Logarithmic advice classes

J.L. Balcázar
U. Schöning

Report LSI-88-12
Abstract: Karp and Lipton introduced the notion of non-uniform complexity classes where a certain amount of "side information", the advice, is given for free. The advice only depends on the length of the input. Karp and Lipton (and also later researchers) concentrated on the study of classes of the form $C/poly$ where $C$ is P, NP, or PSPACE, and $poly$ denotes a polynomial size advice.

This paper starts a study of classes of the form $C/log$. As a main result it is shown that in the context of an NP/log computation a log-bounded advice is equivalent to a sparse oracle in NP. In contrast, it has been shown that a poly-bounded advice corresponds to an arbitrary sparse oracle set.

Furthermore, a general theorem is presented generalizing Karp and Lipton's "round-robin tournament" method.

Resum: La noció de classe no uniforme va ser introduïda per Karp i Lipton per formalitzar classes de complexitat on es pot fer servir sense cost una certa informació adicional (el "consell"); aquesta informació depèn només de la longitud de les dades. Aquests recerquers, i d'altres, van investigar principalment classes com ara P, NP, i PSPACE, amb consells fitats polinòmicament.

A aquest treball estudiem classes amb consells fitats logarítmicament. Els resultats principals mostren que a una computació NP/log, el consell logarítmic és equivalent a un oracle espars a NP. Aquest resultat es pot comparar amb un resultat conegut: consells polinòmics equivalen a oracles esparsos arbitraris.

Finalment, provem un teorema general que mostra com aplicar el mètode del "round-robin tournament" de Karp i Lipton a qualsevol conjunt auto-reduïble.
LOGARITHMIC ADVICE CLASSES

José L. Balcázar
Facultat d’Informàtica
UPC Barcelona
Pau Gargallo 5
E-08028 Barcelona, Spain

and

Uwe Schöning
EWH Koblenz, Rheinau 3-4
D-5400 Koblenz
West Germany

Abstract

Karp and Lipton [9] introduced the notion of non-uniform complexity classes where a certain amount of "side information", the advice, is given for free. The advice only depends on the length of the input. Karp and Lipton (and also later researchers [22,17,2,12]) concentrated on the study of classes of the form $C/poly$ where $C$ is P, NP, or PSPACE, and poly denotes a polynomial size advice.

This paper starts a study of classes of the form $C/log$. As a main result it is shown that in the context of an NP/log computation a log-bounded advice is equivalent to a sparse oracle in NP. In contrast, it has been shown that a poly-bounded advice corresponds to an arbitrary sparse oracle set.

Furthermore, a general theorem is presented that generalizes Karp and Lipton's "round-robin tournament" method.

---

*This work was supported by the Deutsche Forschungsgemeinschaft, grant No. Scho 3/2, and by a grant of CIRIT (Generalitat de Catalunya).
1 PRELIMINARIES

In this section we introduce the relevant notation and review some important results. For more detailed information we refer the reader to [3,18].

All sets considered are languages over some alphabet $\Sigma$, $|\Sigma| > 1$. For a string $x \in \Sigma^*$, $|x|$ denotes its length. For a set $A$ and an integer $n$, $A_n$ denotes the set \( \{ x \in A \mid |x| = n \} \). We call a set $S$ sparse if there is a polynomial $p$ such that for all $n$, the cardinality of $A_n$ is at most $p(n)$.

The classes $P$ and $NP$ have their standard definitions. We also refer to their relativized versions $P(A)$, $NP(A)$ and $P(C)$, $NP(C)$ where $A$ is a set and $C$ is a class of sets. The polynomial-time hierarchy [21] consists of the classes $\Delta_i$, $\Sigma_i$, $\Pi_i$ for $i \geq 0$, and is defined as

\[
\begin{align*}
\Delta_0 &= \Sigma_0 = \Pi_0 = P \\
\Delta_{i+1} &= P(\Sigma_i) \\
\Sigma_{i+1} &= NP(\Sigma_i) \\
\Pi_{i+1} &= co-NP(\Sigma_i)
\end{align*}
\]

The $\Sigma$ and $\Pi$ classes of the polynomial time hierarchy can be equivalently characterized in terms of alternating polynomially length-bounded existential and universal quantifications. For such and other properties of the polynomial time hierarchy, we refer the reader to [21,18,3].

