A Reducibility that Corresponds to Unbalanced Leaf-Language Classes

Christian Glaßer, Stephen Travers, and Klaus W. Wagner

Theoretische Informatik
Julius-Maximilians Universität Würzburg,
97074 Würzburg, Germany

28th February 2005

Abstract

We introduce the polynomial-time tree reducibility (ptt-reducibility). Our main result states that for languages B and C it holds that B ptt-reduces to C if and only if the unbalanced leaf-language class of B is robustly contained in the unbalanced leaf-language class of C. This is the *unbalanced* analogue of the well-known result by Bovet, Crescenzi, Silvestri, and Vereshchagin which connects polylog-time reducibility with *balanced* leaf-languages.

We show that restricted to regular languages, the levels 0, $1/2$, 1, and $3/2$ of the dot-depth hierarchy (DDH) are closed under ptt-reducibility. This gives evidence that with respect to unbalanced leaf-languages, the dot-depth hierarchy and the polynomial-time hierarchy perfectly correspond. Level 0 of the DDH is closed under ptt-reducibility even without the restriction to regular languages. We show that this does not hold for higher levels.

As a consequence of our study of ptt-reducibility, we obtain the first gap theorem of leaf-language definability above the Boolean closure of NP: If $\mathcal{D} = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(\mathcal{C})$ for some $\mathcal{C} \subseteq \mathrm{REG}$, then $\mathcal{D} \subseteq \mathrm{BC(NP)}$ or there exists an oracle O such that $\mathcal{D}^O \not\subseteq \mathrm{P}^{\mathrm{NP}[\epsilon\cdot\log n]^O}$ for every $\epsilon < 1$.

1 Introduction

In their pioneering work for the leaf-language approach, Bovet, Crescenzi, and Silvestri [BCS92] and Vereshchagin [Ver93] independently introduced the notion of polylog-time reducibility (plt-reducibility for short). This reducibility allows an amazing translation between two seemingly independent questions.

1. Are given complexity classes separable by oracles?
2. Are given languages plt-reducible?

Leaf Languages. The translation mentioned above uses the concept of *leaf languages*. Let M be a nondeterministic polynomial-time bounded Turing machine such that every computation path outputs one letter from a fixed alphabet. Let $M(x)$ denote the computation tree of M on input x. Let $\beta_M(x)$ be the concatenation of all leaf-symbols of $M(x)$. For a language B, let $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(B)$ be the class of languages

L such that there exists a nondeterministic polynomial-time-bounded Turing machine M as above such that for all x,

$$x \in L \iff \beta_M(x) \in B.$$

We refer to $\mathrm{Leaf}_\mathrm{u}^\mathrm{p}(B)$ as the *unbalanced leaf-language class* of B. Call a nondeterministic polynomial-time-bounded Turing machine M *balanced* if there exists a polynomial-time computable function that on input (x,n) computes the n-th path of $M(x)$. If we assume M to be balanced in the definition above, then this defines the class $\mathrm{Leaf}_\mathrm{b}^\mathrm{p}(B)$ which we call the *balanced leaf-language class* of B. For any class of languages $\mathcal{C}$ let $\mathrm{Leaf}_\mathrm{u}^\mathrm{p}(\mathcal{C}) = \bigcup_{B\in\mathcal{C}} \mathrm{Leaf}_\mathrm{u}^\mathrm{p}(B)$ and $\mathrm{Leaf}_\mathrm{b}^\mathrm{p}(\mathcal{C}) = \bigcup_{B\in\mathcal{C}} \mathrm{Leaf}_\mathrm{b}^\mathrm{p}(B)$. Call a complexity class $\mathcal{D}$ *unbalanced leaf-language definable* if there exists $\mathcal{C}$ such that $\mathcal{D} = \mathrm{Leaf}_\mathrm{u}^\mathrm{p}(\mathcal{C})$. Analogously define *balanced leaf-language definability*. For a survey on leaf-languages we refer to [Wag04].

BCSV-Theorem. Suppose for given complexity classes $\mathcal{D}_1$ and $\mathcal{D}_2$, there exist languages L_1 and L_2 such that $\mathcal{D}_1 = \mathrm{Leaf}_\mathrm{b}^\mathrm{p}(L_1)$ and $\mathcal{D}_2 = \mathrm{Leaf}_\mathrm{b}^\mathrm{p}(L_2)$. The theorem by Bovet, Crescenzi, Silvestri, and Vereshchagin states the following.

$$L_1 \leq_\mathrm{m}^\mathrm{plt} L_2 \quad\Leftrightarrow\quad \forall O\big(\mathrm{Leaf}_\mathrm{b}^{\mathrm{p}^O}(L_1) \subseteq \mathrm{Leaf}_\mathrm{b}^{\mathrm{p}^O}(L_2)\big) \tag{1}$$

Here $\leq_\mathrm{m}^\mathrm{plt}$ denotes polylog-time reducibility (Definition 2.2). For this equivalence it is crucial that balanced leaf-language classes are used. The theorem does not hold for the unbalanced model: Observe that languages $L, L' \subseteq \{0,1\}^*$ with $L =_\mathrm{def} \{w \mid |w| \text{ is odd}\}$, $L' =_\mathrm{def} 0\{0,1\}^*$ form a counterexample, since $\mathrm{Leaf}_\mathrm{u}^\mathrm{p}(L) = \oplus\mathrm{P}$ is not robustly contained in $\mathrm{Leaf}_\mathrm{u}^\mathrm{p}(L') = \mathrm{P}$ though L plt-reduces to L'. In this paper we introduce a new reducibility (ptt-reducibility) which allows us to prove the following unbalanced analogue.

$$L_1 \leq_\mathrm{m}^\mathrm{ptt} L_2 \quad\Leftrightarrow\quad \forall O\big(\mathrm{Leaf}_\mathrm{u}^{\mathrm{p}^O}(L_1) \subseteq \mathrm{Leaf}_\mathrm{u}^{\mathrm{p}^O}(L_2)\big) \tag{2}$$

Beside the pure academic interest of a Bovet-Crescenzi-Silvestri-Vereshchagin-like theorem (BCSV-theorem for short) for the unbalanced case, further motivation comes from a connection between complexity theory and the theory of finite automata: On the lower levels, the dot-depth hierarchy perfectly corresponds to the polynomial-time hierarchy when we consider *unbalanced* leaf-languages. Below, after the introduction of both hierarchies, we will emphasize that equivalence (2) can be very useful in this respect.

Dot-Depth Hierarchy. *Starfree regular languages* (starfree languages for short) are regular languages that can be built up from single letters by using Boolean operations and concatenation (so iteration is not allowed). SF denotes the class of starfree languages. Brzozowski and Cohen [CB71, Brz76] introduced the *dot-depth hierarchy* (DDH for short) which is a parameterization of the class of starfree languages. The dot-depth counts the minimal number of nested alternations between Boolean operations and concatenation that is needed to define a language. The classes of the dot-depth hierarchy consist of languages that have the same dot-depth. For a class of languages $\mathcal{C}$, let $\mathrm{Pol}(\mathcal{C})$ denote $\mathcal{C}$'s closure under finite union and finite concatenation. Let $\mathrm{BC}(\mathcal{C})$ denote the Boolean closure of $\mathcal{C}$. The classes (or levels) of the dot-depth hierarchy are defined as:

$$\begin{aligned}
\mathcal{B}_0 &=_\mathrm{def} \{L \subseteq A^* \mid A \text{ is a finite alphabet with at least two letters and } L \\
&\qquad\qquad\qquad \text{is a finite union of terms } vA^*w \text{ where } v, w \in A^*\} \\
\mathcal{B}_{n+\frac{1}{2}} &=_\mathrm{def} \mathrm{Pol}(\mathcal{B}_n) \\
\mathcal{B}_{n+1} &=_\mathrm{def} \mathrm{BC}(\mathcal{B}_{n+\frac{1}{2}})
\end{aligned}$$

The dot-depth of a language L is defined as the minimal m such that $L \in \mathcal{B}_m$ where $m = n/2$ for some integer n. All levels of the dot-depth hierarchy are closed under union, under intersection, under taking inverse morphisms, and under taking residuals [PP86, Arf91, PW97]. The dot-depth hierarchy is strict [BK78, Tho84] and exhausts the class of starfree languages [Eil76].

Polynomial-Time Hierarchy. For a complexity class $\mathcal{D}$ let $\mathrm{co}\mathcal{D} = \{\overline{L} \mid L \in \mathcal{D}\}$. Let $\exists{\cdot}\mathcal{D}$ denote the class of languages L such that there exists a polynomial p and $B \in \mathcal{D}$ such that $x \in L \Leftrightarrow \exists y, |y| \leq p(|x|), (x, y) \in B$. Let $\forall{\cdot}\mathcal{D} = \mathrm{co}\exists{\cdot}\mathrm{co}\mathcal{D}$. Define $\exists!{\cdot}\mathcal{D}$ and $\forall!{\cdot}\mathcal{D}$ similarly by using $\exists!$ and $\forall!$ instead of $\exists$ and $\forall$. Stockmeyer [Sto77] introduced the polynomial-time hierarchy (PH for short). We use a definition which is due to Wrathall [Wra77].

$$\begin{aligned} \Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} &=_{\mathrm{def}} \mathrm{P} \\ \Sigma_{n+1}^{\mathrm{P}} &=_{\mathrm{def}} \exists{\cdot}\Pi_n^{\mathrm{P}} \\ \Pi_{n+1}^{\mathrm{P}} &=_{\mathrm{def}} \forall{\cdot}\Sigma_n^{\mathrm{P}} \end{aligned}$$

Connection between DDH and PH. We continue reasoning the better suitability of the unbalanced model for the connection between dot-depth hierarchy and polynomial-time hierarchy. Hertrampf et al. [HLS+93], and Burtschick and Vollmer [BV98] proved that the levels of the polynomial-time hierarchy are connected with the levels of the dot-depth hierarchy. For $n \geq 1$,

$$L \in \mathcal{B}_{n-1/2} \quad \Rightarrow \quad \forall O(\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}^O}(L) \subseteq \Sigma_n^{\mathrm{P}^O}), \tag{3}$$

$$L \in \mathcal{B}_{n-1/2} \quad \Rightarrow \quad \forall O(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}^O}(L) \subseteq \Sigma_n^{\mathrm{P}^O}). \tag{4}$$

In particular, the attraction of this connection comes from the fact that both hierarchies are prominent and well-studied objects. Even more, with the P-NP problem and the dot-depth problem, they represent two of the most fundamental problems in theoretical computer science.

