1\}^n$ then YES.
   > If x is of form 0^{n+i} then NO.

   then let A_{2s+1} be such that all strings of form $1^n\{0,1\}^n \in A_{2s+1}$ and all other strings of length $l, g(2s) \leq l < g(2s+1)$ not in A_{2s+1}.

   > (Note that in this case R_{2i} is satisfied.)
5. Else

   Let S be the set of questions of form $1^n\{0,1\}^n$ asked in the computation by M_i in step 4 above.

   For all $x \in S$, let $x \in A$.

   If x is of form $1^n\{0,1\}^n$ and $x \notin S$ then let $x \notin A$.

   Let y be a question of form $1^n\{0,1\}^n$ not asked by M_i (there exists such a y).

   Let $0^{n+i} \in A \Leftrightarrow y_{n+i+1} = 1$ for $i < n$.

   > (Note that on this A, M_i either accepts incorrectly or rejects.
   > In the latter case, since $A \supseteq A_{2s} \bigcup S$
   > on which M_i accepts, M_i is not a $\leq^P_{lpos}$ reduction.
   > Either way, R_{2i} is satisfied.)

end stage $2s$

Stage $2s + 1$ is similar:

<u>Stage $2s + 1$</u>

6. Let R_{2i+1} be the least unsatisfied odd requirement.

7. Let $n = g(2s + 1)$.

8. If $2^{n/10} \le n^i + i$ then exclude from A all strings of length l,

   $$g(2s + 1) \le l < g(2s + 2).$$

9. Else If

   $M_i^A(1^n)$ accepts when all new questions x (i.e., those not decided in A_{2s})
   are answered as :

   > If x is of form $1^n\{0,1\}^n$ then NO.

   > If x is of form 0^{n+i} then YES.

   then let A_{2s+1} be such that all strings of form $1^n\{0,1\}^n \not\subseteq A_{2s+2}$ and all
   strings of form 0^{n+i}, $i < n$ in A_{2s+2}.

   > (Note that in this case R_{2i+1} is satisfied.)

10. Else

    Let S be the set of questions of form $1^n\{0,1\}^n$ asked in the above
    computation by M_i.

    For all $x \in S$, let $x \notin A$.

    If x is of form $1^n\{0,1\}^n$ and $x \notin S$ then let $x \in A$.

    Let y be a question of form $1^n\{0,1\}^n$ not asked by M_i (there exists such a y).

    Let $0^{n+i} \in A \Leftrightarrow y_{n+i+1} = 1$ for $i < n$.

    > (Note that on this A, either M_i rejects incorrectly, or M_i accepts.
    > In the latter case, since
    > $$A \subseteq A_{2s} \bigcup \{0^{n+i} : i < n\} \bigcup \{1^n z : |z| = n, z \notin S\}$$
    > on which M_i rejects , M_i is not a $\le_{rpos}^P$ reduction. In either case,
    > R_{2i+1} is satisfied.)

<u>end stage $2s + 1$</u>

Let $M^B(1^n) = 1$ iff $n = g(2k)$ for some k and $1^n b_0 b_1 \ldots b_{n-1} \in B$, where $b_j = 1$ iff
$0^{n+j} \in B$. Clearly, $L_A \le_{rpos}^P A$ via M (since 1^n is placed in L_A only in step 4 in which case
all strings of form $1^n z, |z| = n$ are also placed in A). Similarly $L'_A \le_{lpos}^P A$. We claim that
$L_A \not\le_{lpos}^P A$. Suppose by way of contradiction that $L_A \le_{lpos}^P A$ via M_i. Also let M_i be the
least such machine. Then for sufficiently large s in stage $2s$, $2^{n/10} > n^i + i$ and all smaller
even requirements have been satisfied. Thus at this stage by construction M_i will be fooled.
Thus no such machine can exist. It can be similarly shown that $L'_A \not\le_{rpos}^P A$. This proves
the theorem. ∎

Whether $\le_{pos}^P$ and $\le_{rlpos}^P$ are different is at present an open problem.

5 Basic properties of reductions

In this section we consider some of the basic properties of positive reduction in NP. Selman, in [Sel82b], showed that NP is closed downward under globally robust positive Turing reductions. We show that, though Selman's techniques suffice to prove that NP is closed downwards under two of the locally robust reductions, the remaining locally robust reduction fails to leave NP closed downwards in some relativized worlds. As a corollary, we note that rpos and lpos reductions do not share the complementation property of globally robust positive reductions (Proposition 4.4, part 3).

