Leaf Language Classes

A Survey

Klaus W. Wagner*
Institut für Informatik
Julius-Maximilians-Universität Würzburg
wagner@informatik.uni-wuerzburg.de

Abstract

The theory of leaf language classes is a fruitful field of research which has been developed since the beginning of the nineties. The leaf language model, in which one language (or a pair of languages) defines a class of languages, allows a uniform definition and treatment of many complexity classes. The results of this area give new insights into the structure of complexity classes and their relation to other fields of Theoretical Computer Science.

1 Introduction

Complexity classes based on nondeterministic computations have been studied since the beginning of the theory of compuational complexity. Such a complexity class is defined by a class of nondeterministic machines (e.g., nondeterministic polynomial time Turing machines) and a notion of acceptance. Since a nondeterministic machine can have many computation paths on a given input, the notion of acceptance has to make precise how the outcomes of these computation paths determine whether the input is accepted or not. In the simplest case of nondeterministic acceptance there has to be at least one accepting computation path, in the case of probabilistic acceptance there have to be more accepting paths than rejecting paths. During the last forty years more and more different and sophisticated notions of acceptance have been introduced and investigated.

Only at the beginning of the nineties a convincing step to unify these many different approaches was made. Bovet, Crescenzi, and Silvestri [BCS91, BCS92] and independently Vereshchagin [Ve93] developed a method to define notions of acceptance for nondeterministic machines which became later on known as the leaf language approach. (In [Vo03] it is mentioned that Papadimitriou and Sipser used this method in their lectures on complexity theory already around 1979.)

The essence of this method can be described easily. Consider a nondeterministic polynomial time Turing machine M. On input x, every computation path of M produces a symbol from a finite alphabet Σ. Considering the computation paths of M on x in lexicographical order the produced symbols build a finite sequence $\beta_M(x) \in \Sigma^*$, the *leaf word of M on x*. A language $L \subseteq \Sigma^*$ defines the *leaf language class* $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(L)$ of all languages A for which there exists a nondeterministic polynomial time Turing machine M such that $x \in A \Leftrightarrow \beta_M(x) \in L$ for every

*Supported by the Deutsche Forschungsgemeinschaft, grant Wa 847/2

x. In this sense one obtains immediately $\mathrm{NP} = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(0^*1(0 \cup 1)^*)$ and $\mathrm{PP} = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(\{x : x \in \{0,1\}^*$ and there are more 1's than 0's in $x\})$. Also less obvious characterizations of complexity classes like $\mathrm{P}^{\mathrm{NP}} = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}((0 \cup 1 \cup 2)^*10^*)$ can be proven.

Restricting in the above definiton the class of machines to such with balanced computation trees one obtains the *balanced leaf language classes* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L)$ which might differ from the unbalanced classes as can be exemplified by $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(\mathrm{L}_{\mathrm{mid}}) = \mathrm{P}$ and $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(\mathrm{L}_{\mathrm{mid}}) = \mathrm{P}^{\mathrm{PP}}$ where $\mathrm{L}_{\mathrm{mid}} =_{\mathrm{def}} \{x1y : x,y \in \{0,1\}^* \wedge |x| = |y|\}$. Restricting the class of machines to such with leaf words from a given language one can describe so-called *promise complexity classes* like BPP and UP. In fine, it turns out that practically every complexity class considered so far can be described by leaf languages.

The leaf language approach to define complexity classes has proven to be very fruitful over the last dozen of years. Here are some facts:

- All leaf language classes share some common properties. To prove these properties for a given class it is sufficient to show that this class can be described by a leaf language.
- The inclusion between two leaf language classes under every relativization is equivalent to a certain type of reducibility between the corresponding leaf languages. This can be used to shorten dramatically many oracle separation proofs in complexity theory.
- The study of classes defined by regular leaf languages has shown some very unexpected and interesting connections between the algebraic properties of leaf languages and the position of the corresponding leaf language classes in the landscape of complexity classes.

This paper surveys the (in my understanding) most important results on leaf language classes. However, a certain restriction is enforced here by the page restriction of this paper. It should be mentioned that [Vo99] and [Vo03] are further survey papers on leaf language classes. They differ from our survey in the choice and presentation of the results. Also, Heribert Vollmer's Leaf Language Homepage http://www.thi.uni-hannover.de/forschung/leafl/ is a valuable source for this field of research.

2 Notations of Complexity Classes

Since many different complexity classes are used in this paper we will give here a brief summary of the corresponding notations.

For any function $t : \mathbb{N} \to \mathbb{N}$ such that $t(n) \geq n$, let DTIME(t) and NTIME(t) be the classes of languages which can be accepted by deterministic and nondeterministic, resp., Turing machines within time t. For any function $s : \mathbb{N} \to \mathbb{N}$ such that $s(n) \geq \log n$, let DSPACE(s) and NSPACE(s) be the classes of languages which can be accepted by deterministic and nondeterministic, resp., Turing machines within space s. For classes $\mathcal{F} \subseteq \mathbb{N}^{\mathbb{N}}$ define $\mathrm{DTIME}(\mathcal{F}) =_{\mathrm{def}} \bigcup_{t \in \mathcal{F}} \mathrm{DTIME}(t)$, $\mathrm{NTIME}(\mathcal{F}) =_{\mathrm{def}} \bigcup_{t \in \mathcal{F}} \mathrm{NTIME}(t)$, $\mathrm{DSPACE}(\mathcal{F}) =_{\mathrm{def}} \bigcup_{s \in \mathcal{F}} \mathrm{DSPACE}(s)$, and $\mathrm{NSPACE}(\mathcal{F}) =_{\mathrm{def}} \bigcup_{s \in \mathcal{F}} \mathrm{NSPACE}(s)$. Define $\mathrm{Pol} =_{\mathrm{def}} \{n^k : k \geq 1\}$ and $2^{\mathrm{Pol}} =_{\mathrm{def}} \{2^{n^k} : k \geq 1\}$. In particular, we set $\mathrm{P} =_{\mathrm{def}} \mathrm{DTIME}(\mathrm{Pol})$, $\mathrm{NP} =_{\mathrm{def}} \mathrm{NTIME}(\mathrm{Pol})$, $\mathrm{L} =_{\mathrm{def}} \mathrm{DSPACE}(\log)$, $\mathrm{NL} =_{\mathrm{def}} \mathrm{NSPACE}(\log)$, and $\mathrm{PSPACE} =_{\mathrm{def}} \mathrm{DSPACE}(\mathrm{Pol})$. For functions $f, g \in \mathbb{N}$ define $(f \circ g)(n) =_{\mathrm{def}} f(g(n))$ for $n \in \mathbb{N}$, and for classes $\mathcal{F}, \mathcal{G} \subseteq \mathbb{N}^{\mathbb{N}}$ define $\mathcal{F} \circ \mathcal{G} =_{\mathrm{def}} \{f \circ g : f \in \mathcal{F} \wedge g \in \mathcal{G}\}$.

The class PP is defined as the class of languages which can be accepted by probabilistic polynomial time Turing machines in such a way that an input is accepted if the acceptance probability is greater than 1/2. If we replace "greater than 1/2" by "exactly 1/2" we obtain the class $\mathrm{C}_{=}\mathrm{P}$.

The *bounded error* class BPP is also defined by probabilistic polynomial time Turing machines, where an accepted (rejected) input has a acceptance probability greater than 2/3 (less than 1/3, resp.). If we consider the ratio of accepting paths rather than the acceptance probability then this does not change the class PP and $\mathrm{\Subset P}$ but in the bounded error case we obtain the class $\mathrm{BPP}_{\mathrm{path}} \supseteq \mathrm{BPP}$.

For a class $\mathcal{K}$ accepted in a certain way by machines of a certain type and a language X, the *relativized class* $\mathcal{K}^X$ is the class of languages accepted in the same way by machines of the same type which in addition have access to X as an oracle. Note that this general definition needs some more explanation in special cases. For a class $\mathcal{M}$ of languages set $\mathcal{K}^{\mathcal{M}} =_{\mathrm{def}} \bigcup_{X\in\mathcal{M}} \mathcal{K}^X$. In particular, $P^{\mathcal{M}}$ is the class of languages which can be accepted by deterministic Turing machines in polynomial time with an oracle from $\mathcal{M}$. Let further $P^{\mathcal{M}}[1]$ be the class of languages which can be accepted by deterministic Turing machines in polynomial time with one query to an oracle from $\mathcal{M}$.

Let $A\triangle B =_{\mathrm{def}} (A\setminus B)\cup(B\setminus A)$ denote the *symmetric difference* of the sets A and B. Let A, B be languages over the finite alphabet Σ, and let a and b be different symbols from Σ. The set $A \uplus B =_{\mathrm{def}} aA \cup bB$ is said to be the *join* of the sets A and B.

Now we consider special operators on language classes. Let $\mathcal{K}$ and $\mathcal{M}$ be classes of languages, and let $k \geq 2$. We define

$$
\begin{array}{rll}
\text{co-}\mathcal{K} & =_{\mathrm{def}} & \{\overline{A} : A \in \mathcal{K}\},\\
\mathcal{K}\wedge\mathcal{M} & =_{\mathrm{def}} & \{A\cap B : A\in\mathcal{K} \text{ and } B\in\mathcal{M}\},\\
\mathcal{K}\oplus\mathcal{M} & =_{\mathrm{def}} & \{A\triangle B : A\in\mathcal{K} \text{ and } B\in\mathcal{M}\},\\
\mathcal{K}\nabla\mathcal{M} & =_{\mathrm{def}} & \{A : \exists D(D\in\mathrm{P}\wedge A\cap D\in\mathcal{K}\wedge A\cap\overline{D}\in\mathcal{M})\},\\
\exists\cdot\mathcal{K} & =_{\mathrm{def}} & \{A : \text{there exists a polynomial } p \text{ and a } B\in\mathcal{K} \text{ such that}\\
 & & \quad \forall x(x\in A \leftrightarrow \exists z(|z| = p(|x|)\wedge(x,z)\in B))\},\\
\forall\cdot\mathcal{K} & =_{\mathrm{def}} & \{A : \text{there exists a polynomial } p \text{ and a } B\in\mathcal{K} \text{ such that}\\
 & & \quad \forall x(x\in A \leftrightarrow \forall z(|z| = p(|x|)\rightarrow(x,z)\in B))\},\\
\exists!\cdot\mathcal{K} & =_{\mathrm{def}} & \{A : \text{there exists a polynomial } p \text{ and a } B\in\mathcal{K} \text{ such that}\\
 & & \quad \forall x(x\in A \leftrightarrow |\{z : |z| = p(|x|)\wedge(x,z)\in B)\}| = 1)\},\\
\mathrm{U}\cdot\mathcal{K} & =_{\mathrm{def}} & \{A : \text{there exists a polynomial } p \text{ and a } B\in\mathcal{K} \text{ such that}\\
 & & \quad \forall x(x\in A \rightarrow |\{z : |z| = p(|x|)\wedge(x,z)\in B\}| = 1) \text{ and}\\
 & & \quad \forall x(x\notin A \rightarrow |\{z : |z| = p(|x|)\wedge(x,z)\in B\}| = 0)\},\\
\mathrm{Mod}_k\cdot\mathcal{K} & =_{\mathrm{def}} & \{A : \text{there exists a polynomial } p \text{ and a } B\in\mathcal{K} \text{ such that}\\
 & & \quad \forall x(x\in A \leftrightarrow |\{z : |z| = p(|x|)\wedge(x,z)\in B\}| \equiv 0(k))\}.\\
\mathrm{C}\cdot\mathcal{K} & =_{\mathrm{def}} & \{A : \text{there exists a polynomial } p \text{ and a } B\subseteq\Sigma^*\times\Delta^* \text{ such that } B\in\mathcal{K} \text{ and}\\
 & & \quad \forall x(x\in A \leftrightarrow |\{z : |z| = p(|x|)\wedge(x,z)\in B\}| \geq \frac{1}{2}\cdot(\#\Delta)^{p(|x|)})\}.\\
\Subset\cdot\mathcal{K} & =_{\mathrm{def}} & \{A : \text{there exists a polynomial } p \text{ and a } B\subseteq\Sigma^*\times\Delta^* \text{ such that } B\in\mathcal{K} \text{ and}\\
 & & \quad \forall x(x\in A \leftrightarrow |\{z : |z| = p(|x|)\wedge(x,z)\in B\}| = \frac{1}{2}\cdot(\#\Delta)^{p(|x|)})\}.
\end{array}
$$

In particular, there holds $\exists\cdot\mathrm{P} = \mathrm{NP}$, $\forall\cdot\mathrm{P} = \text{co-NP}$, $\mathrm{C}\cdot\mathrm{P} = \mathrm{PP}$, and $\Subset\cdot\mathrm{P} = \mathrm{\Subset P}$. We define $1\text{-NP} =_{\mathrm{def}} \exists!\cdot\mathrm{P}$, $\mathrm{UP} =_{\mathrm{def}} \mathrm{U}\cdot\mathrm{P}$, $\mathrm{Mod}_k\mathrm{P} =_{\mathrm{def}} \mathrm{Mod}_k\cdot\mathrm{P}$ for $k\geq 2$, and $\oplus\mathrm{P} =_{\mathrm{def}} \mathrm{Mod}_2\cdot\mathrm{P}$.

The *boolean hierarchy* over a language class $\mathcal{K}$ is the smallest family of language classes that contains $\mathcal{K}$ and that contains with the classes $\mathcal{M}$ and $\mathcal{L}$ also the classes co-$\mathcal{M}$ and $\mathcal{M}\wedge\mathcal{L}$. If $\mathcal{K}$ is closed under union and intersection then every class of the boolean hierachy over $\mathcal{K}$ coincides with one of the classes $\mathcal{K}(k)$ or co-$\mathcal{K}(k)$ where $\mathcal{K}(1) =_{\mathrm{def}} \mathcal{K}$ and $\mathcal{K}(k+1) =_{\mathrm{def}} \mathcal{K}(k)\oplus\mathcal{K}$ for $k\geq 1$. Notice that $\mathcal{K}(k+1) = \text{co-}\mathcal{K}(k)\wedge\mathcal{K}$ for $k\geq 1$. Let $\mathrm{BH} =_{\mathrm{def}} \bigcup_{k\geq 1}\mathrm{NP}(k)$.

The *polynomial hierarchy* over a language class $\mathcal{K}\supseteq\mathrm{P}$ is the smallest family of language classes that contains $\mathcal{K}$ and that contains with the class $\mathcal{M}$ also the classes co-$\mathcal{M}$, $\exists\cdot\mathcal{M}$, and $\forall\cdot\mathcal{M}$. We define $\Sigma_0^{\mathrm{p}} =_{\mathrm{def}} \Pi_0^{\mathrm{p}} =_{\mathrm{def}} \Delta_0^{\mathrm{p}} =_{\mathrm{def}} \mathrm{P}$, $\Delta_{k+1}^{\mathrm{p}} =_{\mathrm{def}} \mathrm{P}^{\Sigma_k^{\mathrm{p}}}$, $\Sigma_{k+1}^{\mathrm{p}} =_{\mathrm{def}} \exists\cdot\Pi_k^{\mathrm{p}}$, $\Pi_{k+1}^{\mathrm{p}} =_{\mathrm{def}} \forall\cdot\Sigma_k^{\mathrm{p}}$ for $k \geq 0$, and $\mathrm{PH} =_{\mathrm{def}} \bigcup_{k\geq 0}(\Sigma_k^{\mathrm{p}} \cup \Pi_k^{\mathrm{p}} \cup \Delta_k^{\mathrm{p}})$. This hierarchy is called the *polynomial time hierarchy.*

The *modulo hierarchy* over a language class $\mathcal{K} \supseteq \mathrm{P}$ is the smallest family of language classes that contains $\mathcal{K}$ and that contains with the class $\mathcal{M}$ also the classes co-$\mathcal{M}$, $\exists\cdot\mathcal{M}$, $\forall\cdot\mathcal{M}$, and $\mathrm{Mod}_k\cdot\mathcal{M}$ for every $k \geq 2$. Let MOD-PH be the union of all classes of the modulo hierarchy over P. Note that every class of the polynomial hierarchy over $\mathcal{K}$ is also a class of the modulo hierarchy over $\mathcal{K}$, and thus $\mathrm{PH} \subseteq \mathrm{MOD\text{-}PH}$.

The *counting hierarchy* over a language class $\mathcal{K} \supseteq \mathrm{P}$ is the smallest family of language classes that contains $\mathcal{K}$ and that contains with the class $\mathcal{M}$ also the class $\mathrm{C}\cdot\mathcal{M}$.

For $k \geq 1$, let Σ_k-LOGTIME be the class of all languages which can be accepted by a Σ_k-alternating Turing machine in logarithmic time. Such a machine accesses the input as an oracle, it starts with an existing configuration, and it alternates at most k–1 times on every computation path. Let $\mathrm{AC}^0 =_{\mathrm{def}} \bigcup_{k\geq 1} \Sigma_k$-LOGTIME. Alternatively, AC^0 is the class of all languages which can be accepted by uniform unbounded fan-in $\{\wedge, \vee, \neg\}$-circuits of polynomial size and constant depth. Let NC^1 be the class of all languages which can be accepted by an alternating Turing machine in logarithmic time. Alternatively, NC^1 is the class of all languages which can be accepted by uniform bounded fan-in $\{\wedge, \vee, \neg\}$-circuits of polynomial size and logarithmic depth.