Let $E$ ($NE$) be the class of sets recognizable deterministically (nondeterministically, resp.) in exponential (i.e. $2^{O(n)}$) time. It has been shown [5,7] that $E \neq NE$ if and only if there exist sets over a one-letter alphabet in $NP - P$ if and only if there exist sparse sets in $NP - P$.

We assume the existence of some easy-to-compute pairing function (\( \langle \rangle \)) whose inverses are also easy to compute. For a class of sets $C$ and a class of functions $\mathcal{F}$ from $\mathbb{N}$ to $\Sigma^*$ let $C/\mathcal{F}$ [9] be the class of sets $A$ for which there is a set $B \in C$ and a function $h \in \mathcal{F}$ such that for all $x \in \Sigma^*$,

\[ x \in A \iff \langle x, h(n) \rangle \in B \]

where $n = |x|$. For convenience, we write in the following $B(w)$ to denote the set \( \{ x \mid \langle x, w \rangle \in B \} \).

We are particularly interested in two special cases for the class $\mathcal{F}$. Let $\mathcal{F} = poly$ denote the class of polynomially bounded functions (i.e. $h \in poly$ if and only if $|h(n)| \leq p(n)$ for some polynomial $p$, depending on $h$.) Let $\mathcal{F} = log$
denote the class of logarithmically bounded functions (i.e. \( h \in \text{log} \) if and only if \(|h(n)| \leq c \cdot \log n \) for some constant \( c \), depending on \( h \).) Shortly summarized, the main results of Karp and Lipton [9] concerning such non-uniform complexity classes are:

(a) If \( \mathcal{C} \) is included in \( \mathbf{P}/\text{log} \) where \( \mathcal{C} \) is \( \mathbf{NP} \) or \( \mathbf{PSPACE} \) then \( \mathbf{P} = \mathcal{C} \).

(b) If \( \mathbf{NP} \) is included in \( \mathbf{P}/\text{poly} \) then the polynomial hierarchy "collapses" to the class \( \Sigma_2 \) (i.e. \( \Sigma_2 = \Pi_2 = \Sigma_3 = \Pi_3 = \ldots \))

(c) If \( \mathbf{PSPACE} \) is included in \( \mathbf{P}/\text{poly} \) then, in addition to (b), \( \mathbf{PSPACE} = \Sigma_2 \).

A further extension has been obtained by Yap [22]: If \( \text{co-NP} \) is included in \( \mathbf{NP}/\text{poly} \) then the polynomial hierarchy collapses to \( \Sigma_3 \). Furthermore, it has been shown that the classes \( \mathbf{P}/\text{poly}, \mathbf{NP}/\text{poly}, \mathbf{PSPACE}/\text{poly} \) are the same as \( \{ \mathbf{P}(S) | S \text{ is a sparse set} \}, \{ \mathbf{NP}(S) | S \text{ is a sparse set} \}, \{ \mathbf{PSPACE}(S) | S \text{ is a sparse set} \} \), resp. (see [4,17]). Intuitively, attaching a polynomially bounded advice means the same as relativizing the underlying class with an arbitrary sparse oracle set.

Additionally, the class \( \mathbf{P}/\text{poly} \) is known to be the same as the class of sets with polynomial-size circuits [4,18], and similarly, \( \mathbf{NP}/\text{poly} \) can be characterized as the class of sets with polynomial-size generators [22,17]. Several characterizations of the class \( \mathbf{PSPACE}/\text{poly} \) can be found in [2].

For technical reasons, we also introduce the class \( \text{strong-}\mathcal{C}/\mathcal{F} \) (cf. [11]) where \( A \in \text{strong-}\mathcal{C}/\mathcal{F} \) if there is a set \( B \in \mathcal{C} \) and a function \( h \in \mathcal{F} \) such that for all \( x \in \Sigma^* \) and all \( m \geq |x| \),

\[
x \in A \iff (x,h(m)) \in B.
\]

It is easy to see that for any class \( \mathcal{C} \) closed under \( \leq^m_p \)-reducibility (see below), \( \text{strong-}\mathcal{C}/\text{poly} = \mathcal{C}/\text{poly} \). Only for advice lengths smaller than polynomial there is a difference.

