Twenty Questions to a P-selector

Harry Buhrman Leen Torenvliet Peter van Emde Boas

February 1, 1995

Abstract

We show in this note, that any set that is positive Turing reducible to a p-selective set is in fact many-one reducible to this set. Therefore, such a set is itself p-selective.

1 Introduction

Jockusch [Joc68] introduced in 1968 the notion of *semi-recursive* sets in an attempt to construct a solution to Post's program. Though Post's *problem* was already settled in 1957 by Friedberg [Fri57] and Muchnik [Muc56], Post's *program* aimed at finding a property for r.e. sets that would imply T-incompleteness, and subsequently finding an r.e. set that would have this property. It was already known, that hyperhypersimple sets were not Q-complete. Marchenkov [Mar76] showed, that Turing complete semi-recursive r.e. sets are Q-complete. Unfortunately, no hyperhypersimple set can be semi-recursive, as Martin [Mar63] demonstrated, so hyperhypersimplicity had to be refined. In the same paper [Mar76], Marchenkov showed that an η-hyperhypersimple set, for some r.e. relation η, is not Q-complete, and Degtev [Deg73] had constructed a nonrecursive, semi-recursive, and η-hyperhypersimple (in fact η-maximal) r.e. set. Post's program finally reached a conclusion.

The world of structural complexity theory found its origin in a suspected analogy between recursion theoretic concepts and the resource bounded version of these concepts, and many researchers hoped, that one day the analogy between recursive and r.e. sets on one hand, and deterministic and nondeterministic polynomial time computable sets on the other hand, could be stretched to a point where a significant difference between the latter two could be demonstrated. So far, as many papers state, these hopes have been in vain. On the way, many notions from recursion theory have been transferred to the world of complexity theory, and the flurry of notions that was originated by Post's program has not escaped this vogue.

A.L. Selman [Sel79] introduced both the resource bounded version of semi-recursiveness, called p-selectivity, and the notion of polynomial time positive reducibility (in [Sel82]), which translated Jockush's [Joc66] positive reducibility. In his paper [Sel82], Selman showed that the combination of positive truth table reducibility of a set A to a set B, and p-selectivity of B, implies many-one reducibility of A to B and hence p-selectivity of A. In this note, we extend this result to Turing reducibility, and thereby achieve considerable strengthening of many results in [Sel82].

2 Definitions and Notations

Let $\Sigma = \{0,1\}$. Strings are elements of Σ^*, and are denoted by small letters $x, y, u, v, \ldots$. For any string x, the length of x is denoted by $|x|$. Languages are subsets of Σ^*, and are denoted by capital letters $A, B, C, S, \ldots$. The complement of a language A in Σ^* is denoted by $\overline{A}$. For a set A the function χ_A will denote the characteristic function of A, i.e. $\chi_A(x) = 1$ if $x \in A$, and $\chi_A(x) = 0$ otherwise. We assume, that the reader is familiar with the standard Turing machine model, the oracle machine model, and other standard notions of complexity theory as can be found in [BDG88]. The language recognized by a Turing machine M (oracle machine M with oracle set A) is denoted by $L(M)$ ($L(M, A)$). We will write $M(x) = 1$ ($M^A(x) = 1$) iff $x \in L(M)$ ($x \in L(M, A)$), and $M(x) = 0$ ($M^A(x) = 0$) iff $x \notin L(M)$ ($x \notin L(M, A)$). An oracle machine M is called *positive* iff for any two sets A and B, it holds that $A \subseteq B$ implies $L(M, A) \subseteq L(M, B)$. A set A is polynomial time Turing reducible to a set B (notation: $A \leq_T^P B$) iff there exists a polynomial time bounded oracle machine M, such that $A = L(M, B)$. Selman (after Jockusch) extended this notion to: A language A is polynomial time *positive* Turing reducible to a language B (notation: $A \leq_{pos}^P B$) iff $A \leq_T^P B$ via a *positive* oracle machine M.

