Electronic Colloquium on Computational Complexity, Report No. 11 (2004)

Counting with Counterfree Automata

Christian Glaßer
University at Würzburg, Germany
glasser@informatik.uni-wuerzburg.de

16th January 2004

Abstract

We study the power of balanced regular leaf-languages. First, we investigate (i) regular languages that are polylog-time reducible to languages in dot-depth $1/2$ and (ii) regular languages that are polylog-time decidable. For both classes we provide

- forbidden-pattern characterizations, and
- characterizations in terms of regular expressions.

Both classes are decidable. The intersection of class (i) with their complement is exactly class (ii).

We apply our observations and obtain three consequences.

1. Gap theorems for balanced regular-leaf-language definable classes $\mathcal{C}$ and $\mathcal{D}$:
   (a) Either $\mathcal{C}$ is contained in NP, or $\mathcal{C}$ contains coUP.
   (b) Either $\mathcal{D}$ is contained in P, or $\mathcal{D}$ contains UP or coUP.

   Also we extend both theorems such that no promise classes are involved. Formerly, such gap theorems were known only for the unbalanced approach.
2. Polylog-time reductions can tremendously decrease dot-depth complexity (despite that they cannot count). We exploit a weak type of counting possible with counterfree automata, and construct languages of arbitrary dot-depth that are reducible to languages in dot-depth $1/2$.
3. Unbalanced starfree leaf-languages can be much stronger than balanced ones. We construct starfree regular languages L_n such that the balanced leaf-language class of L_n is contained in NP, but the unbalanced leaf-language class of L_n contains level n of the unambiguous alternation hierarchy. This demonstrates the power of unbalanced computations.

1 Introduction

Regular languages are described by regular expressions. These consist of single letters which are combined by three types of operations: Boolean operations, concatenation, and iteration. If we forbid iteration, then these restricted regular expressions define *starfree regular languages* (starfree languages for short). The class of these languages is denoted by SF. SF is a subclass of REG, the class of regular languages.

Throughout the paper all automata are deterministic. Call a minimal finite automaton *permutationfree* if the following holds for all $n \geq 2$, all words w, and all states s: If s on input w^n leads to s, then already s on input w leads to s. While regular languages are accepted by finite automata, starfree languages are accepted by *permutationfree finite automata* [Sch65, MP71] (permutationfree automata for short). By definition, if any state in a permutationfree automaton has a loop w^n, then it has already a loop w. Therefore, permutationfree automata cannot count the length of their inputs modulo some $m > 1$. For this reason, McNaughton and Papert [MP71] call such automata *counterfree*.

ISSN 1433-8092

Dot-Depth Hierarchy. The most interesting aspect of starfree languages is the *dot-depth hierarchy* which was introduced by Brzozowski and Cohen [CB71, Brz76]. The dot-depth measures the complexity of starfree languages. It counts the minimal number of nested alternations between Boolean operations and concatenation that is needed to define a language. Classes of the dot-depth hierarchy consist of languages that have the same dot-depth. Fix some finite alphabet that has at least two letters (the hierarchy collapses for unary alphabets). Define $\mathrm{Pol}(\mathcal{C})$, the polynomial closure of $\mathcal{C}$, to be $\mathcal{C}$'s closure under finite (possibly empty) union and finite (possibly empty) concatenation. Similarly, let $\mathrm{BC}(\mathcal{C})$ be the Boolean closure of $\mathcal{C}$. For $n \geq 0$ define the classes (or levels) of the dot-depth hierarchy:

$$\begin{aligned} \mathcal{B}_0 &\stackrel{df}{=} \{L \mid L \text{ is finite or cofinite}\} \\ \mathcal{B}_{n+\frac{1}{2}} &\stackrel{df}{=} \mathrm{Pol}(\mathcal{B}_n) \\ \mathcal{B}_{n+1} &\stackrel{df}{=} \mathrm{BC}(\mathcal{B}_{n+\frac{1}{2}}) \end{aligned}$$

The dot-depth of some language L is defined as the minimal m such that $L \in \mathcal{B}_m$ where $m = n/2$ for some integer n. At first glance, the definition of levels $n + 1/2$ looks a bit artificial. The reason for this kind of definition is of historic nature: Originally, Brzozowski and Cohen were interested in the full levels $\mathcal{B}_n$ and therefore, defined the dot-depth hierarchy in this way. Later Pin and Weil [PW97] considered both, the levels $\mathcal{B}_n$ and their polynomial closure. To be consistent with Brzozowski and Cohen, they extended the dot-depth hierarchy by half levels $\mathcal{B}_{n+1/2}$.

By definition, all levels of the dot-depth hierarchy are closed under union and it is known that they are also closed under intersection, under taking inverse morphisms, and under taking residuals [PP86, Arf91, PW97]. The dot-depth hierarchy is strict [BK78, Tho84] and it exhausts the class of starfree languages [Eil76].

Does there exist an algorithm that computes the dot-depth on input of a starfree language? This question is known as the *dot-depth problem*. Today, more than 30 years after it was discovered by Brzozowski and Cohen, it is still an open problem. Most researchers consider the dot-depth problem as one of the most difficult problems in automata theory.

The problem remains hard, if we ask for decidability of single classes of the dot-depth hierarchy. However, we know that the 4 lowest classes of the dot-depth hierarchy are decidable. $\mathcal{B}_0$ is decidable for trivial reasons. Pin and Weil [PW97] proved decidability of $\mathcal{B}_{1/2}$, Knast [Kna83] proved decidability of $\mathcal{B}_1$, and Glaßer and Schmitz [GS00] proved decidability of $\mathcal{B}_{3/2}$. Other levels are not known to be decidable, but it is widely believed that they are. The decidability results for $\mathcal{B}_{1/2}$ and $\mathcal{B}_{3/2}$ share an interesting property: Both classes, $\mathcal{B}_{1/2}$ and $\mathcal{B}_{3/2}$, have *forbidden-pattern characterizations*. This means that a language L belongs to $\mathcal{B}_{1/2}$ if and only if L's minimal automaton does *not* have a certain pattern. This immediately implies decidability.

Weak Counting. We come back to the result by Schützenberger [Sch65] and McNaughton and Papert [MP71]: The class of languages accepted by permutationfree automata (or counterfree automata) is exactly the class of starfree languages. This suggests the intuition that starfree languages cannot count (modulo some $m > 1$). For instance the set of even-length words is not starfree. In contrast, there do exist starfree subset of all even-length words. This is possible, since sometimes counting can be reformulated as local properties. For instance $L = (01)^*$ is starfree, since a word belongs to L if and only if it starts with 0, ends with 1, and neither has 00 nor has 11 as factor. This example adjusts our intuition: Starfree languages cannot do arbitrary counting, but counting in a restricted sense is possible. In this paper we will exploit the phenomenon of weak counting.

Leaf Languages. The concept of leaf languages was independently introduced by Bovet, Crescenzi, and Silvestri [BCS92] and Vereshchagin [Ver93]. Let M be any nondeterministic polynomial-time-bounded Turing machine such that every computation path stops and outputs one letter. $M(x)$ denotes the computation tree on input x. 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 $M(x)$. Let $\text{leafstring}^M(x)$ be the concatenation of all outputs of $M(x)$. For any language B, let $\text{Leaf}_u^p(B)$ be the class of languages L such that there exists an (unbalanced) nondeterministic polynomial-time-bounded Turing machine M as above such that for all x,

$$x \in L \iff \text{leafstring}^M(x) \in B.$$

If we assume M to be a balanced, nondeterministic polynomial-time-bounded Turing machine, then this defines the class $\text{Leaf}_b^p(B)$. For any complexity class $\mathcal{C}$ let $\text{Leaf}_u^p(\mathcal{C}) = \bigcup_{B \in \mathcal{C}} \text{Leaf}_u^p(B)$ and $\text{Leaf}_b^p(\mathcal{C}) = \bigcup_{B \in \mathcal{C}} \text{Leaf}_b^p(B)$. If $\mathcal{C} \subseteq \text{REG}$ and $\mathcal{D} = \text{Leaf}_u^p(\mathcal{C})$, then we say that $\mathcal{D}$ is a *unbalanced regular-leaf-language definable class*. Analogously define *balanced regular-leaf-language definable classes*. Since in this paper, $\mathcal{C}$ will always be a subclass of REG, we will use the term (un)balanced leaf-language definable class as abbreviation.

Connection between Hierarchies. Starfree languages have a very nice connection with complexity theory. In the concept of leaf languages, classes of the dot-depth hierarchy correspond exactly to classes of the polynomial-time hierarchy. For $n \geq 1$,

$$\text{Leaf}_b^p(\mathcal{B}_{n-1/2}) = \text{Leaf}_u^p(\mathcal{B}_{n-1/2}) = \Sigma_n^P.$$

This connection allows a translation of knowledge about dot-depth classes into knowledge about complexity classes. Here the forbidden-pattern characterizations come into play. They allows us to identify gaps between leaf-language definable complexity classes. We sketch this very nice approach with help of an example which goes back to Pin and Weil [PW97] and Borchert, Kuske, and Stephan [BKS98].

Consider $\mathcal{B}_{1/2}$. If B belongs to $\mathcal{B}_{1/2}$, then, by the mentioned correspondence, B's leaf-language is contained in NP. Otherwise, B does not belong to $\mathcal{B}_{1/2}$. So B's minimal automaton contains the forbidden pattern [PW97]. This knowledge about B can be exploited to show that B's leaf-language is powerful enough to solve coUP [BKS98]. Therefore, between NP and coUP there are no unbalanced leaf-language definable classes. We call this a *gap theorem*.

Another gap theorem is known for P. Borchert [Bor95][1] showed that the following holds for any unbalanced leaf-language definable class $\mathcal{C}$: Either $\mathcal{C}$ is in P, or $\mathcal{C}$ contains at least one of the following classes: NP, coNP, MOD_pP for some prime p.

Balanced vs. Unbalanced. We are essentially interested in gap theorems similar to the ones showed by Borchert [Bor95] and Borchert, Kuske, and Stephan [BKS98]. However, this time we consider *balanced* leaf languages which show a new situation.