The polynomial time many-one reducibility is denoted by $\leq_{\mathrm{m}}^{\mathrm{p}}$.

For more details on complexity classes the reader is referred to [BDG88], [BDG90], [Pa94], and [Aa04].

3 Definitions and Examples

In this section we introduce the notion of leaf language classes [BCS91, Ve93], we give some examples of leaf language characterizations of well-known complexity classes, and we state some basic properties of leaf language classes.

Let M be a nondeterministic polynomial time machine that, in every step, splits a computation path into at most two computation paths. Hence a computation path of M on input x can be described by a word from $\{0,1\}^*$. For example, a path described by 010 has three branchings, at the the first branching it takes the first way, at the second branching it takes the second way, and at the third branching it takes the first way. Let the nondeterministic polynomial time machine M produce on every computation path z on input x a symbol $M(x,z)$ from a finite alphabet Σ. The manchine M is called an *ε-machine* if it is also allowed to produce the empty word ε, i.e., if $M(x,z) \in \Sigma \cup \{\varepsilon\}$.

Let $r_1, r_2, \ldots, r_k \in \{0,1\}^*$ be the computation paths of M on input x in lexicographical order ($\leq$). The *leaf word* of M on input x is defined as $\beta_M(x) =_{\mathrm{def}} M(x,r_1)M(x,r_2)\ldots M(x,r_k)$.

A machine is called *balanced* if for every input x there exists an $m \geq 0$ and an $r \in \{0,1\}^m$ such that $\{s : s \in \{0,1\}^m \wedge s \leq r\}\cdot\{0,1\} \cup \{s : s \in \{0,1\}^m \wedge s > r\}$ is the set of all computation paths of M on x.

We define three types of *leaf language classes*, the classes $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_2)$ of the *unbalanced model*, the classes $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2)$ of the *balanced model*, and the classes $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2)$ of the *ε-model.* Let L_1 and L_2 be languages such that $L_1 \cap L_2 = \emptyset$, let Σ be a finite alphabet, and let $A \subseteq \Sigma^*$.

$A \in \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_2) \Leftrightarrow_{\mathrm{def}}$ there exists a nondeterministic polynomial time machine M such that, for every $x \in \Sigma^*$,
$x \in A \to \beta_M(x) \in L_1$ and $x \notin A \to \beta_M(x) \in L_2$,

$A \in \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2) \Leftrightarrow_{\mathrm{def}}$ there exists a balanced nondeterministic polynomial time machine M such that, for every $x \in \Sigma^*$,
$x \in A \to \beta_M(x) \in L_1$ and $x \notin A \to \beta_M(x) \in L_2$.

$A \in \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2) \Leftrightarrow_{\mathrm{def}}$ there exists a nondeterministic polynomial time ε-machine M such that, for every $x \in \Sigma^*$,
$x \in A \to \beta_M(x) \in L_1$ and $x \notin A \to \beta_M(x) \in L_2$.

As examples let us consider leaf language characterizations of well-known complexity classes. For an alphabet Σ and $a_1, a_2, \ldots, a_n, a \in \Sigma$ let $|a_1 a_2 \ldots a_n|_a =_{\mathrm{def}} \#\{i : 1 \le i \le n \wedge a_i = a\}$. We define the languages

$$\begin{aligned}
\mathrm{L}_{<\varepsilon} &=_{\mathrm{def}} \{x : x \in \{0,1\}^* \wedge |x|_1 < \varepsilon \cdot |x|\}, \\
\mathrm{L}_{>\varepsilon} &=_{\mathrm{def}} \{x : x \in \{0,1\}^* \wedge |x|_1 > \varepsilon \cdot |x|\}, \\
\mathrm{L}_{\mathrm{mid}} &=_{\mathrm{def}} \{x1y : x, y \in \{0,1\}^* \wedge |x| = |y|\}, \text{ and} \\
\mathrm{L}_{\log} &=_{\mathrm{def}} \{z_1 2 z_2 2 \ldots 2 z_m : m \ge 1 \wedge z_1, \ldots, z_m \in \{0,1\}^{\le \log_2 m} \wedge \max_{i=1,\ldots,m} |z_i|_1 \equiv 1(2)\}.
\end{aligned}$$

Let $\alpha(L)$ be the smallest alphabet Σ such that $L \subseteq \Sigma^*$. Define $\overline{L} =_{\mathrm{def}} \alpha(L)^* \setminus L$.

Theorem 3.1 *The pairs (L_1, L_2) of leaf languages shown in the second and third column define the complexity classes shown in the fourth, fifth, and sixth column. No entry in the third column means $L_2 = \overline{L_1}$.*

	L_1	L_2	$\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1\|L_2)$	$\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1\|L_2)$	$\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1\|L_2)$
1.	$\{1\}$		P	P	1-NP
2.	$0^*1(0 \cup 1)^*$		NP	NP	NP
3.	00^*	11^*	P	P	NP $\cap$ co-NP
4.	$0^*1(0 \cup 1)^*2^*$	$0^*2^*3(2 \cup 3)^*$	NP $\cap$ co-NP	NP $\cap$ co-NP	NP $\cap$ co-NP
5.	0^*10^*	0^*	UP	UP	UP
6.	0^*10^*	0^*20^*	UP $\cap$ co-UP	UP $\cap$ co-UP	UP $\cap$ co-UP
7.	0^*10^*		1-NP	1-NP	1-NP
8.	$00^* \cup 10^*1(0 \cup 1)^*$		$\mathrm{P}^{\mathrm{NP}}[1]$	$\mathrm{P}^{\mathrm{NP}}[1]$	$\mathrm{P}^{\mathrm{NP}}[1]$
9.	$(0 \cup 1 \cup 2)^*10^*$		P^{NP}	P^{NP}	P^{NP}
10.	$\mathrm{L}_{\log}$		Θ_2^{p}	Θ_2^{p}	
11.	$(0 \cup 1)^*11(0 \cup 1)^*$		NP	NP	Σ_2^{p}
12.	$(0^*10^*1)^*0^*$		$\oplus$P	$\oplus$P	$\oplus$P
13.	$(11)^*$		P	$\oplus$P	$\oplus$P
14.	$\mathrm{L}_{>1/2}$		PP	PP	PP
15.	$\{0^n1^n : n \ge 1\}$		co-NP	C=P	C=P
16.	$\{0^n1^m : m \ge n \ge 1\}$		co-NP	PP	PP
17.	$\{ww^R : w \in \{0,1\}^*\}$		co-NP		
18.	$\mathrm{L}_{\mathrm{mid}}$		P	P^{PP}	P^{PP}
19.	$\mathrm{L}_{>2/3}$	$\mathrm{L}_{<1/3}$	BPP	$\mathrm{BPP}_{\mathrm{path}}$	$\mathrm{BPP}_{\mathrm{path}}$

□

Let us consider some special cases.

Proposition 3.2 *Let L be a language such that $L \neq \emptyset$.*

1. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(\emptyset|\emptyset) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(\emptyset|\emptyset) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(\emptyset|\emptyset) = \emptyset$.

2. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(\emptyset|L) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}}(\emptyset|L) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(\emptyset|L) = \{\emptyset\}$.
3. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(L|\emptyset) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}}(L|\emptyset) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(L|\emptyset) = \{\Sigma^* : \Sigma \textit{ finite}\}$.

Let L be a language such that $L \notin \{\emptyset, \{\varepsilon\}\}$, and let $a \notin \alpha(L)$

4. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(\{\varepsilon\}|L) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}}(\{\varepsilon\}|L) = \emptyset$ *and* $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(\{\varepsilon\}|L) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(\{a\}|aL)$.
5. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(L|\{\varepsilon\}) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}}(L|\{\varepsilon\}) = \{\Sigma^* : \Sigma \textit{ finite}\}$ *and* $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(L|\{\varepsilon\}) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(aL|\{a\})$.

It follows from this proposition that, if L_1 or L_2 is in $\{\emptyset, \{\varepsilon\}\}$ then the leaf language class $\mathrm{Leaf}_{\tau}^{\mathrm{P}}(L_1|L_2)$ (for $\tau \in \{\mathrm{u}, \mathrm{b}, \mathrm{e}\}$) is trivial or $\mathrm{Leaf}_{\tau}^{\mathrm{P}}(L_1|L_2) = \mathrm{Leaf}_{\tau}^{\mathrm{P}}(L_1'|L_2')$ for suitable languages L_1' and L_2' such that $L_1', L_2' \notin \{\emptyset, \{\varepsilon\}\}$. Hence in what follows we will ignore these special cases. We will consider only pairs (L_1, L_2) of languages such that $L_1 \cap L_2 = \emptyset$ and $L_1, L_2 \notin \{\emptyset, \{\varepsilon\}\}$. Such pairs are called *pairs of leaf languages*.

For $\tau \in \{\mathrm{u}, \mathrm{b}, \mathrm{e}\}$ we define $\mathrm{Leaf}_{\tau}^{\mathrm{P}}(L) =_{\mathrm{def}} \mathrm{Leaf}_{\tau}^{\mathrm{P}}(L|\overline{L})$. For a class $\mathcal{K}$ of languages we set $\mathrm{Leaf}_{\tau}^{\mathrm{P}}(\mathcal{K}) =_{\mathrm{def}} \bigcup_{L \in \mathcal{K}} \mathrm{Leaf}_{\tau}^{\mathrm{P}}(L)$. We consider the same special cases.

Proposition 3.3 *1.* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(\emptyset) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}}(\emptyset) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(\emptyset) = \{\emptyset\}$.
2. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(\Delta^*) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}}(\Delta^*) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(\Delta^*) = \{\Sigma^* : \Sigma \textit{ finite}\}$.
3. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(\{\varepsilon\}) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}}(\{\varepsilon\}) = \emptyset$ *and* $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(\{\varepsilon\}) = \text{co-NP}$.
4. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(\Delta^+) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{P}}(\Delta^+) = \{\Sigma^* : \Sigma \textit{ finite}\}$ *and* $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(\Delta^+) = \mathrm{NP}$.

Hence, in what follows we consider only languages $L \notin \{\emptyset, \{\varepsilon\}, \alpha(L)^*, \alpha(L)^+\}$ which we call *leaf languages*.

The classes of type $\mathrm{Leaf}_{\tau}^{\mathrm{P}}(L_1|L_2)$ are also called *semantic classes* or *promise classes* (since every machine M accepting a language from $\mathrm{Leaf}_{\tau}^{\mathrm{P}}(L_1|L_2)$ fulfills the promise that $\beta_M(x) \in L_1 \cup L_2$ for every input x), and the classes of type $\mathrm{Leaf}_{\tau}^{\mathrm{P}}(L)$ are also called *syntactic classes* or *complementary classes*.

4 Characterizations for Two Types

The leaf words of a balanced polynomial time machine can also be given by two polynomial time computable functions. This provides the following characterization of $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(L_1|L_2)$.

Theorem 4.1 *Let $(L_1, L_2) \subseteq (\Sigma^*)^2$ be a pair of leaf languages, and let $A \in \Delta^*$.*

1. $A \in \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{P}}(L_1|L_2) \Leftrightarrow$ *there exist polynomial time computable $h : \Delta^* \to \mathbb{N}$ and $g : \Delta^* \times \mathbb{N} \to \Sigma$ such that for all $x \in \Delta^*$,*
$x \in A \to g(x,0)g(x,1)g(x,2)\ldots g(x,h(x)) \in L_1$ *and*
$x \notin A \to g(x,0)g(x,1)g(x,2)\ldots g(x,h(x)) \in L_2$.

2. $A \in \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{P}}(L_1|L_2) \Leftrightarrow$ *there exist polynomial time computable $h : \Delta^* \to \mathbb{N}$ and $g : \Delta^* \times \mathbb{N} \to \Sigma \cup \{\varepsilon\}$ such that for all $x \in \Delta^*$,*
$x \in A \to g(x,0)g(x,1)g(x,2)\ldots g(x,h(x)) \in L_1$ *and*
$x \notin A \to g(x,0)g(x,1)g(x,2)\ldots g(x,h(x)) \in L_2$.

5 Relations Between the Different Types

What about the relations between the three types of leaf language classes? A pair $(L_1, L_2) \subseteq (\Sigma^*)^2$ of leaf languages is *paddable* if there exists an $a \in \Sigma$ such that for all $x, y \in \Sigma^*$ there holds

$xy \in L_1 \leftrightarrow xay \in L_1$ and $xy \in L_2 \leftrightarrow xay \in L_2$. A leaf language $L \subseteq \Sigma^*$ is *paddable* if there exists an $a \in \Sigma$ such that for all $x, y \in \Sigma^*$ there holds $xy \in L \leftrightarrow xay \in L$. For $L \subseteq \Sigma^*$ and $a \notin \Sigma$ define $L^a =_{\text{def}} \bigcup_{b_1 b_2 \ldots b_n \in L} a^* b_1 a^* b_2 a^* \ldots a^* b_n a^*$.

Theorem 5.1 *Let (L_1, L_2) be a pair of leaf languages.*

1. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_2) \subseteq \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2)$.
2. $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_2) \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2)^{\mathrm{PP}}$. *This result is optimal.*
3. $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2) \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2)^{\mathrm{PP}}$. *This result is optimal.*
4. $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_2)^{\mathrm{NP}}$. *This result is optimal.*
5. *If (L_1, L_2) is paddable then* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_2) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2)$.
6. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1^a|L_2^a) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1^a|L_2^a) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1^a|L_2^a) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2)$. $\square$

Corollary 5.2 *Let L be a leaf language.*

1. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \subseteq \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L)$.
2. $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L)^{\mathrm{PP}}$. *This result is optimal.*
3. $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L) \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L)^{\mathrm{PP}}$. *This result is optimal.*
4. $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L) \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L)^{\mathrm{NP}}$. *This result is optimal.*
5. *If L is paddable then* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L)$.
6. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L^a) = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L^a) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L^a) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L)$. $\square$

The Statements 5 and 6 of Theorem 5.1 show that a complexity class is of type $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2)$ if and only if it is of type $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1'|L_2')$ where $(L_1'|L_2')$ is paddable. The same is true for the complementary classes. Hence we will concentrate on investigating the classes of the balanced and unbalanced model.

The difference between the balanced and the unbalanced case can be really considerable. This can be the difference between P and P^{PP} as exemplified by Theorem 3.1.16. Notice that P^{PP} includes all the polynomial time hierarchy. This difference can be explained as follows: Given a bit position in $\beta_M(x)$, one can easily compute the computation path of a balanced machine M on x which produces this bit. For an unbalanced machine this is a counting problem, i.e., this needs oracle queries to a PP set. For other examples where the balanced and the unbalanced leaf language classes for the same leaf language differ see Theorem 3.1.12, Theorem 12.3 and the remark after Conjecture 14.5.

6 The Least Leaf Language Classes

The class P is the minimal class which can be described by leaf languages.

Proposition 6.1 *Let $\tau \in \{b, u\}$, and let (L_1, L_2) be a pair of leaf languages.*

1. $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L_1|L_2) \supseteq \mathrm{P}$.
2. *If L_1 or L_2 is finite then* $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L_1|L_2) = \mathrm{P}$.

Problem 6.2 *The regular languages L such that* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \mathrm{P}$ *and* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) = \mathrm{P}$ *are described in Corollary 11.8 and Corollary 11.10, resp. Theorem 3.1.17 shows that there exist non-regular languages L such that* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) = \mathrm{P}$. *Is there a non-regular language L such that* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \mathrm{P}$*? If not, are there non-regular languages L_1 and L_2 such that* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_2) = \mathrm{P}$*?*

Proposition 6.3 [Bo04]

1. *If* (L_1, L_2) *is a pair of leaf languages then* $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2) \supseteq \mathrm{UP} \cap \text{co-UP}$.
2. *If* L *is a leaf language then* $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L) \supseteq \mathrm{UP}$ *or* $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L) \supseteq \text{co-UP}$.

Problem 6.4 *Theorem 3.1.6 shows that Proposition 6.3.1 cannot be improved. On the other hand, we conjecture that Proposition 6.3.2 can be improved. This is because* $\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L) = \mathrm{UP}$ *implies by Theorem 8.1.5 that* UP *has complete problems. However, there are oracles relative to which* UP *has no complete problems. Can* UP *and* co-UP *be replaced with larger classes in Proposition 6.3.2?*

7 The Size of the Alphabet

What happens to a complementary class when we enlarge the size of the alphabet, i.e., will the class $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L) = \mathrm{Leaf}_{\tau}^{\mathrm{p}}(L|\alpha(L)^* \setminus L)$ differ from the class $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L|\Sigma^* \setminus L)$ for $\Sigma \supset \alpha(L)$? For some important cases this difference can be exactly described.

Theorem 7.1 [Mu05] *Let* $\tau \in \{b, u, e\}$, *and let* L *be a paddable or regular leaf language and let* $\Sigma \supset \alpha(L)$ *be a finite alphabet. Then* $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L|\Sigma^* \setminus L) = \mathrm{Leaf}_{\tau}^{\mathrm{p}}(L) \wedge \text{co-NP}$.

Another question is whether leaf languages over a larger alphabet can be stronger than leaf languages over a smaller alphabet. At least in the complementary case and for the balanced type this is not true.