Selman introduced p-selective sets in [Sel79]. A set A is called p-selective if there exists a polynomial time computable function $f : \Sigma^* \times \Sigma^* \mapsto \Sigma^*$, called a *p-selector*, such that for any $x, y \in \Sigma^*$ both $f(x, y) \in \{x, y\}$ and $\chi_A(f(x, y)) = \max\{\chi_A(x), \chi_A(y)\}$

3 Main Result

This section is dedicated to proving our main result. Let A be a set that is positive Turing reducible to a p-selective set B.

Theorem 1 *For any set A and p-selective set B, if $A \leq_{pos}^P B$ then $A \leq_m^P B$, and A is p-selective.*

Before we prove this theorem, we state two useful results on p-selective sets. For the remainder of this section, let f be a p-selector for B. For $z \in \Sigma^*$, define the following two sets: $B^+(z) = \{x \mid f(x, z) = x\}$ and $B^-(z) = B^+(z) - \{z\}$. The following hold for these sets.

Lemma 2 *For $z \in \Sigma^*$, let $B^+(z)$ and $B^-(z)$ be as above*

1. *$B^+(z)$ and $B^-(z)$ are in P*
2. *$z \in B \Rightarrow B^+(z) \subseteq B$*
3. *$z \notin B \Rightarrow B \subseteq B^-(z)$.*

Proof. (1): Since f can be computed in polynomial time $B^+(z)$ and $B^-(z)$ are in P. (2): Let x be any string in $B^+(z)$. Then $f(x,z) = x$ and hence $z \in B \rightarrow x \in B$. (3): Let x be any string in B. Since $z \notin B$, we have $f(x,z) \neq z$ and hence $x \in B^-(z)$. □

The next lemma is (implicitly) used in many papers. We isolate its statement, because this action will simplify the proof of our main theorem.

Lemma 3 *Let V be a finite nonempty set of strings. There are strings x and y in B such that $x \notin B \rightarrow V \subseteq \overline{B}$ and $y \in B \rightarrow V \subseteq B$. Moreover, x and y can be found in time polynomial in $\|V\| \times \max\{|x| \mid x \in V\}$.*

Proof. We divide V into two sets V_1 and V_2 such that the following *invariant* holds: For all $u \in V_1 : u \in B \rightarrow V_2 \subseteq B$, and for all $v \in V_2 : v \in \overline{B} \rightarrow V_1 \subseteq \overline{B}$. We start with $V_1 = \emptyset$ and $V_2 = V$, so that the conditions trivially hold. As long as V_2 has at least two elements, we execute a *round*, during which we play a *knock-out tournament* between the strings of V_2, where v beats u if $f(u,v) = u$. Then $v \in B \rightarrow u \in B$. Let y be the winner of this tournament. Clearly, $y \in B \rightarrow V_2 \subseteq B$, so we may move y from V_2 to V_1 without violating the first part of the invariant. Since also $\exists x \in V_2 \cap \overline{B} \rightarrow y \notin B$, for each such y moved, the second part of the invariant is also preserved. The winner of the first round establishes the second part of the lemma. Inevitably, a round comes after which $\|V_2\| = 1$. The remaining string in V_2 establishes the first part of the lemma. □

Proof. (of Theorem 1) We will demonstrate a $\leq_{\hat{m}}^P$-reduction from A to B, i.e. a reduction that either determines membership in A without querying the oracle, or behaves as a $\leq_m^P$-reduction otherwise. This reduction can then be transformed into a $\leq_m^P$-reduction through standard means, provided $B \neq \Sigma^*$ and $B \neq \emptyset$. Let M_T witness the positive Turing reduction from A to B, and let x be a string. If $M^{\emptyset}(x) = 1$ or M accepts x without querying the oracle, then $M^A(x) = 1$, for any A so we can accept x. Likewise, we can reject if $M^{\Sigma^*}(x) = 0$, or M rejects x without querying the oracle. We therefore assume, that $M^{\emptyset}$ rejects, that M^{Σ^*} accepts, and that M reaches the query state at least once during the computation on input x. The following algorithm consists of an (outer) simulation of M on input x. Whenever M reaches the query state, then instead of querying the oracle, it performs a pair of (inner) simulations of M on x, using two different polynomial time computable oracle sets, and subsequently either produces the string y and stops, or it produces a continuation state (i.e.