**Proposition 1.1.** For every class \( \mathcal{C} \) of recursive sets, \( \mathbf{P}/\text{log} \not\subseteq \text{strong-}\mathcal{C}/\text{log} \).

**Proof.** Define a set \( A \) over the one-letter alphabet \( \{0\} \) such that its characteristic sequence is an infinite Kolmogorov-random string (that means that every finite initial segment of it has almost linear Kolmogorov complexity). Such a tally language \( A \) is obviously in \( \mathbf{P}/\text{log} \) (even in \( \mathbf{P}/1 \)), but no initial run of \( A \)'s characteristic sequence can be described by \( \log n \) many bits, therefore \( A \not\subseteq \text{strong-}\mathcal{C}/\text{log} \) for every recursive \( \mathcal{C} \).  \( \square \)
If we mention complete sets $A$ for certain classes $\mathcal{C}$ then we mean *many-one* completeness, that is, $A \in \mathcal{C}$ and for all $B \in \mathcal{C}$, $B \leq_{\mathcal{P}}^n A$. Another reducibility mentioned is *strong nondeterministic Turing reducibility* $\leq_{T}^{n}$ which can be defined as $A \leq_{T}^{n} B$ if and only if $A \in \text{NP}(B) \cap \text{co-NP}(B)$. This is equivalent [19,13] to $\text{NP}(A) \subseteq \text{NP}(B)$.

Whenever a relativized class is referred to, we write, e.g., $\text{P}^{\text{NP}}$ instead $\text{P}(\text{NP})$. Additionally, $\text{P}[\log]^{\text{NP}}$ means that the basis oracle machine asks only $O(\log n)$ many queries to the oracle. The class $\text{P}[\log]^{\text{NP}}$ has received some interest because Kadin [8] has shown that $\text{co-NP} \subseteq \text{NP}(S)$ for a sparse set $S$ implies that the polynomial time hierarchy collapses to the class $\text{P}[\log]^{\text{NP}}$.

The classes of the low and high hierarchy in $\text{NP}$ were introduced in [16] and further analyzed in [12]. We give only the definition of the below-mentioned classes $\tilde{L}_i$ for $i \geq 1$,

$$\tilde{L}_i = \{ A \in \text{NP} \mid \Delta_i(A) \subseteq \Delta_i \},$$

and refer the reader to [16,12] for their properties.
2 MAIN RESULTS

We start with a theorem which says that a log advice is, in a sense, equivalent to a sparse oracle in NP. (Notice that a poly advice indeed is equivalent to an arbitrary sparse oracle [4,17]).

Theorem 2.1.

(a) If $A \in \text{co-NP} \cap \text{NP/log}$ then $A \in \text{NP}(S)$ for some sparse set $S \in \text{NP}$.

(b) If $A \in \text{co-NP} \cap \text{P/log}$ then $A \in \text{P}(S)$ for some sparse set $S \in \text{NP}$.

Proof. To prove (a), assume that $A \in \text{co-NP} \cap \text{NP/log}$. Let $B \in \text{NP}$ such that for all $x, x \in A \iff x \in B(w_n)$ where $n = |x|$ and $w_n$ denotes the advice for length $n$; let $|w_n| = c \cdot \log n$. Consider the set $S$ formed by all pairs $(0^n, y)$ where $|y| = c \cdot \log n$, and such that there is an $x$ of length $n$ in $B(y) - A$. Since there are only polynomially many $y$'s of such length, $S$ is sparse. To decide $S$ in NP, guess $x$ and check that it is both in $B(y)$ and in $\overline{A}$. Now we claim that

$$A = \{x \mid \exists y(|y| = c \cdot \log n \wedge (0^n, y) \notin S \wedge x \in B(y))\}$$

Indeed, if $w_n$ is the correct advice for length $n$ then $(0^n, w_n) \notin S$, and thus for $x \in A$ there is $y = w_n$ for which $(0^n, y) \notin S \wedge x \in B(y)$ holds. Conversely, assume $x \notin A$; if any $y$ is found of the appropriate length for which $x \in B(y)$, then $x \in B(y) - A$, therefore $(0^n, y) \in S$ and the right hand side predicate fails. Thus $A \in \text{NP}(S)$.