Can we turn the implications (3) and (4) into equivalences?

The reverse of (3) does not hold, even if we demand L to be starfree: For every $n \geq 1$, there exists a starfree regular language $L_n \notin \mathcal{B}_{n-1/2}$ such that L_n plt-reduces to a language in $\mathcal{B}_{1/2}$ [Gla05]. So by (1), $\forall O, \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}^O}(L_n) \subseteq \Sigma_n^{\mathrm{P}^O}$, but $L_n \notin \mathcal{B}_{n-1/2}$. This shows that the levels of the dot-depth hierarchy are not closed under plt-reducibility even if we restrict ourselves to starfree regular languages. Contrary to that, we will prove that things are different for ptt-reducibility. We can show that restricted to regular languages, the classes $\mathcal{B}_0$, $\mathcal{B}_{1/2}$, $\mathcal{B}_1$, and $\mathcal{B}_{3/2}$ are closed under ptt-reducibility. (Here and in the following, this formulation means that for instance $\mathcal{R}_m^{ptt}(\mathcal{B}_1) \cap \mathrm{REG} = \mathcal{B}_1$ where $\mathcal{R}_m^{ptt}(\mathcal{B}_1)$ denotes $\mathcal{B}_1$'s closure under ptt-reducibility.) It follows that for every regular language L the following holds:

$$L \in \mathcal{B}_0 \quad \Leftrightarrow \quad \forall O(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}^O}(L) \subseteq \mathrm{P}^O) \tag{5}$$

$$L \in \mathcal{B}_{1/2} \quad \Leftrightarrow \quad \forall O(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}^O}(L) \subseteq \mathrm{NP}^O) \tag{6}$$

$$L \in \mathcal{B}_1 \quad \Leftrightarrow \quad \forall O(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}^O}(L) \subseteq \mathrm{BC}(\mathrm{NP})^O) \tag{7}$$

$$L \in \mathcal{B}_{3/2} \quad \Leftrightarrow \quad \forall O(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}^O}(L) \subseteq \Sigma_2^{\mathrm{P}^O}) \tag{8}$$

We consider this observation as evidence that restricted to regular languages, all levels of the dot-depth hierarchy might be closed under ptt-reducibility. This would turn (4) into an equivalence.

By (5)–(8), at least on the lower levels, the dot-depth hierarchy exactly corresponds to robust inclusions of unbalanced leaf-language classes in the polynomial-time hierarchy. This correspondence does not hold for balanced leaf-language classes. So this shows that unbalanced leaf-language classes are indeed well-worth being considered. This motivates the study of ptt-reducibility which is the suitable reducibility for the unbalanced model.

Note that Borchert and Silvestri [BS97] showed that for every class $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L)$, there exists an L' such that $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L')$. So from the plain definability point of view, we can restrict ourselves to balanced leaf-languages. However, a shortcoming of this point of view is that it obliterates the inherent connection of a leaf-language and the complexity class defined by it. Naturally, the leaf-language for a complexity class should be as simple as possible, i.e., the language should capture the properties of the class it describes as close as possible. L' can be much more complex than L, and this brings the unbalanced model into play again. In some cases, the unbalanced model can describe a complexity class by a simpler language than the balanced model can do.

Perfect Correspondence. We describe another aspect of ptt-reducibility. The perfect correspondence between the dot-depth hierarchy and the polynomial-time hierarchy allows to prove statements like the following which are due to Borchert, Kuske, Stephan, and Schmitz.

Theorem 1.1 ([Bor95, BKS99, Sch01]) *Let L be a regular language.*

1. *[Bor95] If $L \in \mathcal{B}_0$, then $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \subseteq \mathrm{P}$. If $L \notin \mathcal{B}_0$, then $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{NP}$ or $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{coNP}$ or $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{MOD}_p\mathrm{P}$ for a prime p.*
2. *[BKS99] If $L \in \mathcal{B}_{1/2}$, then $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \subseteq \mathrm{NP}$. If $L \notin \mathcal{B}_{1/2}$, then $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{coNP}$ or $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{co1NP}$ or $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{MOD}_p\mathrm{P}$ for a prime p.*
3. *[Sch01] If $L \in \mathcal{B}_{3/2}$, then $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \subseteq \Sigma_2^{\mathrm{P}}$. If $L \notin \mathcal{B}_{3/2}$, then $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \forall{\cdot}\mathrm{UP}$ or $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{co}\exists!{\cdot}\mathrm{UP}$ or $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{MOD}_p\mathrm{P}$ for a prime p.*

In view of this theorem we say that $\mathcal{B}_0$ and P (resp., $\mathcal{B}_{1/2}$ and NP, $\mathcal{B}_{3/2}$ and Σ_2^{P}) *perfectly correspond*. For instance, consider $\mathcal{B}_{1/2}$ and NP. By Burtschick and Vollmer [BV98], $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(\mathcal{B}_{1/2}) = \mathrm{NP}$. In addition, Theorem 1.1 states that the languages in $\mathcal{B}_{1/2}$ are the only regular languages L such that $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L)$ is robustly contained in NP. Hence, $\mathcal{B}_{1/2}$ and NP perfectly correspond. With help of plt-reducibility and the new ptt-reducibility, we can make the notion of perfect correspondence precise.

1. A class of regular languages $\mathcal{C}$ and a complexity class $\mathcal{D}$ *perfectly correspond* with respect to balanced leaf-languages if (restricted to regular languages) $\mathcal{C}$ is closed under plt-reducibility and $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(\mathcal{C}) = \mathcal{D}$.
2. A class of regular languages $\mathcal{C}$ and a complexity class $\mathcal{D}$ *perfectly correspond* with respect to unbalanced leaf-languages if (restricted to regular languages) $\mathcal{C}$ is closed under ptt-reducibility and $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(\mathcal{C}) = \mathcal{D}$.

The following perfect correspondences with respect to unbalanced leaf-languages are easily obtained from known results [Bor95, BKS99, Sch01].

- $\mathcal{B}_0$ perfectly corresponds to P.
- $\mathcal{B}_{1/2}$ perfectly corresponds to NP.
- $\mathcal{B}_{3/2}$ perfectly corresponds to Σ_2^{P}.

We show that restricted to regular languages, $\mathcal{B}_1$ is closed under ptt-reducibility. From this we obtain a new perfect correspondence with respect to unbalanced leaf-languages:

- $\mathcal{B}_1$ perfectly corresponds to the Boolean closure of NP.

It follows that above the Boolean hierarchy over NP there is a gap in unbalanced leaf-language definability: If $\mathcal{D} = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(\mathcal{C})$ for some class $\mathcal{C}$ of regular languages, then $\mathcal{D} \subseteq \mathrm{BC(NP)}$ or there exists an oracle O such that $\mathcal{D}^O \not\subseteq \mathrm{P}^{\mathrm{NP}[\epsilon\cdot\log n]^O}$ for all $\epsilon < 1$.

Our investigations of the ptt-reducibility show the following phenomenon: While we can (unconditionally) prove that level 0 of the dot-depth hierarchy is closed under ptt-reducibility, we can show the similar property for higher levels only if we restrict ourselves to regular languages. We can construct a language $B \in \mathrm{NP} \setminus \mathrm{REG}$ that is ptt-reducible to a language in $\mathcal{B}_{1/2}$. The exception of level 0 allows to improve the correspondence between $\mathcal{B}_0$ and P: Not only that $\mathcal{B}_0$ and P perfectly correspond, but in fact it even holds that for any language $L \notin \mathcal{B}_0$ (this includes all nonregular languages) there exists an oracle O such that $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}O}(L) \not\subseteq \mathrm{P}^O$.

Organization of the Paper. Section 3 defines ptt-reducibility. In section 4 we formulate and prove the central result of this paper, the unbalanced analogue of the BCSV-theorem. Section 5 studies the ptt-closure of classes of the dot-depth hierarchy, and it shows that on some lower levels, the dot-depth hierarchy perfectly corresponds to the polynomial-time hierarchy.

2 Preliminaries

For a machine or automaton M, let $L(M)$ denote the accepted language. For a finite alphabet Σ, the *initial word relation* $\sqsubseteq$ on Σ^* is defined by

$$u \sqsubseteq v \overset{df}{\Longleftrightarrow} \exists w(w \in \Sigma^* \wedge uw = v).$$

We write $u \sqsubset v$ if and only if $u \sqsubseteq v$ and $u \neq v$. The lexicographical order on $\{0,1\}^*$ is defined by

$$x \preceq y \overset{df}{\Longleftrightarrow} x \sqsubseteq y \vee \exists u(u0 \sqsubseteq x \wedge u1 \sqsubseteq y).$$

The quasi-lexicographical order on $\{0,1\}^*$ is defined by

$$x \leq y \overset{df}{\Longleftrightarrow} |x| < |y| \vee (|x| = |y| \wedge \exists u(u0 \sqsubseteq x \wedge u1 \sqsubseteq y)) \vee x = y.$$

In what follows we identify the set $\{0,1\}^*$ with the set $\mathbb{N}$ of natural numbers according to the quasi-lexicographical order. So $\{0,1\}^*$ inherits operations like $+$ from the natural numbers. Furthermore, we identify a set $O \subseteq \mathbb{N}$ with the characteristic sequence $c_O(0)c_O(1)c_O(2)\cdots \in \{0,1\}^\omega$ where c_O is the characteristic function of O. For a set $O \subseteq \{0,1\}^\omega$ and $u \in \{0,1\}^*$ we define the following sets.

$$\begin{aligned} u|O &=_{\text{def}} uc_O(|u|)c_O(|u|+1)c_O(|u|+2)\cdots \\ uO &=_{\text{def}} uc_O(0)c_O(1)c_O(2)\cdots \end{aligned}$$

The following theorem shows the close relation between the dot-depth hierarchy and the polynomial-time hierarchy. Here $\mathrm{NP}(n)$ denotes level n of the Boolean hierarchy over NP. PLT is the class of languages that have polylog-time computable characteristic functions where the input is accessed as an oracle.