For the unbalanced case the following holds: For any regular B in dot-depth $1/2$, $\text{Leaf}_u^p(B)$ is in NP; for any regular B not in dot-depth $1/2$, $\text{Leaf}_u^p(B)$ is not in NP (unless $\text{coUP} \subseteq \text{NP}$). This does not hold anymore for the balanced case. It is possible to construct a starfree language C (Example 3.7) such that C is outside dot-depth $1/2$, but $\text{Leaf}_b^p(C) \subseteq \text{NP}$ (and therefore does not robustly contain[2] coUP). Even more, there is a regular D that is not starfree, but still $\text{Leaf}_b^p(D) \subseteq \text{NP}$ (e.g., $D = (AA)^*$ for any alphabet A). In this sense, the classes of the dot-depth hierarchy do not fit to balanced leaf languages. The reason for this becomes clear with help of a theorem discovered by Bovet, Crescenzi, and Silvestri [BCS92] and Vereshchagin [Ver93].

$$B \leq^{plt} C \;\Leftrightarrow\; \text{for all oracles } O,\ \text{Leaf}_b^p(B)^O \subseteq \text{Leaf}_b^p(C)^O$$

So $\text{Leaf}_b^p(B) \subseteq \text{NP}$ not only for all B in dot-depth $1/2$, but also for all B that are polylog-time reducible to a language in dot-depth $1/2$.

Our Contribution. We start the paper with a study of the power of polylog-time reductions restricted to regular languages. More precisely, we study two classes:

[1] In contrast to the chronological order, we first mentioned the result by Borchert, Kuske, and Stephan [BKS98] and then Borchert's result [Bor95]. The reason is that our paper first proves a result similar to the one by Borchert, Kuske, and Stephan, and then derives a result similar to Borchert.

[2] This means containment relative to all oracles.

1. $\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG}$, the class of regular languages that are polylog-time reducible to a language in dot-depth $1/2$ (Section 3), and

2. $\mathrm{PLT} \cap \mathrm{REG}$, the class of regular languages that are polylog-time decidable (Section 4).

For both classes we prove two characterizations:

- a forbidden-pattern characterization, and

- a characterization in terms of regular expressions.

This immediately implies decidability of the classes. Moreover, we show that both classes are strongly connected:

$$\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{co}\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG} = \mathrm{PLT} \cap \mathrm{REG}.$$

We derive three consequences from the characterizations above.

Consequence 1: Two gap theorems. We obtain gap theorems for *balanced* leaf-language definable classes $\mathcal{C}$ and $\mathcal{D}$:

1. Either $\mathcal{C}$ is contained in NP, or $\mathcal{C}$ contains coUP.

2. Either $\mathcal{D}$ is contained in P, or $\mathcal{D}$ contains UP or coUP.

We translate both theorems into gap theorems that do not involve promise classes.

1. Either $\mathcal{C}$ is contained in NP, or $\mathcal{C}$ contains at least one of the following classes: coNP, $\mathrm{co1NP}$, $\mathrm{MOD}_p\mathrm{P}$ for some prime p.

2. Either $\mathcal{D}$ is contained in P, or $\mathcal{D}$ contains at least one of the following classes: NP, coNP, $\mathrm{1NP}$, $\mathrm{co1NP}$, $\mathrm{MOD}_p\mathrm{P}$ for some prime p.

Formerly, such gap theorems were known only for the unbalanced case [Bor95, BKS98].

Consequence 2: Polylog-time reductions can decrease dot-depth complexity. We come back to counting with counterfree automata. Here we are interested in the question to what extend polylog-time reductions can use weak counting. We mentioned a starfree language outside dot-depth $1/2$ that is polylog-time reducible to a language in dot-depth $1/2$ (Example 3.7). Hence, although polylog-time reductions cannot count (e.g., the number of letters a in a word), they can decrease dot-depth complexity. We show that this decrease can be tremendous: For $n \geq 1$ there exist starfree languages L_n that are not in $\mathcal{B}_n$ but still in $\mathcal{R}^{plt}(\mathcal{B}_{1/2})$.

This is only possible, since counterfree automata can do weak counting. We use starfree languages with high dot-depth complexity. These languages have the property that words not in the language have a regular[3] pattern of letters b. We can locally test whether a given word has this regular pattern. If so, then by looking at the position of the last b we can gain information about the number of a's and b's in the word. This tells us immediately whether the word belongs to the language. Otherwise, if a word does not have the regular pattern, then it is in the language by definition. All these computations can be done by a polylog-time reduction function. So polylog-time reductions can drastically decrease the dot-depth complexity of these languages.

Consequence 3: Unbalanced starfree leaf-languages can be much stronger than balanced ones. Remember that $L_n \notin \mathcal{B}_n$, but still $L_n \in \mathcal{R}^{plt}(\mathcal{B}_{1/2})$. We exploit this to obtain conclusions for leaf-language definable complexity classes. We prove lower bounds for the complexity of $\mathrm{Leaf}_u^p(L_n)$:

- $\mathrm{Leaf}_b^p(L_n) \subseteq \mathrm{NP}$, but

[3] Here we aim at the natural meaning of the word 'regular'.

- $\mathrm{Leaf}_u^p(L_n)$ contains level n of the unambiguous alternation hierarchy.

It is expected that level n of the unambiguous alternation hierarchy is not contained in level $n-1$ of the polynomial-time hierarchy. If this is true, then for every $n \geq 1$, $\mathrm{Leaf}_b^p(L_n) \subseteq \mathrm{NP}$, yet $\mathrm{Leaf}_u^p(L_n) \not\subseteq \Sigma_{\mathrm{n-1}}^{\mathrm{P}}$. Therefore, our result gives evidence that for starfree languages, unbalanced leaf-languages are much stronger than balanced ones.

2 Preliminaries

$\mathbb{N}$ denotes the set of natural numbers. We fix a finite alphabet A such that $|A| \geq 2$. A^* denotes the set of words (including the empty word ε). Throughout the paper all languages and all classes of the dot-depth hierarchy are considered with respect to A. Polylog-time reductions are defined as follows.

Definition 2.1 *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 A is polylog-time reducible *to a language B, $A \leq^{plt} B$, if there exists a polylog-time computable f such that for all x, $x \in A \Leftrightarrow f(x) \in B$.*

PLT denotes the class of languages that have polylog-time computable characteristic functions. We summarize connections between dot-depth hierarchy and polynomial-time hierarchy.

Theorem 2.2 ([HLS$^+$93, BV98, BKS98]) *The following holds for $n \geq 1$ and relative to all oracles.*

1. $\mathrm{P} = \mathrm{Leaf}_b^p(\mathrm{PLT}) = \mathrm{Leaf}_b^p(\mathcal{B}_0) = \mathrm{Leaf}_u^p(\mathcal{B}_0)$
2. $\Sigma_{\mathrm{n}}^{\mathrm{P}} = \mathrm{Leaf}_b^p(\mathcal{B}_{n-1/2}) = \mathrm{Leaf}_u^p(\mathcal{B}_{n-1/2})$
3. $\Pi_{\mathrm{n}}^{\mathrm{P}} = \mathrm{Leaf}_b^p(\mathrm{co}\mathcal{B}_{n-1/2}) = \mathrm{Leaf}_u^p(\mathrm{co}\mathcal{B}_{n-1/2})$
4. $\mathrm{BC}(\Sigma_{\mathrm{n}}^{\mathrm{P}}) = \mathrm{Leaf}_b^p(\mathcal{B}_n) = \mathrm{Leaf}_u^p(\mathcal{B}_n)$
5. $\mathrm{NP}(n) = \mathrm{Leaf}_b^p(\mathcal{B}_{1/2}(n)) = \mathrm{Leaf}_u^p(\mathcal{B}_{1/2}(n))$

Bovet, Crescenzi, and Silvestri [BCS92] and Vereshchagin [Ver93] showed an important connection between polylog-time reducibility and *balanced* leaf-language definable classes.

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

$$B \leq^{plt} C \ \Leftrightarrow \ \text{for all oracles } O,\ \mathrm{Leaf}_b^p(B)^O \subseteq \mathrm{Leaf}_b^p(C)^O.$$

Definition 2.4 *Let $\mathcal{A}$ be a finite automaton with set of states S and extended transition function δ. For every word w, let δ^w be the function $S \to S$ such that $\delta^w(s) \stackrel{df}{=} \delta(s, w)$. A nonempty word u is called* idempotent *if $\delta^{uu} = \delta^u$.*

Note that all δ^w form the syntactic monoid of $\mathcal{A}$. The next lemma shows that for every fixed automaton, we can factorize any word into idempotents of constant length.

Lemma 2.5 ([GS00]) *For every finite automaton there exists a constant c such that every nonempty word w can be factorized as $w = v_0u_1v_1 \cdots u_mv_m$ where v_i and u_i are nonempty words of length $< c$ and all u_i are idempotent.*

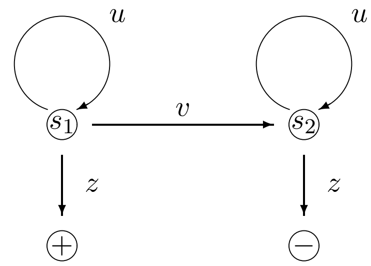

Figure 1: Forbidden pattern for $\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG}$ where $|v| = |u|$.

3 Regular Languages that are $\leq^{plt}$-reducible to $\mathcal{B}_{1/2}$

In this section, with Theorem 3.2, we prove two characterizations of regular languages that belong to the polylog-time closure of $\mathcal{B}_{1/2}$:

- a forbidden-pattern characterization, and
- a characterization in terms of regular expressions.

In Theorem 3.1 we separately formulate and prove the most difficult direction of these characterizations. As a consequence, we obtain a gap theorem for balanced leaf-language definable complexity classes $\mathcal{C}$: Either $\mathcal{C}$ is contained in NP, or $\mathcal{C}$ contains coUP. Additionally, we describe this gap so that no promise classes are involved: Either $\mathcal{C}$ is contained in NP, or $\mathcal{C}$ contains at least one of the following classes: coNP, co1NP, $\mathrm{MOD}_p\mathrm{P}$ for some prime p. Formerly, such gap theorems were known only for the unbalanced case [BKS98].

Moreover, in this section we see that the regular part of the polylog-time closure of $\mathcal{B}_{1/2}$ coincides with the regular part of $\mathcal{B}_{1/2}$'s closure under a weaker form of polylog-time reduction: Here the reduction function is only allowed to ask a constant number of queries.

Finally, the forbidden-pattern characterization implies decidability of $\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG}$.