Finally, to prove (b), observe that under the hypothesis that $B \in \text{P}$, the right hand side predicate becomes $\text{P}(S)$, since the quantification is logarithmically bounded.

The reader should notice the unusual “one-sided correctness” that the sparse oracle $A$ above has. This one-sided correctness is the reason for the membership of $A$ in NP, and on the other hand, it is in this context still sufficiently strong to allow the basis algorithm to obtain “full correctness” by asking several oracle queries.

Note that $S$ can be taken as well in co-NP, by defining it in terms of the “opposite kind of mistakes”: form $S'$ with all pairs $(0^n, y)$ where $|y| = c \cdot \log n$, and such that every $x$ of length $n$ in $B(y)$ is also in $A$. This can be checked in co-NP, and now $A$ is defined by

$$A = \{x \mid \exists y(|y| = c \cdot \log n \wedge (0^n, y) \in S \wedge x \in B(y))\}$$

The argument is essentially the same. Also, by simple complementation, we can obtain that if $A \in \text{NP} \cap \text{co-NP/log}$ then $A \in \text{co-NP}(S)$ where $S \in \text{NP}$.
We use this theorem to obtain the following equivalences.

**Corollary 2.2.** The following are equivalent.

(a) \( \text{co-NP} \subseteq \text{NP}/\log \).

(b) \( \text{co-NP} \subseteq \text{NP}(S) \) for some sparse \( S \in \text{NP} \).

(c) There is a sparse \( \leq_{\text{mp}} \)-complete set for \( \text{NP} \).

**Proof.** (a) \( \Rightarrow \) (b). Let TAUT be the set of boolean tautologies, which is \( \leq_{\text{mp}}^{\text{n}} \)-complete for co-NP. By (a), TAUT \( \in \text{NP}/\log \), therefore by Theorem 2.1, TAUT \( \in \text{NP}(S) \) for a sparse set \( S \) in \( \text{NP} \). Since \( \text{NP}(S) \) is closed under \( \leq_{\text{mp}}^{\text{n}} \)-reducibility, co-NP \( \subseteq \text{NP}(S) \).

(b) \( \Rightarrow \) (c). Assume co-NP \( \subseteq \text{NP}(S) \) for sparse \( S \in \text{NP} \). We show that \( S \) is \( \leq_{\text{mp}}^{\text{n}} \)-hard for NP. Since co-NP \( \subseteq \text{co-NP}(S) \) trivially, co-NP \( \subseteq \text{NP}(S) \cap \text{co-NP}(S) \), and by complementation \( \text{NP} \subseteq \text{NP}(S) \cap \text{co-NP}(S) \). Thus \( A \leq_{\text{mp}}^{\text{n}} S \) for every set \( A \in \text{NP} \).

(c) \( \Rightarrow \) (a). By (c) and complementation, co-NP \( \subseteq \text{NP}(S) \cap \text{co-NP}(S) \subseteq \text{NP}(S) \). The advice will consist of the census of \( S \) up to \( p(n) \) for a polynomial \( p \); then nondeterminism is used to find out all of \( S \) up to \( p(n) \), obtaining an NP algorithm without oracle. Formally, let \( A \in \text{NP}(S) \) via a nondeterministic oracle machine \( M \) whose queries on inputs of length \( n \) are of length at most \( p(n) \). On input \( (x, k) \) where \( |x| = n \) and \( k \) is the census of \( S \) up to length \( p(n) \), the following nondeterministic algorithm accepts \( A \):

\[
\text{INPUT} \ (x, k) \\
\text{GUESS} \ a \ \text{set} \ s \ \text{of} \ k \ \text{different strings of length at most} \ p(n) \\
\text{FOR} \ \text{each} \ \text{string} \ w \ \text{in} \ s \ \text{DO} \\
\quad \text{nondeterministically check that} \ w \in S \\
\text{CHECK} \ \text{that} \ x \ \text{is} \ \in L(M, s) \ \text{and if so, halt accepting}
\]