Theorem 2.1 ([HLS+93, BV98, BKS99]) *The following holds for $n \geq 1$ and relative to all oracles.*

1. $\mathrm{P} = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\mathrm{PLT}) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\mathcal{B}_0) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathcal{B}_0)$

2. $\Sigma^{\mathrm{P}}_n = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\mathcal{B}_{n-1/2}) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathcal{B}_{n-1/2})$

3. $\Pi^{\mathrm{P}}_n = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\mathrm{co}\mathcal{B}_{n-1/2}) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathrm{co}\mathcal{B}_{n-1/2})$

4. $\mathrm{BC}(\Sigma^{\mathrm{P}}_n) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\mathcal{B}_n) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathcal{B}_n)$

5. $\mathrm{NP}(n) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\mathcal{B}_{1/2}(n)) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathcal{B}_{1/2}(n))$

Bovet, Crescenzi, and Silvestri [BCS92] and Vereshchagin [Ver93] showed that polylog-time reducibility exactly corresponds to balanced leaf-language definable classes.

Definition 2.2 *A function $f : A^* \to A^*$ is* polylog-time computable *if there exist two polynomial-time-bounded oracle transducers $R : A^* \times \mathbb{N} \to A$ and $l : A^* \to \mathbb{N}$ such that for all x,*

$$f(x) = R^x(|x|, 1)R^x(|x|, 2) \cdots R^x(|x|, l^x(|x|))$$

where R and l access the input x as an oracle. A language B is polylog-time reducible *(plt-reducible) to a language C, $B \leq^{\mathrm{plt}}_{\mathrm{m}} C$ for short, if there exists a polylog-time computable f such that for all x, $x \in B \Leftrightarrow f(x) \in C$.*

Theorem 2.3 ([BCS92, Ver93]) *For all languages B and C,*

$$B \leq^{\mathrm{plt}}_{\mathrm{m}} C \;\Leftrightarrow\; \forall O\big(\,\mathrm{Leaf}^{\mathrm{p}^O}_{\mathrm{b}}(B) \subseteq \mathrm{Leaf}^{\mathrm{p}^O}_{\mathrm{b}}(C)\big).$$

Let $\mathcal{D}$ be a complexity class. A language L belongs to the class $\exists^{\mathrm{u}}\cdot\mathcal{D}$ if there exist a polynomial p and $B \in \mathcal{D}$ such that:

$$\begin{array}{lcl} x \in L & \Rightarrow & (\exists! y, |y| \leq p(|x|))[(x, y) \in B] \\ x \notin L & \Rightarrow & (\forall y, |y| \leq p(|x|))[(x, y) \notin B] \end{array}$$

Analogously, L belongs to $\forall^{\mathrm{u}}\cdot\mathcal{D}$ if there exist a polynomial p and $B \in \mathcal{D}$ such that:

$$\begin{array}{lcl} x \in L & \Rightarrow & (\forall y, |y| \leq p(|x|))[(x, y) \in B] \\ x \notin L & \Rightarrow & (\exists! y, |y| \leq p(|x|))[(x, y) \notin B] \end{array}$$

3 Polynomial-Time Tree Reducibility

With polynomial-time tree reducibility (ptt-reducibility for short) we introduce the unbalanced analog of polylog-time reducibility (plt-reducibility). For the representation of a balanced computation tree it suffices to think of a leaf string such that each symbol is accessible in polylog-time in the length of the leaf string. Representations of unbalanced computation trees are more complicated. Here the particular structure of the tree must be taken into account. This makes it necessary to define suitable representations of trees. Intuitively, a language B ptt-reduces to a language C if there exists a polynomial-time (in the height of the tree) computable function that transforms trees such that for every tree t, the leafstring of t belongs to B if and only if the leafstring of $f(t)$ belongs to C.

We start with representations of trees. Let Σ be a finite alphabet. A triple $t = (T, h, m)$ is called a Σ-*tree* if $T \subseteq \{0,1\}^*$ is finite, $h : T \to \Sigma$, and $m \in \mathbb{N}$ such that $\forall z \forall u((u \sqsubseteq z \wedge z \in T) \to u \in T)$ and $\forall z(z \in T \to |z| \le m)$. Let T_Σ be the set of all Σ-trees. A *leaf* of t is a $z \in T$ such that there is no $u \in T$ with $z \sqsubset u$. For a Σ-tree $t = (T, h, m)$, we define the *leaf word* of t as $\beta(t) =_{\text{def}} h(z_1)h(z_2)\cdots h(z_s)$ where $\{z_1, z_2, \ldots, z_s\}$ is the set of all leaves of t and $z_1 \prec z_2 \prec \cdots \prec z_s$.

Choose $r \ge 1$ such that $|\Sigma| \le 2^r$, and let $e : \Sigma \to \{0,1\}^r$ be an injective mapping. A Σ-tree $t = (T, h, m)$ is encoded by the set $O_t =_{\text{def}} \{ze(h(z)) \mid z \in T\}$ and the number $m_t =_{\text{def}} m$. On the other hand, an arbitrary set $O \subseteq \{0,1\}^*$ and a number $m \in \mathbb{N}$ define a Σ-tree $t_{O,m} =_{\text{def}} (T_{O,m}, h_{O,m}, m)$ where

$$\begin{aligned} T_{O,m} &=_{\text{def}} \{z \mid |z| \le m \wedge \forall u(u \sqsubseteq z \to \exists v(v \in e(\Sigma) \wedge uv \in O))\} \quad \text{and} \\ h_{O,m}(z) &=_{\text{def}} e^{-1}(\text{lexicographically first } v \in e(\Sigma) \text{ such that } zv \in O). \end{aligned}$$

It is easy to see that $t_{O_t, m_t} = t$ for every Σ-tree t. Now let us define functions that transform unbalanced computation trees.

Definition 3.1 *Let Σ_1 and Σ_2 be finite alphabets. A function $f : T_{\Sigma_1} \to T_{\Sigma_2}$ is called a* polynomial-time tree function *(ptt-function for short) if there exist $k > 0$ and functions $g_1 : T_{\Sigma_1} \times \{0,1\}^* \times \mathbb{N} \to \{0,1\}$ and $g_2 : T_{\Sigma_1} \times \{0,1\}^* \times \mathbb{N} \to \Sigma_2$ such that:*

- *There exists a polynomial $p(\cdot,\cdot)$ such that $g_1(t, z, m)$ and $g_2(t, z, m)$ are computable in time $p(|z|, m)$ where the tree t is accessed as the oracle O_t.*
- *It holds that $f(t) = (T', h', m_t^k + k)$ where $T' =_{\text{def}} \{z \mid g_1(t, z, m_t) = 1\}$ and $h'(z) =_{\text{def}} g_2(t, z, m_t)$.*

We will also write $g_1^{O_t}(z, m)$ and $g_2^{O_t}(z, m)$ instead of $g_1(t, z, m)$ and $g_2(t, z, m)$, respectively. Finally we define polynomial-time tree reducibility.

Definition 3.2 *For $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$, we define $L_1 \le_{\text{m}}^{\text{ptt}} L_2$ (L_1 is* ptt-reducible *to L_2) if there exists a ptt-function $f : T_{\Sigma_1} \to T_{\Sigma_2}$ such that for all $t \in T_{\Sigma_1}$,*

$$\beta(t) \in L_1 \leftrightarrow \beta(f(t)) \in L_2.$$

Proposition 3.3 $\le_{\text{m}}^{\text{ptt}}$ *is reflexive and transitive.*

We describe how parts of the characteristic sequence of a set O can be interpreted as trees: For any $x \in \{0,1\}^*$ and any $O \subseteq \{0,1\}^*$, let

$$\begin{aligned} O - x &=_{\text{def}} \{y \mid y + x \in O\} \quad \text{and} \\ O + x &=_{\text{def}} \{y \mid y - x \in O\}. \end{aligned}$$

Observe that $O - x + x = \{y \in O \mid y \ge x\}$ and $O + x - x = O$. Moreover, note that the characteristic sequence of $O - x$ is exactly $c_O(x)c_O(x+1)c_O(x+2)\cdots$. For $L \subseteq \Sigma^*$ and $O \subseteq \{0,1\}^*$ we define

$$\begin{aligned} L^O &=_{\text{def}} \{x \mid x \in \{0,1\}^* \text{ and } \beta(t_{O-x,|x|}) \in L\} \quad \text{and} \\ L^\circ &=_{\text{def}} \{(O, x) \mid x \in \{0,1\}^* \text{ and } \beta(t_{O-x,|x|}) \in L\}. \end{aligned}$$

Proposition 3.4 *Let $L \in \Sigma^*$, $O_1, O_2 \subseteq \{0,1\}^*$, and $x, u, v \in \{0,1\}^*$.*

1. *If $|u| = |v| \leq x$, then $x \in L^{uO_1} \leftrightarrow x \in L^{vO_1}$.*
2. *If $|u| > 2^{r+3} \cdot x$, then $x \in L^{uO_1} \leftrightarrow x \in L^{uO_2}$. (Note that r is the constant that was chosen at the beginning of this section such that $|\Sigma| \leq 2^r$.*

Proof Assume we want to check $x \in L^{O_1}$. So we have to consider the tree $t_{O_1 - x, |x|}$ which consists of paths of length $\leq |x|$. The latter are described by the words in $O_1 - x$ that are of length $\leq |x| + r$. There are less than $2^{|x|+r+1} \leq x \cdot 2^{r+2}$ such words. So in order to figure out whether $x \in L^{O_1}$, we only need to know $i \in O_1$ for $i \in \{x, x+1, \ldots, x + x \cdot 2^{r+2}\}$. □

Some more notations are needed for the proof of the unbalanced BCSV-theorem in section 4. Let M be a nondeterministic polynomial-time Turing machine (NPTM, for short) such that on input x, M produces on every computation path z a symbol $M(x,z)$ from a finite alphabet Σ. Let k be the smallest natural number such that $n^k + k$ bounds the running time of M. For every computation path z of M on input x, let

$$\begin{aligned} T_M(x) &=_{\text{def}} \{u \,|\, \exists z (z \text{ computation path of } M \text{ on } x \text{ and } u \sqsubseteq z)\} \quad \text{and} \\ h_M(x)(z) &=_{\text{def}} M(x,z). \end{aligned}$$

For the other $z \in T_M(x)$ the value of $h_M(x)(z)$ is chosen arbitrarily from Σ. The computation tree of M on x is

$$t_M(x) =_{\text{def}} (T_M(x), h_M(x), |x|^k + k).$$

Note that $t_M(x)$ is a Σ-tree. For a nondeterministic polynomial-time oracle Turing machine (NPTOM, for short) M, we define the computation tree of M on x with oracle O as

$$t_M^O(x) =_{\text{def}} (T_M^O(x), h_M^O(x), |x|^k + k).$$

For a language $L \subseteq \Sigma^*$, define $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L)$ as the class of all languages B for which there exists an NPTM M such that for all x,

$$x \in B \Leftrightarrow \beta(t_M(x)) \in L.$$

For a fixed oracle $O \subseteq \{0,1\}^*$, let $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}O}(L)$ be the class of all languages B for which there exists an NPTOM M such that for all x,

$$x \in B \Leftrightarrow \beta(t_M^O(x)) \in L.$$

Finally, let $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}\circ}(L)$ be the class of all sets B for which there exists an NPTOM M such that for all x and all oracles O,

$$(O,x) \in B \Leftrightarrow \beta(t_M^O(x)) \in L.$$

Proposition 3.5 *Let L be a language.*

1. *$L^O \in \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}O}(L)$ for every oracle O.*
2. *$L^\circ \in \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}\circ}(L)$.*

A language $L \subseteq \Sigma^*$ is called *nontrivial* if $L \neq \emptyset$ and $L \neq \Sigma^*$.