Theorem 5.1 NP *is closed downward under* $\leq^P_{lpos}$ *reductions.*

Corollary 5.2

  1. coNP *is closed downward under* $\leq^P_{rpos}$ *reductions.*

  2. NP *and* coNP *are closed downward under* $\leq^P_{rlpos}$ *reductions.*

  3. *[Sel82b]* NP *and* coNP *are closed downward under* $\leq^P_{pos}$ *reductions.*

Proof: Let $A \leq^P_{lpos} B, B \in NP$. We give an NP algorithm for A. Let $A \leq^P_{lpos} B$ via M.

On input x

    Simulate M, guessing answers for each questions asked.

    Verify the answers guessed YES.

    Accept iff M accepts.

Now if $x \in A$, then there exists a sequence of right guesses by which the above algorithm accepts.

Now we consider the case in which $x \notin A$. Clearly the guessed oracle for which the above algorithm simulates M is a subset of B. Since $x \notin M^B$, $x \notin M^C$ for all $C \subseteq B$ (since M is locally left positive with respect to B). Thus the above algorithm does not accept x. ∎

However the proof does not work for right positive reductions. We give a relativized world in which NP is not closed downward under locally right robust positive reductions.

Theorem 5.3 *There is a recursive oracle* B *such that* NP^B *is not closed downward under* $\leq^P_{rpos}$ *reductions. That is, there are recursive sets* A, B *and* C *such that* $C \leq^P_{rpos} A$, $A \in NP^B$, *and* $C \notin NP^B$.

Proof: This proof is similar to the proof of Theorem 4.9 . Let $g(0) = 1, g(n + 1) = 2^{g(n)}$. We will define sets A and B. Let $A = \{x : (\exists y)|y| = |x|, xy \in B\}$. Clearly $A \in NP^B$. Let $L_A = \{1^n : n = g(k)$ for some k and $1^n b_0 b_1 \ldots b_{n-1} \in A$ where $b_j = 1 \Leftrightarrow 0^{n+j} \in A\}$. If $[1^n b_0 b_1 \ldots b_{n-1} \in A \wedge b_j = 1 \Rightarrow d_j = 1] \Rightarrow [1^n d_0 d_1 \ldots d_{n-1} \in A]$, where $b_j = 1 \Leftrightarrow 0^{n+j} \in A$. then $L_A \leq^P_{rpos} A$ (via machine M which with oracle B accepts 1^n iff $n = g(k)$ for some k and $1^n a_0 a_1 \ldots a_{n-1} \in B$, where $a_i = \chi_B(0^{n+i})$). We will construct B so that A satisfies the above property. In addition we will ensure that $L_A \notin NP^B$. Taking $C = L_A$ proves the theorem.

Let R_i be the requirement that $L(N_i^B) \neq L_A$. A will contain strings of form $1^n z, |z| = n$ and $0^{n+i}, i < n$ where $n = g(k)$ for some k (this thus restricts some elements to be out of B. We assume that such elements are not in B without explicitly mentioning so). At stage s we decide the membership of strings of length l, $g(s) \leq l < g(s + 1)$ in A (and strings of length l, $2g(s) \leq l < 2g(s + 1)$ in B). Below B_s denotes the strings of B determined before stage s. Go to stage 0.