Theorem 3.1 *Let L be a regular language such that the pattern in Figure 1 does not appear in the minimal automaton of L. There exists $d \geq 1$ such that L is a finite union of languages of the form $w_0(A^d)^* w_1 \cdots (A^d)^* w_n$ where $n \geq 0$ and $w_i \in A^*$.*

Proof Let $\mathcal{A}$ denote the minimal automaton of L with transition function δ and initial state s_0. Let c be the constant from Lemma 2.5.

$$d \stackrel{df}{=} c!$$

For every idempotent u define $B(u) \stackrel{df}{=} ((A^d)^* \cap A^* u) \cup \{\varepsilon\}$.

Claim: Let $E = w_0 u_0 B(u_0) \cdot w_1 u_1 B(u_1) \cdots w_{n-1} u_{n-1} B(u_{n-1}) \cdot w_n$ such that each u_i is idempotent of length $< c$ and $E \subseteq L$. There exists $E' = w'_0 u'_0 B(u'_0) \cdot w'_1 u'_1 B(u'_1) \cdots w'_{m-1} u'_{m-1} B(u'_{m-1}) \cdot w'_m$ such that each u'_i is idempotent of length $< c$,

1. $|w'_0 u'_0 \cdots w'_{m-1} u'_{m-1} w'_m| \leq 2c|A|^c d|\mathcal{A}|$, and
2. $E \subseteq E' \subseteq L$.

With this claim at hand we argue as follows. For any $w \in L$ let E'_w be the expression we obtain when we apply the claim to $E = w$. By statement 1 of the claim, the length of E' is bounded by a constant. Therefore,

$$L = \bigcup_{w \in L} E'_w,$$

where the union is finite. It remains to show that every E'_w is a finite union of languages of the form $w_0 {(A^d)}^* w_1 \cdots {(A^d)}^* w_n$. This is easy to observe, since for every u,

$$B(u) = \bigcup_{\substack{v \in A^*, \\ |v| = d - |u|}} {(A^d)}^* v u \cup \{\varepsilon\}.$$

This proves the theorem.

Proof of the Claim: Assume the claim does not hold. So there exists a counter example E such that there is no E' as stated in the claim. We choose E minimal in the sense that $|w_0 u_0 \cdots w_{n-1} u_{n-1} w_n|$ is minimal. Hence the claim holds for all shorter expressions. Note that

$$|w_0 u_0 \cdots w_{n-1} u_{n-1} w_n| > 2c|A|^c d|\mathcal{A}|. \tag{1}$$

Case 1: $n \geq |A|^c d|\mathcal{A}|$.

There are less than $|A|^c$ idempotents of length $< c$. Hence there exists an idempotent u such that $u = u_i$ for more than $d|\mathcal{A}|$ words u_i. We mark $d|\mathcal{A}| + 1$ such appearances of u. Say these are the factors u_{i_0}, u_{i_2}, and so on. Consider the word $w \stackrel{df}{=} w_0 u_0 \cdots w_{n-1} u_{n-1} w_n$ and denote the prefix that reaches up to the end of u_{i_j} by

$$y_j \stackrel{df}{=} w_0 u_0 \cdots w_{i_j} u_{i_j}.$$

We are now interested in two things:

- the length of y_j modulo d, and
- the state in $\mathcal{A}$ we reach after reading y_j.

Since we consider $d|\mathcal{A}|+1$ words y_j, there exist two such words y_j and y_k such that $|y_j| \equiv |y_k| \pmod{d}$ and $\delta(s_0, y_j) = \delta(s_0, y_k)$. This means that (i) the positions of u_{i_j} and u_{i_k} in w are equivalent modulo d, and (ii) when $\mathcal{A}$ reads w, then it reaches u_{i_j} with the same state it reaches u_{i_k}. $\tilde{E}$ is defined as the expression obtained from E when we replace the part between u_{i_j} and w_{i_k+1} by $B(u)$. So the expressions E and $\tilde{E}$ can be aligned as follows.

$$\begin{aligned} E &= w_0 u_0 B(u_0) \cdots w_{i_j} u_{i_j} B(u_{i_j}) \cdots w_{i_k} u_{i_k} B(u_{i_k}) w_{i_k+1} u_{i_k+1} \cdots w_{n-1} u_{n-1} B(u_{n-1}) \cdot w_n \\ \tilde{E} &= w_0 u_0 B(u_0) \cdots w_{i_j} u_{i_j} \qquad \cdot B(u) \cdot \qquad w_{i_k+1} u_{i_k+1} \cdots w_{n-1} u_{n-1} B(u_{n-1}) \cdot w_n \end{aligned}$$

We show

$$E \subseteq \tilde{E} \subseteq L. \tag{2}$$

Assume for the moment that (2) has been proved. Then $\tilde{E}$ satisfies the requirements for E in our claim. Moreover, $\tilde{E}$ is shorter than E, i.e.,

$$|w_0 u_0 \cdots w_{i_j} u_{i_j} \cdot w_{i_k+1} u_{i_k+1} w_{n-1} u_{n-1} w_n| < |w_0 u_0 \cdots w_{n-1} u_{n-1} w_n|.$$

Therefore, the claim holds for $\tilde{E}$. We obtain an E' that corresponds to $\tilde{E}$. Together with (2) it follows that $E \subseteq \tilde{E} \subseteq E' \subseteq L$. This contradicts the choice of E.

To finish Case 1 it remains to show (2). For $E \subseteq \tilde{E}$ it suffices to observe

$$B(u_{i_j})w_{i_j+1}u_{i_j+1}B(u_{i_j+1})\cdots w_{i_k}u_{i_k}B(u_{i_k}) \subseteq B(u).$$

Note that $u = u_{i_j} = u_{i_k}$. Therefore, it is enough to show

$$B(u_{i_j})w_{i_j+1}u_{i_j+1}B(u_{i_j+1})\cdots w_{i_k}u_{i_k}B(u_{i_k}) \subseteq (A^d)^*.$$

Words in any $B(\cdot)$ are of length $\equiv 0 \pmod d$. So it remains to show

$$|w_{i_j+1}u_{i_j+1}\cdots w_{i_k}u_{i_k}| \equiv 0 \pmod d.$$

This is easy to see, since $y_k = y_j \cdot w_{i_j+1}u_{i_j+1}\cdots w_{i_k}u_{i_k}$ and $|y_k| \equiv |y_j| \pmod d$. This shows $E \subseteq \tilde{E}$.

Finally we show $\tilde{E} \subseteq L$. The word $w_0u_0\cdots w_{n-1}u_{n-1}w_n$ belongs to E and therefore to L. From $\delta(s_0, y_j) = \delta(s_0, y_k)$ it follows that

$$w_0u_0\cdots w_{i_j}u_{i_j}\cdot w_{i_k+1}u_{i_k+1}\ldots w_{n-1}u_{n-1}w_n \in L.$$

We show that there does not exist $v \in B(u_{n-1})$ such that

$$w_0u_0\cdots w_{i_j}u_{i_j}\cdot w_{i_k+1}u_{i_k+1}\ldots w_{n-1}u_{n-1}\cdot v\cdot w_n \notin L.$$

If such v exists, then $|v| \equiv 0 \pmod d$ and u_{n-1} is suffix of v. Therefore, after reading

$$w_0u_0\cdots w_{i_j}u_{i_j}\cdot w_{i_k+1}u_{i_k+1}\ldots w_{n-1}u_{n-1}$$

the automaton reaches a state s_1 such that

- s_1 has a loop u_{n-1} (since u_{n-1} is idempotent),
- $s_2 \overset{df}{=} \delta(s_1, v)$ has a loop u_{n-1} (since u_{n-1} is suffix of v),
- $\delta(s_1, w_n)$ accepts, and $\delta(s_2, w_n)$ rejects.

Since $|v|$ is divisible by $d = c!$, it is in particular divisible by $|u_{n-1}|$. Let $k \overset{df}{=} |v|/|u_{n-1}|$. Both states s_1 and s_2 have a loop u_{n-1}^k where $|u_{n-1}^k| = |v|$. So the pattern in Figure 1 appears in $\mathcal{A}$; this contradicts our assumption. This shows that our choice of v is not possible. Therefore,

$$w_0u_0\cdots w_{i_j}u_{i_j}\cdot w_{i_k+1}u_{i_k+1}\ldots w_{n-1}u_{n-1}\cdot B(u_{n-1})\cdot w_n \subseteq L. \tag{3}$$

Analogously we obtain:

$$w_0u_0\cdots w_{i_j}u_{i_j}\cdot w_{i_k+1}u_{i_k+1}\cdots w_{n-2}u_{n-2}B(u_{n-2})w_{n-1}u_{n-1}B(u_{n-1})w_n \subseteq L$$

$$\vdots$$

$$w_0u_0B(u_0)\cdots w_{i_j}u_{i_j}B(u_{i_j})\cdot w_{i_k+1}u_{i_k+1}B(u_{i_k+1})\cdots w_{n-1}u_{n-1}B(u_{n-1})w_n \subseteq L$$

This shows $\tilde{E} \subseteq L$ and finishes Case 1.

Case 2: $n < |A|^c d|\mathcal{A}|$.

We reduce this case to Case 1 where we already obtained a contradiction. The sum of lengths of all u_i is $\leq cn$. From equation (1) it follows that the sum of lengths of all w_i is $> c|A|^c d|\mathcal{A}|$. So there exists j such that $|w_j| > c$. By Lemma 2.5, there exist nonempty words v_0, v_1, and u such that $w_j = v_0uv_1$

where u is idempotent of length $< c$. We define $\tilde{E}$ to be the expression obtained from E when we insert $B(u)$ between u and v_1. The expressions E and $\tilde{E}$ can be aligned as follows.

$$\begin{array}{rcl} E & = & w_0u_0B(u_0)\cdots w_{j-1}u_{j-1}B(u_{j-1})\cdot v_0u \quad\quad \cdot \quad\quad v_1\cdot u_jB(u_j)\cdots w_{n-1}u_{n-1}B(u_{n-1})\cdot w_n \\ \tilde{E} & = & w_0u_0B(u_0)\cdots w_{j-1}u_{j-1}B(u_{j-1})\cdot v_0u\cdot B(u)\cdot v_1\cdot u_jB(u_j)\cdots w_{n-1}u_{n-1}B(u_{n-1})\cdot w_n \end{array}$$

Assume there exists $w \in \tilde{E} - L$. Let $w = x \cdot v_0uvv_1 \cdot z$ be the factorization according to expression $\tilde{E}$ where $v \in B(u)$. Analogous to the argumentation for equation (3) we find the pattern from Figure 1 between $s_1 \stackrel{df}{=} \delta(s_0, xv_0u)$ and $s_2 \stackrel{df}{=} \delta(s_0, xv_0uv)$. (Here v_1z leads from s_1 to an accepting state and from s_2 to a rejecting state.) This is a contradiction and it follows that $E \subseteq \tilde{E} \subseteq L$. Therefore, by our assumption, also for $\tilde{E}$ there does not exist an expression E' as stated in the claim. Note that E and $\tilde{E}$ have the same size, i.e., when dropping the expressions $B(\cdot)$ in E and $\tilde{E}$, then the resulting words have same size. So $\tilde{E}$ is a minimal counter example for the claim, and compared to E it contains one more expression $B(\cdot)$. By iterating this procedure we obtain a minimal counter example that contains $|A|^c d|\mathcal{A}|$ expressions $B(\cdot)$. For this counter example we obtain a contradiction by Case 1. This finishes the proof the claim. ☐

Theorem 3.2 *For every regular L the following are equivalent.*

1. *$L \in \mathcal{R}^{plt}(\mathcal{B}_{1/2})$.*
2. *The pattern in Figure 1 does not appear in the minimal automaton of L.*
3. *There exists $d \geq 1$ such that L is a finite union of languages of the form $w_0(A^d)^*w_1\cdots(A^d)^*w_n$ where $n \geq 0$ and $w_i \in A^*$.*

Proof 2 $\Rightarrow$ 3: Follows from Theorem 3.1.