4 The BCSV-Theorem for Unbalanced Leaf Languages

Let B and C be languages. Bovet, Crescenzi, and Silvestri [BCS92] and Vereshchagin [Ver93] proved that B polylog-time reduces to C if and only if for all oracles O, $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}^O}(B) \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}^O}(C)$. So plt-reducibility corresponds to robust inclusions of balanced leaf-language classes. We show that ptt-reducibility and unbalanced leaf-language classes share the same connection.

Theorem 4.1 *For nontrivial $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ the following are equivalent:*

(1) $L_1 \leq_{\mathrm{m}}^{\mathrm{ptt}} L_2$

(2) $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^\circ}(L_1) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^\circ}(L_2)$

(3) $L_1^\circ \in \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^\circ}(L_2)$

Proof (1) $\Rightarrow$ (2). Let $L_1 \leq_{\mathrm{m}}^{\mathrm{ptt}} L_2$ via ptt-function f. For $B \in \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^\circ}(L_1)$ there exists an NPTOM M such that $(O, x) \in B \Leftrightarrow \beta(t_M^O(x)) \in L_1$ for all $x \in \Sigma^*$ and oracles O. It is easy to construct an NPTOM M' such that $\beta(t_{M'}^O(x)) = \beta(f(t_M^O(x)))$ for all x and O.[1] Consequently,

$$(O, x) \in B \Leftrightarrow \beta(t_M^O(x)) \in L_1 \Leftrightarrow \beta(f(t_M^O(x))) \in L_2 \Leftrightarrow \beta(t_{M'}^O(x)) \in L_2,$$

and hence $B \in \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^\circ}(L_2)$.

(2) $\Rightarrow$ (3) is obvious because of Proposition 3.5.2

(3) $\Rightarrow$ (1). Let $L_1^\circ \in \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^\circ}(L_2)$. There exists an NPTOM M such that $(O, x) \in L_1^\circ \Leftrightarrow \beta(t_M^O(x)) \in L_2$. Let k be the smallest natural number such that $n^k + k$ bounds the running time of M. For a Σ-tree t we obtain $O_t = O_t + 0^{m_t} - 0^{m_t}$ and therefore,

$$\beta(t) \in L_1 \Leftrightarrow \beta(t_{O_t, m_t}) \in L_1 \Leftrightarrow (O_t + 0^{m_t}, 0^{m_t}) \in L_1^\circ \Leftrightarrow \beta(t_M^{O_t + 0^{m_t}}(0^{m_t})) \in L_2.$$

Define $f(t) =_{\mathrm{def}} t_M^{O_t + 0^{m_t}}(0^{m_t}) = (T_M^{O_t + 0^{m_t}}(0^{m_t}), h_M^{O_t + 0^{m_t}}(0^{m_t}), m_t^k + k)$. Observe that there exist polynomial-time computable functions g_1, g_2 such that $T_M^{O_t + x}(0^{m_t}) = \{z \mid g_1^t(z, m_t) = 1\}$ and $h_M^{O_t + x}(0^{m_t}) = g_2^t(z, m_t)$. Hence $L_1 \leq_{\mathrm{m}}^{\mathrm{ptt}} L_2$. $\square$

Theorem 4.2 *For nontrivial $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ the following are equivalent:*

(1) $L_1 \leq_{\mathrm{m}}^{\mathrm{ptt}} L_2$

(2) $\forall O(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L_1) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L_2))$

(3) $\forall O(L_1^O \in \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L_2))$

[1] Note that we cannot guarantee $t_{M'}^O(x) = f(t_M^O(x))$. Every inner node of $t_{M'}^O(x)$ has exactly two children, since $t_{M'}^O(x)$ is a computation tree. In contrast, $f(t_M^O(x))$ is an arbitrary tree which by our definition can contain inner nodes that have only one child. However, $\beta(t_{M'}^O(x)) = \beta(f(t_M^O(x)))$ is possible, since from $f(t_M^O(x))$ we obtain a computation tree by deleting such nodes.

Proof (1) $\Rightarrow$ (2) follows from (1) $\Rightarrow$ (2) in Theorem 4.1, and (2) $\Rightarrow$ (3) is obvious because of Proposition 3.5.1.

(3) $\Rightarrow$ (1). Because of Theorem 4.1 it suffices to prove $L_1^\circ \in \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}\circ}(L_2)$. Let $M_0, M_1, M_2, \dots$ be an enumeration of all NPTOMs. Assume $L_1^\circ \notin \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}\circ}(L_2)$, i.e., for every k there exist an oracle O_k and a word x_k such that

$$x_k \in L_1^{O_k} \leftrightarrow \beta(t_{M_k}^{O_k}(x_k)) \notin L_2. \tag{9}$$

Claim 4.3 *There exist $u_0, u_1, u_2, \dots \in \{0,1\}^*$ and an $z_0, z_1, z_2, \dots \in \mathbb{N}$ such that*

- *u_{k-1} is a proper initial word of u_k, and*
- *$z_k \in L_1^O \leftrightarrow \beta(t_{M_k}^O(z_k)) \notin L_2$ for all $O \in u_k \cdot \{0,1\}^\omega$.*

Proof of Claim 4.3. By induction on k. For $k = 0$, let $z_0 = x_0$, and let u_0 be the shortest initial word of $O_0 \in \{0,1\}^\omega$ such that $|u_0| > 2^{r+3} \cdot x_0$ and $\{0, 1, \dots, |u_0|\}$ contains every query of M_0 to the oracle O_0 during its work on x_0. For every $O \in u_0 \cdot \{0,1\}^\omega$ we obtain using Proposition 3.4.2 and (9)

$$\begin{aligned} z_0 \in L_1^O &\Leftrightarrow x_0 \in L_1^O \Leftrightarrow x_0 \in L_1^{O_0} \Leftrightarrow \beta(t_{M_0}^{O_0}(x_0)) \notin L_2 \\ &\Leftrightarrow \beta(t_{M_0}^{O}(x_0)) \notin L_2 \Leftrightarrow \beta(t_{M_0}^{O}(z_0)) \notin L_2. \end{aligned}$$

Now assume that we have already constructed $u_0, u_1, \dots, u_k \in \{0,1\}^*$ and $z_0, x_1, \dots, x_k \in \mathbb{N}$ that satisfy the claim. We construct u_{k+1} and z_{k+1} as follows. Consider an NPTOM M such that for every input x and every oracle O the following holds. If $x \geq |u_k|$, then M works as M_{k+1} on x with oracle $u_k|O$. If $x < |u_k|$, then M works in such a way that $x \in L_1^O \leftrightarrow \beta(t_{M_r}^O(x)) \in L_2$. Choose r such that $M = M_r$.

If $x_r < |u_k|$, then $x_r \in L_1^{O_r} \leftrightarrow \beta(t_{M_r}^{O_r}(x_r)) \in L_2$ which contradicts (9). Therefore, $x_r \geq |u_k|$, and consequently, using Proposition 3.4.1 and (9),

$$x_r \in L_1^{u_k|O_r} \Leftrightarrow x_r \in L_1^{O_r} \Leftrightarrow \beta(t_{M_r}^{O_r}(x_r)) \notin L_2 \Leftrightarrow \beta(t_{M_{k+1}}^{u_k|O_r}(x_r)) \notin L_2. \tag{10}$$

Now define $z_{k+1} =_{\mathrm{def}} x_r$ and let u_{k+1} be the shortest initial word of $u_k|O_r$ such that $|u_{k+1}| > |u_k|$, $|u_{k+1}| > 2^{r+3} \cdot x_r$, and $\{0, 1, \dots, |u_{k+1}|\}$ contains every query of M_{k+1} to the oracle $u_k|O_r$ during its work on input x_r. Hence u_k is a proper initial word of u_{k+1}, and by Proposition 3.4.2 and (10) we obtain for all $O \in u_{k+1} \cdot \{0,1\}^\omega$

$$\begin{aligned} z_{k+1} \in L_1^O &\Leftrightarrow x_r \in L_1^O \Leftrightarrow x_r \in L_1^{u_k|O_r} \Leftrightarrow \beta(t_{M_{k+1}}^{u_k|O_r}(x_r)) \notin L_2 \\ &\Leftrightarrow \beta(t_{M_{k+1}}^{O}(x_r)) \notin L_2 \Leftrightarrow \beta(t_{M_{k+1}}^{O}(z_{k+1})) \notin L_2. \end{aligned}$$

This completes the induction and proves Claim 4.3.

Now define $O' =_{\mathrm{def}} \lim_{k\to\infty} u_k \cdot 0^\omega$, from which we obtain $O' \in u_k \cdot \{0,1\}^\omega$ for all $k \geq 0$. By the claim, $z_k \in L_1^{O'} \leftrightarrow \beta(t_{M_k}^{O'}(z_k)) \notin L_2$ for every $k \geq 0$. This means $L_1^{O'} \notin \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}O'}(L_2)$ which contradicts the assumption of (3). $\square$

5 ptt-Reducibility and the Dot-Depth Hierarchy

By Theorem 2.1 the levels of the dot-depth hierarchy and the levels of the polynomial-time hierarchy are closely related. Note that this connection exists for both models, balanced and unbalanced leaf-languages. In this section we discuss evidence that for the unbalanced model this connection is even closer than that stated in Theorem 2.1.