Theorem 7.2 [Mu05]

1. *For every alphabet* Σ *and every leaf language* $L \subseteq \Sigma^*$ *there exists a leaf language* $L' \subseteq \{0,1\}^*$ *such that* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L|\Sigma^* \setminus L) = \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L'|\{0,1\}^* \setminus L')$.
2. *For every leaf language* $L \subseteq \Sigma^*$ *there exists a leaf language* $L' \subseteq \{0,1\}^*$ *such that* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) = \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L')$.

8 Common Properties of Leaf Language Classes

All leaf language classes share some basic properties. A class $\mathcal{K}$ of languages is *recursively presentable* if there exists a recursive set A such that $\mathcal{K} = \{\{x : (i, x) \in A\} : i \in \mathbb{N}\}$.

Theorem 8.1 [BCS91, Bo94b] *Let* $\tau \in \{b, u, e\}$, *let* (L_1, L_2) *be a pair of leaf languages, and let* L *be a leaf language.*

1. $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L_1|L_2)$ *is closed under* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-reducibility.*
2. $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L_1|L_2)$ *is closed under marked union.*
3. $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L_1|L_2)$ *is countable.*
4. $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L)$ *has* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-complete sets.*
5. *If* L_1 *and* L_2 *are recursive then* $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L_1|L_2)$ *is recursively presentable.*
6. *If* L *is recursive then* $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L)$ *has recursive* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-complete sets.*

The statements remain valid relative to every oracle, where in the statements 5 and 6 the oracle B *has to be assumed to be recursive.* □

As a $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete language for $\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L)$ a succinct version of L (via description by circuits) can be chosen. For the balanced case this is shown in [BL96] and [Ve98].

In fact, the leaf language classes of type $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L)$ are the only balanced leaf language classes which have $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete sets.

Theorem 8.2 [BCS92, BS97] *The following are equivalent for (recursive) pair (L_1, L_2) of leaf languages.*

(1) $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2)$ *has a $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete set.*

(2) $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2)^X$ *has a $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete set for every oracle X.*

(3) *There exists a (recursive) language L such that* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2) = \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L)$.

Problem 8.3 *Is Theorem 8.2 valid also for unbalanced leaf language classes?*

9 Families of Leaf Language Classes

In this section we will get an impression about the structure of the most important families of leaf language classes. Let $\tau \in \{\mathrm{b}, \mathrm{u}, \mathrm{e}\}$. For a class $\mathcal{K}$ of languages we define

$$\mathcal{L}_{\tau}^{\mathrm{p}}(\mathcal{K}) =_{\mathrm{def}} \{\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L_1|L_2) : (L_1, L_2) \in \mathcal{K}^2 \text{ is a pair of leaf languages}\}$$

and the "complementary" family

$$\mathcal{C}_{\tau}^{\mathrm{p}}(\mathcal{K}) =_{\mathrm{def}} \{\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L) : L \in \mathcal{K} \text{ is a leaf language}\}.$$

Let ALL, REC, and PAD be the classes of all, all recursive and all paddable, resp., languages. Set $\mathcal{L}_{\tau}^{\mathrm{p}} =_{\mathrm{def}} \mathcal{L}_{\tau}^{\mathrm{p}}(\mathrm{ALL})$ and $\mathcal{C}_{\tau}^{\mathrm{p}} =_{\mathrm{def}} \mathcal{C}_{\tau}^{\mathrm{p}}(\mathrm{ALL})$. Because of Theorem 5.1 and Theorem 5.2 we have

Proposition 9.1 *1.* $\mathcal{L}_{\mathrm{b}}^{\mathrm{p}}(\mathrm{PAD}) = \mathcal{L}_{\mathrm{u}}^{\mathrm{p}}(\mathrm{PAD}) = \mathcal{L}_{\mathrm{e}}^{\mathrm{p}}(\mathrm{PAD}) = \mathcal{L}_{\mathrm{e}}^{\mathrm{p}}$.

2. $\mathcal{C}_{\mathrm{b}}^{\mathrm{p}}(\mathrm{PAD}) = \mathcal{C}_{\mathrm{u}}^{\mathrm{p}}(\mathrm{PAD}) = \mathcal{C}_{\mathrm{e}}^{\mathrm{p}}(\mathrm{PAD}) = \mathcal{C}_{\mathrm{e}}^{\mathrm{p}}$.

Converting in a sense Theorem 8.1 one gets complete characterizations of these families of balanced leaf language classes.

Theorem 9.2 [BS97]

1. $\mathcal{L}_{\mathrm{b}}^{\mathrm{p}} = \{\mathcal{K} : \mathcal{K}$ *is countable and is closed under marked union and* $\leq_{\mathrm{m}}^{\mathrm{p}}\}$
2. $\mathcal{C}_{\mathrm{b}}^{\mathrm{p}} = \{\mathcal{K} : \mathcal{K}$ *has* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-complete sets and is closed under* $\leq_{\mathrm{m}}^{\mathrm{p}}\}$
3. $\mathcal{L}_{\mathrm{b}}^{\mathrm{p}}(\mathrm{REC}) = \{\mathcal{K} : \mathcal{K}$ *is recursively presentable and is closed under marked union and* $\leq_{\mathrm{m}}^{\mathrm{p}}\}$
4. $\mathcal{C}_{\mathrm{b}}^{\mathrm{p}}(\mathrm{REC}) = \{\mathcal{K} : \mathcal{K}$ *has recursive* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-complete sets and is closed under* $\leq_{\mathrm{m}}^{\mathrm{p}}\}$

As a consequence of Theorem 9.2 and Theorem 8.1 we obtain

Corollary 9.3 *1. For every leaf languages L and every B there exists a leaf language L' such that* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L)^B = \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L')$. *If L and B are recursive then L' can also be chosen as a recursive language.*

2. For every pair (L_1, L_2) of leaf languages and every B there exists a pair (L_1', L_2') of leaf languages such that $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_2)^B = \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1'|L_2')$. *If L_1, L_2 and B are recursive then L_1' and L_2' can also be chosen as recursive languages.*

The following theorem clarifies the algebraic nature of some families of leaf language classes. Notice that, if $\mathcal{K}, \mathcal{M} \supseteq \mathrm{P}$ are closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$, then

$$\mathcal{K} \nabla \mathcal{M} = \{A : \text{there exists } B \in \mathcal{K} \text{ and } C \in \mathcal{M} \text{ such that } A \leq_{\mathrm{m}}^{\mathrm{p}} B \,\dot\cup\, C\}.$$

Theorem 9.4 *1.* [Bo94b] $(\mathcal{L}^{\mathrm{p}}_{\mathrm{b}}, \subseteq)$ *and* $(\mathcal{L}^{\mathrm{p}}_{\mathrm{b}}(\mathrm{REC}), \subseteq)$ *are distributive lattices.*
2. [BS97] $(\mathcal{C}^{\mathrm{p}}_{\mathrm{b}}, \subseteq)$ *and* $(\mathcal{C}^{\mathrm{p}}_{\mathrm{b}}(\mathrm{REC}), \subseteq)$ *are distributive upper semi-lattices.* $\square$

The least upper bound and the greatest lower bound (if any) are given by the operations ∇ *and* $\cap$*, resp.*

Leaf language classes of type $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L_1|L_2)$ can be characterized as intersections of leaf language classes of type $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L)$.

Theorem 9.5 [BS97] $\mathcal{L}^{\mathrm{p}}_{\mathrm{b}} = \mathcal{C}^{\mathrm{p}}_{\mathrm{b}} \wedge \mathcal{C}^{\mathrm{p}}_{\mathrm{b}}$ *and* $\mathcal{L}^{\mathrm{p}}_{\mathrm{b}}(\mathrm{REC}) = \mathcal{C}^{\mathrm{p}}_{\mathrm{b}}(\mathrm{REC}) \wedge \mathcal{C}^{\mathrm{p}}_{\mathrm{b}}(\mathrm{REC})$

Problem 9.6 *Are Theorem 9.2, Corollary 9.3, Theorem 9.4, and Theorem 9.5 also valid for unbalanced model and the* ε*-model?*

Between the classes $\mathcal{C}^{\mathrm{p}}_{\tau}$ and $\mathcal{L}^{\mathrm{p}}_{\tau}$ the following inclusions hold true.

Theorem 9.7 [Mu05]
1. $\mathcal{C}^{\mathrm{p}}_{\mathrm{e}} \subseteq \mathcal{C}^{\mathrm{p}}_{\mathrm{u}} \subseteq \mathcal{C}^{\mathrm{p}}_{\mathrm{b}} \subseteq \mathcal{L}^{\mathrm{p}}_{\mathrm{b}}$
2. $\mathcal{C}^{\mathrm{p}}_{\mathrm{e}} \subseteq \mathcal{L}^{\mathrm{p}}_{\mathrm{e}} \subseteq \mathcal{L}^{\mathrm{p}}_{\mathrm{u}} = \mathcal{L}^{\mathrm{p}}_{\mathrm{b}}$

Based on some commonly believed assumptions, further inclusions between these classes do not exist:

Theorem 9.8 [Mu05]
1. If $\mathrm{P} \neq \mathrm{UP}$ *then* $\mathcal{C}^{\mathrm{p}}_{\mathrm{e}} \subset \mathcal{C}^{\mathrm{p}}_{\mathrm{u}}$.
2. If $\mathrm{P} \neq \mathrm{UP} \cap \text{co-UP}$ *then* $\mathcal{L}^{\mathrm{p}}_{\mathrm{e}} \subset \mathcal{L}^{\mathrm{p}}_{\mathrm{u}}$ *and* $\mathcal{C}^{\mathrm{p}}_{\mathrm{u}} \not\subseteq \mathcal{L}^{\mathrm{p}}_{\mathrm{e}}$.
3. If $\mathrm{NP} \cap \text{co-NP}$ *has no* $\leq^{\mathrm{p}}_{\mathrm{m}}$*-complete sets then* $\mathcal{C}^{\mathrm{p}}_{\mathrm{b}} \subset \mathcal{L}^{\mathrm{p}}_{\mathrm{b}}$, $\mathcal{C}^{\mathrm{p}}_{\mathrm{e}} \subset \mathcal{L}^{\mathrm{p}}_{\mathrm{e}}$ *and* $\mathcal{L}^{\mathrm{p}}_{\mathrm{e}} \not\subseteq \mathcal{C}^{\mathrm{p}}_{\mathrm{b}}$.

10 Closure Properties

10.1 Boolean Operators

In this subsection we will study the closure properties of our families of leaf language classes w.r.t. *boolean operators*. Examples for boolean operators are the operators co-, $\wedge$, $\vee$ and $\oplus$. Let us introduce this notion more generally. For an n-ary boolean function f and languages $A_1, A_2, \ldots, A_n \subseteq \Sigma^*$ we define the language $f(A_1, A_2, \ldots, A_n) \subseteq \Sigma^*$ by

$$f(A_1, A_2, \ldots, A_n) =_{\mathrm{def}} \{x : f(c_{A_1}, c_{A_2}, \ldots, c_{A_n}) = 1\}.$$

For an n-ary boolean function f and a classes $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_n$, of languages we define the class $f(\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_n)$ by

$$f(\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_n) =_{\mathrm{def}} \{f(A_1, A_2, \ldots, A_n) : A_i \in \mathcal{K}_i \text{ for } i = 1, 2, \ldots, n\}.$$

In such a way the function f maps classes of languages to classes of languages and is therefore called a *boolean operator*.

Defining the boolean functions non, et, vel, and aut by $\mathrm{non}(x) = 1 \Leftrightarrow_{\mathrm{def}} x = 0$, $\mathrm{et}(x, y) = 1 \Leftrightarrow_{\mathrm{def}} x = y = 1$, $\mathrm{vel}(x, y) = 0 \Leftrightarrow_{\mathrm{def}} x = y = 0$, and $\mathrm{aut}(x, y) = 0 \Leftrightarrow_{\mathrm{def}} x = y$, resp., we obtain $\mathrm{non}(\mathcal{K}) = \text{co-}\mathcal{K}$, $\mathrm{et}(\mathcal{K}, \mathcal{M}) = \mathcal{K} \wedge \mathcal{M}$, $\mathrm{vel}(\mathcal{K}, \mathcal{M}) = \mathcal{K} \vee \mathcal{M}$, and $\mathrm{aut}(\mathcal{K}, \mathcal{M}) = \mathcal{K} \oplus \mathcal{M}$.

The effect of a boolean operators to the classes $\mathrm{Leaf}^{\mathrm{p}}_{\tau}(L_1|L'_1)$ can be modelled directly by the leaf languages.

Lemma 10.1 [Mu05] *Let $\tau \in \{\mathrm{b}, \mathrm{u}, \mathrm{e}\}$, let f be an n-ary boolean function, and let (L_i^1, L_i^0) be a pair of leaf languages for $i = 1, 2, \ldots, n$ where the alphabets $\alpha(L_i^1 \cup L_i^0)$ are pairwise disjoint. Then*

$$f(\mathrm{Leaf}_\tau^{\mathrm{p}}(L_1^1|L_1^0), \mathrm{Leaf}_\tau^{\mathrm{p}}(L_2^1|L_2^0), \ldots, \mathrm{Leaf}_\tau^{\mathrm{p}}(L_n^1|L_n^0)) = \mathrm{Leaf}_\tau^{\mathrm{p}}(L_f^1|L_f^0)$$

where $L_f^j =_{\mathrm{def}} \bigcup_{f(a_1,\ldots,a_n)=j} L_1^{a_1} L_2^{a_2} \ldots L_n^{a_n}$ for $j = 1, 2$.

Again we obtain immediately

Theorem 10.2 [Mu05] *If the class $\mathcal{K}$ is closed under union and concatenation then the families $\mathcal{L}_{\mathrm{b}}^{\mathrm{p}}(\mathcal{K})$, $\mathcal{L}_{\mathrm{u}}^{\mathrm{p}}(\mathcal{K})$, and $\mathcal{L}_{\mathrm{e}}^{\mathrm{p}}(\mathcal{K})$ are closed under all boolean operators.*

Corollary 10.3 *The families $\mathcal{L}_{\mathrm{b}}^{\mathrm{p}}$, $\mathcal{L}_{\mathrm{u}}^{\mathrm{p}}$, and $\mathcal{L}_{\mathrm{e}}^{\mathrm{p}}$ are closed under all boolean operators.*

In the complementary case there is a similar result for the ε-model. For a language L let $L^1 =_{\mathrm{def}} L$ and $L^0 =_{\mathrm{def}} \overline{L}$. For alphabets Σ and Δ such that $\Delta \subset \Sigma$ we define the homomorphism $\varphi_{\Sigma,\Delta} : \Sigma^* \to \Delta^*$ by $\varphi_{\Sigma,\Delta}(a) =_{\mathrm{def}} a$ for $a \in \Delta$ and $\varphi_{\Sigma,\Delta}(a) =_{\mathrm{def}} \varepsilon$ for $a \in \Sigma \setminus \Delta$.

Lemma 10.4 [Mu05] *Let f be an n-ary boolean function, let L_i be a leaf languages for $i = 1, 2, \ldots, n$ where the $\alpha(L_i)$ are pairwise disjoint, and let $\Sigma =_{\mathrm{def}} \bigcup_{i=1}^n \alpha(L_i)$. Then*

$$f(\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1), \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_2), \ldots, \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_n)) = \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_f)$$

where $L_f =_{\mathrm{def}} \bigcup_{f(a_1,\ldots,a_n)=1} (\bigcap_{i=1}^n \varphi_{\Sigma,\alpha(L_i)}^{-1}(L_i^{a_i}))$.

Theorem 10.5 [Mu05] *If the class $\mathcal{K}$ is closed under union, complementation, and concatenation then the family $\mathcal{C}_{\mathrm{e}}^{\mathrm{p}}(\mathcal{K})$ is closed under all boolean operators.*

Corollary 10.6 *The family $\mathcal{C}_{\mathrm{e}}^{\mathrm{p}}$ is closed under all boolean operators.*

Using the characterization of $\mathcal{C}_{\mathrm{b}}^{\mathrm{p}}$ in Theorem 9.2 we obtain

Theorem 10.7 [Mu05] *The family $\mathcal{C}_{\mathrm{b}}^{\mathrm{p}}$ is closed under all boolean operators.*

From Corollary 10.3, Corollary 10.6, and Theorem 10.7 we obtain

Corollary 10.8

1. *Let $\tau \in \{\mathrm{b}, \mathrm{u}, \mathrm{e}\}$. For every pair (L_1, L_1') of leaf languages and every class $\mathcal{K}$ of the boolean hierarchy over $\mathrm{Leaf}_\tau^{\mathrm{p}}(L_1|L_1')$ there exists a pair (L_2, L_2') of leaf languages such that $\mathrm{Leaf}_\tau^{\mathrm{p}}(L_2|L_2') = \mathcal{K}$.*
2. *Let $\tau \in \{\mathrm{b}, \mathrm{e}\}$. For every leaf language L_1 and every class $\mathcal{K}$ of the boolean hierarchy over $\mathrm{Leaf}_\tau^{\mathrm{p}}(L_1)$ there exists a leaf language L_2 such that $\mathrm{Leaf}_\tau^{\mathrm{p}}(L_2) = \mathcal{K}$.*

For the family $\mathcal{C}_{\mathrm{u}}^{\mathrm{p}}$ we only know

Proposition 10.9 *The family $\mathcal{C}_{\mathrm{u}}^{\mathrm{p}}$ is closed under* co-.