3 $\Rightarrow$ 1: Let a, b be different letters from A. It suffices to show $w_0(A^d)^*w_1\cdots(A^d)^*w_n \leq^{plt} A^*bA^*$. On input w the reduction produces a word w' such that

$$|w'| = 2^{\lceil \log |w| \rceil (n+1)}.$$

The k-th letter of w' is computed as follows: We interpret k as $(n+1)$-tuple $(p_0, \ldots p_n)$ of positions in w. First we make sure that $p_0 = 0$, $p_n = |w| - |w_n|$, and $p_0 \leq p_1 \leq \cdots \leq p_n$. If either of these conditions does not hold, then output letter a. Otherwise, for every i we verify that factor w_i appears at position p_i in w. Again, output a if either of these does not hold. Finally, for $i \geq 1$, verify that $p_i - p_{i-1} - w_{i-1} \equiv 0 \pmod{d}$. If this does not hold, then output a. Otherwise output b. Observe that this polylog-time function reduces $w_0(A^d)^*w_1\cdots(A^d)^*w_n$ to A^*bA^*.

1 $\Rightarrow$ 2: There exists $L' \in \mathcal{B}_{1/2}$ such that $L \leq^{plt} L'$. By Theorem 2.3, relative to all oracles, $\mathrm{Leaf}^p_b(L) \subseteq \mathrm{Leaf}^p_b(L')$. By Theorem 2.2, relative to all oracles, $\mathrm{Leaf}^p_b(L') \subseteq \mathrm{NP}$. Hence, relative to all oracles,

$$\mathrm{Leaf}^p_b(L) \subseteq \mathrm{NP}. \tag{4}$$

Assume that the pattern in Figure 1 appears in L's minimal automaton $\mathcal{A}$ (with initial state s_0 and transition function δ). Choose x such that $\delta(s_0, x) = s_1$. We exploit the pattern and show that $\mathrm{Leaf}^p_b(L)$ contains coUP. For this, consider any UP machine M. We modify M by adding an artificial first path which outputs x and an artificial last path which outputs z. Moreover, any rejecting path outputs u and any accepting path outputs v. If $w \in U$, then the leaf word of $M(w)$ belongs to $xu^*vu^*z \subseteq \overline{L}$. If $w \notin U$, then the leaf word of $M(w)$ belongs to $xu^*z \subseteq L$. Therefore, $\overline{L(M)} \in \mathrm{Leaf}^p_b(L)$ and hence, $\mathrm{coUP} \subseteq \mathrm{Leaf}^p_b(L)$. Since our construction relativizes, $\mathrm{coUP} \subseteq \mathrm{Leaf}^p_b(L)$ relative to all oracles. From equation (4) we obtain that relative to all oracles,

$$\mathrm{coUP} \subseteq \mathrm{NP}. \tag{5}$$

This is a contradiction, since an oracle relative to which $\mathrm{coUP} \not\subseteq \mathrm{NP}$ is known [GW03]. □

Let $\leq_{const}^{plt}$ denote the restricted type of polylog-time reductions where the reduction is only allowed to ask a constant number of queries (i.e., can access only constantly many letters of the input word). Clearly, we can assume that these are nonadaptive queries.

Corollary 3.3 $\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG} = \mathcal{R}_{const}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG}.$

Proof The inclusion $\supseteq$ holds trivially. If $L \in \mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG}$, then by Theorem 3.2, L is a finite union of languages of the form $w_0(A^d)^* w_1 \cdots (A^d)^* w_n$. In the proof of Theorem 3.2 we look at the implication 3 $\Rightarrow$ 1. There we actually show $w_0(A^d)^* w_1 \cdots (A^d)^* w_n \leq_{const}^{plt} A^*bA^*$. □

Corollary 3.4 *Let* $\mathcal{C} = \mathrm{Leaf}_b^p(L)$ *for some regular* L.

1. *If* $L \in \mathcal{R}^{plt}(\mathcal{B}_{1/2})$, *then* $\mathcal{C} \subseteq \mathrm{NP}$.
2. *If* $L \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$, *then* $\mathrm{coUP} \subseteq \mathcal{C}$.

Proof If $L \in \mathcal{R}^{plt}(\mathcal{B}_{1/2})$, then from the Theorems 2.2 and 2.3 it follows that $\mathcal{C} \subseteq \mathrm{Leaf}_b^p(\mathcal{B}_{1/2}) = \mathrm{NP}$. Otherwise, $L \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$. Let $\mathcal{A}$ be the minimal automaton of L. By Theorem 3.2, $\mathcal{A}$ contains the pattern in Figure 1. We can exploit this pattern to show $\mathrm{coUP} \subseteq \mathcal{C}$ (see direction 1 $\Rightarrow$ 2 in the proof of Theorem 3.2). □

Corollary 3.4 shows a gap for balanced leaf-language definable classes above NP: Any such class higher than NP contains coUP. Since coUP is a promise class, it would be most welcome to show a similar gap that does not involve any promise class. Borchert, Kuske, and Stephan [BKS98] show how to do this. By iterating the coUP pattern they obtain a list of non-promise complexity classes such that every *unbalanced* leaf-language definable class higher than NP contains at least one class from the list. The same idea works here for the balanced setting.

Corollary 3.5 *Let* $\mathcal{C} = \mathrm{Leaf}_b^p(L)$ *for some regular* L.

1. *If* $L \in \mathcal{R}^{plt}(\mathcal{B}_{1/2})$, *then* $\mathcal{C} \subseteq \mathrm{NP}$.
2. *If* $L \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$, *then* $\mathrm{coNP} \subseteq \mathcal{C}$, *or* $\mathrm{co1NP} \subseteq \mathcal{C}$, *or for some prime* p, $\mathrm{MOD}_p\mathrm{P} \subseteq \mathcal{C}$.

Proof The proof is based on an idea of Borchert, Kuske, and Stephan [BKS98]. If $L \in \mathcal{R}^{plt}(\mathcal{B}_{1/2})$, then by Theorems 2.2 and 2.3, $\mathcal{C} \subseteq \mathrm{Leaf}_b^p(\mathcal{B}_{1/2}) = \mathrm{NP}$. Now assume $L \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$. Let $\mathcal{A}$ be the minimal automaton of L with transition function δ. By Theorem 3.2, $\mathcal{A}$ contains the pattern in Figure 1. Without loss of generality, we may assume that if we start in any state in $\mathcal{A}$ and read v, then we end in a state that has a loop u. When we are in s_1 and read v, then we reach s_2. In s_2 we read v once again and reach some state s_3. We continue this repeated reading of v's until one of the following cases holds.

Case 1: We reach a state s_m such that $\delta(s_m, v) = s_m$.

We are interested in the acceptance behavior of $\delta(s_i, z)$ for $0 \leq i \leq m$. Choose the largest j such that $\delta(s_j, z)$ accepts and $\delta(s_{j+1}, z)$ rejects.

Case 1a: For all $i > j$, $\delta(s_i, z)$ rejects.

If we use u^m and v^m instead of u and v, then we see that $\mathcal{A}$ has the balanced coNP pattern (Figure 2). It follows that $\mathrm{coNP} \subseteq \mathrm{Leaf}_b^p(L) = \mathcal{C}$.

Case 1b: There exists $k > j$ such that $\delta(s_k, z)$ rejects and $\delta(s_{k+1}, z)$ accepts.

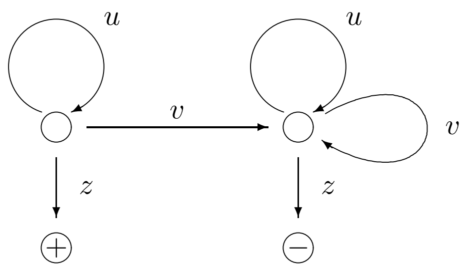

Figure 2: Balanced coNP pattern where $|v| = |u|$.

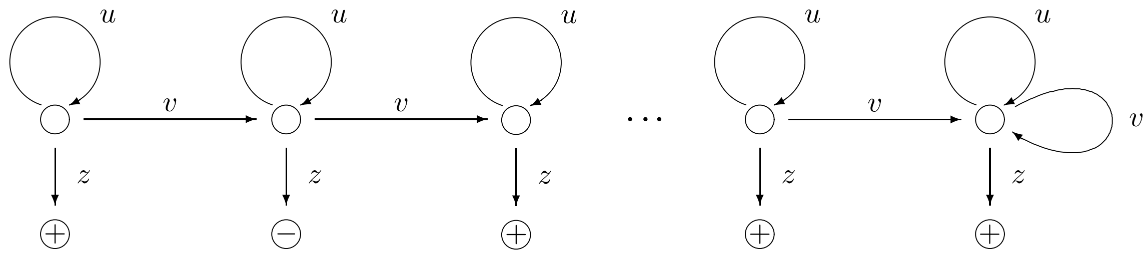

Figure 3: Balanced $\mathrm{co1NP}$ pattern where $|v| = |u|$.

By the choice of j, for all $i \geq k$, $\delta(s_i, z)$ accepts. Therefore, if we use u^k and v^k instead of u and v, then we see that $\mathcal{A}$ has the balanced $\mathrm{co1NP}$ pattern (Figure 3). It follows that $\mathrm{co1NP} \subseteq \mathrm{Leaf}_b^p(L) = \mathcal{C}$.