Definition 5.1 *A class of regular languages $\mathcal{C}$ and a complexity class $\mathcal{D}$* perfectly correspond *with respect to balanced leaf-languages if (restricted to regular languages) $\mathcal{C}$ is closed under plt-reducibility and* $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\mathcal{C}) = \mathcal{D}$*. A class of regular languages $\mathcal{C}$ and a complexity class $\mathcal{D}$* perfectly correspond *with respect to unbalanced leaf-languages if (restricted to regular languages) $\mathcal{C}$ is closed under ptt-reducibility and* $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathcal{C}) = \mathcal{D}$*.*

Perfect correspondences are connections closer than those stated in Theorem 2.1.

Proposition 5.2 *If $\mathcal{C}$ perfectly corresponds to $\mathcal{D}$ with respect to balanced leaf-languages, then for every regular $L \notin \mathcal{C}$ there exists an oracle relative to which* $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\mathcal{C}) \not\subseteq \mathcal{D}$*. The similar statement holds for unbalanced leaf-languages.*

Proof Follows from Theorems 2.3 and 4.2. □

The levels of the dot-depth hierarchy and the levels of the polynomial-time hierarchy do not perfectly correspond with respect to balanced leaf-languages. In particular, for $n \geq 1$, $\mathcal{B}_{n/2}$ is not closed under plt-reducibility even if we restrict ourselves to starfree regular languages.

Theorem 5.3 *For every $n \geq 1$, $\mathcal{B}_{n-1/2}$ does not perfectly correspond to Σ_n^{P} with respect to balanced leaf-languages.*

Proof For every $n \geq 1$, there exists $L_n \in \mathrm{SF} - \mathcal{B}_{n-1/2}$ such that L_n plt-reduces to a language in $\mathcal{B}_{1/2}$ [Gla05]. □

In contrast, we will see that restricted to regular languages, the classes $\mathcal{B}_0$, $\mathcal{B}_{1/2}$, $\mathcal{B}_1$, and $\mathcal{B}_{3/2}$ are closed under ptt-reducibility. In particular, these classes perfectly correspond to the classes of the polynomial-time hierarchy. While for $\mathcal{B}_0$, $\mathcal{B}_{1/2}$, and $\mathcal{B}_{3/2}$ the latter is easily obtained from known results [Bor95, BKS99, Sch01], this is a new result for $\mathcal{B}_1$. We consider these results as evidence that restricted to regular languages, all levels of the dot-depth hierarchy might be closed under ptt-reducibility and therefore, perfectly correspond to the levels of the polynomial-time hierarchy.

Unlike all classes $\mathcal{B}_{n/2}$ for $n \geq 1$ (see Theorem 5.12), the class $\mathcal{B}_0$ is closed under ptt-reducibility even without the restriction to regular languages.

Theorem 5.4 $\mathcal{R}_m^{ptt}(\mathcal{B}_0) = \mathcal{B}_0$.

Proof Let $L \subseteq \Sigma^*$ be $\leq_{\mathrm{m}}^{\mathrm{ptt}}$-reducible to a language in $\mathcal{B}_0$. Hence $L \leq_{\mathrm{m}}^{\mathrm{ptt}} \{1\}$, i.e., there exists a ptt-function f such that for all Σ-trees t,

$$\beta(t) \in L \Leftrightarrow \beta(f(t)) = 1.$$

Without loss of generality we can assume that $\beta(t) \notin L$ implies $\beta(f(t)) = 0$. Let g_1 and g_2 be the polynomial-time computable functions that define f. Let M_1 and M_2 be polynomial-time machines computing g_1 and g_2, resp., in time n^k for suitable $k \geq 1$.

Claim 5.5 *For every $wxv \in L$ with $|x| > \max((3\log_2|wxv|)^{k+1}, 1)$ there exist $r > 1$ and $v_1, v_2, \ldots, v_r$ such that*

1. $|v_1 v_2 \cdots v_r| \leq (3\log_2|wxv|)^{k+1}$,
2. $wxv \in wv_1\Sigma^* v_2\Sigma^* \cdots \Sigma^* v_r v$, *and*
3. $wv_1\Sigma^* v_2\Sigma^* \cdots \Sigma^* v_r v \cap \Sigma^{\leq 2|wxv|} \subseteq L$.

The same holds true for $\overline{L}$.

Proof of Claim 5.5. Consider wxv such that $|x| > \max((3\log_2|wxv|)^{k+1}, 1)$, and let $m =_{\mathrm{def}} 3\log_2|wxv|$. From $|x| \geq 2$ we obtain $m \geq 3$. Let $t = (T, h, m)$ be a balanced Σ-tree such that $\beta(t) = wxv$. Since $\beta(f(t)) \in \{0, 1\}$, the tree $f(t)$ has only one path. To compute this path, at most $2m^k \leq m^{k+1} - 2$ paths of t are queried by M_1 or M_2. So we can factorize wxv as

$$wxv = wv_1 u_1 v_2 u_2 \cdots u_{r-1} v_r v$$

such that all u_i and v_i are nonempty, $|v_1 v_2 \cdots v_r| \leq m^{k+1}$, and no queried path goes through a symbol in u_i. Since $|x| > m^{k+1}$ we have $r \geq 2$. Also note that $r \leq m^{k+1}$, since the v_i are nonempty. Now we cut all paths that correspond to symbols in u_i. This results in a new Σ-tree $t_1 = (T_1, h_1, m)$ such that $\beta(t_1) = wv_1 v_2 \cdots v_r v$. Clearly,

$$wxv \in wv_1\Sigma^* v_2\Sigma^* \cdots \Sigma^* v_{r-1}\Sigma^* v_r v.$$

Now consider arbitrary $z_1, z_2, \ldots, z_{r-1}$ such that $|wv_1 z_1 v_2 z_2 \cdots z_{r-1} v_r v| \leq 2|wxv|$. If π_i is the path corresponding to the first symbol of u_i, then we add π_i to the tree, and additionally, we attach a tree with leaf word z_i to π_i. This results in a Σ-tree $t_2 = (T_2, h_2, m)$ such that

$$\beta(t_2) = wv_1 z_1 v_2 z_2 v_3 \cdots v_{r-1} z_{r-1} v_r v.$$

Note that height m still suffices, since the length of π_i is less than or equal to $\lceil \log_2|wxv| \rceil$ and since $|z_i| \leq 2|wxv|$. M_1 and M_2 do not query paths that go through symbols in u_i. Therefore, these machines cannot distinguish between the Σ-trees t and t_2. Consequently,

$$wxv \in L \Leftrightarrow wv_1 z_1 v_2 z_2 v_3 \cdots v_{r-1} z_{r-1} v_r v \in L.$$

This proves Claim 5.5.

Define N_0 as a natural number such that $N_0 > \max(2(3\log_2 N_0)^{k+1}, 1)$, and consequently $N > 2(3\log_2 N)^{k+1}$ for every $N \geq N_0$.

Claim 5.6 *For every $x \in L$ with $|x| \geq 2^{N_0}$ there exist w, v such that $|w| = |v| = (3\log_2|x|)^{k+1}$, $x \in w\Sigma^* v$, and $w\Sigma^* v \subseteq L$.*

Proof of Claim 5.6. Let $x \in L$ such that $|x| \geq 2^{N_0} \geq N_0 > 1$, and let $m =_{\text{def}} 3\log_2|x|$. Hence $|x| > 2m^{k+1} \geq m^{k+1}$. By Claim 5.5, there exist $r > 1$ and $w_1, w_2, \cdots, w_r$ such that $|w_1 w_2 \cdots w_r| \leq m^{k+1}$ and

$$x \in w_1\Sigma^* w_2\Sigma^* \cdots \Sigma^* w_r \cap \Sigma^{\leq |x|} \subseteq L.$$

Because of $|x| \geq 2m^{k+1}$ and $|w_1 w_r| \leq m^{k+1}$, we can choose words u_1, u_2, u_3 such that $x = w_1 u_1 u_2 u_3 w_r$ and $|w_1 u_1| = |u_3 w_r| = m^{k+1}$. Assume there exists a z' such that $w_1 u_1 z' u_3 w_r \in \overline{L}$. Let $z =_{\text{def}} u_1 z' u_3$. So $w_1 z w_r \in \overline{L}$ and

$$|z| \geq |u_1 u_3| = |w_1 u_1| + |u_3 w_r| - |w_1 w_r| \geq 2m^{k+1} - m^{k+1} = m^{k+1}.$$

Thus there exists a z such that $w_1 z w_r \in \overline{L}$ and $|z| \geq m^{k+1}$ (and hence $|z| \geq N_0$). Let z be of minimum length with this property. Observe $|z| \geq m^{k+1} \geq |w_1 w_r|$ and $|z| \geq 3\log_2|x| \geq 3$. Since $2|z| \geq N_0$ we obtain $2|z| > 2(3\log_2 2|z|)^{k+1}$ and hence

$$|z| > (3\log_2 2|z|)^{k+1} \geq (3\log_2|w_1 z w_r|)^{k+1}.$$

From Claim 5.5 we obtain $s > 1$ and $v_1, v_2, \cdots, v_s$ such that

$$w_1 v_1\Sigma^* v_2\Sigma^* \cdots \Sigma^* v_s w_r \cap \Sigma^{\leq 2|w_1 z w_r|} \subseteq \overline{L}$$

and

$$|v_1 v_2 \cdots v_s| \leq (3\log_2|w_1 z w_r|)^{k+1} < |z|.$$

From $w_1 v_1 v_2 \cdots v_s w_r \in \overline{L}$ and from the minimality of z we obtain $|v_1 v_2 \cdots v_s| < m^{k+1}$.

So far we have seen

$$w_1\Sigma^* w_2\Sigma^* \cdots \Sigma^* w_r \cap \Sigma^{\leq |x|} \subseteq L \tag{11}$$

and

$$w_1 v_1\Sigma^* v_2\Sigma^* \cdots \Sigma^* v_s w_r \cap \Sigma^{\leq 2|z|} \subseteq \overline{L}. \tag{12}$$

Now observe that

$$\begin{aligned} |w_1 v_1 w_2 w_3 \cdots w_{r-1} v_2 v_3 \cdots v_s w_r| = |w_1 w_2 \cdots w_r| + |v_1 v_2 \cdots v_s| &\leq 2m^{k+1} \\ &\leq \min\{|x|, 2|z|\}. \end{aligned}$$

Together with (11) and (12) this implies $w_1 v_1 w_2 w_3 \cdots w_{r-1} v_2 v_3 \cdots v_s w_r \in L \cap \overline{L}$ which is a contradiction. This means that there is no z' such that $w_1 u_1 z' u_3 w_r \in \overline{L}$. Consequently, the statement of the claim is fulfilled by $w =_{\text{def}} w_1 u_1$ and $v =_{\text{def}} u_3 w_r$. This proves Claim 5.6.