Problem 10.10 *The families $\mathcal{L}_{\mathrm{b}}^{\mathrm{p}}$, $\mathcal{L}_{\mathrm{u}}^{\mathrm{p}}$, $\mathcal{L}_{\mathrm{e}}^{\mathrm{p}}$, $\mathcal{C}_{\mathrm{e}}^{\mathrm{p}}$, and $\mathcal{C}_{\mathrm{b}}^{\mathrm{p}}$ are closed under all boolean operators, but what about the class $\mathcal{C}_{\mathrm{u}}^{\mathrm{p}}$? For example, is $\mathcal{C}_{\mathrm{u}}^{\mathrm{p}}$ closed under $\wedge$?*

10.2 Intersection

Lemma 10.11 [BCS91] *Let $\tau \in \{\mathrm{b}, \mathrm{u}, \mathrm{e}\}$ and let (L_1, L_1'), (L_2, L_2') be pairs of leaf languages such that $\alpha(L_1 \cup L_1') \cap \alpha(L_2 \cup L_2') = \emptyset$. Then*

$$\mathrm{Leaf}^{\mathrm{p}}(L_1|L_1') \cap \mathrm{Leaf}^{\mathrm{p}}(L_2|L_2') = \mathrm{Leaf}^{\mathrm{p}}(L_1 L_2|L_1' L_2').$$

Theorem 10.12 *For $\tau \in \{\mathrm{b}, \mathrm{u}, \mathrm{e}\}$, if the class $\mathcal{K}$ is closed under concatenation then the family $\mathcal{L}^{\mathrm{p}}_{\tau}(\mathcal{K})$ is closed under intersection.*

Corollary 10.13 *The families $\mathcal{L}^{\mathrm{p}}_{\mathrm{b}}$, $\mathcal{L}^{\mathrm{p}}_{\mathrm{u}}$, and $\mathcal{L}^{\mathrm{p}}_{\mathrm{e}}$ are closed under intersection.*

Since $\mathrm{NP}, \text{co-NP} \in \mathcal{C}^{\mathrm{p}}_{\mathrm{e}}$ and all classes in $\mathcal{C}^{\mathrm{p}}_{\mathrm{b}}$ have $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets, we obtain

Theorem 10.14 *If $\mathrm{NP} \cap \text{co-NP}$ does not have $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets then the families $\mathcal{C}^{\mathrm{p}}_{\mathrm{b}}$, $\mathcal{C}^{\mathrm{p}}_{\mathrm{u}}$, and $\mathcal{C}^{\mathrm{p}}_{\mathrm{e}}$ are not closed under intersection.*

Note that there is some evidence that $\mathrm{NP} \cap \text{co-NP}$ does not have $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets.

10.3 Union

Lemma 10.15 [BCS91] *If $\mathcal{K}_1, \mathcal{K}_2 \in \mathcal{C}^{\mathrm{p}}_{\mathrm{b}}$ such that $\mathcal{K}_1 \not\subseteq \mathcal{K}_2$ and $\mathcal{K}_2 \not\subseteq \mathcal{K}_1$ then $\mathcal{K}_1 \cup \mathcal{K}_2 \notin \mathcal{L}^{\mathrm{p}}_{\mathrm{b}}$.*

Let RE be the class of recursive enumerable languages. Since $\mathrm{RE}, \text{co-RE} \in \mathcal{C}^{\mathrm{p}}_{\mathrm{e}}$ we obtain

Theorem 10.16 *The families $\mathcal{C}^{\mathrm{p}}_{\mathrm{b}}$, $\mathcal{C}^{\mathrm{p}}_{\mathrm{u}}$, $\mathcal{C}^{\mathrm{p}}_{\mathrm{e}}$, $\mathcal{L}^{\mathrm{p}}_{\mathrm{b}}$, $\mathcal{L}^{\mathrm{p}}_{\mathrm{u}}$, and $\mathcal{L}^{\mathrm{p}}_{\mathrm{e}}$ are not closed under union.*

A further consequence of Lemma 10.15 is

Theorem 10.17 *There holds $\mathrm{NP} \cup \text{co-NP} \in \mathcal{L}^{\mathrm{p}}_{\mathrm{b}}$ if and only if $\mathrm{NP} = \text{co-NP}$.*

10.4 The ∇-Operator

Lemma 10.18 *Let $\tau \in \{\mathrm{b}, \mathrm{u}, \mathrm{e}\}$ and let (L_1, L_1'), (L_2, L_2') be pairs of leaf languages such that $\alpha(L_1 \cup L_1') \cap \alpha(L_2 \cup L_2') = \emptyset$. Then*

1. $\mathrm{Leaf}^{\mathrm{p}}(L_1|L_1') \nabla \mathrm{Leaf}^{\mathrm{p}}(L_2|L_2') = \mathrm{Leaf}^{\mathrm{p}}(L_1 \cup L_2|L_1' \cup L_2')$
2. $\mathrm{Leaf}^{\mathrm{p}}(L_1) \nabla \mathrm{Leaf}^{\mathrm{p}}(L_2) = \mathrm{Leaf}^{\mathrm{p}}(L_1 \cup L_2)$. $\square$

Theorem 10.19 *If the class $\mathcal{K}$ is closed under union or marked union then the families $\mathcal{C}^{\mathrm{p}}_{\mathrm{b}}(\mathcal{K})$, $\mathcal{C}^{\mathrm{p}}_{\mathrm{u}}(\mathcal{K})$, $\mathcal{L}^{\mathrm{p}}_{\mathrm{b}}(\mathcal{K})$, and $\mathcal{L}^{\mathrm{p}}_{\mathrm{u}}(\mathcal{K})$ are closed under ∇.*

Corollary 10.20 *The families $\mathcal{C}^{\mathrm{p}}_{\mathrm{b}}$, $\mathcal{C}^{\mathrm{p}}_{\mathrm{u}}$, $\mathcal{L}^{\mathrm{p}}_{\mathrm{b}}$, and $\mathcal{L}^{\mathrm{p}}_{\mathrm{u}}$ are closed under ∇.*

Lemma 10.21 [Mu05] *If $\mathrm{NP} \neq \text{co-NP}$ and $\mathrm{UP} \wedge \text{co-UP} \not\subseteq \mathrm{P}^{\mathrm{NP}}[1]$ then $\mathrm{NP} \nabla \text{co-NP} \notin \mathcal{C}^{\mathrm{p}}_{\mathrm{e}}$.*

Theorem 10.22 [Mu05] *If $\mathrm{NP} \neq \text{co-NP}$ and $\mathrm{UP} \wedge \text{co-UP} \not\subseteq \mathrm{P}^{\mathrm{NP}}[1]$ then $\mathcal{C}^{\mathrm{p}}_{\mathrm{e}}$ is not closed under ∇.*

In [Mu05] it is shown that there exists an oracle A such that $\mathrm{UP}^A \wedge \text{co-UP}^A \not\subseteq (\mathrm{P}^{\mathrm{NP}}[1])^A$.

Problem 10.23 *Is $\mathcal{L}^{\mathrm{p}}_{\mathrm{e}}$ closed under ∇?*

10.5 The Turing Operator

We consider the *Turing operator* $\mathrm{P}^{\bullet}$ which is defined by $\mathrm{P}^{\bullet}(\mathcal{K}) =_{\mathrm{def}} \mathrm{P}^{\mathcal{K}}$. Analogously, $\mathrm{P}^{\bullet}[1](\mathcal{K}) =_{\mathrm{def}} \mathrm{P}^{\mathcal{K}}[1]$.

The operator $\mathrm{P}^{\bullet}[1](\mathcal{K})$ special case of the operator ∇.

Lemma 10.24 *If the class $\mathcal{K}$ of languages is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$, marked union, intersection with* P*-sets, and union with* P*-sets then $\mathcal{K}\nabla\text{co-}\mathcal{K} = \mathrm{P}^{\mathcal{K}}[1]$.*

From Theorem 10.19 we get

Theorem 10.25 *If the class $\mathcal{K}$ is closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$, marked union, intersection with* P*-sets, and union with* P*-sets then the families $\mathcal{C}_{\mathrm{b}}^{\mathrm{p}}(\mathcal{K})$, $\mathcal{C}_{\mathrm{u}}^{\mathrm{p}}(\mathcal{K})$, $\mathcal{L}_{\mathrm{b}}^{\mathrm{p}}(\mathcal{K})$, and $\mathcal{L}_{\mathrm{u}}^{\mathrm{p}}(\mathcal{K})$ are closed under $\mathrm{P}^{\bullet}[1]$.*

Corollary 10.26 *The families $\mathcal{C}_{\mathrm{b}}^{\mathrm{p}}$, $\mathcal{C}_{\mathrm{u}}^{\mathrm{p}}$, $\mathcal{L}_{\mathrm{b}}^{\mathrm{p}}$, and $\mathcal{L}_{\mathrm{u}}^{\mathrm{p}}$ are closed under $\mathrm{P}^{\bullet}[1]$.*

By Lemma 10.21 and Lemma 10.24 we have

Theorem 10.27 [Mu05] *If $\mathrm{NP} \neq \text{co-NP}$ and $\mathrm{UP} \wedge \text{co-UP} \not\subseteq \mathrm{P}^{\mathrm{NP}}[1]$ then $\mathcal{C}_{\mathrm{e}}^{\mathrm{p}}$ is not closed under $\mathrm{P}^{\bullet}[1]$.*

Problem 10.28 *Is $\mathcal{L}_{\mathrm{e}}^{\mathrm{p}}$ closed under $\mathrm{P}^{\bullet}[1]$?*

Finally we consider the full Turing operator.

Theorem 10.29
1. [BCS91] *The families $\mathcal{L}_{\mathrm{b}}^{\mathrm{p}}$ and $\mathcal{L}_{\mathrm{u}}^{\mathrm{p}}$ are closed under $\mathrm{P}^{\bullet}$.*
2. [Mu05] *The familiy $\mathcal{C}_{\mathrm{b}}^{\mathrm{p}}$ is closed under $\mathrm{P}^{\bullet}$.*

Theorem 10.30 [Tr02] *If $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_1')$ is closed under positive polynomial time tt-reducibility then there exists a pair (L_3, L_3') such that $\mathrm{P}^{\exists\cdot\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_1')} = \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_3|L_3')$. If $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_1')$ is complementary then $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_3|L_3')$ is complementary too.*

10.6 Quantifier Operators

In this subsection we will study the closure properties of our leaf language families w.r.t. the operators $\exists$, $\forall$, $\mathrm{Mod}_k (k \geq 2)$, C, and $\Subset$ which we will call *quantifier operators.*

Let L be a language, and let a be a symbol not in $\alpha(L)$. We define the languages

$$\begin{array}{rcl}
\exists L & =_{\mathrm{def}} & \{a^1 w_1 a^2 w_2 a^3 w_3 \ldots a^r w_r a^{r+1} : \exists i (1 \leq i \leq r \wedge w_i \in L)\}, \\
\forall L & =_{\mathrm{def}} & \{a^1 w_1 a^2 w_2 a^3 w_3 \ldots a^r w_r a^{r+1} : \forall i (1 \leq i \leq r \rightarrow w_i \in L)\}, \\
\mathrm{Mod}_k^1 L & =_{\mathrm{def}} & \{a^1 w_1 a^2 w_2 a^3 w_3 \ldots a^r w_r a^{r+1} : \#\{i : 1 \leq i \leq r \wedge w_i \in L\} \equiv 1(k)\} \text{ for } k \geq 2, \\
\mathrm{Mod}_k^0 L & =_{\mathrm{def}} & \{a^1 w_1 a^2 w_2 a^3 w_3 \ldots a^r w_r a^{r+1} : \#\{i : 1 \leq i \leq r \wedge w_i \in L\} \not\equiv 1(k)\} \text{ for } k \geq 2, \\
\mathrm{C}_{\geq} L & =_{\mathrm{def}} & \{a^1 w_1 a^2 w_2 a^3 w_3 \ldots a^r w_r a^{r+1} : \#\{i : 1 \leq i \leq r \wedge w_i \in L\} \geq \frac{r}{2}\}, \\
\mathrm{C}_{<} L & =_{\mathrm{def}} & \{a^1 w_1 a^2 w_2 a^3 w_3 \ldots a^r w_r a^{r+1} : \#\{i : 1 \leq i \leq r \wedge w_i \in L\} < \frac{r}{2}\}, \\
\mathrm{C}_{=} L & =_{\mathrm{def}} & \{a^1 w_1 a^2 w_2 a^3 w_3 \ldots a^r w_r a^{r+1} : \#\{i : 1 \leq i \leq r \wedge w_i \in L\} = \frac{r}{2}\}, \text{ and} \\
\mathrm{C}_{\neq} L & =_{\mathrm{def}} & \{a^1 w_1 a^2 w_2 a^3 w_3 \ldots a^r w_r a^{r+1} : \#\{i : 1 \leq i \leq r \wedge w_i \in L\} \neq \frac{r}{2}\}
\end{array}$$

Lemma 10.31 *Let $\tau \in \{\mathrm{b}, \mathrm{u}, \mathrm{e}\}$, and let (L, L') be a pair of leaf languages.*

1. $\exists\cdot\mathrm{Leaf}_{\tau}^{\mathrm{p}}(L|L') = \mathrm{Leaf}_{\tau}^{\mathrm{p}}(\exists L|\forall L')$

2. $\forall\cdot\mathrm{Leaf}^{\mathrm{p}}_{\tau}(L|L') = \mathrm{Leaf}^{\mathrm{p}}_{\tau}(\forall L|\exists L')$
3. $\mathrm{Mod}_k\cdot\mathrm{Leaf}^{\mathrm{p}}_{\tau}(L|L') = \mathrm{Leaf}^{\mathrm{p}}_{\tau}(\mathrm{Mod}^1_k L|\mathrm{Mod}^0_k L')$ *for* $(k \geq 2)$.
4. $\mathrm{C}\cdot\mathrm{Leaf}^{\mathrm{p}}_{\tau}(L|L') = \mathrm{Leaf}^{\mathrm{p}}_{\tau}(\mathrm{C}_{\geq} L|\mathrm{C}_{<} L')$
5. $\Subset\cdot\mathrm{Leaf}^{\mathrm{p}}_{\tau}(L|L') = \mathrm{Leaf}^{\mathrm{p}}_{\tau}(\mathrm{C}_{=} L|\mathrm{C}_{\neq} L')$

Theorem 10.32 *The families* $\mathcal{L}^{\mathrm{p}}_{\mathrm{b}}$, $\mathcal{L}^{\mathrm{p}}_{\mathrm{u}}$, $\mathcal{L}^{\mathrm{p}}_{\mathrm{e}}$, $\mathcal{L}^{\mathrm{p}}_{\mathrm{b}}(\mathrm{REC})$, $\mathcal{L}^{\mathrm{p}}_{\mathrm{u}}(\mathrm{REC})$, *and* $\mathcal{L}^{\mathrm{p}}_{\mathrm{e}}(\mathrm{REC})$ *are closed under* $\exists$, $\forall$, $\mathrm{Mod}_k (k \geq 2)$, C, *and* $\Subset$. □

From Theorem 10.32 one obtains the following.

Corollary 10.33
1. *Let* $\tau \in \{\mathrm{b}, \mathrm{u}, \mathrm{e}\}$. *For every pair* (L_1, L'_1) *of leaf languages and every class* $\mathcal{K}$ *of the modulo hierarchy over* $\mathrm{Leaf}^{\mathrm{p}}_{\tau}(L_1|L'_1)$ *there exists a pair* (L_2, L'_2) *of leaf languages such that* $\mathrm{Leaf}^{\mathrm{p}}_{\tau}(L_2|L'_2) = \mathcal{K}$.
2. *Let* $\tau \in \{\mathrm{b}, \mathrm{u}, \mathrm{e}\}$. *For every pair* (L_1, L'_1) *of leaf languages and every class* $\mathcal{K}$ *of the counting hierarchy over* $\mathrm{Leaf}^{\mathrm{p}}_{\tau}(L_1|L'_1)$ *there exists a pair* (L_2, L'_2) *of leaf languages such that* $\mathrm{Leaf}^{\mathrm{p}}_{\tau}(L_2|L'_2) = \mathcal{K}$.
3. *For every leaf language* L_1 *and every class* $\mathcal{K}$ *of the counting hierarchy over* $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(L_1)$ *there exists a leaf language* L_2 *such that* $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(L_2) = \mathcal{K}$.

11 Regular Leaf Languages

Which classes can be defined by regular leaf languages? Which is the largest such class? The latter questions is answered by the following theorem. Let A_5 be the group of even permutations of $(1, 2, 3, 4, 5)$, and let a_1 be the identical permutation of $(1, 2, 3, 4, 5)$. Obviously, the language

$M_5 =_{\mathrm{def}} \{p_1 p_2 \ldots p_m : m \geq 1 \wedge p_1, p_2, \ldots, p_m \in \mathrm{A}_5 \wedge p_1 \cdot p_2 \cdot \ldots \cdot p_m = a_1\}$
is regular. Let REG denote the class of all regular languages.