Case 2: We reach a state s_m such that $\delta(s_m, v) = s_i$ for some $i < m$.

Therefore, for $n = m - i + 1$, $\mathcal{A}$ has the balanced n-counting pattern (Figure 4). We may assume that n is a prime (otherwise $\mathcal{A}$ has the balanced p-counting pattern for every prime factor p of n). Since n is prime, any $\mathrm{MOD}_n\mathrm{P}$ computation can be modified such that the number of accepting paths either is $\equiv 0 \pmod{n}$ (acceptance) or is $\equiv 1 \pmod{n}$ (rejection). This shows $\mathrm{MOD}_n\mathrm{P} \subseteq \mathrm{Leaf}_b^p(L) = \mathcal{C}$. □

Corollary 3.6 *It is decidable whether a given regular language is $\leq^{plt}$ reducible to a language in $\mathcal{B}_{1/2}$.*

Proof This follows from Theorem 3.2, since it is decidable (in nondeterministic logarithmic space) whether a given automaton contains the pattern in Figure 1. □

Example 3.7 *A starfree language outside $\mathcal{B}_{1/2}$ that is $\leq^{plt}$ reducible to a language in $\mathcal{B}_{1/2}$.* [4]

We consider automaton $\mathcal{E}$ (Figure 5). $\mathcal{E}$ is minimal and permutationfree. So $L(\mathcal{E})$ is starfree. The automaton contains the forbidden pattern for $\mathcal{B}_{1/2}$ [PW97]. Therefore, $L(\mathcal{E}) \notin \mathcal{B}_{1/2}$. Moreover, $\mathcal{E}$

[4] Some of the following properties of this example were discovered during a discussion with Bernhard Schwarz, Victor Selivanov, and Klaus W. Wagner.

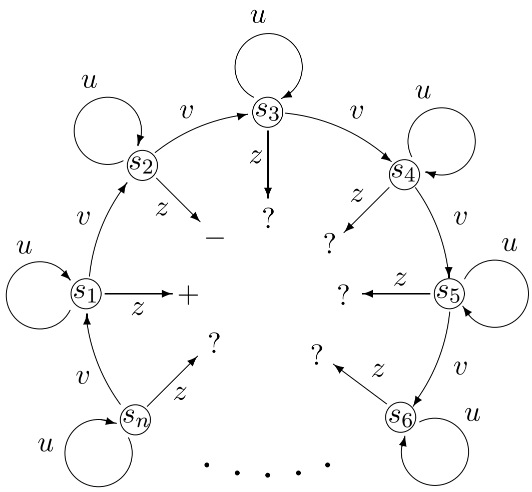

Figure 4: Balanced n-counting pattern where $|v| = |u|$.

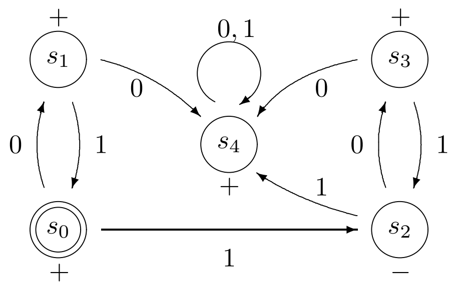

Figure 5: Automaton $\mathcal{E}$ with initial state s_0.

does not contain the pattern in Figure 1. Therefore, by Theorem 3.2, $L(\mathcal{E}) \in \mathcal{R}^{plt}(\mathcal{B}_{1/2})$ (e.g., $L(\mathcal{E})$ polylog-time reduces to A^*1A^*). $L(\mathcal{E})$ can be characterized in different ways:

$$\begin{aligned} L &= (AA)^* \cup L_0 \\ &= (01)^* \cup L_0 \\ &= \overline{(01)^*1(01)^*} \end{aligned}$$

where

$$L_0 \stackrel{df}{=} A^*00A^* \cup A^*111A^* \cup A^*11A^*11A^* \cup A^*0 \cup 11A^* \cup 1A^*11A^*.$$

It follows that $L(\mathcal{E}) \in \mathcal{B}_{1/2} \vee \mathrm{co}\mathcal{B}_{1/2}$ which is the complement of the second level of the Boolean hierarchy over $\mathcal{B}_{1/2}$. In particular, $L(\mathcal{E}) \in \mathcal{B}_1$. Moreover,

$$\begin{aligned} \mathrm{Leaf}^p_b(L) &= \mathrm{NP} \qquad \text{and} \\ \mathrm{Leaf}^p_u(L) &= \mathrm{co1NP}. \end{aligned}$$

4 Regular Languages that are Polylog-Time Decidable

This section is similar to Section 3. Here we consider $\mathrm{PLT} \cap \mathrm{REG}$ instead of $\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG}$. First, with Theorem 3.2, we provide a characterization of $\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{co}\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG}$ which immediately implies

$$\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{co}\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG} = \mathrm{PLT} \cap \mathrm{REG}.$$

This strong connection between $\mathcal{R}^{plt}(\mathcal{B}_{1/2})$ and PLT allows a translation of results about $\mathcal{R}^{plt}(\mathcal{B}_{1/2})$ (Section 3) to results about PLT. Beside the equation above we obtain two characterizations of regular languages that belong to PLT:

- a forbidden-pattern characterization, and
- a characterization in terms of regular expressions.

While the first characterization is new, the latter one is already known [Wag01]. As a consequence of the forbidden-pattern characterization, we obtain a gap theorem for balanced leaf-language definable complexity classes $\mathcal{C}$: Either $\mathcal{C}$ is contained in P, or $\mathcal{C}$ contains UP or coUP. Additionally, we describe this gap so that no promise classes are involved: Either $\mathcal{C}$ is contained in P, or $\mathcal{C}$ contains at least one of the following classes: NP, coNP, 1NP, co1NP, $\mathrm{MOD}_p\mathrm{P}$ for some prime p. Formerly, such gap theorems were known only for the unbalanced case [Bor95].

Finally, the forbidden-pattern characterization implies decidability of the class $\mathrm{PLT} \cap \mathrm{REG}$.

Theorem 4.1 *If $L \in \mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{co}\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{REG}$, then there exists $d \geq 1$ such that L is a finite union of singletons $\{u\}$ and languages $v(A^d)^* w$ where $u, v, w \in A^*$.*

Proof Choose L according to the theorem. By Theorem 3.2, there exists $d \geq 1$ such that L is a finite union of languages $v_0(A^d)^* v_1 \cdots (A^d)^* v_m$. Call these languages the *terms of* L. Let L' be the complement of L. So there exists $e \geq 1$ such that L' is a finite union of languages $w_0(A^e)^* w_1 \cdots (A^e)^* w_n$. Call these languages the *terms of* L'.

Claim 4.2 *We may assume that $d = e$.*

Proof In the terms of L and L' we replace $(A^d)^*$ and $(A^e)^*$ according to the following equations.

$$(A^d)^* = \bigcup_{0 \leq i < e} \bigcup_{w \in A^{d \cdot i}} (A^{d \cdot e})^* w \tag{6}$$

$$(A^e)^* = \bigcup_{0 \leq i < d} \bigcup_{w \in A^{e \cdot i}} (A^{d \cdot e})^* w \tag{7}$$

This implies the claim, since the unions in (6) and (7) are finite. □

Let $T = v_0(A^d)^* v_1 \cdots (A^d)^* v_m$ be a term of L, and let $T' = w_0(A^d)^* w_1 \cdots (A^d)^* w_n$ be a term of L'. We say that T and T' are *compatible* if all of the following holds:

1. $m > 0$ and $n > 0$
2. v_0 is prefix of w_0 or w_0 is prefix of v_0
3. v_m is suffix of w_n or w_n is suffix of v_m
4. $|v_0 v_1 \cdots v_m| \equiv |w_0 w_1 \cdots w_n| \pmod{d}$

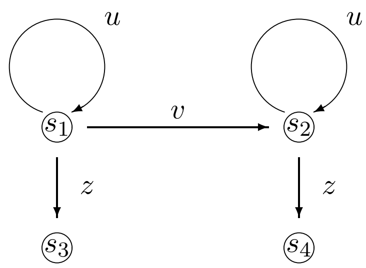

Figure 6: Forbidden pattern for PLT $\cap$ REG where $|v| = |u|$ and s_3 accepts $\Leftrightarrow$ s_4 rejects.

Claim 4.3 *If T is a term of L and T' is a term of L', then T and T' are not compatible.*

Proof Let $T = v_0(A^d)^* v_1 \cdots (A^d)^* v_m$ be a term of L, and let $T' = w_0(A^d)^* w_1 \cdots (A^d)^* w_n$ be a term of L' such that T and T' are compatible. Hence $m > 0$ and $n > 0$. Moreover, v_0 is prefix of w_0 or w_0 is prefix of v_0. So there exists a word v such that v_0 and w_0 are prefixes of v and $|v| \equiv |v_0| \pmod{d}$. Similarly, there exists w such that v_m and w_n are suffixes of w and $|w| \equiv |w_n| \pmod{d}$. Let u be any word that has a length $\equiv -|v_0 v_1 \cdots v_{m-1}| \pmod{d}$.

$$z \stackrel{df}{=} v\ v_1 v_2 \cdots v_{m-1}\ u\ w_0 w_1 \cdots w_{n-1}\ w$$

Observe that $|uw_0 w_1 \cdots w_{n-1} w| \equiv |uw_0 w_1 \cdots w_n| \equiv |uv_0 v_1 \cdots v_m| \equiv |v_m| \pmod{d}$. Therefore, $uw_0 w_1 \cdots w_{n-1} w \in (A^d)^* v_m$. Together with $v \in v_0(A^d)^*$ this shows $z \in T$. Similarly, observe $|vv_1 v_2 \cdots v_{m-1} uw_0| \equiv |v_0 v_1 \cdots v_{m-1} uw_0| \equiv |w_0| \pmod{d}$. Hence $vv_1 v_2 \cdots v_{m-1} uw_0 \in w_0(A^d)^*$. Together with $w \in (A^d)^* w_n$ this shows $z \in T'$. Hence $T \cap T' \neq \emptyset$ and therefore $L \cap L' \neq \emptyset$. $\square$

Claim 4.4 *Let $T = v_0(A^d)^* v_1 \cdots (A^d)^* v_m$ be a term of L such that $m > 0$. Define $\tilde{T} \stackrel{df}{=} v_0(A^d)^* A^r v_m$ where $r \stackrel{df}{=} |v_1 v_2 \cdots v_{m-1}|$. Then $T \subseteq \tilde{T}$ and $\tilde{T} \cap L'$ is finite.*