By Claim 5.6, for every $x \in L$ with $|x| \geq 2^{N_0}$ there exist words w, v such that $|w| = |v| = (3\log_2|x|)^{k+1}$ and $x \in w\Sigma^* v \subseteq L$. Since $wv \in L$ and $|wv| = 2(3\log_2|x|)^{k+1} < |x|$ we can apply Claim 5.6 repeatedly until we obtain w', v' such that $|w'| = |v'| = 2^{N_0}$ and $x \in w\Sigma^* v w\Sigma^* v \subseteq L$. Hence

$$L = \bigcup_{\substack{x \in L,\\ |x| < 2^{N_0}}} \{x\} \cup \bigcup_{\substack{wv \in L,\\ |w|=|v|=2^{N_0}}} w\Sigma^* v.$$

This shows $L \in \mathcal{B}_0$. $\square$

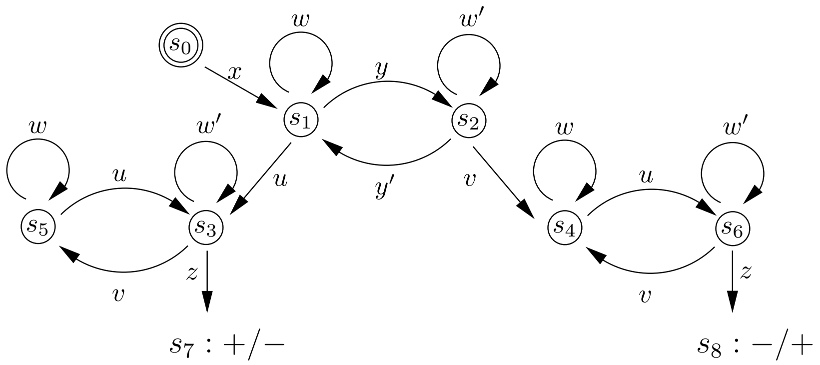

Figure 1: Pattern P_1 where w, w' are nonempty - Nonexistence of this pattern characterizes $\mathcal{B}_1$.

Theorem 5.7 $\mathcal{R}_m^{ptt}(\mathcal{B}_{1/2}) \cap \mathrm{REG} = \mathcal{B}_{1/2}$.

Proof It suffices to argue for the inclusion from left to right. Assume there exists $L \in \mathcal{R}_m^{ptt}(\mathcal{B}_{1/2}) \cap \mathrm{REG}$ such that $L \notin \mathcal{B}_{1/2}$. So there exists $L' \in \mathcal{B}_{1/2}$ such that $L \leq_{\mathrm{m}}^{\mathrm{ptt}} L'$. Hence for all oracles O, $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L') \subseteq \mathrm{NP}^O$. By Borchert, Kuske, and Stephan [BKS99], for all oracles O, $\mathrm{coUP}^O \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L)$. By Theorem 4.2, for all oracles O, $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L')$ and therefore, $\mathrm{coUP}^O \subseteq \mathrm{NP}^O$. This contradicts an oracle construction by Eppstein et al. [EHTY92]. $\square$

Lemma 5.8 *Let $L \in \mathrm{REG} \setminus \mathcal{B}_1$. Then there exists an oracle B such that $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^B}(L) \not\subseteq \mathrm{P}^{\mathrm{NP}[\epsilon \cdot \log n]^B}$ for all $\epsilon < 1$.*

Proof Let A be an alphabet with $|A| \geq 2$ and $L \subseteq A^*$ such that $L \in \mathrm{REG} \setminus \mathcal{B}_1$. Hence, the minimal automaton of L contains pattern P_1 (see Figure 1) and there exist $u, v, x, y, y', z \in A^*$ and $w, w' \in A^+$ as apparent in Figure 1. Without loss of generality, we assume that the minimal automaton contains the first version of the pattern, i.e., state s_7 is accepting and state s_8 is rejecting. Let L_{P_1} be the language of all words in $x\{u, v, w, w', y, y'\}^* z$ such that the minimal automaton of L moves along the paths drawn in Figure 1 and finally reaches s_7. Let L'_{P_1} be the similar set of words leading to s_8. Clearly, $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_{P_1}, L'_{P_1}) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L)$.[2] We construct B such that for all $\epsilon < 1$,

$$\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^B}(L_{P_1}, L'_{P_1}) \not\subseteq \mathrm{P}_{\|}^{\mathrm{NP}[n^\epsilon]^B}.$$

This implies that for all $\epsilon < 1$,

$$\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^B}(L_{P_1}, L'_{P_1}) \not\subseteq \mathrm{P}^{\mathrm{NP}[\epsilon \cdot \log(n)]^B}.$$

Let $e \notin A$ be a new letter. For $n \in \mathbb{N}$ let $\alpha_{0,n} \prec \alpha_{1,n} \prec \ldots \prec \alpha_{2^n-1,n}$ be the words of $\{0,1\}^n$ in lexicographical order. For any set $D \subseteq \{0,1\}^*$ with characteristic function c_D, the *characteristic sequence of D restricted to words of length n* is defined as $C_D(n) =_{\mathrm{def}} c_D(\alpha_{0,n}) c_D(\alpha_{1,n}) \ldots c_D(\alpha_{2^n-2,n}) c_D(\alpha_{2^n-1,n})$. Such a characteristic sequence can be considered as

[2] Note that here a pair of languages (L_{P_1}, L'_{P_1}) defines a leaf-language class. A language belongs to $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_{P_1}, L'_{P_1})$ if and only if there exists a nondeterministic polynomial-time-bounded Turing machine M such that for all x: If $x \in L$ then $\beta_M(x) \in L_{P_1}$; if $x \notin L$ then $\beta_M(x) \in L'_{P_1}$. This can be used to define promise classes, as in this case.

a sequence of letters from $A \cup \{e\}$ where $\lceil \log(|A|+1) \rceil$ bits of $C_D(n)$ encode a letter from $A \cup \{e\}$. Denote this new sequence by $C'_D(n)$ and observe that its length is greater than $2^{n-|A|}$. Let $C'_D(n)_{|A}$ be the sequence obtained by removing all e's from $C'_D(n)$. We say that the sequence $C_D(n)$ is *valid for pattern* P_1 if the following holds:

- $C'_D(n)$ does not contain a factor e^{n+1}, and
- $C'_D(n)_{|A} \in L_{P_1} \cup L'_{P_1}$.

We call a valid sequence *accepted (resp., rejected) by pattern* P_1 if it belongs to L_{P_1} (resp., L'_{P_1}). Hence, a valid sequence $C_D(n)$ encodes a sequence $C'_D(n)$ over $A \cup \{e\}$ which may contain only short e-blocks.

We will define a fast-growing tower function $t : \mathbb{N} \to \mathbb{N}$ such that $t(n+1) = 2^{t(n)}$ for $n \geq 0$. For an arbitrary oracle O, we define our witness language W^O as follows:

$$W^O =_{\text{def}} \{0^{t(n)} \,\big|\, n \geq 0 \text{ and } C_O(t(n)) \text{ is accepted by pattern } P_1\}$$

Throughout the construction we will ensure that for all n, the sequence $C_B(t(n))$ is valid for pattern P_1. This implies $W^B \in \text{Leaf}_{\text{u}}^{\text{p}B}(L_{P_1}, L'_{P_1})$: On input 0^m, an unbalanced machine first verifies that $m = t(n)$ for some n, and then produces a computation tree with leaf string $C'_O(m)$. Since $C'_O(m)$ only contains short blocks of e's, this machine can reorganize its computation tree such that all e's are removed from the leaf string. So it remains to show that $W^B \notin \text{P}^{\text{NP}[\epsilon \cdot \log(n)]^B}$.

Our oracle B will be defined as the union of (finite) oracle stages $B_i, i \geq 1$, which are constructed iteratively. Each stage B_n is characterized by oracle words of length $t(n)$ and therefore by the sequence $C_B(t(n))$. Let $B[k,j] =_{\text{def}} \bigcup_{k \leq i \leq j} B_i$ denote an interval of oracle stages.

We enumerate $\text{P}_{\|}^{\text{NP}[m^\epsilon]}$-machines as follows. Consider an enumeration of all tuples (M, N, p, ϵ) such that M is a deterministic polynomial-time oracle Turing machine, N is a nondeterministic polynomial-time oracle Turing machine, p is a polynomial and $\epsilon < 1$. We interpret M as the base machine and N as the oracle machine.

By defining the first value $t(0)$ of the tower function sufficiently large and $t(n+1) =_{\text{def}} 2^{t(n)}$, we can ensure that the enumeration satisfies the following technical requirements. For the n-th tuple of the enumeration, (M, N, p, ϵ), all of the following holds:

1. $p(t(n)) \leq 2^{\log^2 t(n)}$
2. $3 \log^2 t(n) \leq t(n)^{(1-\epsilon)/2}$
3. $2^{t(n)} / 2^{t(n)^{(1+\epsilon)/2}} \geq 2 \cdot |A| \cdot |ww'yy'uv|$
4. Let the running times of M and N be bounded by polynomials q and r, respectively. Then it holds that $r(q(n)) \leq p(n)$.
5. M on input x asks at most $|x|^\epsilon$ nonadaptive queries to the oracle $L(N)$.

Let (M, N, p, ϵ) be the n-th tuple in our enumeration and let $m = t(n)$. We diagonalize against (M, N, p, ϵ) through ensuring

$$L(M^{B[1,n], L(N^{B[1,n]})}) \neq W^{B[1,n]}. \tag{13}$$

Notice that M can access both oracles, $B[1,n]$ and $L(N^{B[1,n]})$.