Theorem 11.1 [HLSVW93] $\mathrm{Leaf}^{\mathrm{p}}(M_5) = \mathrm{Leaf}^{\mathrm{p}}(\mathrm{REG}) = \mathrm{PSPACE}$.

Which regular leaf languages describe the class PSPACE? This can be answered by looking at certain algebraic properties of their syntactic monoids. (For a survey on the connections between languages, monoids, and finite model theory see [Pi96].) Notice that it is a widely believed conjecture that MOD-PH is a proper subclass of PSPACE. Hence Statement 1 and Statement 2 of the following theorem have a dichotomic character.

Theorem 11.2 [HLSVW93]

1. *If the syntactic monoid of* L *is not solvable then* $\mathrm{Leaf}^{\mathrm{p}}(L) = \mathrm{PSPACE}$.
2. *If the syntactic monoid of* L *is solvable then* $\mathrm{Leaf}^{\mathrm{p}}(L) \subseteq \mathrm{MOD\text{-}PH}$.
3. *If the syntactic monoid of* L *is aperiodic then* $\mathrm{Leaf}^{\mathrm{p}}(L) \subseteq \mathrm{PH}$.

There is an interesting correspondence between classes of languages with an aperiodic syntactic monoid and the classes of the polynomial time hierarchy. The syntactic monoid of L is aperiodic if and only if L is starfree (Schützenberger 1965). The class SF of starfree languages is defined as the smallest class of languages which containes all finite languages and which is closed under union, complementation, and concatenation. Two so-called *dot-depth hierarchies* within the class of starfree languages have been studied intensively. Roughly speaking, the levels of these hierarchies are defined by the number of alternations between concatenations and boolean set operations in the generation of a starfree language. For a class $\mathcal{K}$, let $\mathrm{BC}(\mathcal{K})$ be the *boolean closure* of $\mathcal{K}$, i.e., the closure of $\mathcal{K}$ under union and complementation, and let $\mathrm{Pol}(\mathcal{K})$ be the closure of $\mathcal{K}$ under union and concatenation. The *Cohen-Brzozowski hierarchy* (Cohen and Brzozowski 1971) consists of the classes $\mathcal{B}_{k/2}$ for $k \geq 0$ and the *Straubing-Thérien hierarchy* (Straubing and Thérien 1981) consists of the classes $\mathcal{L}_{k/2}$ for $k \geq 1$. These classes are defined by

$$
\begin{array}{llll}
\mathcal{B}_{k/2} & =_{\mathrm{def}} \bigcup_{A \text{ finite alphabet}} \mathcal{B}^A_{k/2} & \mathcal{L}_{k/2} & =_{\mathrm{def}} \bigcup_{A \text{ finite alphabet}} \mathcal{L}^A_{k/2} \\
\mathcal{B}^A_0 & =_{\mathrm{def}} \text{class of finite unions of languages of} & & \\
 & \quad \text{the form } \{w\} \text{ and } wA^*v \text{ where } w, v \in A^* & & \\
\mathcal{B}^A_{1/2} & =_{\mathrm{def}} \mathrm{Pol}(\{\{w\} : w \in A^*\} \cup \{A^*\}) & \mathcal{L}^A_{1/2} & =_{\mathrm{def}} \mathrm{Pol}(\{A^*aA^* : a \in A\}) \\
\mathcal{B}^A_k & =_{\mathrm{def}} \mathrm{BC}(\mathcal{B}_{k-1/2}) & \mathcal{L}^A_k & =_{\mathrm{def}} \mathrm{BC}(\mathcal{L}_{k-1/2}) \\
\mathcal{B}^A_{k+1/2} & =_{\mathrm{def}} \mathrm{Pol}(\mathcal{B}_k) & \mathcal{L}^A_{k+1/2} & =_{\mathrm{def}} \mathrm{Pol}(\mathcal{L}_k)
\end{array}
$$

It is well known that $\mathcal{L}_{k-1/2} \subset \mathcal{B}_{k-1/2} \subset \mathcal{L}_{k+1/2}$ and $\mathcal{L}_k \subset \mathcal{B}_k \subset \mathcal{L}_{k+1}$ for $k \geq 1$, and hence $\bigcup_{k\geq 1} \mathcal{L}_{k/2} = \bigcup_{k \geq 1} \mathcal{B}_{k/2} = \mathrm{SF}$. Define $A_k =_{\mathrm{def}} \{0, 1, \ldots, k\}$ for $k \geq 1$ and

$L_1 =_{\mathrm{def}} 0^*1(0 \cup 1)^*$,
$L_k =_{\mathrm{def}} A^*_k k(A^*_{k-1} \setminus L_{k-1})kA^*_k$ for $k \geq 2$,
$D_2 =_{\mathrm{def}} (0 \cup 1 \cup 2)10^*$, and
$D_k =_{\mathrm{def}} A^*_k\{k-1, k\}(A^*_{k-2} \setminus L_{k-1})k(L_{k-2}\{k-1, k\})^*$ for $k \geq 3$.

Proposition 11.3 *1. $L_k \in \mathcal{L}_{k-1/2}$ for $k \geq 1$.*
2. $D_k \in \mathcal{L}_{k-1/2} \cap \text{co-}\mathcal{L}_{k-1/2}$ for $k \geq 2$. □

Proposition 11.3 is optimal as shown in Corollary 11.5.

In the leaf language concept the levels of the dot-depth hierarchies correspond exactly to the levels of the polynomial time hierarchy.

Theorem 11.4 *1.* [HLSVW93] $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{SF}) = \mathrm{PH}$.
2. [HLSVW93] $\mathrm{Leaf}^{\mathrm{p}}(\mathcal{B}_0) = \mathrm{P}$.
3. [BV98] $\mathrm{Leaf}^{\mathrm{p}}(L_k) = \mathrm{Leaf}^{\mathrm{p}}(\mathcal{L}_{k-1/2}) = \mathrm{Leaf}^{\mathrm{p}}(\mathcal{B}_{k-1/2}) = \Sigma^{\mathrm{p}}_k$ *for $k \geq 1$.*
4. [HLSVW93] $\mathrm{Leaf}^{\mathrm{p}}(\mathcal{L}_k) = \mathrm{Leaf}^{\mathrm{p}}(\mathcal{B}_k) = \mathrm{BC}(\Sigma^{\mathrm{p}}_k)$ *for $k \geq 1$.*
5. [BSS99, BLSTT04, Tr02]
$\mathrm{Leaf}^{\mathrm{p}}(D_{k+1}) = \mathrm{Leaf}^{\mathrm{p}}(\mathcal{L}_{k+1/2} \cap \text{co-}\mathcal{L}_{k+1/2}) = \mathrm{Leaf}^{\mathrm{p}}(\mathcal{B}_{k+1/2} \cap \text{co-}\mathcal{B}_{k+1/2}) = \Delta^{\mathrm{p}}_{k+1}$ *for $k \geq 0$.*

These statements remain valid relative to every oracle.

In [Sch00] it is shown that $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathrm{RTL}) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathrm{RPTL}) = \Delta^{\mathrm{p}}_2$, where RTL and RPTL are defined by restricted temporal logic, and there holds $\mathcal{B}_1 \subset \mathrm{RTL} \cap \mathrm{RPTL} \subset \mathrm{RTL} \subset \mathcal{B}_{3/2} \cap \text{co-}\mathcal{B}_{3/2}$ and $\mathcal{B}_1 \subset \mathrm{RTL} \cap \mathrm{RPTL} \subset \mathrm{RPTL} \subset \mathcal{B}_{3/2} \cap \text{co-}\mathcal{B}_{3/2}$. □

Corollary 11.5 *1.* [SW04] $L_k \in \mathcal{L}_{k-1/2} \setminus \text{co-}\mathcal{B}_{k-1/2}$ *for $k \geq 1$.*
2. $D_k \in (\mathcal{L}_{k-1/2} \cap \text{co-}\mathcal{L}_{k-1/2}) \setminus \mathcal{B}_{k-1}$ for $k \geq 1$. □

The correspondence between the dot-depth hierarchies on the one side and the polynomial time hierarchy on the other side is even stronger than indicated by the previous theorem. It applies also to the boolean hierachies over the various levels of the polynomial time hierarchy.

Theorem 11.6 *1.* [SW98, BKS99] $\mathrm{Leaf}^{\mathrm{p}}(\mathcal{L}_{1/2}(m)) = \mathrm{Leaf}^{\mathrm{p}}(\mathcal{B}_{1/2}(m)) = \mathrm{NP}(m)$ *for $m \geq 1$.*
2. [Se01] $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathcal{B}_{k-1/2}(m)) = \Sigma^{\mathrm{p}}_k(m)$ *for $k, m \geq 1$.*

It is also shown in [Se01] that some refinements of the $\mathcal{B}_{m-1/2}(k)$ hierarchy can be related to the corresponding refinements of the $\Sigma_m^{\mathrm{p}}(k)$ hierarchy in the same way as in Theorem 11.6.

We have seen that P is the minimal class which can be defined by leaf languages. We do also know the "next larger" classes which can be defined by regular leaf languages. Let us start with the unbalanced case.

Theorem 11.7 *Let L be a regular language.*

1. [Bo94a] $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \mathrm{P}$ *or* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{NP}$ *or* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \text{co-NP}$ *or* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{Mod}_p\mathrm{P}$ *for some prime number p.*
2. [Bo94a] *If $L \in \mathcal{B}_0$ then* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \mathrm{P}$.
3. [BKS99] *If $L \in \mathcal{B}_{1/2} \setminus \mathcal{B}_0$ then* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \mathrm{NP}$.
4. [BKS99] *If $L \in \text{co-}\mathcal{B}_{1/2} \setminus \mathcal{B}_0$ then* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \text{co-NP}$.
5. [BKS99] *If $L \in \mathrm{SF} \setminus \mathcal{B}_{1/2}$ then* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \forall\mathrm{P} = \text{co-NP}$ *or* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \text{co-}\exists!\cdot\mathrm{P}$.
6. [BLSTT04] *If $L \in (\mathcal{L}_{3/2} \cap \text{co-}\mathcal{L}_{3/2}) \setminus \mathcal{L}_1$ then* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \Delta_2^{\mathrm{p}}$.
7. [Sch00] *If $L \in \mathrm{SF} \setminus \mathcal{B}_{3/2}$ then* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \forall\cdot\mathrm{UP}$ *or* $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \text{co-}\exists!\cdot\mathrm{P}$.

Let us hint at the similar structure of Statement 5 and Statement 7 which are proved using *forbidden pattern* characterizations of the classes $\mathcal{B}_{1/2}$ and $\mathcal{B}_{3/2}$. Similar results for higher classes $\mathcal{B}_{k/2}$ could be obtained from forbidden pattern characterizations of these classes which are unfortunately not known so far.

For the proper subclasses RTL and RPTL of $\mathcal{B}_{3/2} \cap \text{co-}\mathcal{B}_{3/2}$ (see the remark after Theorem 11.4) the following is shown in [Sch00]. If $L \notin \mathrm{RTL} \cap \mathrm{RPTL}$ then $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \Delta_2^{\mathrm{p}}$ or $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}((0^*10^*2)^*0^*)$ or $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \text{co-}\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}((0^*10^*2)^*0^*)$ or $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) \supseteq \mathrm{Mod}_p\mathrm{P}$ for some prime number p.

Under the assumption that the polynomial time hierarchy does not collapse one can conclude from Theorem 11.7 characterizations of those regular leaf languages which define the classes P, NP and co-NP, resp.

Corollary 11.8 [BKS99] *Assume that the polynomial time hierarchy does not collapse. Let L be a regular language.*

1. $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \mathrm{P} \Leftrightarrow L \in \mathcal{B}_0$.
2. $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \mathrm{NP} \Leftrightarrow L \in \mathcal{B}_{1/2} \setminus \mathcal{B}_0$.
3. $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L) = \text{co-NP} \Leftrightarrow L \in \text{co-}\mathcal{B}_{1/2} \setminus \mathcal{B}_0$.

Now we consider the balanced case. For a class $\mathcal{K}$ of languages, let $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{K})$ be the class of languages which are $\leq_{\mathrm{m}}^{\mathrm{plt}}$-reducible (see Section 13) to a language in $\mathcal{K}$.

Theorem 11.9 [Gl04] *Let L be a regular language.*

1. $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) = \mathrm{P}$ *or* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) \supseteq \mathrm{NP}$ *or* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) \supseteq \text{co-NP}$ *or* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) \supseteq \mathrm{Mod}_p\mathrm{P}$ *for some prime number p.*
2. *If $L \in \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_0)$ then* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) = \mathrm{P}$.
3. *If $L \in \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_{1/2}) \setminus \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_0)$ then* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) = \mathrm{NP}$.
4. *If $L \in \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\text{co-}\mathcal{B}_{1/2}) \setminus \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_0)$ then* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) = \text{co-NP}$.
5. *If $L \in \mathrm{SF} \setminus \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_{1/2})$ then* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) \supseteq \text{co-NP}$ *or* $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L) \supseteq \text{co-1-NP}$.

Corollary 11.10 *Assume that the polynomial time hierarchy does not collapse. Let L be a regular language.*

1. $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L) = \mathrm{P} \Leftrightarrow L \in \mathcal{R}^{\mathrm{plt}}_{\mathrm{m}}(\mathcal{B}_0)$.
2. $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L) = \mathrm{NP} \Leftrightarrow L \in \mathcal{R}^{\mathrm{plt}}_{\mathrm{m}}(\mathcal{B}_{1/2}) \smallsetminus \mathcal{R}^{\mathrm{plt}}_{\mathrm{m}}(\mathcal{B}_0)$.
3. $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L) = \text{co-NP} \Leftrightarrow L \in \mathcal{R}^{\mathrm{plt}}_{\mathrm{m}}(\text{co-}\mathcal{B}_{1/2}) \smallsetminus \mathcal{R}^{\mathrm{plt}}_{\mathrm{m}}(\mathcal{B}_0)$.

Problem 11.11 *From Theorem 5.1, Corollary 11.8, and Corollary 11.10 we obtain for every regular paddable language L that $L \in \mathcal{R}^{\mathrm{plt}}_{\mathrm{m}}(\mathcal{B}_{k/2})$ implies $L \in \mathcal{B}_{k/2}$ for $k = 0, 1$. Is this true for every $k \geq 2$?*

In [Ga98] the classes $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(L)$ and $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L)$ was investigated for regular languages L which can be accepted by finite deterministic automata with at most three states.

12 Other Leaf Languages

So far we have looked at classes which are defined by regular leaf languages. What happens if we choose leaf languages from other classes, in particular from given complexity classes? Since the leaf word $\beta_M(x)$ of a machine M can be exponentially longer than the input word x, one can expext a maximally exponential blowup of the complexity when going from a leaf language to the languages accepted with this leaf language. Indeed this exponential blowup does really happen. In particular, we have

Theorem 12.1 [HLSVW93] *Let $s \geq \log$ and $t \geq \mathrm{id}$.*

1. $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{DSPACE}(s)) = \mathrm{DSPACE}(s \circ 2^{\mathrm{Pol}})$.
2. $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{NSPACE}(s)) = \mathrm{NSPACE}(s \circ 2^{\mathrm{Pol}})$.
3. $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{DTIME}(t)) = \mathrm{DTIME}(\mathrm{Pol} \circ t \circ 2^{\mathrm{Pol}})$.
4. $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{NTIME}(t)) = \mathrm{NTIME}(\mathrm{Pol} \circ t \circ 2^{\mathrm{Pol}})$.

Corollary 12.2
1. $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{PSPACE}) = \mathrm{DSPACE}(2^{\mathrm{Pol}})$.
2. $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{NP}) = \mathrm{NTIME}(2^{\mathrm{Pol}})$.
3. $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{P}) = \mathrm{DTIME}(2^{\mathrm{Pol}})$.
4. $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{L}) = \mathrm{Leaf}^{\mathrm{p}}(\mathrm{NL}) = \mathrm{Leaf}^{\mathrm{p}}(\mathrm{DSPACE}(\log^k)) = \mathrm{PSPACE}$.

Because of Theorem 11.1 and Statement 4 of the preceding theorem we have also $\mathrm{Leaf}^{\mathrm{p}}(\mathrm{CFL}) = \mathrm{Leaf}^{\mathrm{p}}(\mathrm{DCFL}) = \mathrm{Leaf}^{\mathrm{p}}(\mathrm{NC}^1) = \mathrm{PSPACE}$.

The next theorem gives another example of the phenomenon that the balanced and the unbalanced leaf language classes for the same leaf languages can differ (unless the polynomial time hierarchy collapses). Note that $\mathcal{B}_{k-1/2} \subseteq \Sigma_k$-LOGTIME for $k \geq 0$ and $\mathrm{SF} \subseteq \mathrm{AC}^0$.

Theorem 12.3 [JMT94, HVW96]

1. $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\Sigma_k\text{-LOGTIME}) = \Sigma^{\mathrm{p}}_k$ *and* $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\Sigma_k\text{-LOGTIME}) = (\Sigma^{\mathrm{p}}_k)^{\mathrm{PP}}$ *for* $k \geq 0$.
2. $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(\mathrm{AC}^0) = \mathrm{PH}$ *and* $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(\mathrm{AC}^0) = \mathrm{PH}^{\mathrm{PP}}$.

In [Vo98] classes $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L)$ were studied where L is accepted by special circuits.