Proof Clearly $T \subseteq \tilde{T}$. Assume $\tilde{T} \cap L'$ is infinite. There must exist $T' = w_0(A^d)^* w_1 \cdots (A^d)^* w_n$, a term of L', such that $\tilde{T} \cap T'$ is infinite. Hence $n > 0$. Words in $\tilde{T}$ have lengths $\equiv |v_0 v_1 \ldots v_m| \pmod{d}$. Words in T' have lengths $\equiv |w_0 w_1 \ldots w_n| \pmod{d}$. Therefore,

$$|v_0 v_1 \ldots v_m| \equiv |w_0 w_1 \ldots w_n| \pmod{d}.$$

By Claim 4.3, T and T' are not compatible. So at least one of the following is false:

1. v_0 is prefix of w_0 or w_0 is prefix of v_0
2. v_m is suffix of w_n or w_n is suffix of v_m

It follows that $\tilde{T} \cap T' = \emptyset$. This contradicts our assumption. $\square$

Consider the terms of L and replace all $T = v_0(A^d)^* v_1 \cdots (A^d)^* v_m$ where $m > 0$ by $\tilde{T}$. Let $\tilde{L}$ denote the language defined in this way. $\tilde{L}$ is a finite union of singletons $\{u\}$ and languages $v(A^d)^* w$ where $u, v, w \in A^*$. By Claim 4.4, $L \subseteq \tilde{L}$ and $\tilde{L} \cap L'$ is finite. So L is a finite modification of $\tilde{L}$. Therefore, L is a finite union of singletons $\{u\}$ and languages $v(A^d)^* w$ as well. $\square$

Now we are going to show an analog of Theorem 3.2. This time we provide characterizations of PLT. We want to point out that the equivalence of statements 1 and 4 in Corollary 4.5 has been shown by Wagner [Wag01].

Corollary 4.5 *For every regular L the following are equivalent.*

1. $L \in \mathrm{PLT}$.
2. $L \in \mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{co}\mathcal{R}^{plt}(\mathcal{B}_{1/2})$
3. *The pattern in Figure 6 does not appear in the minimal automaton of L.*
4. *There exists $d \geq 1$ such that L is a finite union of singletons $\{u\}$ and languages $v(A^d)^*w$ where $u, v, w \in A^*$.*

Proof Let L' be the complement of L. $\mathcal{A}$ (resp., $\mathcal{A}'$) denotes the minimal automaton of L (resp., L'). Note that $\mathcal{A}'$ is obtained from $\mathcal{A}$ just by inverting the acceptance behavior.

1 $\Rightarrow$ 3: Assume $L \in \mathrm{PLT}$ and $\mathcal{A}$ contains the pattern from Figure 6. So $L' \in \mathrm{PLT}$. If s_3 accepts and s_4 rejects, then $\mathcal{A}$ has the pattern from Figure 1. Otherwise, $\mathcal{A}'$ has the pattern from Figure 1. By Theorem 3.2, $L \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$ or $L' \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$. So $L \notin \mathrm{PLT}$ or $L' \notin \mathrm{PLT}$. This is a contradiction.

3 $\Rightarrow$ 2: Assume the pattern in Figure 6 does not appear in $\mathcal{A}$. Therefore, neither $\mathcal{A}$ nor $\mathcal{A}'$ contain the pattern in Figure 1. By Theorem 3.2, L and L' belong to $\mathcal{R}^{plt}(\mathcal{B}_{1/2})$.

2 $\Rightarrow$ 4: By Theorem 4.1.

4 $\Rightarrow$ 1: Trivial. □

Corollary 4.6 *Let $\mathcal{C} = \mathrm{Leaf}_b^p(L)$ for some regular L.*

1. *If $L \in \mathrm{PLT}$, then $\mathcal{C} \subseteq \mathrm{P}$.*
2. *If $L \notin \mathrm{PLT}$, then $\mathrm{UP} \subseteq \mathcal{C}$ or $\mathrm{coUP} \subseteq \mathcal{C}$.*

Proof If $L \in \mathrm{PLT}$, then by Theorem 2.2, $\mathcal{C} \subseteq \mathrm{P}$. Otherwise, $L \notin \mathrm{PLT}$. Let L' be the complement of L. By Corollary 4.5, $L \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$ or $L' \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$. By Corollary 3.4, $\mathrm{coUP} \subseteq \mathrm{Leaf}_b^p(L)$ or $\mathrm{coUP} \subseteq \mathrm{Leaf}_b^p(L') = \mathrm{coLeaf}_b^p(L)$. □

So we found a gap for balanced leaf-language definable classes above P: Any such class higher than P contains UP or coUP. Similar to the gap above NP (Corollary 4.6) we obtain a gap that does not involve any promise class.

Corollary 4.7 *Let $\mathcal{C} = \mathrm{Leaf}_b^p(L)$ for some regular L.*

1. *If $L \in \mathrm{PLT}$, then $\mathcal{C} \subseteq \mathrm{P}$.*
2. *If $L \notin \mathrm{PLT}$, then at least one of the following classes is contained in $\mathcal{C}$: NP, coNP, $\mathrm{1NP}$, $\mathrm{co1NP}$, $\mathrm{MOD}_p\mathrm{P}$ for some prime p.*

Proof By Corollary 4.6, Item 1 holds. Let $L \notin \mathrm{PLT}$ and let L' be the complement of L. By Corollary 4.5, $L \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$ or $L' \notin \mathcal{R}^{plt}(\mathcal{B}_{1/2})$. By Corollary 3.5, at least one of the following holds:

1. $\mathrm{coNP} \subseteq \mathrm{Leaf}_b^p(L)$, or $\mathrm{co1NP} \subseteq \mathrm{Leaf}_b^p(L)$, or for some prime p, $\mathrm{MOD}_p\mathrm{P} \subseteq \mathrm{Leaf}_b^p(L)$
2. $\mathrm{coNP} \subseteq \mathrm{Leaf}_b^p(L') = \mathrm{coLeaf}_b^p(L)$, or $\mathrm{co1NP} \subseteq \mathrm{Leaf}_b^p(L') = \mathrm{coLeaf}_b^p(L)$, or for some prime p, $\mathrm{MOD}_p\mathrm{P} \subseteq \mathrm{Leaf}_b^p(L') = \mathrm{coLeaf}_b^p(L)$

This proves the corollary, since for every prime p, $\mathrm{MOD}_p\mathrm{P}$ is closed under complement. □

Corollary 4.8 *It is decidable whether a given regular language belongs to* PLT.

Proof This follows from Corollary 4.5, since it is decidable (in nondeterministic logarithmic space) whether a given automaton contains the pattern in Figure 6. □

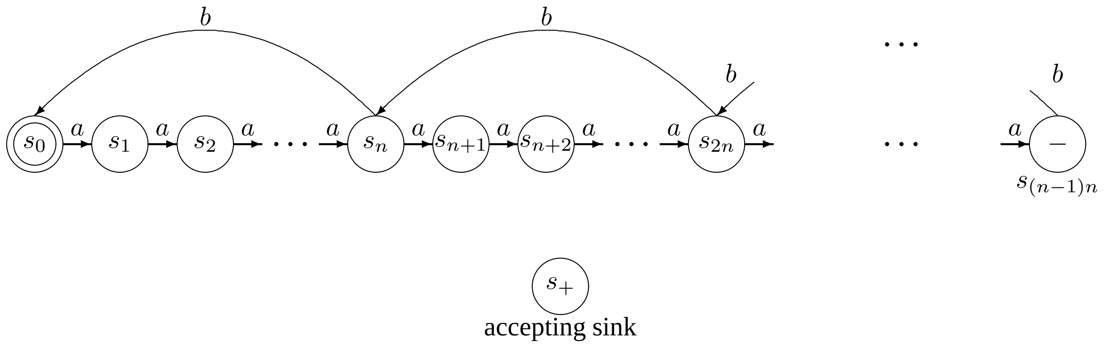

Figure 7: Automaton $\mathcal{A}_p$ where $p \geq 3$, $n = p - 1$, s_0 is initial state, $s_{(n-1)n}$ is the only rejecting state, and all undefined transitions lead to the accepting sink s_+. All $\mathcal{A}_p$ are minimal. For any prime $p \geq 3$, $L(\mathcal{A}_p) \in \mathrm{SF} \cap \mathcal{R}^{plt}(\mathcal{B}_{1/2})$ but $L(\mathcal{A}_p) \notin \mathcal{B}_{p-3}$.

5 Balanced versus Unbalanced Computations

In Example 3.7 we have seen that there exist starfree languages L that are not in $\mathcal{B}_{1/2}$, but $\leq^{plt}$ reducible to languages in $\mathcal{B}_{1/2}$. For such L, $\mathrm{Leaf}_b^p(L) \subseteq \mathrm{NP}$. These observations raise two questions:

1. Does $\mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{SF}$ fall into some level of the dot-depth hierarchy?
2. Can we characterize the complexity of $\mathrm{Leaf}_u^p(L)$ for $L \in \mathcal{R}^{plt}(\mathcal{B}_{1/2}) \cap \mathrm{SF}$?

In this section we give a 'no' answer to the first question. For $n \geq 1$ there exist starfree languages L_n that are not in $\mathcal{B}_n$ but still in $\mathcal{R}^{plt}(\mathcal{B}_{1/2})$. Regarding the second question, we prove lower bounds for the complexity of $\mathrm{Leaf}_u^p(L_n)$. More precisely,

- $\mathrm{Leaf}_b^p(L_n) \subseteq \mathrm{NP}$, but
- $\mathrm{Leaf}_u^p(L_n)$ contains level n of the unambiguous alternation hierarchy.

It is expected that level n of the unambiguous alternation hierarchy is not contained in level $n-1$ of the polynomial-time hierarchy. If this is true, then for every $n \geq 1$, $\mathrm{Leaf}_b^p(L_n) \subseteq \mathrm{NP}$, yet $\mathrm{Leaf}_u^p(L_n) \not\subseteq \Sigma_{n-1}^{\mathrm{P}}$. Therefore, our result gives evidence that for starfree languages unbalanced leaf-languages are much stronger than balanced ones.

Before we present the formal proof, we want to give some intuition why it is possible to construct languages of arbitrary dot-depth that are still polylog-time reducible to languages in $\mathcal{B}_{1/2}$. Choose any prime $p \geq 3$. We argue that the language L accepted by automaton $\mathcal{A}_p$ (defined in Figure 7) is not in dot-depth $p-3$ but is polylog-time reducible to A^*aA^*.