We describe the main idea behind the diagonalization against (M,N,p,ϵ): We start with an oracle B_n such that $C_B(m)$ is accepted by P_1. After that we simulate M with the so-far constructed oracle ($B[1,n]$) on input 0^m and determine segments in B_n that have to be reserved. If M rejects 0^m we are done for this stage. Otherwise we change B_n on non-reserved positions, such that $C_B(m)$ is still *valid* but now rejected by P_1 (here the e's compensate length differences). We then repeat the simulation of M on input 0^m with the modified oracle and update the list of reserved segments. If M still accepts we are done, otherwise we modify non-reserved positions such that $C_B(m)$ remains valid but accepted by P_1 again. We will show that after $\epsilon \cdot \log m$ such rounds, M on input 0^m will err in its decision.

The detailed construction of the diagonalization against (M,N,p,ϵ) follows.

We define

$$\beta =_{\text{def}} xw\gamma wuz,$$

such that $\gamma \in \{w,e\}^*$, γ does not contain a factor e^{m+1}, and $|\beta| = 2^{m-|A|}$. We start with $B_n \subseteq \{0,1\}^m$ such that $C'_{B_n}(m) = \beta$. Clearly, $C_{B_n}(m)$ is accepted by pattern P_1: Whether a valid sequence is accepted or rejected is determined by the first occurrence of a word from $\{u,v\}$ in the encoded sequence; for u the sequence is accepted, for v it is rejected.

Let F denote the set of reserved segments; $F=\emptyset$ at the beginning. F is supposed to contain words of length m that we will not modify in the further construction. Simulate $M^{B[1,n]}$ on input 0^m. If M rejects, (13) is fulfilled and the construction of stage B_n is complete. So assume M accepts. Let Q_1 be the set of M's queries to B_n on input 0^m. Thus, $|Q_1| \leq p(m)$. Let $q_1,\dots,q_k$ be M's nonadaptive queries to N where $k \leq m^\epsilon$. Let $Q_+ \subseteq \{q_1,\dots,q_k\}$ be the set of positively answered queries. Hence, for $q \in Q_+$, the nondeterministic machine N on input q produces at least one accepting path. We define $Q_2 =_{\text{def}} \{q \,\big|\, \exists q' \in Q_+(N \text{ on input } q' \text{ queries } q \text{ on its leftmost accepting path})\}$. Observe that $|Q_2| \leq p(m)^2$. We now set $F = F \cup Q_1 \cup Q_2$. Since $|F| \leq p(m)^3$ and $|C_{B_n}(m)| = 2^m$, there exist $2^m/p(m)^3$ consecutive words of length m that are not in F. These words represent a segment s in β. By the construction of β, $s \in \{w,e\}^*$. In the next step, s is replaced by a segment $s' \in y\{w',e\}^*v$ such that $|s'| = |s|$ and s does not contain a factor e^{m+1}. Observe that the purpose of e in this construction is to compensate differences in the lengths of y, w, w' and v. After this modification, $C_{B_n}(m)$ is still valid but now rejected by P_1. Since all further modifications in later rounds will be restricted to the segment s', we reserve all the rest of the oracle at this stage, i.e., F now contains all words from $\{0,1\}^m$ except those encoding s'.

Again, we simulate $M^{B[1,n]}$ on input 0^m and now assume that it has noticed the deception and thus rejects. Let Q_3 be the set of queries to B_n during this simulation. Since $Q_2 \subseteq F$, no query in Q_+ can have flipped from positive to negative. Consequently, there have to be queries in $\{q_1,\dots,q_k\} \setminus Q_+$ which have been answered positively by N during the second simulation of M. Let Q'_+ be the set of these queries. We repeat the above construction by defining the set $Q_4 =_{\text{def}} \{q \,\big|\, \exists q' \in Q'_+(N \text{ on input } q' \text{ queries } q \text{ on its leftmost accepting path})\}$. We have $|Q_3| \leq p(m)$ and $|Q_4| \leq p(m)^2$. Set $F = F \cup Q_3 \cup Q_4$. Hence, we still find

$$\frac{2^m}{p(m)^3 \cdot p(m)^3}$$

consecutive words of length m that are not in F. These correspond to a segment $s_1 \in \{w',e\}^*$ which has not been reserved yet. This segment is replaced by a segment $s'_1 \in y'\{w,e\}^*u$ with $|s_1| = |s'_1|$. This modification causes $C_{B_n}(m)$ to be accepted by P_1.

We can deceive M again by repeating the above procedure. After at most k rounds, no more of M's queries to N can flip from negative to positive. At that point, M cannot change its behavior any longer. Each round the size of the non-reserved area of $\{0,1\}^m$ is divided by at most $p(m)^3$. Hence after k rounds we still have a segment of size

$$\frac{2^m}{p(m)^{3k}} \geq \frac{2^m}{p(m)^{3m^\epsilon}} \geq \frac{2^m}{2^{3m^\epsilon(\log^2 m)}} \geq \frac{2^m}{2^{m^{(1-\epsilon)/2}m^\epsilon}} = \frac{2^m}{2^{m^{(1+\epsilon)/2}}} \geq 2 \cdot |A| \cdot |ww'yy'uv|.$$

Therefore, after k rounds we can still find a sufficiently large non-reserved area. We can then modify this segment to deceive M one final time. $\square$

Utilizing Theorem 4.2, we can translate this oracle separation into a statement about the ptt-closure of $\mathcal{B}_1$.

Theorem 5.9 $\mathcal{R}_m^{ptt}(\mathcal{B}_1) \cap \mathrm{REG} = \mathcal{B}_1$.

Proof It suffices to argue for the inclusion from left to right. Assume there exists $L \in \mathcal{R}_m^{ptt}(\mathcal{B}_1) \cap \mathrm{REG}$ such that $L \notin \mathcal{B}_1$. So there exists $L' \in \mathcal{B}_1$ such that $L \leq_{\mathrm{m}}^{\mathrm{ptt}} L'$. By Theorem 4.2, for all oracles O, we then have $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L')$. Theorem 2.1 holds relative to all oracles. Therefore, for all oracles O, it holds that $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L') \subseteq BC(\mathrm{NP})^O$. This contradicts Lemma 5.8. $\square$

As a consequence, we obtain the first gap theorem of leaf-language definability above the Boolean closure of NP.

Corollary 5.10 *Let $\mathcal{D} = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(\mathcal{C})$ for some $\mathcal{C} \subseteq \mathrm{REG}$. Then $\mathcal{D} \subseteq \mathrm{BC(NP)}$ or there exists an oracle O such that $\mathcal{D}^O \not\subseteq \mathrm{P}^{\mathrm{NP}[\epsilon \cdot \log n]^O}$ for all $\epsilon < 1$.*

Theorem 5.11 $\mathcal{R}_m^{ptt}(\mathcal{B}_{3/2}) \cap \mathrm{REG} = \mathcal{B}_{3/2}$.

Proof It suffices to argue for the inclusion from left to right. Assume there exists $L \in \mathcal{R}_m^{ptt}(\mathcal{B}_{3/2}) \cap \mathrm{REG}$ such that $L \notin \mathcal{B}_{3/2}$. So there exists $L' \in \mathcal{B}_{3/2}$ such that $L \leq_{\mathrm{m}}^{\mathrm{ptt}} L'$. Hence for all oracles O, $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L') \subseteq \Sigma_2^{pO}$. By Schmitz [Sch01], for all oracles O, $\forall^{\mathrm{u}} \cdot \exists^{\mathrm{u}} \cdot \mathrm{P}^O \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L)$. By Theorem 4.2, for all oracles O, $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}^O}(L')$ and therefore, $\forall^{\mathrm{u}} \cdot \exists^{\mathrm{u}} \cdot \mathrm{P}^O \subseteq \Sigma_2^{pO}$. This contradicts an oracle construction by Spakowski and Tripathi [ST04]. $\square$

By Theorem 5.4, $\mathcal{B}_0$ is closed under ptt-reducibility. As stated in the Theorems 5.7, 5.9, and 5.11, the classes $\mathcal{B}_{1/2}$, $\mathcal{B}_1$, and $\mathcal{B}_{3/2}$ are closed under ptt-reducibility if we restrict ourselves to regular languages. We explain this difference and show that the restriction to regular languages is crucial: For $k \geq 1$, $\mathcal{B}_{k/2}$ is not closed under ptt-reducibility.

Theorem 5.12 *There exists $B \in \mathrm{NP} \setminus \mathrm{REG}$ such that $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(B) \subseteq \mathrm{NP}$.*

Proof We use the pairing function $\langle\cdot,\cdot\rangle$ that is defined as follows for letters a_i and b_i.

$$\langle a_1a_2\cdots a_k, b_1b_2\cdots b_l\rangle =_{\text{def}} 0a_10a_2\cdots 0a_k1b_11b_2\cdots 1b_l$$

Let $N_1, N_2, \ldots$ be an enumeration of nondeterministic polynomial-time-bounded Turing machines such that N_i on inputs of length n has running time $n^i + i$. We may assume that given i, one can determine the machine N_i in polynomial-time in $|i|$.

Every word appears as leaf string of a suitable computation. This changes if we demand that the leaf string is generated by a short input. A word w is called *honestly generated* if it is generated by a machine N_i on input of a sufficiently small word x. We make this precise with the definition of B which consists of all honestly generated words.

$$B =_{\text{def}} \{w \mid (\exists i \leq |w|/2)(\exists x \in A^*, |x|^i + i < |w|)[\beta_{N_i}(x) = w]\}$$

Assume we are given w, i, and x as above. The running time of N_i on x is $|x|^i + i < |w|$. Therefore, in time $O(|w|^2)$ we can determine the machine N_i, can simulate the first $|w|$ computation paths of $N_i(x)$, and can test whether $\beta_{N_i}(x) = w$. This shows $B \in \mathrm{NP}$.

Let $n \geq 2$ and $1 \leq i \leq n/2$. We estimate $|B \cap A^n|$ as follows.

$$|B \cap A^n| \leq \sum_{i=1}^{n/2} |\{x \in A^* \mid |x| \leq (n-i-1)^{1/i}\}| \leq \sum_{i=1}^{n/2} 2^{n-i} = 2^n \sum_{i=1}^{n/2} 2^{-i} < 2^n$$

This shows that at least one word of any length belongs to $\overline{B}$. In particular, $\overline{B}$ is infinite.