Stage s

1. Let R_i be the least unsatisfied requirement.
2. Let $n = g(s)$.
3. If $2^{n/10} \leq n^i + i$ then exclude from A all strings of length l, $n \leq l < 2^n$.
4. Else If $N_i^{B_s \cup \{0^{2(n+i)}: i \geq n/2\} \cup \{1^n z 0^{2n}: |z|=n, z \geq 0^{n/2} 1^{n/2}\}}(1^n)$ rejects then let
   $$B_{s+1} = B_s \cup \{0^{2(n+i)} : i \geq n/2\} \cup \{1^n z 0^{2n} : |z| = n, z \geq 0^{n/2} 1^{n/2}\}.$$
   (Note that in this case we have already fooled N_i, since $1^n \in L_A - L(N_i^B)$).
5. Else
   Fix an accepting path of $N_i^{B_s \cup \{0^{2(n+i)}: i \geq n/2\} \cup \{1^n z 0^{2n}: |z|=n, z \geq 0^{n/2} 1^{n/2}\}}(1^n)$
   Let S be the set of questions asked by N_i which are in
   $$\{0^{(n+i)} w : |w| = n + i\} \cup \{1^n z 0^{2n} : |z| = n, z \geq 0^{n/2} 1^{n/2}\}.$$
   Let $y, |y| = n, y \in \{0,1\}^{n/2} 1^{n/2}$ be such that $1^n y 0^{2n} \notin S$ (clearly, such a y exists).
   Let s_{n+i} be a string of length $n + i$ such that $0^{(n+i)} s_{n+i} \notin S$.
   Let $B_{s+1} = B_s \cup \{w : w \in \{0^{2(n+i)} | i \geq n/2\}$
   $\cup \{1^n z 0^{2n} : |z| = n, z \geq 0^{n/2} 1^{n/2}$ and $1^n z 0^{2n} \in S\}\} \cup \{0^{(n+i)} s_{n+i} : y_i = 1\}$.
   (Note that in this case $1^n \in L(N_i^B) - L_A$).
end stage s.

Clearly, $L_A \leq^P_{rpos} A$. If all requirements are satisfied then clearly $L_A \notin NP^B$. So assume that R_i is the least requirement not satisfied. But then let s be so large that $2^{n/10} > n^i + i$.

13

and all requirements less that i are satisfied by stage s. Then by construction R_i will be satisfied at stage s. Thus all the requirements are satisfied. ∎

Though $A \leq^P_{pos} B \Rightarrow \overline{A} \leq^P_{pos} \overline{B}$ (Proposition 4.4), the analog of this result fails for rpos and lpos reductions, as an immediate corollary of Theorems 5.1 and 5.3 and Corollary 5.2.

Corollary 5.4

1. *There exist recursive oracles A and B such that $A \leq^P_{rpos} B$ but $\overline{A} \not\leq^P_{rpos} \overline{B}$.*

2. *There exist recursive oracles A and B such that $A \leq^P_{lpos} B$ but $\overline{A} \not\leq^P_{lpos} \overline{B}$.*

6 P-selectivity and positive reductions

Selman, in [Sel79], introduced the notion of P-selectivity. Intuitively, A is P-selective if given two strings x and y, a polynomial-time function can determine which of x or y is more "likely" to be an element of A.

Definition 6.1 *[Sel79] A is P-selective if there exists a polynomial-time computable function f such that:*

1. *$(\forall x, y) f(x, y) \in \{x, y\}$, and*

2. *$x \in A \lor y \in A \Rightarrow f(x, y) \in A$.*

Selman [Sel82b] showed that if $A \leq^P_{pos} \overline{A}$ and A is P-selective then A is in P. Selman's proof can be easily seen to generalize to the following:

Theorem 6.2 *$A \in P$ if and only if $A \leq^P_{rlpos} \overline{A}$, and A is P-selective.*

We leave it as an open problem whether $\leq^P_{rpos}$ or $\leq^P_{lpos}$ reducibility suffice to obtain the above theorem. Below, we show that weak P-selectivity does not suffice.

Ko [Ko83] generalized Selman's notion of P-selectivity.

Definition 6.3 *[Ko83]*

A preorder R on Σ is partially polynomial-time computable if there is a polynomial-time computable function f such that

1. *$f(x, y) = f(y, x) = x$ if xRy but not yRx,*

2. *$f(x, y) = f(y, x) \in \{x, y\}$ if xRy and yRx, and*

3. *$f(x, y) = \#$ otherwise.*

Let xSy if and only if xRy and yRx. Let R' be an induced ordering on Σ^*/S, i.e., $\overline{x}R'\overline{y}$ iff xRy, where $\overline{x}$ denotes the equivalence class of x under the relation S.

Definition 6.4 *[Ko83] A partial ordering R is p-linear if for all n, the set $\Sigma_n = \{x : |x| \leq n\}$ can be decomposed into at most $p(n)$ many pairwise disjoint sets $B_1, \ldots B_m, m \leq p(n)$. for some polynomial p such that:*

    1. If x and y are in the same set B_i then $xRy \vee yRx$, and

    2. if x and y are in two different sets then neither xRy nor yRx.

Definition 6.5 *[Ko83] A is weakly P-selective if and only if there is a partially polynomial-time computable preorder R with the induced equivalence relation S and partial ordering R' such that*

    1. R' is p-linear, and

    2. for all n, $A_n = \{x \in A : |x| \leq n\}$ is the union of initial segments of R' chains in Σ_n.