Since \( k \) can be written down in \( O(\log n) \) bits, we obtain that \( A \in \text{NP}/\log \). \( \square \)

The class \( \text{NP}/\log \) is closed under polynomially length-bounded existential quantification. Therefore, all of the following statements are easily seen to be equivalent to co-NP \( \subseteq \text{NP}/\log \):

(a) There is a sparse \( \leq_{\text{mp}}^{\text{n}} \)-complete set for co-NP.
(b) $\text{co-NP} \subseteq \text{NP}(S)$ for some sparse $S \in \text{co-NP}$.

(c) $\Sigma_2 \subseteq \text{NP}(S)$ for some sparse $S \in \text{NP}$.

(d) $\Sigma_2 \subseteq \text{NP}(S)$ for some sparse $S \in \text{co-NP}$.

(e) $\Sigma_2 \subseteq \text{NP/log}$.

(f) All the facts obtained by complementing these facts, such as $\text{NP} \subseteq \text{co-NP/log}$, $\Pi_2 \subseteq \text{co-NP/log}$, and the like.

A second consequence of the theorem is the inclusion of $\text{NP} \cap \text{co-NP/log}$ in the low hierarchy [17,12]. We need the following easy lemma.

**Lemma 2.3.** $L_i$ for $i \geq 1$, and $\hat{L}_i$ for $i \geq 2$, are closed under $\leq^{\text{NP}}_i$.

**Proof.** Assume $B \in L_i$ with $i \geq 1$. It is known [13,19] that $A \leq^{\text{NP}}_i B$ if and only if $\text{NP}(A) \subseteq \text{NP}(B)$. Then $\Sigma_i(A) = \Sigma_{i-1}(NP(A)) \subseteq \Sigma_{i-1}(NP(B)) = \Sigma_i(B) = \Sigma_i$, and therefore $A \in L_i$. The argument for $\hat{L}_i$ is the same. $\square$

Using this fact, and the fact that sparse sets in $\text{NP}$ are in $\hat{L}_2$ [12], we obtain:

**Corollary 2.4.** $\text{NP} \cap \text{co-NP/log} \subseteq \hat{L}_2$.

**Proof.** Let $A \in \text{NP} \cap \text{co-NP/log}$. By the theorem, $A \in \text{co-NP}(S)$ where $S \in \text{NP}$. Since $A \in \text{NP}$, we can state that $A \in \text{NP}(S) \cap \text{co-NP}(S)$, i.e. $A \leq^{\text{NP}}_i S \in \hat{L}_2$. By the lemma, $A \in \hat{L}_2$. $\square$

Thus, if $\text{NP} \subseteq \text{co-NP/log}$ then the polynomial time hierarchy collapses to $\Delta_2$. It should be observed that this collapse can be improved.

**Corollary 2.5.** If $\text{NP} \subseteq \text{co-NP/log}$ then the polynomial time hierarchy collapses to $\text{P}[\log]^{\text{NP}}$.

**Proof.** In [8] it is shown that if $\text{co-NP} \subseteq \text{NP}(S)$ for some sparse $S \in \text{NP}$ then the indicated collapse holds. Corollary 2.2 asserts that Kadin's hypothesis is equivalent to $\text{NP} \subseteq \text{co-NP/log}$. $\square$
Corollary 2.5 could be expressed as a form of lowness, in the sense that what it proves is something like: if \( A \in \text{co-NP/log} \) then \( (\text{P/log})^\text{NP} A = (\text{P/log})^\text{NP} \). This does not properly correspond to any of the lowness properties defined in [12], but the statement has the same structure.

Using the characterization by means of sparse sets in \( \text{NP} \), we can relate existence of unfeasible sets in these classes to the inequality of exponential time classes.

**Theorem 2.6.**

(a) \( (\text{NP} \cap \text{P/log}) - \text{P} \neq \emptyset \) if and only if \( \text{E} \neq \text{NE} \).