We argue that $B \notin \mathrm{REG}$. For this we start with the description of a nondeterministic machine N on input $\langle \mathcal{M}, k\rangle$ where k is a natural number and $\mathcal{M}$ is a deterministic finite automaton. First, N deterministically computes nonempty words u, v, z such that for all $i \geq 0$, $uv^iz \notin L(\mathcal{M})$. If such words do not exist, then N generates the leaf string 0. Otherwise, in a nondeterministic way N generates the leaf string uv^kz. Observe that the words u, v, z, if they exist, can be computed in polynomial-time which shows that N is polynomial-time bounded. Therefore, $N = N_j$ for some $j \geq 1$.

Assume $B \in \mathrm{REG}$, i.e., $B = L(\mathcal{M})$ for some finite automaton $\mathcal{M}$. Choose l sufficiently large such that $l \geq 2j$ and $l > |\langle \mathcal{M}, l\rangle|^j + j$. Let $x =_{\text{def}} \langle \mathcal{M}, l\rangle$ and $w =_{\text{def}} \beta_{N_j}(x)$. Since $\overline{B}$ is infinite, there exist nonempty words u, v, z such that for all $i \geq 0$, $uv^iz \notin L(\mathcal{M})$. Therefore, for suitable such words it holds that $w = uv^lz \notin L(\mathcal{M})$. So $j \leq |w|/2$ and $|w| > |x|^j + j$. It follows that $w \in B - L(\mathcal{M})$ which contradicts the assumption $B = L(\mathcal{M})$ and which shows $B \notin \mathrm{REG}$.

Finally we show $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(B) \subseteq \mathrm{NP}$. Fix any $j \geq 1$ and let $L = \{x \mid \beta_{N_j}(x) \in B\}$. It suffices to show $L \in \mathrm{NP}$. Let x be an arbitrary word of length ≥ 2. Define $w =_{\text{def}} \beta_{N_j}(x)$ and observe

$$\begin{aligned} x \in L \quad &\Leftrightarrow \quad w \in B \\ &\Leftrightarrow \quad (|x|^j + j < |w|) \vee (|x|^j + j \geq |w| \wedge w \in B). \end{aligned}$$

The first $|x|^j + j$ letters of the leaf string w can be determined in polynomial-time in $|x|$. So the condition $|x|^j + j < |w|$ is decidable in polynomial-time in $|x|$. If $|x|^j + j \geq |w|$, then $w \in B$ can be decided in nondeterministic polynomial-time in $|x|$. Hence the condition on the right-hand side is decidable in NP which shows $L \in \mathrm{NP}$. □

Corollary 5.13 *1. There exists $B \in \mathrm{NP} \setminus \mathrm{REG}$ such that $B \in \mathcal{R}_m^{ptt}(\mathcal{B}_{1/2})$.*

2. *For every $k \geq 1$, $\mathcal{B}_{k/2}$ is not closed under $\leq_{\mathrm{m}}^{\mathrm{ptt}}$-reducibility.*

Proof Let $C =_{\mathrm{def}} \{0,1\}^*1\{0,1\}^*$ and define B as in Theorem 5.12. There we show $B \in \mathrm{NP} \setminus \mathrm{REG}$ and $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(B) \subseteq \mathrm{NP}$. The argument for the latter inclusion is relativizable. Therefore, for all oracles O, $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}O}(B) \subseteq \mathrm{NP}^O = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}O}(C)$. By Theorem 4.2, $B \leq_{\mathrm{m}}^{\mathrm{ptt}} C$ and hence $B \in \mathcal{R}_m^{ptt}(\mathcal{B}_{1/2})$. This shows the first statement and the second one follows immediately. □

We state an upper bound for the complexity of the $\leq_{\mathrm{m}}^{\mathrm{ptt}}$-closure of regular languages.

Theorem 5.14 $\mathcal{R}_m^{ptt}(\mathrm{REG}) \subseteq \bigcup_{k \geq 1} \mathrm{DSPACE}(\log^k n)$.

Proof Let $L \in \mathcal{R}_m^{ptt}(\mathrm{REG})$, i.e., there exists $L' \in \mathrm{REG}$ such that $L \leq_{\mathrm{m}}^{\mathrm{ptt}} L'$ via ptt-function f. So there exist $k > 0$ and functions g_1 and g_2 as in Definition 3.1. Both functions are polynomial-time computable when the tree is accessed as an oracle. For a word x, let t_x denote the balanced binary tree that has leaf string x.

Let $m = \lceil \log |x| \rceil^k + k$. We describe an algorithm that computes $\beta(f(t_x))$: Consider all strings z of length $\leq m$ in lexicographical order. If $g_1(t_x, z, \lceil \log |x| \rceil) = 1$, then output $g_2(t_x, z, \lceil \log |x| \rceil)$. Consider the next string z.

This algorithm computes $\beta(f(t_x))$, since it exactly simulates f. If t_x is accessed as oracle, then $g_1(t_x, z, \lceil \log |x| \rceil)$ and $g_1(t_x, z, \lceil \log |x| \rceil)$ are computable in polynomial time in $\log |x|$. Given x, an oracle access to t_x can be simulated in logarithmic space. Therefore, the algorithm above can be simulated in polylogarithmic space in $|x|$. Given $\beta(f(t_x))$, we can test in constant space whether $\beta(f(t_x)) \in L'$. The theorem follows, since

$$x \in L \Leftrightarrow \beta(t_x) \in L \Leftrightarrow \beta(f(t_x)) \in L'.$$

□

Due to this theorem, we can now specify the complexity of nonregular sets C such that $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(C) \subseteq \mathrm{NP}$.[3] Accordingly it is unlikely that such sets are NP-complete. In particular, this applies to the set B that was used in Theorem 5.12 and Corollary 5.13.

Corollary 5.15 *Let C be a set. Then the following holds: If $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}O}(C) \subseteq \mathrm{NP}^O$ for all oracles O, then $C \in \bigcup_{k \geq 1} \mathrm{DSPACE}(\log^k n)$.*

Proof For all oracles O, $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}O}(C) \subseteq \mathrm{NP}^O = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}O}(0^*1\{0,1\}^*)$. So $C \leq_{\mathrm{m}}^{\mathrm{ptt}} 0^*1\{0,1\}^*$ and hence $C \in \mathcal{R}_m^{ptt}(\mathrm{REG}) \subseteq \bigcup_{k \geq 1} \mathrm{DSPACE}(\log^k n)$. □

Since $\mathrm{PSPACE} = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(\mathrm{REG})$ [HLS+93], the last corollary remains valid if we replace NP by PSPACE.

[3] Recall that for regular sets, we already know by Theorem 5.7 that only languages in $\mathcal{B}_{1/2}$ come into question.

Acknowledgments

We thank Bernd Borchert, Heinz Schmitz, Victor Selivanov, and Pascal Tesson for helpful discussions about leaf languages.

References

[Arf91] M. Arfi. Opérations polynomiales et hiérarchies de concaténation. *Theoretical Computer Science*, 91:71–84, 1991.

[BCS92] D. P. Bovet, P. Crescenzi, and R. Silvestri. A uniform approach to define complexity classes. *Theoretical Computer Science*, 104:263–283, 1992.

[BK78] J. A. Brzozowski and R. Knast. The dot-depth hierarchy of star-free languages is infinite. *Journal of Computer and System Sciences*, 16:37–55, 1978.

[BKS99] B. Borchert, D. Kuske, and F. Stephan. On existentially first-order definable languages and their relation to NP. *Theoretical Informatics and Applications*, 33:259–269, 1999.

[Bor95] B. Borchert. On the acceptance power of regular languages. *Theoretical Computer Science*, 148:207–225, 1995.

[Brz76] J. A. Brzozowski. Hierarchies of aperiodic languages. *RAIRO Inform. Theor.*, 10:33–49, 1976.

[BS97] B. Borchert and R. Silvestri. A characterization of the leaf language classes. *Information Processing Letters*, 63(3):153–158, 1997.

[BV98] H.-J. Burtschick and H. Vollmer. Lindström quantifiers and leaf language definability. *International Journal of Foundations of Computer Science*, 9:277–294, 1998.

[CB71] R. S. Cohen and J. A. Brzozowski. Dot-depth of star-free events. *Journal of Computer and System Sciences*, 5:1–16, 1971.

[EHTY92] D. Eppstein, L. A. Hemachandra, J. Tisdall, and B. Yener. Simultaneous strong separations of probabilistic and unambiguous complexity classes. *Mathematical Systems Theory*, 25:23–36, 1992.

[Eil76] S. Eilenberg. *Automata, languages and machines*, volume B. Academic Press, New York, 1976.

[Gin66] A. Ginzburg. About some properties of definite, reverse-definite and related automata. *IEEE Transactions on Electronic Computers EC-15*, pages 806–810, 1966.

[Gla05] C. Glaßer. Polylog-time reductions decrease dot-depth. In *Proceedings 22nd Symposium on Theoretical Aspects of Computer Science*, volume 3404 of *Lecture Notes in Computer Science*. Springer Verlag, 2005.

[HLS$^+$93] U. Hertrampf, C. Lautemann, T. Schwentick, H. Vollmer, and K. W. Wagner. On the power of polynomial time bit-reductions. In *Proceedings 8th Structure in Complexity Theory*, pages 200–207, 1993.

[PP86] D. Perrin and J. E. Pin. First-order logic and star-free sets. *Journal of Computer and System Sciences*, 32:393–406, 1986.

[PW97] J. E. Pin and P. Weil. Polynomial closure and unambiguous product. *Theory of computing systems*, 30:383–422, 1997.

[Sch01] H. Schmitz. *The Forbidden-Pattern Approach to Concatenation Hierarchies*. PhD thesis, Fakultät für Mathematik und Informatik, Universität Würzburg, 2001.

[ST04] H. Spakowski and R. Tripathi. On the power of unambiguity in alternating machines. Technical Report 851, University of Rochester, 2004.

[Sto77] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[Tho84] W. Thomas. An application of the Ehrenfeucht–Fraïssé game in formal language theory. *Société Mathématique de France, mémoire 16*, 2:11–21, 1984.

[Ver93] N. K. Vereshchagin. Relativizable and non-relativizable theorems in the polynomial theory of algorithms. *Izvestija Rossijskoj Akademii Nauk*, 57:51–90, 1993. In Russian.

[Wag04] K. W. Wagner. Leaf language classes. In *Proceedings International Conference on Machines, Computations, and Universality*, volume 3354 of *Lecture Notes in Computer Science*. Springer Verlag, 2004.

[Wra77] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3:23–33, 1977.