In contrast to Theorem 6.2 we show that:

Theorem 6.6 *There exists recursive oracle Q and a recursive set A such that A is weakly P^Q-selective, $A \leq_{pos}^P \overline{A}$ but $A \notin P^Q$.*

Proof: We will define A and Q in the following. Q will act as a weak P-selector for A. Thus A will be trivially weakly P^Q-selective.

Let $g(0) = 1$, $g(n+1) = 8^{g(n)}$. A and Q will be such that

1) $A \subseteq S$ where $S = [\{1^{g(n)} : n \in \mathcal{N}\} \cup \{1^{g(n)}0^k : n \in \mathcal{N} \wedge 0 < k \leq 2g(n)\} \cup \{1^{4g(n)}y : n \in \mathcal{N} \wedge |y| = 1 + g(n)\}]$.

2) $1^{g(n)}0^{2k+1} \in A \Leftrightarrow 1^{g(n)}0^{2k+2} \notin A$, for $k < g(n)$.

3) for $|y| = g(n)$, $1^{4g(n)}y1 \in A \Leftrightarrow 1^{4g(n)}y0 \notin A$.

4) $1^{g(n)} \in A \Leftrightarrow 1^{4g(n)}y1 \in A$ where $y = \chi_A(1^{g(n)}0^2)\chi_A(1^{g(n)}0^4)\ldots\chi_A(1^{g(n)}0^{2g(n)})$.

For partial ordering R we have

5) $B_n = \{x : |x| = n\}$ (for B_i in the definition of p-linear partial ordering).

6) $\langle x, y \rangle \in Q$ if and only if $|x| = |y|$ and xRy.

Clearly, $A \leq_{pos}^P \overline{A}$ and A is weakly P^Q-selective.

Following construction diagonalizes to ensure that every P^Q machine fails to accept A. Assume without loss of generality that M_i^Q queries only strings of form $\langle x, y \rangle$ such that

15

$|x| = |y|, x \in S$ and $y \in S$. At stage s we determine the membership in A for all strings of length $l, g(s) \le l < g(s+1)$. We also define Q on all pairs of strings of length between $g(s)$ and $g(s+1)$. We explicitly give the membership in A only for strings in S. Also we define the relation R only for strings in S which are of same length. A and R on other values can be predetermined using (1) and (5) above.

Let R_i be the requirement that $L(M_i^Q) \ne A$. Go to stage 0.

Stage s

    1. Let $x = 1^{g(s)}$.

    2. Let i be the least requirement not satisfied until now.
       Let $n = g(s)$.

    3. If $n^i + i \ge 8^{n/10}$ then let $1^n \notin A$, $1^n 0^{2k} \in A, k \le n$,
       and $1^{4n} y1 \notin A$ for all $y, |y| = n$.
       Define Q in some way consistent with A.

    4. Else run M^i on 1^n. Answering all questions $\langle z, y \rangle$ in the following way.
       If $5g(s) + 1 = |z| = |y|$, then let $z = uc, y = wr; r, c \in \{0, 1\}$.
       Let $u0, w0 \in A$ and $u1, w1 \notin A$.
       Answer the question in a way consistent with the previous answers
       and A determined until now.

    5. Let y be such that neither $1^{4n} y0$ nor $1^{4n} y1$ has appeared
       in any query until now. Let $\chi_A(1^n 0^2) \chi_A(1^n 0^4) \ldots \chi_A(1^n 0^{2n}) = y$.

    6. Let $1^{4n} y1 \in A$ if and only if M rejected in the above simulation.
       (Note that M_i has been fooled in this stage)

end stage s.

Clearly, $A \le_{pos}^P \overline{A}$. Also A is weak P^Q-selective. Suppose by way of contradiction that $M_i^Q = A$. Also let M_i be the least such machine. Then for sufficiently large s in stage s. $8^{n/10} > n^i + i$ and all smaller requirements are satisfied. Thus at this stage by construction M_i will be fooled. Thus no such machine can exist. $\blacksquare$

Selman [Sel82b] showed that if $A \le_{pos}^P B$ ($B \ne \emptyset, B \ne \Sigma^*$) and B is P-selective, then there exists an algorithm that runs in time polynomial in the number of queries in the computation tree of the reducer and outputs a set I such that $x \in A \Leftrightarrow I \subseteq B$. We observe that Selman's proof holds even for $\le_{rlpos}^P$ reductions.