YES/NO) for the simulation. If it reaches a final configuration, it also produces the string y, and stops. Therefore, it is polynomial time bounded and produces a string y. We will claim that $x \in A$ iff $y \in B$. Let $z_1, \ldots, z_{k-1}$ be query strings previously produced by the algorithm and consider the next time the algorithm either reaches a query state or a final state.
Algorithm: We treat the query and final state separately:

1. Suppose the state is QUERY and that z_k is the string written on the query tape. There are four cases:

   (a) $M^{B^+(z_k)}(x) = 1$ and $M^{B^-(z_k)}(x) = 0$. Then $y = z_k$; stop.

   (b) $M^{B^+(z_k)}(x) = M^{B^-(z_k)}(x) = 1$. Continue the simulation in the NO state;

   (c) $M^{B^+(z_k)}(x) = M^{B^-(z_k)}(x) = 0$. Continue the simulation in the YES state;

   The case where $M^{B^+(z_k)}(x) = 0$ and $M^{B^-(z_k)}(x) = 1$ cannot occur since we assumed that M is positive.

2. Suppose the state is ACCEPT. Let $W = \{z_1, \ldots, z_{k-1}\}$, and let $Y \subseteq W$ be the set of strings in W for which the simulation was continued in the YES state. $Y \neq \emptyset$ since we assumed that $M^{\emptyset}(x) = 0$. Let $y \in Y$ be such that $y \in B \rightarrow Y \subseteq B$. Such a string exists by Lemma 3.

3. Suppose the state is REJECT. Let $W = \{z_1, \ldots, z_{k-1}\}$, and let $N \subseteq W$ be the set of strings in W for which the simulation was continued in the NO state. $N \neq \emptyset$ since we assumed that $M^{\Sigma^*}(x) = 1$. Let $y \in N$ be such that $y \notin B \rightarrow N \subseteq \overline{B}$. Such a string exists again by Lemma 3.

CLAIM 4 *The string y produced by the algorithm has the property that $x \in A$ iff $y \in B$.*

Proof. We treat the cases separately. Suppose first, that y is produced by the algorithm terminating in case 1a. If $y \in B$ then $B^+(y) \subseteq B$, by Lemma 2, and so $M^B(x) = M^{B^+(y)}(x) = 1$, and thus $x \in A$. Likewise, if $y \notin B$ then $B \subseteq B^-(y)$ and therefore $M^B(x) = M^{B^-(y)}(x) = 0$ and so $x \notin A$. Suppose next, that the algorithm reaches case 2. Then $M^{B^-(y)}(x) = 0$. By Lemma 2, $y \notin B \rightarrow B \subseteq B^-(y) \rightarrow M^B(x) \leq M^{B^-(y)}(x) = 0$. On the other hand, $y \in B \rightarrow Y \subseteq B \rightarrow (B - W) \cup Y \subseteq B \rightarrow M^B(x) \geq M^{(B-W)\cup Y}(x) = 1$. Suppose finally, that the algorithm reaches case 3. Then $M^{B^+(y)}(x) = 1$. Hence $y \in B \rightarrow M^B(x) = 1$. However $y \notin B \rightarrow N \subseteq \overline{B} \rightarrow M^B(x) \leq M^{B\cup(W-N)}(x) = 0$. □

Finally, since A is $\leq_m^P$-reducible to B, it is p-selective. Given two strings x and y we can either directly compute for one of the two strings if it is in A

or not, or we can compute two strings x' and y' such that $x \in A \leftrightarrow x' \in B$, and $y \in A \leftrightarrow y' \in B$. Let $f'(x,y) = x$ if $f(x',y') = x'$ and y otherwise, then $\chi_A(f'(x,y)) = \chi_B(f(x',y')) = \max \chi_B(x'), \chi_B(y') = \max \chi_A(x), \chi_A(y)$, so f' is a p-selector for A. □