(b) \( (\text{co-NP} \cap \text{NP/log}) - \text{NP} \neq \emptyset \) if and only if \( \text{NE} \neq \text{co-NE} \).

**Proof.** Right to left implications are immediate from the fact that all tally sets are in \( \text{P/log} \). Assume \( \text{E} = \text{NE} \). By [7], no sparse sets exist in \( \text{NP} - \text{P} \). By Theorem 2.1, using complementation, we know that the sets in \( \text{NP} \cap \text{P/log} \) are in \( \text{P}(S) \) for a sparse set \( S \in \text{NP} \), and therefore \( S \in \text{P} \). Thus \( \text{NP} \cap \text{P/log} = \text{P} \). Similarly, if \( \text{NE} \) is closed under complementation then all sparse sets in \( \text{NP} \) are also in \( \text{co-NP} \). Thus the theorem yields that sets in \( \text{co-NP} \cap \text{NP/log} \) are in \( \text{NP}(S) \) where \( S \in \text{NP} \cap \text{co-NP} \), and therefore \( \text{co-NP} \cap \text{NP/log} \subseteq \text{NP} \).

It is also interesting to note that \( (\text{NP} \cap \text{P/log}) \) has \( \leq^p_T \)-complete sets, namely, those tally sets that are \( \leq^p_T \)-complete for the class of sparse sets in \( \text{P} \)(see [7]).

**Proposition 2.7.** Let \( T_0 = \text{tally}(K_E) \) where \( K_E \) is any \( \leq^p_n \)-complete set for \( \text{E} \). Then \( T_0 \) is \( \leq^p_T \)-complete for \( (\text{NP} \cap \text{P/log}) \).

**Proof.** It is known [7] that \( T_0 \) is \( \leq^p_T \)-complete for the class of the sparse sets in \( \text{NP} \). Of course \( T_0 \in \text{NP} \cap \text{P/log} \). Given \( A \in \text{NP} \cap \text{P/log} \), we see from the theorem that \( A \in \text{P}(S) \) for a sparse \( S \in \text{NP} \), and therefore \( S \in \text{P}(T_0) \). Thus \( A \in \text{P}(T_0) \).

Further, we want to point out the incomparability of \( \text{NP/log} \) and \( \text{P/poly} \) under reasonable assumptions.

**Theorem 2.8.**

(a) If \( \Sigma_2 \neq \Pi_2 \) then \( \text{NP/log} \not\subseteq \text{P/poly} \).
(b) \( P/\text{poly} \not\subseteq \text{NP/log} \).

**Proof.** Statement (a) follows from [9] since \( \text{NP} \subseteq \text{NP/log} \), and \( \text{NP} \subseteq P/\text{poly} \) implies \( \Sigma_2 = \Pi_2 \). Statement (b) can be proved by considering a sparse set \( S \) whose characteristic function at each length \( n \) consists of a string of length \( n \) of high (unbounded) Kolmogorov complexity, followed by \( 2^n - n \) zeros. Such a set cannot be decided with an \( O(\log n) \) advice in a recursive manner, since otherwise the advice plus a constant length program would allow to recover the first \( n \) bits of the characteristic function of \( S \), contradicting its high Kolmogorov complexity.

\[ \square \]

Note that Theorem 2.8 (b) can be generalized, similar to Proposition 1.1: For every class \( C \) of recursive sets, \( P/\text{poly} \not\subseteq C/\log \).

Several of Karp and Lipton's proofs [9] take advantage of the self-reducibility structure of typical \( C \)-complete sets where \( C \) is \( P, \text{NP} \), or \( \text{PSPACE} \). The following definition refers to the length order, more general definitions can be found in [10,15].

**Definition 2.9** A set \( A \) is **self-reducible** if there is a deterministic, polynomial-time oracle machine \( M \) such that \( A = L(M, A) \), and for each \( x \), \( M \) on input \( x \) queries the oracle only for strings \( y, |y| < |x| \).