Proposition 6.7 *[implicit from techniques of [Sel82b]]* *Let $A \le_{rlpos}^P B$ via machine M. $B \ne \emptyset, B \ne \Sigma^*$. Let B be P-selective. Then there exists an algorithm that runs in time*

polynomial in the length of the input x and the number of queries in the computation tree of M on x, that outputs a set I such that $x \in A \Leftrightarrow I \subseteq B$.

Proof: We restate Selman's algorithm for completeness, modified for $\leq^P_{rlpos}$ reduction. Let f be a P-selector for B. Let $f'(x_1, x_2, \ldots, x_r) = f(x_1, (f(x_2, \ldots f(x_{r-1}, x_r) \ldots)))$. Let $a \in B$ and $b \notin B$.

On input x.
    1. If $M^\emptyset(x)$ accepts let $I = \{a\}$.
    2. If $M^{\Sigma^*}(x)$ rejects then $I = \{b\}$.
    3. Else

        Let $Q = T = \emptyset$

        Repeat

            I=T;

            Simulate M on x with oracle I.

            Q = set of all queries asked in the above computation.

            If M rejects then

            begin

                $u = f'(Q - I)$;

                $T = I \cup \{u\}$

            end

        Until $T = I$

Clearly, the above algorithm on input x outputs a set I such that $x \in A \Leftrightarrow I \subseteq B$ (by induction on the number of times the repeat loop is executed). Also since each time the repeat loop is executed cardinality of T increases at least by 1, we have that the number of times repeat loop is executed is at most the number of queries in the computation tree of M on input x. Also since each loop runs in time polynomial in the length of x and the number of queries in the computation tree of M on x, the whole algorithm runs in time polynomial in $|x|$ and the number of queries in computation tree of M on input x. ∎

Corollary 6.8 *If $A \leq^P_{rlpos} B$, $B \neq \emptyset, B \neq \Sigma^*$ and B is P-selective then for some polynomial p, $A \leq_m B$ by a function g computable in time $2^{p(|x|)}$.*

Proof: Use the above algorithm to get I such that $x \in A \Leftrightarrow I \subseteq B$. Since $\overline{B}$ is P-selective, we select an element from I which is most likely to be in $\overline{B}$ (say y). Now $x \in A \Leftrightarrow y \in B$. ∎

It is easy to convert a $\leq_{rpos}^P$ computation tree to a $\leq_{rlpos}^P$ computation tree in exponential time. Thus we also have:

Corollary 6.9 *If* $A \leq_{lpos}^P B$, $B \neq \emptyset, B \neq \Sigma^*$ *and* B *is P-selective then for some polynomial* p, $A \leq_m B$ *by a function* g *computable in time* $2^{p(|x|)}$

Corollary 6.10 *If* $A \leq_{rpos}^P B$, $B \neq \emptyset, B \neq \Sigma^*$ *and* B *is P-selective then for some polynomial* p, $A \leq_m B$ *by a function* g *computable in time* $2^{p(|x|)}$

For *rlptt* reductions the number of queries in the computation tree is polynomial in the length of the input; thus we have:

Corollary 6.11 *If* $A \leq_{rlptt}^P B$, $B \neq \emptyset, B \neq \Sigma^*$ *and* B *is P-selective then* $A \leq_m^P B$.

Corollary 6.12 *If* A *is* $\leq_{rlptt}^P$ *self reducible and* A *is P-selective then* A *is in* P.

Proof: From corollary 6.11 we have A is $\leq_m^P$ self reducible. Since any $\leq_m^P$ self reducible set is in P, $A \in P$. ∎

Theorem 6.13 *[Sel82b] For every tally language* A *there exist sets* A *and* B *such that:*

1. $B \leq_{ptt}^P A \leq_T^P B$.
2. $C \leq_{tt}^P A \leq_T^P C$.
3. $B \leq_{ptt}^P C \leq_{tt}^P B$.
4. B *is P-selective, and*
5. C *P-selective* $\Rightarrow C \in P$.

As a corollary we obtain:

Corollary 6.14 *Let* A *be a tally language not in* P. *Then there exist* $\leq_T^P$ *equivalent sets* B, C *such that* $C \leq_{tt}^P B$ *but* $C \nleq_{rlptt}^P B$. *Also* $\overline{B} \nleq_{rlpos}^P B$.