Remark: In the above algorithm, the query z_k is not necessarily encountered in the computation of $M^{B^+(z_k)}(x)$ or $M^{B^-(z_k)}(x)$. Suppose for instance, that $z_k \notin B$ and that all queries z_i in the path leading to z_k are not in B, then for each of these queries, both $f(z_i, z_k) = z_k$ and $f(z_i, z_k) = z_i$ are consistent with the behavior of the p-selector. It is therefore consistent to assume, that the computation of $M^{B^+(z)}(x)$, at the node where z_i is queried, fails to enter the branch of the computation tree that contains z_k. This situation does not occur if the p-selector computes a true linear order on the strings involved. In that case, cases 2 and 3 are contradictory, and the algorithm always ends up in case 1a.

From the theorem, and the main theorem in [BvHT92] we derive the following corollaries.

Corollary 5 *A is self-reducible and $A \leq^P_{pos} B$ iff $A \in P$.*

Corollary 6 *There are no $\leq^P_{pos}$-hard sets for EXP.*

Corollary 7 *Let C be any of the classes P, PP, $\oplus$P or PSPACE. $C = P$ iff there exists a $\leq^P_{pos}$-hard set for C.*

4 An open problem?

Ogiwara and Watanabe have shown [OW90] that existence of a sparse *NP*-complete set under bounded truth-table reductions would imply $P = NP$, and therefore that the existence of such a set is highly unlikely. Their proof involved extensive use of the left set technique. With this new result, a similar result for positive Turing reductions seems within reach. Suppose SATISFIABILITY would be positive Turing reducible to a sparse set. Then, by the recent result of Buhrman et al. [BLS92], this would imply a positive Turing reducibility of SATISFIABILITY to a Tally set. Selman showed [Sel82], that for any tally set T, there exists a p-selective set B such that $T \leq^P_T B$. By our theorem, changing the last reduction in the chain to a positive reduction would imply the desired result, since SAT is positively self-reducible and thus many-one self-reducible.

Unfortunately, our new theorem both opens and closes this line of attack. If a tally set T would be positive Turing reducible to a p-selective set, then our theorem implies that this tally set is itself p-selective. Non-p-selective tally sets are however easily constructed.

References

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, 1988.

[BLS92] H. Buhrman, L. Longpré, and E. Spaan. Sparse reduces conjunctively to tally. Technical Report NU-CCS-92-8, College of Computer Science, Northeastern University, Boston, 02215 MA, 1992. To appear in proc. 8th Structure in Complexity Theory.

[BvHT92] H. Buhrman, P. van Helden, and L. Torenvliet. Selective self-reducible sets: A new characterization of P. To appear in proc. 8th structure in complexity theory conf., 1992.

[Deg73] A.N. Degtev. *tt* and *m*-degrees. *Alg. Log.*, 12:143–161, 1973.

[Fri57] R.M. Friedberg. Two recursively enumerable sets of incomparable degrees of unsolvability. In *Proc. Nat. Acad. Sci.*, volume 43, pages 236–238, 1957.

[Joc66] C.G. Jockusch. *Reducibilities in Recursive Function Theory.* PhD thesis, M.I.T., 1966.

[Joc68] C.G. Jockusch. Semirecursive sets and positive reducibility. *Trans. of the Amer. Math. Soc.*, 131:420–436, 1968.

[Mar63] D.A. Martin. A theorem on hypersimple sets. *J. Symb. Log.*, 28:273–278, 1963.

[Mar76] S.S. Marchenkov. One class of partial sets. *Mat. Zam.*, 20:473–478, 1976.

[Muc56] A.A. Muchnik. Negative answer to the problem of reducibility in the theory of algorithms. *Dokl. Acad. Nauk*, 108:194–197, 1956.

[OW90] M. Ogiwara and O. Watanabe. On polynomial time bounded truth-table reducibility of NP sets to sparse sets. In *Proc. 22nd Annual ACM Symposium on Theory of Computing*, pages 457–467, 1990.

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

[Sel82] A. L. Selman. Analogues of semicursive sets and effective reducibilities to the study of NP complexity. *Information and Control*, 52(1):36–51, January 1982.