**Theorem 2.10.** If \( A \) is a self-reducible set with \( A \in \text{strong-P/log} \), then \( A \in P \).

**Proof.** Let \( M \) be the self-reducing machine for \( A \) according to Definition 2.9.

Since \( A \in \text{strong-P/log} \), there is a set \( B \in P \) together with a sequence \( \{w_n\}_{n \in \mathbb{N}} \) of strings such that \( |w_n| \leq c \cdot \log n \) for some constant \( c \) and all \( n \), and \( A_n = B(w_m)_n \) for all \( n \) and \( m \geq n \).

Let a polynomial-time computable 2-placed predicate **consistent** be defined as

\[
\text{consistent}(x, w) \iff (x \in B(w) \iff x \in L(M, B(w))). 
\]

Intuitively, \( \text{consistent}(x, w) \) is true if the "advice" \( w \) – together with the "advice interpreter" \( B \) – behaves consistent (locally), i.e. on the first level of the self-reduction structure of \( x \) induced by \( M \). Notice that \( \text{consistent}(x, w_m) \) is true for all \( m \geq |x| \).

Now, the following algorithm will be shown to recognize \( A \) in polynomial-time, hence \( A \in P \).
INPUT x ; \{ |x| = n \}
FOR u, |u| \leq c \cdot \log n DO
    BEGIN
        b := TRUE ;
        FOR v, |v| \leq c \cdot \log n DO
            b := b AND test(x, u, v) ;
        IF b THEN
            IF x \in L(M, B(u))
                THEN halt accepting
            ELSE halt rejecting ;
        END ;
    END ;

Hereby, the recursive procedure test(x, u, v) operates as follows.

PROCEDURE test(x, u, v) : BOOLEAN ;
BEGIN
    IF (x \in L(M, B(u)) \iff x \in L(M, B(v)))
        THEN RETURN TRUE
    ELSE
        BEGIN
            y := the first query string in the computations of M^{B(u)}
            and M^{B(v)} on input x where the oracles B(u) and
            B(v) disagree in their respective answers;
            IF NOT consistent(y, u) THEN RETURN FALSE ;
            IF NOT consistent(y, v) THEN RETURN TRUE ;
            RETURN test(y, u, v) ;
        END ;
END ;

By polynomial well-foundedness of the self-reduction structure of A, it is clear that
the (tail) recursion of test is polynomially bounded in depth. Whenever a string
in the self-reducibility structure of M on x is reached where M does not query
its oracle, then x \in L(M, B(u)) \iff x \in L(M, B(v)) is true, and the recursion
ends. Therefore, the procedure test runs in polynomial time (in |x| + |u| + |v|).
The main program is therefore polynomial-time in |x|.

Regarding the correctness, we prove two claims:

Claim 1. For every string x (|x| = n), every m \geq n, and every v (|v| \leq c \cdot \log n),
test(x, w_m, v) evaluates to TRUE.

Proof. Suppose the contrary. Choose x minimal such that for some m and some
\( v, \text{test}(x, w_m, v) = \text{FALSE}. \) This string \( x \) cannot be a "leaf" in the self-reducibility structure of \( M \) on \( x \), otherwise the procedure \( \text{test} \) would return \text{TRUE} in the first line. That is, \( M \) on input \( x \) has to have at least one oracle query, and the ELSE branch is entered. Since the oracles \( B(\cdot) \) and \( B(v) \) disagree what the status of \( x \) is concerned, there must be a first oracle query, which is assigned to \( y \), on which the oracles disagree. The test consistent\((y, w_m)\) is true by definition of \( w_m \). Also, the next test for consistent\((y, w_m)\) is true since test is assumed to return \text{FALSE}. Therefore, the control reaches the recursive call of \( \text{test}(y, w_m, v) \) which, by assumption, evaluates to \text{FALSE}. But this is a contradiction to the minimality choice of \( x \).