13 Relativized Leaf Language Classes

In this section we will consider relativized leaf language classes, i.e., the classes of type $\mathrm{Leaf}_u^p(L_1|L_2)^X$, $\mathrm{Leaf}_b^p(L_1|L_2)^X$, $\mathrm{Leaf}_u^p(L)^X$, and $\mathrm{Leaf}_b^p(L)^X$. The use of oracles is an important tool in complexity theory. Several central problems about the inclusion between complexity classes are unresolved, e.g., the famous problem of whether $\mathrm{P} = \mathrm{NP}$. However, it was possible to prove that there are oracles X, Y such that $\mathrm{P}^X \neq \mathrm{NP}^X$ and $\mathrm{P}^Y = \mathrm{NP}^Y$. That means that we cannot prove $\mathrm{P} = \mathrm{NP}$ or $\mathrm{P} \neq \mathrm{NP}$ by using relativizable methods, i.e., methods which work in the same way when using additionally an oracle set. We will present here a relatively easy to handle criterion for the fact that there are oracles X such that $\mathrm{Leaf}_b^p(L_1|L_1')^X \not\subseteq \mathrm{Leaf}_b^p(L_2|L_2')^X$ and $\mathrm{Leaf}_u^p(L_1|L_1')^X \not\subseteq \mathrm{Leaf}_u^p(L_2|L_2')^X$.

To get a criterion for $\forall X(\mathrm{Leaf}_b^p(L_1|L_1')^X \subseteq \mathrm{Leaf}_b^p(L_2|L_2')^X)$ we define the *polylogtime reducibility* $\leq_m^{plt}$. A function $g : \Sigma_1^* \to \Sigma_2^*$ is *polylogtime computable* if there exist a $k \geq 1$ such that g is computed in $\mathcal{O}(\log^k n)$ time by a Turing machine which accesses the input as an oracle. For pairs $(L_1, L_1') \subseteq (\Sigma_1^*)^2$ and $(L_2, L_2') \subseteq (\Sigma_2^*)^2$ of leaf languages let

$(L_1, L_1') \leq_m^{plt} (L_2, L_2') \Leftrightarrow_{\mathrm{def}}$ there exist polylogtime computable functions g, h such that for all words $x \in \Sigma_1^*$,
- $x \in L_1 \to g(x,1)g(x,2)\dots g(x,h(x)) \in L_2$ and
- $x \in L_1' \to g(x,1)g(x,2)\dots g(x,h(x)) \in L_2'$.

Instead of $(L_1|\overline{L_1}) \leq_m^{plt} (L_2|\overline{L_2})$ we also write $L_1 \leq_m^{plt} L_2$. As examples we consider

Proposition 13.1 *1.* $(11)^* \leq_m^{plt} 1$.

2. $0^*10^* \not\leq_m^{plt} 1$ *and* $(0^*10^*1)^*0^* \not\leq_m^{plt} (11)^*$. □

Theorem 13.2 [BCS92, Ve93] *For all pairs* (L_1, L_1') *and* (L_2, L_2') *there holds* $(L_1, L_1') \leq_m^{plt} (L_2, L_2')$ *if and only if* $\forall X(\mathrm{Leaf}_b^p(L_1|L_1')^X \subseteq \mathrm{Leaf}_b^p(L_2|L_2')^X)$.
In particular, $L_1 \leq_m^{plt} L_2$ *if and only if* $\forall X(\mathrm{Leaf}_b^p(L_1)^X \subseteq \mathrm{Leaf}_b^p(L_2)^X)$.

Theorem 13.2 can be used very elegantly to show that there are relativized worlds in which some inclusion between complexity classes does not hold. As a consequence of Theorem 13.2 and Proposition 13.1 we obtain

Corollary 13.3 *There exists an oracle* X *such that* $\text{co-NP}^X \not\subseteq \mathrm{NP}^X$ *(and hence* $\mathrm{P}^X \neq \mathrm{NP}^X$*).* □

A language L is called *commutative* if the membership of a word x to L depends only on the number of occurences of every alphabet symbol in x and not on the order in which the symbols occur in x. For the special case of commutative, paddable leaf languages the criterion $L_1 \leq_m^{plt} L_2$ for $\forall X(\mathrm{Leaf}_b^p(L_1)^X \subseteq \mathrm{Leaf}_b^p(L_2)^X)$ can be formulated in a much more easy and applicable way. A commutative, paddable language $L \subseteq \{0, 1, \dots k\}^*$ with "padding symbol" 0 is determined by the set $V_L =_{\mathrm{def}} \{(n_1, n_2, \dots, n_k) : 1^{n_1}2^{n_2}\dots k^{n_k} \in L\}$ of *Parikh vectors*. Such a language L is said to be *of bounded significance* if there exists an $m \in \mathbb{N}$ such that $(n_1, n_2, \dots, n_k) \in V_L \leftrightarrow (\min(n_1, m), \min(n_2, m), \dots, \min(n_k, m)) \in V_L$ for all $n_1, n_2, \dots, n_k \in \mathbb{N}$. For $u, v \in \mathbb{N}^k$, we define the *multinomial coefficient* $\binom{u}{v} =_{\mathrm{def}} \prod_{i=1}^{k} \binom{u_i}{v_i}$.

Theorem 13.4 [CHVW98] *For commutative, paddable language of bounded significance* $L_1 \subseteq \{0, 1, \dots k\}^*$ *and* $L_2 \subseteq \{0, 1, \dots m\}^*$ *the following are equivalent:*

(1) $\forall X(\mathrm{Leaf}^{\mathrm{p}}(L_1)^X \subseteq \mathrm{Leaf}^{\mathrm{p}}(L_2)^X)$
(2) $L_1 \leq_{\mathrm{m}}^{\mathrm{plt}} L_2$
(3) *there exist* $p_1, \ldots, p_m : \mathbb{N}^k \to \mathbb{N}$ *which are positive linear combinations of multinomial coefficients such that* $v \in V_{L_1} \leftrightarrow (p_1(v), \ldots, p_m(v)) \in V_{L_2}$.

For a commutative, paddable language $L \subseteq \{0,1,\ldots k\}^*$ define $\mathrm{m}^+(L)$ ($\mathrm{m}^-(L)$) to be the maximum natural number r such that there exist $v_1, v_2, \ldots, v_r$ such that $v_1 \leq v_2 \leq \cdots \leq v_r$, $v_1 \in V_L$ ($v_1 \notin V_L$), and $v_i \in V_L \leftrightarrow v_{i+1} \notin V_L$ for $i = 1, 2, \ldots, k-1$. If there is not such a maximum natural number then $\mathrm{m}^+(L) =_{\mathrm{def}} \mathrm{m}^-(L) =_{\mathrm{def}} \infty$.

Theorem 13.5 [CHVW98] *Let L_1 and L_2 be commutative, paddable languages, and let L_2 be of bounded significance. If $\forall X(\mathrm{Leaf}^{\mathrm{p}}(L_1)^X \subseteq \mathrm{Leaf}^{\mathrm{p}}(L_2)^X)$ then $\mathrm{m}^+(L_1) \leq \mathrm{m}^+(L_2)$ and $\mathrm{m}^-(L_1) \leq \mathrm{m}^-(L_2)$.*

Theorem 13.6 [CHVW98] *For a commutative, paddable language L there holds $\forall X(\mathrm{Leaf}^{\mathrm{p}}(L)^X \subseteq \mathrm{NP}(k)^X)$ if and only if $\mathrm{m}^+(L) \leq k$.*

A similar criterion for $\forall X(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L)^X \supseteq \mathrm{NP}(k)^X)$ is also proved in [CHVW98].

In [GKV03] a modification of Theorem 13.2 was proved that concerns *generic* oracles. We do not give a formal definition of genericity, but, informally speaking, a generic oracle is a typical oracle in the sense that it has all the properties that can be enforced by a stage construction (the main technique to construct oracles with certain desired properties). A generic oracle which separates e.g. P and NP also separates all other complexity classes which are separated by an oracle which is built by a stage construction. For arithmetic sets L_1 and L_2 the following three statements are equivalent: (1) $L_1 \leq_{\mathrm{m}}^{\mathrm{plt}} L_2$, (2) $(\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1)^X \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_2)^X)$ for every generic oracle X, and (3) $(\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1)^X \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_2)^X)$ for some generic oracle X.

Theorem 13.2 shows that if $(L_1, L_1') \not\leq_{\mathrm{m}}^{\mathrm{plt}} (L_2, L_2')$ then there exists an oracle X such that $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_1')^X \not\subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_2|L_2')^X$. In [BCS92] an additional condition to (L_1, L_1') is given such that if $L_1 \not\leq_{\mathrm{m}}^{\mathrm{plt}} L_2$ then there exists an oracle X such that $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_2|L_2')^X$ is even strongly separated from $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_1')^X$, i.e., there exists an infinite set in $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_2|L_2')^X$ which has no infinite subset in $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_1')^X$.

In [BCS91] conditions are given under which the inclusion $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_1') \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_2|L_2')$ implies the inclusion $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_1')^X \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_2|L_2')^X$ for every sparse oracle X.

In [Ve93] a polylogtime Turing reducibility $\leq_{\mathrm{T}}^{\mathrm{plt}}$ is defined in a natural way, and it is shown that $(L_1, L_1') \leq_{\mathrm{T}}^{\mathrm{plt}} (L_2, L_2')$ if and only if $\forall X(\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_1|L_1')^X \subseteq \mathrm{P}^{\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_2|L_2')^X})$.

To get a similar criterion like the one in Theorem 13.2 for the unbalanced leaf language model, i.e., a criterion for $\forall X(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_2)^X \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1'|L_2')^X)$ a new reducibilty is necessary: the *polynomial time tree reducibility* $\leq_{\mathrm{m}}^{\mathrm{ptt}}$ [Wa04]. It is based on paths in finite trees rather than bits of words. Let Σ be a finite alphabet. We call a triple $t = (T, h, m)$ a Σ*-tree* if $T \subseteq \{0,1\}^*$ is finite (the set of paths), $h : T \to \Sigma$, and $m \in \mathbb{N}$ such that $\forall z \forall u((u$ is an initial word of $z \wedge z \in T) \to u \in T)$ and $\forall z(z \in T \to |z| \leq m)$. Let T_Σ be the set of all Σ-trees. The *leaf word* of a Σ-tree $t = (T, h, m)$ is defined as $\beta(t) =_{\mathrm{def}} h(r_1)h(r_2)\ldots h(r_k)$ where $r_1, r_2, \ldots, r_k \in \{0,1\}^*$ are the maximal paths of t in lexicographical order.

A function $f : T_{\Sigma_1} \to T_{\Sigma_2}$ is called a *polynomial time tree function* (for short: ptt function) if and only if there exist functions $g_1 : T_{\Sigma_1} \times \{0,1\}^* \times \mathbb{N} \to \{0,1\}$, $g_2 : T_{\Sigma_1} \times \{0,1\}^* \times \mathbb{N} \to \Sigma_2$, and a $k > 0$ such that

- g_1, g_2 are polynomial time computable in the length of the $\{0,1\}^*$ argument and in the value of the $\mathbb{N}$ arguments where a tree $t \in T_{\Sigma_1}$ is accessed as an oracle, and
- $f(t) = (T', h', m^k)$ where $T' =_{\mathrm{def}} \{z : g_1^t(z,m) = 1\}$ and $h'(z) =_{\mathrm{def}} g_2^t(z,m)$ for every tree $t = (T,h,m) \in T_{\Sigma_1}$.

For pairs $(L_1, L_1') \subseteq (\Sigma_1^*)^2$ and $(L_2, L_2') \subseteq (\Sigma_2^*)^2$ of leaf languages let

$(L_1, L_1') \leq_{\mathrm{m}}^{\mathrm{ptt}} (L_2, L_2') \Leftrightarrow_{\mathrm{def}}$ there exists a ptt function $f : T_{\Sigma_1} \to T_{\Sigma_2}$ such that for all $x \in \Sigma_1^*$,
- $\beta(t) \in L_1 \to \beta(f(x)) \in L_2$ and
- $\beta(t) \in L_1' \to \beta(f(x)) \in L_2'$.

Instead of $(L_1, \overline{L_1}) \leq_{\mathrm{m}}^{\mathrm{ptt}} (L_2, \overline{L_2})$ we also write $L_1 \leq_{\mathrm{m}}^{\mathrm{ptt}} L_2$. As examples we consider

Proposition 13.7 *1.* $(0^*10^*1)^*0^* \leq_{\mathrm{m}}^{\mathrm{ptt}} (11)^*$ *and* $1 \leq_{\mathrm{m}}^{\mathrm{ptt}} (11)^*$.
2. $0^*10^* \not\leq_{\mathrm{m}}^{\mathrm{ptt}} 1$ *and* $(11)^* \not\leq_{\mathrm{m}}^{\mathrm{ptt}} 1$. □

Theorem 13.8 [Wa04] *For pairs* (L_1, L_1') *and* (L_2, L_2') *of leaf languages there holds* $(L_1, L_1') \leq_{\mathrm{m}}^{\mathrm{ptt}} (L_2, L_2')$ *if and only if* $\forall X(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1|L_1')^X \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_2|L_2')^X)$.
In particular, $L_1 \leq_{\mathrm{m}}^{\mathrm{ptt}} L_2$ *if and only if* $\forall X(\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_1)^X \subseteq \mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_2)^X)$.

To get a similar criterion like the one in Theorem 13.2 for the ε-model, i.e., a criterion for $\forall X(\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_2)^X \subseteq \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1'|L_2')^X)$ we introduce the $\leq_{\mathrm{m}}^{\mathrm{pte}}$- reducibility [Wa04]. For a finite alphabet Σ and $a \notin \Sigma$, the homomorphism $h_{\Sigma,a} : (\Sigma \cup \{\varepsilon\})^* \to \Sigma^*$ is defined by $h_{\Sigma,a}(b) =_{\mathrm{def}} b$ for $b \in \Sigma$ and $h_{\Sigma,a}(a) =_{\mathrm{def}} \varepsilon$.

For pairs $(L_1, L_1') \subseteq (\Sigma_1^*)^2$ and $(L_2, L_2') \subseteq (\Sigma_2^*)^2$ of leaf languages let

$(L_1, L_1') \leq_{\mathrm{m}}^{\mathrm{pte}} (L_2, L_2') \Leftrightarrow_{\mathrm{def}}$ there exists a $a \notin \Sigma_1 \cup \Sigma_2$ and a plt function
$f : (\Sigma_1 \cup \{a\})^* \to (\Sigma_2 \cup \{a\})^*$ such that for all $x \in (\Sigma_1 \cup \{a\})^*$,
- $h_{\Sigma_1,a}(x) \in L_1 \to h_{\Sigma_2,a}(f(x)) \in L_2$ and
- $h_{\Sigma_1,a}(x) \in L_1' \to h_{\Sigma_2,a}(f(x)) \in L_2'$.

Instead of $(L_1, \overline{L_1}) \leq_{\mathrm{m}}^{\mathrm{pte}} (L_2, \overline{L_2})$ we also write $L_1 \leq_{\mathrm{m}}^{\mathrm{pte}} L_2$. As examples we consider

Proposition 13.9 *1.* $0^*10^* \leq_{\mathrm{m}}^{\mathrm{pte}} 1$.
2. $(11)^* \not\leq_{\mathrm{m}}^{\mathrm{pte}} 1$ *and* $1 \not\leq_{\mathrm{m}}^{\mathrm{pte}} (11)^*$. □

For $L \subseteq \Sigma^*$ and $a \notin \Sigma$ define $L^a =_{\mathrm{def}} \bigcup_{b_1 b_2 \ldots b_n \in L} a^* b_1 a^* b_2 a^* \ldots a^* b_n a^*$.

Proposition 13.10 [Wa04] *For pairs* $(L_1, L_1') \subseteq (\Sigma_1^*)^2$ *and* $(L_2, L_2') \subseteq (\Sigma_2^*)^2$ *of leaf languages and* $a \notin \Sigma_1 \cup \Sigma_2$, *there holds* $(L_1, L_1') \leq_{\mathrm{m}}^{\mathrm{pte}} (L_2, L_2')$ *if and only if* $(L_1^a, (L_1')^a) \leq_{\mathrm{m}}^{\mathrm{plt}} (L_2^a, (L_2')^a)$.

Theorem 13.11 [Wa04] *For pairs* (L_1, L_1') *and* (L_2, L_2') *of leaf languages there holds* $(L_1, L_1') \leq_{\mathrm{m}}^{\mathrm{pte}} (L_2, L_2')$ *if and only if* $\forall X(\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1|L_1')^X \subseteq \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_2|L_2')^X)$.
In particular, $L_1 \leq_{\mathrm{m}}^{\mathrm{pte}} L_2$ *if and only if* $\forall X(\mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_1)^X \subseteq \mathrm{Leaf}_{\mathrm{e}}^{\mathrm{p}}(L_2)^X)$.

Problem 13.12 *Can one define a leaf language model that "fits" to the* $\leq_{\mathrm{m}}^{\mathrm{qf}}$*-reducibility in the sense of Theorem 13.2?* □

14 Reducibilities and the Doth-Depth-Hierarchies

It became clear in Section 13 that the reducibilities $\leq_{\mathrm{m}}^{\mathrm{plt}}$, $\leq_{\mathrm{m}}^{\mathrm{ptt}}$, and $\leq_{\mathrm{m}}^{\mathrm{pte}}$ are very important for the study of leaf language classes. In this section we will consider the closures of the classes of the dot-depth hierarchies under these reducibilities.