Why does L not belong to dot-depth $p-3$? Thomas [Tho84] constructed a family of languages that separate dot-depth classes. From this family we use a language L' that is not in dot-depth $p-3$. It is easy to see that L is the image of L' under the morphism that maps $a \mapsto a^{p-1}$ and $b \mapsto b$. Since dot-depth levels are closed under taking inverse morphisms, we obtain that L is not in dot-depth $p-3$.

Why is L polylog-time reducible to A^*aA^*? Let $n \stackrel{df}{=} p-1$. In $\mathcal{A}_p$, the number of a's between s_{in} and $s_{(i+1)n}$ is $\equiv -1 \pmod{p}$. All loops in $\mathcal{A}_p$ that do not go through s_+ are of length $\equiv 0 \pmod{p}$. Therefore, whenever we reach s_{in}, then the number of letters that has been read so far is $\equiv -i \pmod{p}$. Call a word *well-formed* if it does not lead from s_0 to s_+.

In every well-formed word, after $(n-1)n$ consecutive a's there must follow a letter b. $\quad (*)$

Let w be well-formed. Consider any b in w. This b must be read in some state s_{in} where $i \geq 1$. It follows that the number of letters left of this b is $\equiv -i \pmod p$. This shows:

$$\text{If } w \text{ is well-formed and } w = w_1 b w_2, \text{ then } w_1 \text{ leads from } s_0 \text{ to } s_{in} \text{ where } i \stackrel{df}{=} (-|w_1| \bmod p). \quad (**)$$

Hence in a well-formed word, the position (modulo p) of some letter b tells us the state in which this letter is read. This shows that we can locally test whether a word is well-formed: Just guess all neighboring b's, make sure that their distance is small $(*)$, determine the states in which these b's must be read $(**)$, and test whether these states fit to the factor between the b's. This local test shows that the set of words that are not well-formed is polylog-time reducible to A^*aA^*. It remains to argue that the set of words that are in L and that are not well-formed is polylog-time reducible to A^*aA^*. This is easy, since by $(**)$, the position of the last letter b tells us the state in which this b is read. So we just have to verify that the remaining part of the word (which is short) does not lead to $s_{(n-1)n}$.

Theorem 5.1 *For any prime $p \geq 3$, $L(\mathcal{A}_p) \in \mathrm{SF} \cap \mathcal{R}^{plt}(\mathcal{B}_{1/2})$ but $L(\mathcal{A}_p) \notin \mathcal{B}_{p-3}$.*

Proof The theorem is proved by the following series of claims.

Claim 5.2 *For any $p \geq 3$, $\mathcal{A}_p$ is minimal.*

Proof Otherwise $\mathcal{A}_p$ must have two different but equivalent states s and s'. None of them can be s_+. So $s = s_i$ and $s' = s_j$. Let $n = p-1$ and let $\delta_\mathcal{A}$ be the transition function of $\mathcal{A}_p$. Note that $\delta_\mathcal{A}(s, a^{(n-1)n-i})$ is the rejecting state $s_{(n-1)n}$. Therefore, $\delta_\mathcal{A}(s', a^{(n-1)n-i})$ must be rejecting as well, and therefore, must be equal to $s_{(n-1)n}$. It follows that $s' = s_i = s$ which is a contradiction. □

Claim 5.3 *For any $p \geq 3$, $L(\mathcal{A}_p)$ is starfree.*

Proof Let $n = p - 1$ and let $\delta_\mathcal{A}$ be the transition function of $\mathcal{A}_p$. If $L(\mathcal{A}_p)$ is not starfree, then there exists a state s, a word w, and $k \geq 2$ such that for all m, $\delta_\mathcal{A}(s, w^{km}) = s$ but $\delta_\mathcal{A}(s, w) \neq s$. Note that $s \neq s_+$. So $s = s_i$ for some i. Since $\delta_\mathcal{A}(s, w^{km}) = s$, in w^{km} the number of a's is equal to n times the number of b's. The same holds for w. Since $\delta_\mathcal{A}(s, w) \neq s_+$, this implies $\delta_\mathcal{A}(s, w) = s$. □

Claim 5.4 *For any $p \geq 3$, $L(\mathcal{A}_p) \in \mathcal{R}^{plt}(\mathcal{B}_{1/2})$.*

Proof Otherwise, by Theorem 3.2, $\mathcal{A}_p$ contains the pattern from Figure 1. To avoid confusion, we rename the states s_1 and s_2 in Figure 1; now they are called s and s'. Since s and s' can reach a rejecting state, both states are different from s_+. So $s = s_i$ and $s' = s_j$ for suitable i and j.

Let $n = p - 1$. Since $|u| > 0$, u must contain at least one letter b. So $u \in a^k b A^*$ for some k. It follows that $i + k \equiv 0 \pmod n$ and $j + k \equiv 0 \pmod n$. Therefore,

$$i \equiv j \pmod n. \tag{8}$$

We already observed in Claim 5.3, that in every loop that does not pass s_+, the number of a's is equal to n times the number of b's. Therefore,

$$\text{every loop that does not pass } s_+ \text{ has length } \equiv 0 \pmod{n+1}. \tag{9}$$

In particular,

$$|v| \equiv |u| \equiv 0 \pmod{n+1}. \tag{10}$$

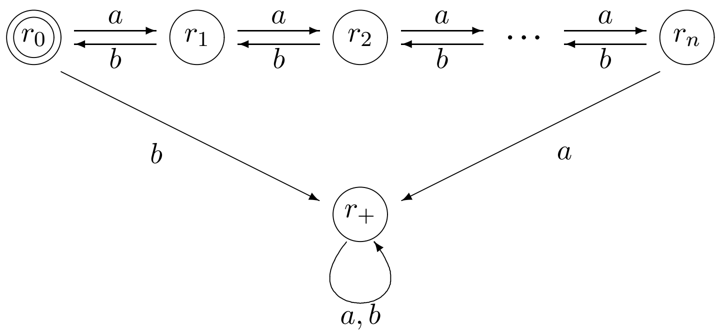

Figure 8: Automaton $\mathcal{D}_n$ where $n \geq 1$, r_0 is initial state, and r_n is the only rejecting state. All $\mathcal{D}_n$ are minimal, and for $n \geq 1$ it holds that $L(\mathcal{D}_n) \in \mathcal{B}_n - \mathcal{B}_{n-1}$ [Tho84].

Now consider the path from s_i to s_j induced by the word v. Along this way we cut out possible loops. Recall that these are of length $\equiv 0 \pmod{n+1}$. We obtain a direct path from s_i to s_j. By (8), this path must be of length kn where $1 \leq k \leq n-1$. Therefore, $|v| \equiv kn \pmod{n+1}$. From (10) it follows that

$$kn \equiv 0 \pmod{n+1}. \tag{11}$$

This is impossible, since $k \not\equiv 0 \pmod{n+1}$, $n \not\equiv 0 \pmod{n+1}$, and $n+1 = p$ is a prime. □

Claim 5.5 *For any prime $p \geq 3$, $L(\mathcal{A}_p) \notin \mathcal{B}_{p-3}$.*

Proof Let $n = p - 1$. Consider the morphism $\varphi : A^* \mapsto A^*$ such that:

$$a \mapsto a^n$$
$$b \mapsto b$$

Denote $\mathcal{A}_p$'s transition function by $\delta_{\mathcal{A}}$ and $\mathcal{D}_{n-1}$'s transition function by $\delta_{\mathcal{D}}$. For every w,

$$\text{if } \delta_{\mathcal{D}}(s_0, w) = r_i, \text{ then } \delta_{\mathcal{A}}(s_0, \varphi(w)) = s_{in},$$
$$\text{if } \delta_{\mathcal{D}}(s_0, w) = r_+, \text{ then } \delta_{\mathcal{A}}(s_0, \varphi(w)) = s_+.$$

Moreover, r_{n-1} is the only rejecting state in $\mathcal{D}_{n-1}$, and $s_{(n-1)n}$ is the only rejecting state in $\mathcal{A}_p$. It follows that $L(\mathcal{D}_{n-1}) = \varphi^{-1}(L(\mathcal{A}_p))$.

It is known that all classes of the dot-depth hierarchy are closed under inverse morphisms. Therefore, if $L(\mathcal{A}_p)$ belongs to $\mathcal{B}_{p-3}$, then so does $L(\mathcal{D}_{n-1})$. This is not possible, since Thomas [Tho84] showed $L(\mathcal{D}_{n-1}) \in \mathcal{B}_{n-1} - \mathcal{B}_{n-2}$. It follows that $L(\mathcal{A}_p) \notin \mathcal{B}_{p-3}$. □

This finishes the proof of Theorem 5.1. □

Corollary 5.6 *For every n, there exists a starfree language L such that L is polylog-time reducible to a language in $\mathcal{B}_{1/2}$, but L does not belong to $\mathcal{B}_n$.*

Niedermeier and Rossmanith introduced unambiguous alternating polynomial-time-bounded Turing machines.

Definition 5.7 ([NR98]) *An alternating Turing machine is called* unambiguous, *if for all inputs its computation tree neither contains existential nodes with more than one accepting successor nor contains universal nodes with more than one rejecting successor.*

The levels of the unambiguous alternation hierarchy *are defined as follows: Level* $\mathrm{AU\Sigma_k^P}$, $k \geq 1$, *is the set of languages accepted by unambiguous alternating polynomial-time-bounded Turing machines that have at most* $k-1$ *alternations between existential and universal configurations, starting with an existential one. Analogously define* $\mathrm{AU\Pi_k^P}$*; here the machines start with universal configurations.*

In their paper, Niedermeier and Rossmanith [NR98] cite Hemaspaandra [unpublished] for a characterization of $\mathrm{AU\Sigma_k^P}$ and $\mathrm{AU\Pi_k^P}$ in terms of unambiguous alternating quantifiers. For any complexity class $\mathcal{C}$, define $\exists!{\cdot}\mathcal{C}$ as the class of languages L such that there exist a polynomial p and $L' \in \mathcal{C}$ such that for all x,

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \text{there exists exactly one } y \in A^{=p(|x|)} \text{ such that } (x,y) \in L' \\ x \notin L \quad &\Rightarrow \quad \text{there exists no } y \in A^{=p(|x|)} \text{ such that } (x,y) \in L' \end{aligned}$$

Analogously define $\forall!{\cdot}\mathcal{C}$.