**Claim 2.** For every string \( x \) (\(|x| = n\)), every \( u \) (\(|u| \leq c \cdot \log n\)), and every \( m \geq n \), if \( \text{test}(x, u, w_m) \) evaluates to \text{TRUE}, then

\[
x \in L(M, B(u)) \iff x \in L(M, B(w_m)).
\]

(Therefore, \( x \in L(M, B(u)) \iff x \in A. \))

**Proof.** Assume the contrary. Fix a string \( u \) for which the predicate fails, and choose \( x \) (\(|x| = n\)) minimal such that for some \( m \), \( \text{test}(x, u, w_m) = \text{TRUE} \), but

\[
x \in L(M, B(u)) \iff x \notin L(M, B(w_m)).
\]

Consider the execution of the procedure call \( \text{test}(x, u, w_m) \). By the above assumption, the ELSE branch is entered. Therefore, \( M \) on input \( x \) has at least one oracle query, and a query string \( y \) can be determined such that

\[
y \in B(u) \iff y \notin B(w_m). \tag{1}
\]

By the assumption that \( \text{test}(x, u, w_m) = \text{TRUE} \), it must be the case that consistent\((y, u)\) is true, i.e.

\[
y \in L(M, B(u)) \iff y \in B(u). \tag{2}
\]

Also, by definition of \( w_m \), consistent\((y, w_m)\) is true, i.e.

\[
y \in L(M, B(w_m)) \iff y \in B(w_m). \tag{3}
\]

Therefore, the control reaches the recursive call of \( \text{test}(y, u, w_m) \). By the assumption, this call returns \text{TRUE}. By minimality of \( x \), and since \( y < x \), the assertion of the claim holds for \( y \), i.e.

\[
y \in L(M, B(u)) \iff y \in L(M, B(w_m)). \tag{4}
\]
Now, combining the statements (1), (2), (3), and (4) gives a contradiction, which proves the claim. □

The correctness of the algorithm follows from Claim 1 and 2 as follows. Whenever the outer for-loop in the main program finds some \( u \) with \( \text{test}(x, u, v) = \text{TRUE} \) for all \( v \) (especially for \( v = w_n \)), then, by Claim 2, the decision of the algorithm for accepting or rejecting is correct.

On the other hand, Claim 1 guarantees that at least one such \( u \), e.g. \( u = w_n \), will be found in the outer for-loop.

This completes the proof of the Theorem. □

For simplicity, the above proof was given w.r.t. self-reducibility defined on the length order. It is easy to see that this is not essential. The same proof goes through when using the more general definitions from [10,15] where just a polynomially well founded order, not necessarily length-respecting, on the "self-reduction tree" is required.

Furthermore, if the self-reduction tree can, by padding properties of the considered language \( A \), be assumed to consist only of strings of the same length as the input string \( x \), then the assumption "\( A \in \text{P} \\cap \text{P/log} \)" can be weakened to "\( A \in \text{P} \\cap \text{log} \). Since all "natural" complete sets for classes \( C \) in \( \text{PSPACE} \) do have such padding properties, we immediately obtain (for the cases \( C = \text{NP} \) and \( C = \text{PSPACE} \)):

**Corollary 2.11 [9].** If \( \text{NP} \subseteq \text{P/log} \) then \( \text{NP} = \text{P} \).

**Corollary 2.12 [9].** If \( \text{PSPACE} \subseteq \text{P/log} \) then \( \text{PSPACE} = \text{P} \).

**Proof.** Use the fact that the set \( QBF \) has a very simple (with appropriate encoding, even length-respecting) self-reducibility structure, and that \( QBF \) is \( \text{PSPACE} \)-complete. □

Actually, this simple (namely, 2-truth-table and positive) self-reducibility structure of \( QBF \) has been used by Karp and Lipton directly to prove this corollary. Our theorem extends their "round-robin tournament" method to the more general situation of an adaptive self-reduction. Karp and Lipton's original argument uses positiveness of the reduction, and is therefore not directly applicable to this more general situation. For example, the language \( \#SAT \) which is complete for the class \( \text{PP} \) (see [20,6]) has a more general (adaptive) self-reducibility structure (see
[1,18]). The following application of Theorem 2.10 is therefore new.

Corollary 2.13. If $PP \subseteq P/log$ then $PP = P$.

Acknowledgement

The authors would like to thank Rainer Schuler for pointing out an error in an earlier version of this paper.
References