Theorem 14.1 *1.* $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(L_k) = \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{L}_{k-1/2}) = \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_{k-1/2})$ *for every* $k \geq 1$.

2. $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(L_k) = \mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{L}_{k-1/2}) = \mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{B}_{k-1/2})$ *for every* $k \geq 1$.
3. $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(D_k) = \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{L}_{k-1/2} \cap \text{co-}\mathcal{L}_{k-1/2}) = \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_{k-1/2} \cap \text{co-}\mathcal{B}_{k-1/2})$ *for every* $k \geq 2$.
4. $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(D_k) = \mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{L}_{k-1/2} \cap \text{co-}\mathcal{L}_{k-1/2}) = \mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{B}_{k-1/2} \cap \text{co-}\mathcal{B}_{k-1/2})$ *for every* $k \geq 2$.
5. $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{L}_k) = \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_k)$ *for every* $k \geq 1$.
6. $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{L}_k) = \mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{B}_k)$ *for every* $k \geq 1$. $\square$

Problem 14.2 *Are the following statements true:*
For every $L \in \mathrm{SF}$ *there exists an* L' *such that* $L \equiv_{\mathrm{m}}^{\mathrm{plt}} L'$ *and* $L \in \mathcal{B}_{k/2} \Leftrightarrow L' \in \mathcal{L}_{k/2}$ *for all* $k \geq 0$.
For every $L \in \mathrm{SF}$ *there exists an* L' *such that* $L \equiv_{\mathrm{m}}^{\mathrm{ptt}} L'$ *and* $L \in \mathcal{B}_{k/2} \Leftrightarrow L' \in \mathcal{L}_{k/2}$ *for all* $k \geq 0$.

Theorem 14.3 *For* $k \geq 1$, *the classes* $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(L_k)$ *and* $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(L_k)$ *are closed under union, intersection and concatenation.* $\square$

However, the classes $\mathcal{B}_0$ and $\mathcal{B}_{1/2}$ are not closed under $\leq_{\mathrm{m}}^{\mathrm{plt}}$-reducibility as shown in the following theorem.

Theorem 14.4 *1.* [Wa01] *A regular language* $L \in \Sigma^*$ *is in* $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_0)$ *if and only if there exists an* $r \geq 0$ *such that* L *is the finite union of sets* $w(\Sigma^r)^*v$.

2. [Gl04] $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_{1/2}) \cap \mathrm{REG} = \mathrm{Pol}(\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_0) \cap \mathrm{REG})$.
3. $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathrm{SF}) \cap \mathrm{REG} = \mathrm{SF}$.

Define $\mathcal{H}_0 =_{\mathrm{def}} \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_0) \cap \mathrm{REG}$, $\mathcal{H}_{k+1/2} =_{\mathrm{def}} \mathrm{Pol}(\mathcal{H}_k)$ and $\mathcal{H}_{k+1} =_{\mathrm{def}} \mathrm{BC}(\mathcal{H}_{k+1/2})$ for $k \geq 0$. By Theorem 14.4 we have $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_{k/2}) \cap \mathrm{REG} = \mathcal{H}_{k/2}$ for $k = 0, 1$.

Conjecture 14.5 $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_{k/2}) \cap \mathrm{REG} = \mathcal{H}_{k/2}$ *for every* $k \geq 0$.

By Theorem 14.4.1 we obtain that the non-starfree language $(11)^*$ is in $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_0)$. However, $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_0) \cap \mathrm{SF} = \mathcal{B}_0$ [Wa01]. The latter is not true for $\mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_{1/2})$. It was proved in [Gl04] that, for every $k \geq 1$, there exists a starfree language $L_k \in \mathcal{R}_{\mathrm{m}}^{\mathrm{plt}}(\mathcal{B}_{1/2}) \setminus \mathcal{B}_k$. For these languages the balanced and the unbalanced leaf language classes could differ because $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{p}}(L_k) = \mathrm{NP}$ and $\mathrm{Leaf}_{\mathrm{u}}^{\mathrm{p}}(L_k) \supseteq \mathrm{U}\Sigma_k^{\mathrm{p}}$ where $\mathrm{U}\Sigma_k^{\mathrm{p}}$ is the k-the level of the UP-*hierarchy* defined by $\mathrm{U}\Sigma_1^{\mathrm{p}} =_{\mathrm{def}} \mathrm{UP}$ and $\mathrm{U}\Sigma_{k+1}^{\mathrm{p}} =_{\mathrm{def}} \mathrm{U}\cdot\text{co-}\mathrm{U}\Sigma_k^{\mathrm{p}}$ for $k \geq 1$. There are oracles relative to which $\mathrm{U}\Sigma_k^{\mathrm{p}} \not\subseteq \Sigma_{k-1}^{\mathrm{p}}$.

What about the closure of the classes $\mathcal{B}_{k/2}$ under polynomial time tree reducibility $\leq_{\mathrm{m}}^{\mathrm{ptt}}$?

Theorem 14.6 [GTW04]

1. $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{B}_0) = \mathcal{B}_0$.
2. $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{B}_{1/2})$ *includes nonregular languages. Consequently,* $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{B}_{k/2}) \supset \mathcal{B}_{k/2}$ *for* $k > 0$.
3. $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{B}_{k/2}) \cap \mathrm{REG} = \mathcal{B}_{k/2}$ *for* $k = 1, 2, 3$.

Conjecture 14.7 $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{B}_{k/2}) \cap \mathrm{REG} = \mathcal{B}_{k/2}$ *for every* $k > 0$.

Problem 14.8 *Characterize* $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathcal{B}_{k/2})$ *and* $\mathcal{R}_{\mathrm{m}}^{\mathrm{ptt}}(\mathrm{SF})$.

In [SW04] the *quantifier free reducibility* $\leq_{\mathrm{m}}^{\mathrm{qf}}$ is defined under which all classes of the Cohen-Brzozowski hierarchy are closed. We use the finite model framework as described e.g. in [Im99]). With a finite alphabet Σ we relate the signature $\sigma_\Sigma =_{\mathrm{def}} \{\leq, \perp, \top, S\} \cup \{Q_a : a \in \Sigma\}$ where $\leq$ and S are binary symbols, the Q_a are unary symbols, and $\perp$ and $\top$ are constants. Formulae over σ_Σ speak about words $x = x_1x_2\ldots x_n$ with $x_1, x_2, \ldots, x_n \in \Sigma$ in the following way: variables i stand for positions in x and have therefore values from $\{1, 2, \ldots, n\}$. The constants $\perp$ and $\top$ stand for the first position (1) in x and the last position (n) in x. The atomic formulae $i \leq j$, $S(i,j)$, and $Q_a(i)$ are true if $i \leq j$ in $\{1, 2, \ldots, n\}$, $i + 1 = j$, and $x_i = a$, resp. For example, the $\sigma_{\{0,1,2\}}$-formula $Q_2(\perp) \wedge Q_2(\top) \wedge \forall i \forall j((i < j \wedge Q_2(i) \wedge Q_2(j)) \rightarrow \exists k(i < k \wedge k < j \wedge Q_1(k)))$ describes exactly the words from $2(0^*1(0 \cup 1)^*2)^*$.

A *sound interpretation I over the alphabets* Σ_1 *and* Σ_2 consists of an $m \geq 1$ and a collection $\{\varphi_U, \varphi_\leq, \varphi_\perp, \varphi_\top, \varphi_S\} \cup \{\varphi_b : b \in \Sigma_2\}$ of quantifier-free σ_{Σ_1}-formulae such that φ_U and the φ_b have m variables and $\varphi_\leq$ has $2m$ variables, and for every word x

- $U_{I,x} =_{\mathrm{def}} \{u : u \in \{1, \ldots, |x|\}^m \wedge \varphi_U(u)\}$ is non-empty,
- defining $u \leq_I v \Leftrightarrow_{\mathrm{def}} \varphi_\leq(u, v)$, we get a linear order $u_1 <_I u_2 <_I \cdots <_I u_r$ on $U_{I,x}$,
- $\varphi_\perp(u) \Leftrightarrow u = u_1$,
- $\varphi_\top(u) \Leftrightarrow u = u_r$,
- $\varphi_S(u, v) \Leftrightarrow$ there exists an $i \in \{1, 2, \ldots, r-1\}$ such that $u = u_i$ and $v = u_{i+1}$, and
- for every $u \in U_{I,x}$ the formula $\varphi_b(u)$ becomes true for exactly one $b \in \Sigma$ which we denote by $b_{I,x}(u)$.

Define $g_I(x) =_{\mathrm{def}} b_{I,x}(u_1)b_{I,x}(u_2)\ldots b_{I,x}(u_r)$.

A function $f : \Sigma_1^+ \rightarrow \Sigma_2^+$ is a *qf-function* if there exists a sound interpretation I over the alphabets Σ_1 and Σ_2 such that $f(x) = g_I(x)$ for allmost all x. For pairs $(L_1, L_1') \subseteq (\Sigma_1^+)^2$ and $(L_2, L_2') \subseteq (\Sigma_2^+)^2$ of leaf languages let

$(L_1|L_1') \leq_{\mathrm{m}}^{\mathrm{qf}} (L_2|L_2') \Leftrightarrow_{\mathrm{def}}$ there exists a qf-function $f : \Sigma_1^+ \rightarrow \Sigma_2^+$ such that for all $x \in (\Sigma_1)^+$,
- $x \in L_1 \rightarrow f(x) \in L_2$ and
- $x \in L_1' \rightarrow f(x) \in L_2'$.

Instead of $(L_1|\overline{L_1}) \leq_{\mathrm{m}}^{\mathrm{qf}} (L_2|\overline{L_2})$ we also write $L_1 \leq_{\mathrm{m}}^{\mathrm{qf}} L_2$. As an example we consider

Proposition 14.9 *1.* $(0^*10^*1)^*0^* \leq_{\mathrm{m}}^{\mathrm{qf}} (11)^*$ *and* $1 \leq_{\mathrm{m}}^{\mathrm{qf}} (11)^*$.
2. $0^*10^* \not\leq_{\mathrm{m}}^{\mathrm{qf}} 1$ *and* $(11)^* \not\leq_{\mathrm{m}}^{\mathrm{qf}} 1$. □

Theorem 14.10 [SW04]

1. *For $k \geq 0$ and $m \geq 1$, all classes $\mathcal{B}_{k/2}$ and all classes $\mathcal{B}_{k+1/2}(m)$ are closed under $\leq_{\mathrm{m}}^{\mathrm{qf}}$.*
2. *For $k \geq 1$, the language L_k is $\leq_{\mathrm{m}}^{\mathrm{qf}}$-complete for $\mathcal{B}_{k-1/2}$.*

Problem 14.11 *Are there $\leq_{\mathrm{m}}^{\mathrm{qf}}$-complete languages for $\mathcal{B}_{k-1/2} \cap \mathcal{B}_{k-1/2}$ for $k \geq 2$? In particular, is D_k complete languages for $\mathcal{B}_{k-1/2} \cap \mathcal{B}_{k-1/2}$ w.r.t. $\leq_{\mathrm{m}}^{\mathrm{qf}}$?* □

How do the reducibilities $\leq_{\mathrm{m}}^{\mathrm{plt}}$, $\leq_{\mathrm{m}}^{\mathrm{ptt}}$, $\leq_{\mathrm{m}}^{\mathrm{pte}}$, and $\leq_{\mathrm{m}}^{\mathrm{qf}}$ compare with each other?

Theorem 14.12 *If $\alpha, \beta \in \{\mathrm{plt}, \mathrm{ptt}, \mathrm{pte}, \mathrm{qf}\}$ such that $\alpha \neq \beta$ and $(\alpha, \beta) \neq (\mathrm{qf}, \mathrm{ptt})$ then $\leq_{\mathrm{m}}^{\alpha} \not\subseteq \leq_{\mathrm{m}}^{\beta}$.* □

Problem 14.13 *Do we have $\leq_{\mathrm{m}}^{\mathrm{qf}} \subset \leq_{\mathrm{m}}^{\mathrm{ptt}}$ or $\leq_{\mathrm{m}}^{\mathrm{qf}} \not\subseteq \leq_{\mathrm{m}}^{\mathrm{ptt}}$?.*

15 Other Leaf Language Models

In [JMT94] leaf language models are considered which rest on logarithmic space and logarithmic time machines rather than polynomial time machines. We will present here some of their results about the logspace leaf language model. In the same systematics as for polynomial time machines we denote also the leaf language classes for logspace machines but we use an upper script log. In fact, we will need only classes $\mathrm{Leaf}^{\log}(L)$ because the following statements are valid for unbalanced as well as the balanced case.

Theorem 15.1 [JMT94]

1. $\mathrm{Leaf}^{\log}(\mathrm{PSPACE}) = \mathrm{EXPSPACE}$.
2. $\mathrm{Leaf}^{\log}(\mathrm{NP}) = \mathrm{NEXPTIME}$.
3. $\mathrm{Leaf}^{\log}(\mathrm{P}) = \mathrm{EXPTIME}$.
4. $\mathrm{Leaf}^{\log}(\mathrm{L}) = \mathrm{Leaf}^{\log}(\mathrm{NL}) = \mathrm{Leaf}^{\log}(\mathrm{DSPACE}(\log^k)) = \mathrm{PSPACE}$.
5. $\mathrm{Leaf}^{\log}(\mathrm{CFL}) = \mathrm{Leaf}^{\log}(\mathrm{DCFL}) = \mathrm{Leaf}^{\log}(\mathrm{NC}^1) = \mathrm{PSPACE}$.
6. $\mathrm{Leaf}^{\log}(\Sigma_k\text{-LOGTIME}) = \Sigma_k^{\mathrm{p}}$ *for* $k \geq 1$.
7. $\mathrm{Leaf}^{\log}(\mathrm{AC}^0) = \mathrm{PH}$.
8. $\mathrm{Leaf}^{\log}(\mathrm{REG}) = \mathrm{P}$.
9. $\mathrm{Leaf}^{\log}(\mathrm{SF}) = \mathrm{Leaf}^{\log}(\mathcal{B}_{k/2}) = \mathrm{NL}$ *for* $k \geq 1$.
10. $\mathrm{Leaf}^{\log}(\mathcal{B}_0) = \mathrm{L}$.

Interestingly, the results for the polynomial time leaf language model and the logspace leaf language model differ only for small classes. In particular the results for regular leaf languages are clearly different (unless P = PSPACE).

As in the polynomial time case the minimal leaf language class in the logspace model is L. Also the "next larger" such leaf languages classes are known.

Theorem 15.2 [Bo94a] *For a regular language L there holds* $\mathrm{Leaf}_{\mathrm{u}}^{\log}(L) = \mathrm{L}$ *or* $\mathrm{Leaf}_{\mathrm{u}}^{\log}(L) \supseteq \mathrm{NL}$ *or* $\mathrm{Leaf}_{\mathrm{u}}^{\log}(L) \supseteq \mathrm{Mod}_p\mathrm{L}$ *for some prime number p.*

In [CMTV98] a leaf language model on the base of NC^1 computations was considered with leaf language classes $\mathrm{Leaf}^{\mathrm{NC}^1}(L)$. The results are similar to those for the logspace model. Simple context-free languages L were found for which still $\mathrm{Leaf}^{\mathrm{NC}^1}(L) = \mathrm{PSPACE}$ holds true.

In [GV03] a leaf language model to compute functions was considered. Let D be the set of all (3×3)-matrices with entries from $\{-1,0,1\}$. For a string $x \in \mathrm{D}^*$ let $\langle x\rangle$ be the result of multiplying the matrices in x in the given order. The following result is the function analog of the *bottleneck* characterization of PSPACE (Cai und Furst 1991).

Theorem 15.3 [GV03] *A function f is computable in polynomial space if and only if there exist a nondeterministic polynomial time machine M that computes on every computation path a matrix from D such that*

$$f(x) = (1\,0\,0)\cdot\langle\beta_M(x)\rangle\cdot\begin{pmatrix}1\\0\\0\end{pmatrix}.$$

A general leaf language model for computing functions was defined and investigated in [KSV00].

In [Ko00] a leaf language model to describe classes of finite partitions was considered.

16 Leaf Languages for Recursion-Theoretic Classes

Is it possible to define recursion-theoretic classes with simple leaf languages? Since we understand the polynomial time hierarchy as the complexity-theoretic analogue of the arithmetic hierarchy we look for a leaf language model in which, analogously to Theorem 11.4, the classes of the Cohen-Brzozowski hierarchy generate the corresponding classes of the arithmetic hierarchy. We restrict ourselves to the case of balanced and complementary leaf language classes.

To reflect the quantifier structure in the arithmetic hierarchy where the quantifiers range over an infinite domain, leaf languages with finite objects like words are not suitable. We use infinite sequences from Σ^{ω^m} with $m \geq 1$, i.e., infinite sequences of ordinality ω^m over the finite alphabet Σ. Such sequences can be represented by total functions $\mathbb{N}^m \to \Sigma$. We generalize the leaf language concept to ω^m-languages. For $B \in \Sigma_1^*$ and $L \in \Sigma_2^{\omega^m}$ let

$$B \in \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(L) \Leftrightarrow_{\mathrm{def}} \text{there exists a computable total } g : \Sigma_1^* \times \mathbb{N}^m \to \Sigma_2 \text{ such that } x \in B \leftrightarrow g(x) \in L \text{ for all } x \in \Sigma_1^*$$

where $g(x)(i_1, \ldots, i_m) =_{\mathrm{def}} g(x, i_1, \ldots, i_m)$ for $x \in \Sigma_1^*$ and $i_1, \ldots, i_m \in \mathbb{N}$.