Theorem 5.8 (Hemaspaandra [unpublished]) *For every* $k \geq 1$,

$$\begin{aligned} \mathrm{AU\Sigma_{k+1}^P} &= \exists!{\cdot}\mathrm{AU\Pi_k^P} \quad \textit{and} \\ \mathrm{AU\Pi_{k+1}^P} &= \forall!{\cdot}\mathrm{AU\Sigma_k^P}. \end{aligned}$$

Theorem 5.9 *For every* $k \geq 1$ *and every* $p \geq 4k+2$, $\mathrm{AU\Sigma_k^P} \subseteq \mathrm{Leaf}_u^p(L(\mathcal{A}_p))$.

Proof Define the following subclasses of SPP.[5] For $k \geq 2$, let $\mathrm{SPP_k}$ be the class of languages L such that there exists a nondeterministic, polynomial-time-bounded Turing machine M such that for all x,

1. $M(x)$ is a complete binary tree whose leafs are labeled with 0 (reject) or 1 (accept),
2. for every prefix v of the leaf word of $M(x)$, $|\#_0(v) - \#_1(v)| \leq k$, and
3. $x \in L \Rightarrow \mathrm{acc}_M(x) - \mathrm{rej}_M(x) = 2$,
   $x \notin L \Rightarrow \mathrm{acc}_M(x) - \mathrm{rej}_M(x) = 0$.

Claim 5.10 *For every* $k \geq 2$, $\mathrm{SPP_k} = \mathrm{coSPP_k}$.

Proof Let $L \in \mathrm{SPP_k}$ via machine M. Define a machine M' as follows: On input x the machine splits the computation into two branches. On the left branch M' simulates $M(x)$ where 0's are replaced by 1's, and 1's are replaced by 0's. On the right branch M' generates a binary tree of same size as $M(x)$ such that the leaf word is $110101\cdots01$. Observe that $\overline{L} \in \mathrm{SPP_k}$ via machine M'. □

Claim 5.11 *For* $k \geq 2$, $\exists!{\cdot}\mathrm{SPP_k} \subseteq \mathrm{SPP_{k+2}}$.

Proof Let $L \in \exists!{\cdot}\mathrm{SPP_k}$ via polynomial p and $L' \in \mathrm{SPP_k}$. Let M' be a nondeterministic machine that witnesses $L' \in \mathrm{SPP_k}$. Define M to be the following nondeterministic machine: On input x, M guesses $y \in A^{=|x|}$ and then simulates $M'(x,y)$.

[5] SPP was introduced in 1991 independently by Fenner *et al.* [FFK94], Gupta [Gup91] (under the name ZUP), and Ogiwara and Hemachandra [OH93] (under the name XP). $L \in \mathrm{SPP}$ if and only if there exists $f \in \mathrm{GapP}$ such that for all x: If $x \in L$, then $f(x) = 1$. If $x \notin L$, then $f(x) = 0$.

Clearly, M is a nondeterministic, polynomial-time-bounded Turing machine such that on input x the computation tree is a complete binary tree labeled with 0's and 1's.

$$\begin{aligned} x \in L &\Rightarrow \text{ there exists exactly one } y \in A^{=p(|x|)} \text{ such that } (x,y) \in L' \\ &\Rightarrow \mathrm{acc}_M(x) - \mathrm{rej}_M(x) = 2 \\ x \notin L &\Rightarrow \text{ for all } y \in A^{=p(|x|)},\ (x,y) \notin L' \\ &\Rightarrow \mathrm{acc}_M(x) - \mathrm{rej}_M(x) = 0 \end{aligned}$$

For every x, there exists at most one $y \in A^{=p(|x|)}$ such that $\mathrm{acc}_{M'}(x,y) - \mathrm{rej}_{M'}(x,y)$ differs from 0, and for this y it holds that $\mathrm{acc}_{M'}(x,y) - \mathrm{rej}_{M'}(x,y) = 2$. Moreover, for every $y \in A^{=p(|x|)}$ and every prefix v of the leaf word of $M'(x,y)$, $|\#_0(v) - \#_1(v)| \le k$. It follows that for every prefix v of the leaf word of $M(x)$, $|\#_0(v) - \#_1(v)| \le k+2$. This shows $L \in \mathrm{SPP_{k+2}}$. □

Claim 5.12 *For* $k \ge 1$, $\mathrm{AU\Sigma_k^P} \subseteq \mathrm{SPP_{2k}}$.

Proof The proof is by induction. For $k=1$, $\mathrm{AU\Sigma_k^P} = \mathrm{UP}$. We can modify any UP-machine such that accepting paths output 11 and rejecting paths output 01. This shows $\mathrm{AU\Sigma_1^P} \subseteq \mathrm{SPP_2}$.

By Theorem 5.8, for $k \ge 2$, $\mathrm{AU\Sigma_k^P} = \exists!\cdot\mathrm{AU\Pi_{k-1}^P}$. By induction hypothesis, $\mathrm{AU\Sigma_{k-1}^P} \subseteq \mathrm{SPP_{2k-2}}$. Hence, by Claim 5.10, $\mathrm{AU\Pi_{k-1}^P} \subseteq \mathrm{SPP_{2k-2}}$. From Claim 5.11 it follows $\mathrm{AU\Sigma_k^P} \subseteq \mathrm{SPP_{2k}}$. □

It remains to show:

Claim 5.13 *For every* $k \ge 2$ *and every* $p \ge 2k+2$, $\mathrm{SPP_k} \subseteq \mathrm{Leaf}_u^p(L(\mathcal{A}_p))$.

Proof Let $n = p-1$. Observe that the following $2k+1$ states appear in $\mathcal{A}_p$: $s_0, s_n, s_{2n}, \ldots, s_{2kn}$. Denote these states as $z_{-k}, z_{-k+1}, \ldots, z_k$, i.e., $z_i \stackrel{df}{=} s_{(i+k)n}$ where $-k \le i \le k$. Moreover, $z_- \stackrel{df}{=} s_{(n-1)n}$ is a rejecting state, and $z_+ \stackrel{df}{=} s_+$ is an accepting state. Let δ be $\mathcal{A}_p$'s transition function. Define $y \stackrel{df}{=} a^{nk}$ and $z \stackrel{df}{=} a^{(n-1)n-nk}$. Observe that $\delta(s_0, y) = z_0$, $\delta(z_0, z) = z_-$, and $\delta(z_2, z) = z_+$.

Define $h : A^* \mapsto A^*$ to be the morphism mapping 0 with b, and 1 to a^n. Let $L \in \mathrm{SPP_k}$ via machine M. For every x, let w_x denote the leaf word of $M(x)$. Let M' be an unbalanced nondeterministic Turing machine that generates the following leaf word on input x:

$$w'_x \stackrel{df}{=} yh(w_x)z.$$

Clearly, after reading prefix y of w'_x, the automaton $\mathcal{A}_p$ is in state z_0. Now the following holds:

1. Whenever $\mathcal{A}_p$ is in state z_j and there is a 0 in w_x, then $\mathcal{A}_p$ reads b and goes to state z_{j-1}.
2. Whenever $\mathcal{A}_p$ is in state z_j and there is a 1 in w_x, then $\mathcal{A}_p$ reads a^n and goes to state z_{j+1}.

So $\mathcal{A}_p$ counts $\#_1(w_x) - \#_0(w_x)$. Since M is an $\mathrm{SPP_k}$ machine, for every prefix v of w_x, $|\#_0(v) - \#_1(v)| \le k$. Therefore,

$$\begin{aligned} x \in L &\Rightarrow \#_1(w_x) - \#_0(w_x) = 2 \\ &\Rightarrow \delta(s_0, yh(w_x)) = z_2 \\ &\Rightarrow \delta(s_0, w'_x) = z_+, \quad \text{and} \\ x \notin L &\Rightarrow \#_1(w_x) - \#_0(w_x) = 0 \\ &\Rightarrow \delta(s_0, yh(w_x)) = z_0 \\ &\Rightarrow \delta(s_0, w'_x) = z_-. \end{aligned}$$

So $x \in L$ if and only if $w'_x \in L(\mathcal{A}_p)$. It follows that $L \in \mathrm{Leaf}_u^p(L(\mathcal{A}_p))$. □

The theorem follows from the Claims 5.12 and 5.13. □

Corollary 5.14 *For all* $k \ge 1$ *there exists* $L \in \mathrm{SF}$ *such that* $\mathrm{AU\Sigma_k^P} \subseteq \mathrm{Leaf}_u^p(L)$ *but* $\mathrm{Leaf}_b^p(L) \subseteq \mathrm{NP}$.

Acknowledgments

The author thanks Bernhard Schwarz, Victor Selivanov, and Klaus W. Wagner for exciting discussions about leaf languages and reducibility notions for starfree languages. In particular, Example 3.7 was discussed in depth in this group.

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.

[BKS98] B. Borchert, D. Kuske, and F. Stephan. On existentially first-order definable languages and their relation to NP. In *Proc. 25rd International Colloquium on Automata, Languages and Programming*, volume 1443 of *Lecture Notes in Computer Science*, pages 17–28. Springer Verlag, 1998.

[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.

[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.

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

[FFK94] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48:116–148, 1994.

[GS00] C. Glaßer and H. Schmitz. Languages of dot-depth 3/2. In *Proceedings 17th Symposium on Theoretical Aspects of Computer Science*, volume 1770 of *Lecture Notes in Computer Science*, pages 555–566. Springer Verlag, 2000.

[Gup91] S. Gupta. The power of witness reduction. In *Proceedings 6th Structure in Complexity Theory*, pages 43–59. IEEE Computer Society Press, 1991.

[GW03] C. Glaßer and G. Wechsung. Relativizing function classes. *Journal of Universal Computer Science*, 9(1):34–50, 2003.

[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.

[Kna83] R. Knast. A semigroup characterization of dot-depth one languages. *RAIRO Inform. Théor.*, 17:321–330, 1983.

[MP71] R. McNaughton and S. Papert. *Counterfree Automata*. MIT Press, Cambridge, 1971.

[NR98] R. Niedermeier and P. Rossmanith. Unambiguous computations and locally definable acceptance types. *Theoretical Computer Science*, 194(1-2):137–161, 1998.

[OH93] M. Ogiwara and L. Hemachandra. A complexity theory of feasible closure properties. *Journal of Computer and System Sciences*, 46:295–325, 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.

[Sch65] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.

[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.

[Wag01] K. W. Wagner. A reducibility and complete sets for the dot-depth hierarchy. Manuscript, 2001.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'