Corollary 6.15 *If* $E \neq NE$ *then there exists sets* B *and* C *such that:*

1. $B \in NP - P$,
2. $B \leq_{ptt}^P C$,
3. $C \leq_{tt}^P B$,
4. $C \nleq_{rlptt}^P B$, *and*
5. $\overline{B} \nleq_{rlpos}^P B$.

18

Proof: Follows from the above theorem since if $E \neq NE$ then there are tally sets in $NP-P$ ([Boo74, BWSD78], see also [HIS85]) ∎

Corollary 6.16 *If $NE \neq E$ then there exists a $\leq_{tt}^P$ degree in NP that does not consist of single $\leq_{rlpos}^P$ degree.*

Corollary 6.17 *If $NE \neq E$ then there exists a $\leq_{rlptt}^P$ degree in NP which consists of a single $\leq_m^P$ degree.*

Corollary 6.18 *If $NE \cap coNE \neq E$ then there exists sets B and C in NP such that $B \leq_{ptt}^P C$ and $C \leq_{tt}^P B$ but $C \not\leq_{rlptt}^P B$.*

7   Conclusions and Open Problems

In this paper we defined locally positive reductions as more moderate versions of Selman's (globally) positive reductions. We compared the different locally positive polynomial-time Turing reductions and identified the properties required by positive reductions to obtain the results of Selman—thus enriching the domain in which his results are applicable and delimiting the boundaries of their application.

Some open problems arise out of our work. It is open at present whether there exist relativized worlds in which $\leq_{rlpos}^P$ and $\leq_{pos}^P$ are different. The construction of such worlds would show that rlpos reductions are indeed more flexible than pos reductions; in this paper, we proved that rpos and lpos reductions are mutually incomparable, and that each is more flexible than rlpos and pos reductions. Another open problem is whether Theorem 6.2 fails for $\leq_{rpos}^P$ or $\leq_{lpos}^P$ in some relativized world.

8   Acknowledgements

We thank William Gasarch for helpful comments on the paper.

References

[All]   E. Allender. Limitations of the upward separation technique. In *Automata, Languages, and Programming (ICALP 1989)*. Springer-Verlag *Lecture Notes in Computer Science* . To appear.

19

[Bl87]   M. Blum and R. Impagliazzo. Generic oracles and oracle classes. In *28th Annual IEEE Symposium on Foundations of Computer Science*, October 1987.

[BK88]   R. Book and K. Ko. On sets truth-table reducible to sparse sets. *SIAM Journal on Computing*, 17(5):903–919, 1988. own.

[Boo74]   R. Book. Tally languages and complexity classes. *Information and Control*, 26:186–193, 1974.

[BWSD78] R. Book, C. Wrathall, A. Selman, and D. Dobkin. Inclusion languages and the Berman-Hartmanis conjecture. *Mathematical Systems Theory*, 11:1–8, 1978.

[HH87]   J. Hartmanis and L. Hemachandra. One-way functions, robustness, and the non-isomorphism of NP-complete sets. In *Proceedings 2nd Structure in Complexity Theory Conference*, pages 160–174. IEEE Computer Society Press, June 1987.

[HIS85]   J. Hartmanis, N. Immerman, and V. Sewelson. Sparse sets in NP-P: EXPTIME versus NEXPTIME. *Information and Control*, 65(2/3):159–181, May/June 1985.

[HU79]   J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[Ko83]   K. Ko. On self-reducibility and weak P-selectivity. *Journal of Computer and System Sciences*, 26:209–221, 1983.

[Ko87]   K. Ko. On helping by robust oracle machines. *Theoretical Computer Science*, 52:15–36, 1987.

[LLS75]   R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–124, 1975.

[Sch85]   U. Schöning. Robust algorithms: A different approach to oracles. *Theoretical Computer Science*, 40:57–66, 1985.

[Sel79]   A. Selman. P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Mathematical Systems Theory*, 13:55–65, 1979.

[Sel82a]   A. Selman. Analogues of semirecursive sets. *Information and Control*, 52:36–51, 1982.

[Sel82b]   A. Selman. Reductions on NP and P-selective sets. *Theoretical Computer Science*, 19:287–304, 1982.

[Tar87]   G. Tardos. Query complexity, or why is it difficult to separate $NP^A \cap coNP^A$ from P^A by random oracles A. Manuscript, July 1987.