Regular ω^m-languages ($m \geq 1$) are well studied (Büchi 1965, Choueka 1978, Wojciechowski 1985, Bedon and Carton 1998). One possible definition is the following: An ω^m-language is regular if it can be described by a monadic second order formula with the relations $<$, s (successor relation), min and max. According to the corresponding results on starfree languages of words (Thomas 1982) we define: An ω^m-language is starfree if it can be described by a monadic first order formula with the same relations. An ω^m-language is in the class $\mathcal{B}_{k-1/2}^{\omega,m}$ if it can be described by a monadic first order formula with the same relations whose quantifiers have a Σ_k-structure. Further, let $\mathcal{B}_k^{\omega,m} =_{\mathrm{def}} \mathrm{BC}(\mathcal{B}_{k-1/2}^{\omega,m})$ and $\mathcal{B}_{k/2}^{\omega} =_{\mathrm{def}} \bigcup_{m \geq 1} \mathcal{B}_{k/2}^{\omega,m}$ for $k \geq 1$. Finally, for $k \geq 1$, let Σ_k^{ar} be the k-th class of the arithmetic hierarchy.

Theorem 16.1 *For $k \geq 1$, there holds $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(\mathcal{B}_{k-1/2}^{\omega}) = \Sigma_k^{ar}$ and $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(\mathcal{B}_k^{\omega}) = \mathrm{BC}(\Sigma_k^{ar})$.*

We conjecture that the preceding result is not valid for fixed m. For $m = 1$ it is definitively not valid because we can prove that $\bigcup_{k \geq 1} \mathcal{B}_k^{\omega,1} = \mathrm{BC}(\Sigma_2^{ar})$. Generally, we conjecture $\bigcup_{k \geq 1} \mathcal{B}_k^{\omega,m} = \mathrm{BC}(\Sigma_{2m}^{ar})$ for $m \geq 2$.

We say that the leaf set $L \in \Sigma^{\omega^m}$ characterizes the m-degree of the set B if $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(L) = \{C : C \leq_{\mathrm{m}} B\}$. Which leaf sets characterize m-degrees? Which m-degrees can be characterized by leaf sets? It is not hard to see that, if one modifies the definition of $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(L)$ by omitting "total" then every leaf set L characterizes the m-degree of $R(L) =_{\mathrm{def}} \{c(r, x) : u(r, x) \in L\}$ where u is a universal function for the $(n + 1)$-ary computable functions.

It would be very useful to have a reducibility $\leq^*$ between ω^n-sets such that

$$L_1 \leq^* L_2 \Leftrightarrow \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(L_1) \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(L_2).$$

A first idea is as follows. For $L_1 \subseteq \Sigma_1^{\omega^m}$ and $L_2 \subseteq \Sigma_2^{\omega^n}$ we define $L_1 \leq_{\mathrm{m}}^{\mathrm{rec}} L_2 \Leftrightarrow_{\mathrm{def}}$ there exists a total computable oracle function g such that $\xi \in L_1 \leftrightarrow g^{\xi}(0)g^{\xi}(1)g^{\xi}(2)\ldots \in L_2$ for all $\xi \in \Sigma_1^{\omega^m}$

Theorem 16.2 *If $L_1 \leq_{\mathrm{m}}^{\mathrm{rec}} L_2$ then $\mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(L_1) \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(L_2)$.*

Problem 16.3 *Does $L_1 \leq_{\mathrm{m}}^{\mathrm{rec}} L_2 \Leftrightarrow \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(L_1) \subseteq \mathrm{Leaf}_{\mathrm{b}}^{\mathrm{rec}}(L_2)$ hold true or do we need a weaker reducibility to achieve that?*

17 Further General Models and the Leaf Language Model

In this section we will consider other general models to define complexity classes.

In [GP86] and [He92a] the model of *locally definable acceptance types* was introduced. Let $k \geq 2$, and let F be a finite set of functions over $\{1, 2, \ldots, k\}$. A nondeterministic polynomial time machine M is called an F-machine if it assigns to every node of indegree r of its computation tree on input x an r-ary function from F. The input is accepted if a circuit-like evaluation of the computation tree results in 1. The class $(F)P$ consists of all languages which can be accepted in this way by an F-machine. For every F there holds $\mathrm{P} \subseteq (F)P \subseteq \mathrm{PSPACE}$, and there are single finite functions f such that $(\{f\})P = \mathrm{PSPACE}$. The connection to the leaf language model is as follows. For every regular language L there is a binary associative function f such that $(\{f\})P = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(L)$, and for every binary associative function f there is a regular language L such that $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L) = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{u}}(L) = (\{f\})P$. More results about locally definable acceptance types can be found in [GP86, He92a, He92b, He94a, He94b, NR98, PV01]

In [BS01] the leaf language concept is generalized to operators, the *dot operators*. For a pair (L_1, L_2) of leaf languages over the alphabet $\{0, 1\}$ the dot operator $(L_1, L_2)\cdot$ is defined by

$$A \in (L_1, L_2) \cdot \mathcal{K} =_{\mathrm{def}} \text{there exist a } B \in \mathcal{K} \text{ and a polynomial time computable function } h \text{ such that for all } x,$$
$$x \in A \rightarrow \mathrm{c}_B(x,0)\mathrm{c}_B(x,1)\ldots\mathrm{c}_B(x,h(x)) \in L_1 \text{ and}$$
$$x \notin A \rightarrow \mathrm{c}_B(x,0)\mathrm{c}_B(x,1)\ldots\mathrm{c}_B(x,h(x)) \in L_2.$$

Furthermore, for a leaf language L define $L \cdot \mathcal{K} =_{\mathrm{def}} (L, \overline{L}) \cdot \mathcal{K}$. For example, $(0^*1(0\cup1)^*)\cdot\mathcal{K} = \exists\cdot\mathcal{K}$ for every class $\mathcal{K}$ of languages. The observation $(L_1, L_2) \cdot \mathrm{P} = \mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L_1|L_2)$ makes clear that the dot operator $(L_1, L_2)\cdot$ is a generalization of the leaf language class $\mathrm{Leaf}^{\mathrm{p}}_{\mathrm{b}}(L_1|L_2)$. It is proved that $\forall\mathcal{K}(L \cdot \mathcal{K} \subseteq (L_1, L_2) \cdot \mathcal{K})$ if and only if $(L, \overline{L})$ is polylogtime monotone projection reducible to (L_1, L_2). This can be considered as a generalization of Theorem 13.2. This model can also be used to characterize reducibility notions. For example, there exists a language L_T such that for all languages A and B there holds $A \leq^{\mathrm{p}}_{\mathrm{T}} B \Leftrightarrow A \in L_T \cdot \{C : C \leq^{\mathrm{p}}_{\mathrm{m}} B\}$. Many reducibilities can be characterized in such a way.

In [BV98] and [GV01] close connections between leaf language definable complexity classes and classes defined by certain *Lindström quantifiers* were established.

Acknowledgements. I am grateful to Bernd Borchert, Tübingen, for many valuable remarks on preliminary versions of this survey. Also I am thankful to Christian Glaßer, Stephen Travers, and Fabian Müller, Würzburg, for the interesting discussions about this paper.

References

[Aa04] S. Aaronson. The complexity zoo. http://www.complexityzoo.com/

[BCS91] D.P. Bovet, P. Crescenzi, R. Silvestri. Complexity classes and sparse oracles. Proc. 6th IEEE Structure in Complexity Theory Conference, 1991, 102–108. Journal of Computer and System Sciences 50, 1995, 382–390.

[BCS92] D.P. Bovet, P. Crescenzi, R. Silvestri. A uniform approach to define complexity classes. *Theoret. Comp. Sci*, 104 (1992), 263–283.

[BDG88] J.L. Balcázar, J. Díaz and J. Gabarró. *Structural Complexity I*, volume 11 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1988.

[BDG90] J.L. Balcázar, J. Díaz and J. Gabarró. *Structural Complexity II*, volume 11 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1990.

[Bo94a] B. Borchert. On the Acceptance Power of Regular Languages. Proc. 11th Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Computer Science 775, Springer Verlag, 1994, pp. 533-542. Final version: Theoretical Computer Science 148, 1995, 207–225.

[Bo94b] B. Borchert. Predicate Classes, Promise Classes, and the Acceptance Power of Regular Languages. Ph.D. thesis, Universität Heidelberg, 1994.

[Bo04] B. Borchert. Personal communication.

[BKS99] B. Borchert, D. Kuske and F. Stephan. On existentially first–order definable languages and their relation to NP. *Theor. Informatics Appl.*, 33 (1999), 259–269.

[BLSTT04] B. Borchert, K-J. Lange, F. Stephan, P. Tesson, D. Thérien. The dot-depth and the polynomial hierarchy correspond on the delta levels. Technical Report 2004-03, Wilhelm-Schickard-Institut, Universität Tübingen. To appear in the proceedings of DLT 2004.

[BL96] B. Borchert, A. Lozano. Succinct Circuit Representations and Leaf Language Classes are Basically the Same Concept. Information Processing Letters 58, 1996, 211–215.

[BS97] B. Borchert, R. Silvestri. A Characterization of the Leaf Language Classes. Information Processing Letters 63, 1997, 153–158. Preliminary version: B. Borchert. Predicate classes and promise classes, Proc. 9th Structure in Complexity Theory Conference, 1994, 235–241.

[BS01] B. Borchert, R. Silvestri. Dot Operators. Theoretical Computer Science 262(1), 2001, 501–523.

[BV98] H.-J. Burtschick and H. Vollmer. Lindström Quatifiers and Leaf Language Definability. *Int. J. of Foundations of Computer Science*, 9 (1998), 277–294.

[BSS99] B. Borchert, H. Schmitz, F. Stephan. Unpublished manuscript, 1999.

[CHVW98] K. Cronauer, U. Hertrampf, H. Vollmer and K.W. Wagner. The chain method to separate counting classes. *Theory Comput. Systems*, 31 (1998), 93–108.

[CMTV98] H. Caussinus, P. McKenzie, D. Thérien, H. Vollmer. Nondeterministic NC1 Computation. Journal of Computer and System Sciences 57, 1998, 200–212.

[Ga98] M. Galota, Blattsprachen und endliche Automaten. Technical Report No. 205, University of Würzburg, Department of Computer Science, 1.

[GKV03] M. Galota, S. Kosub, H. Vollmer. Generic Separations and Leaf Languages. to appear in Mathematical Logic Quarterly, 2003.

[Gl04] Ch. Glaßer. Counting with Counterfree Automata Electronic Colloquium on Computational Complexity, Report TR04-011, 2004. Technical Report No. 315, University of Würzburg, 2004.

[GTW04] Ch. Glaßer., St. Travers, K Wagner. On $\leq_{\mathrm{m}}^{\mathrm{ptt}}$-reducibility. Manuscript 2004.

[GP86] L.M. Goldschlager, I. Parberry. On the Construction of Parallel Computers from Various Bases of Boolean Functions. Theoretical Computer Science 43, 1986, 43–58.

[GV01] M. Galota, H. Vollmer. A generalization of the Büchi-Elgot-Trakhtenbrot theorem. Proc. 15th Computer Science Logic 2001, Lecture Notes in Computer Science Vol. 2142, 2001, 355–368.

[GV03] M. Galota, H. Vollmer. Functions computable in polynomial space. Manuscript 2003.

[He92a] U. Hertrampf. Locally Definable Acceptance Types for Polynomial Time Machines. Proc. 9th Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Computer Science 577, Springer Verlag, 1992, 199–207.

[He92b] U. Hertrampf. Locally Definable Acceptance Types - the Three Valued Case. 1st Latin American Symposium in Theoretical Informatics, Lecture Notes in Computer Science 583, Springer Verlag, 1992, 262–271.

[He94a] U. Hertrampf. Complexity Classes with Finite Acceptance Types. Proc. 11th Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Computer Science 775, Springer Verlag, 1994, 543–553.

[He94b] U. Hertrampf. Complexity Classes Defined via k-Valued Functions. Proc. 9th Structure in Complexity Theory Conference, 1994, 224–234.

[HHH99] E. Hemaspaandra, L.A. Hemaspaandra, H. Hempel. A downward collapse within the polynomial hierarchy. SIAM Journal on Computing, 28(1999), 383–393.

[HLSVW93] U. Hertrampf, C. Lautemann, T. Schwentick, H. Vollmer and K.W. Wagner. On the power of polynomial time bit-reductions. *Proc. 8th Structure in Complexity Theory*, 1993, 200–207.

[HVW96] U. Hertrampf, H. Vollmer and K.W. Wagner. On balanced vs. unbalanced computation trees. *Math. Systems Theory* 29 (1996), 411–421.

[Im99] N. Immermann. Descriptive Complexity. Berlin, Springer, 1999.

[JMT94] B. Jenner, P. McKenzie and D. Thérien. Logspace and logtime leaf languages. 9th Annual Conference on Structure in Complexity Theory 1994, 242–254. *Information and Computation*, 129 (1996), 21–33.

[Ka88] J. Kadin. The polynomial time hierarchy collapses if the boolean hierarchy collapses. SIAM Journal on Computing, 17(1988), 1263–1282.

[Ko00] S. Kosub. On NP-partitions over posets with an application to reducing the set of solutions of NP problems. Proc. 25th Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science Vol. 1893, 2000, 467–476.

[KSV00] S. Kosub, H. Schmitz, H. Vollmer. Uniformly defining complexity classes of functions. International Journal of Foundations of Computer Science 11(4), 2000, 525–551.

[Mu05] F. Müller. On families of leaf language classes. Diploma thesis, University of Würzburg 2005.

[NR98] R. Niedermeier, P. Rossmanith. Unambiguous computations and locally definable acceptance types. Theoretical Computer Science 194, 1998, 137–161.

[Pa94] Ch.H. Papadimitriou. Computational Complexity. Addison Wesley, 1994.

[PV01] T. Peichl, H. Vollmer. Finite automata with generalized acceptance criteria. Discrete Mathematics and Theoretical Computer Science 4, 2001, 179–192.

[Pi96] J.-E. Pin. Syntactic semigroups. In: G. Rozenberg and A. Salomaa (editors), Handbook of formal languages, Volume 1. 679–746. Springer, 1996.

[RW01] St. Reith, K.W. Wagner, On high and low sets for the boolean hierarchy. Theoretical Computer Science 261(2001), 305-321.

[Sch00] H. Schmitz. The forbidden pattern approach to concatenation hierachies. Ph.D. Thesis, University of Würzburg 2000.

[SW98] H. Schmitz, K. Wagner. The Boolean Hierarchy over Level 1/2 of the Straubing-Therien Hierarchy. Technical Report No. 201, University of Würzburg, Department of Computer Science, 1998.

[Se01] V.L. Selivanov. Relating automata-theoretic hierarchies to complexity-theoretic hierarchies. Proc. 13th Conference on Fundamentals of Computation Theory 2001, Lecture Notes in Computer Science Vol. 2138, 2001, 323–334. Final version: Theoret. Informatics Appl. 36 (2002), 29–42.

[SW04] V.L. Selivanov, K.W. Wagner. A reducibility for the dot-depth hierarchy. Proc. of the 29th Intern. Symp. on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science Vol. 3153, 2004, 783–793.

[Tr02] St. Travers. Blattsprachen Komplexitätsklassen: Über Turing-Abschluss und Counting-Operatoren Studienarbeit, Universität Würzburg, Dezember 2002.

[Ve93] N.K. Vereshchagin. Relativizable and non-relativizable theorems in the polynomial theory of algorithms. *Izvestiya Rossiiskoi Akademii Nauk*, 57 (1993), 51–90 (in Russian).

[Ve98] H. Veith. Succinct Representation, Leaf Languages and Projection Reductions. Information and Computation 142 (1998), 207–236. Preliminary version: "Succinct Representation and Leaf Languages" in: Proc. 11th Annual IEEE Conference on Computational Complexity (CCC), 1996, 118–126.

[Vo98] H. Vollmer. Relating polynomial time to constant depth. Theoretial Computer Science 207 (1998), 159–170

[Vo99] H. Vollmer. Uniform characterizations of complexity classes. ACM SIGACT-Newsletter 30(1), 1999, 17–27.

[Vo03] H. Vollmer. Complexity theory made easy - the formal language approach to the definition of complexity classes. Proc. 7th Developments in Language Theory, 2003.

[Vo04] H. Vollmer. The Leaf Language Homepage.
http://www.thi.uni-hannover.de/forschung/leafl/

[Wa01] K.W. Wagner. A reducibility and complete sets for the dot-depth hierarchy. Manuscript.

[Wa04] K.W. Wagner. New BCSV theorems. Technical Report 337, Institut für Informatik, Universität Würzburg.

[Ya85] A.C. Yao. Separating the polynomial-time hierarchy by oracles. Proc. 26th IEEE Symposium on Foundations of Computer Science (1985), 1–10.