Relating Automata-theoretic Hierarchies to Complexity-theoretic Hierarchies

V. L. Selivanov*
Theoretische Informatik, Universität Würzburg†

Abstract

We show that some natural refinements of the Straubing and Brzozowski hierarchies correspond (via the so called leaf-languages) step by step to similar refinements of the polynomial-time hierarchy. This extends a result of H.-J. Burtschik and H. Vollmer on relationship between the Straubing and the polynomial hierarchies. In particular, this applies to the boolean hierarchy and the plus–hierarchy.

1 Introduction

In complexity theory, the so called leaf-language approach to defining complexity classes became recently rather popular. Let us recall some relevant definitions. Consider a polynomial-time nondeterministic Turing machine M working on an input word x over some alphabet X and printing a letter from another alphabet A after finishing any computation path. These values are the leaves of the binary tree defined by the nondeterministic choices of M on input x. An ordering of the tuples in the program of M determines a left-to-right ordering of all the leaves. In this way, M may be considered as a deterministic transducer computing a total function $M : X^* \to A^*$ from the set of words X^* over X to the set of words over A. Now, relate to any language $L \subseteq A^*$ (called in this situation a leaf language) the language $M^{-1}(L) \subseteq X^*$. Denote by $Leaf(L)$ the set of languages $M^{-1}(L)$, for all machines M specified above. For a set of languages $\mathcal{C}$, let $Leaf(\mathcal{C})$ be the union of $Leaf(L)$, for all $L \in \mathcal{C}$.

It turns out that many inportant complexity classes have natural and useful descriptions in terms of leaf languages (see e.g. [BCS92, V93, HL+93, B95, HVW95, HVW96, JMT96]). In particular, a close relationship between some classes of regular leaf languages and complexity classes within $PSPACE$ was established in [HL+93]. In [BV98], a close relationship between the Straubing hierarchy $\{\mathcal{L}_n\}$ and the polynomial hiearachy $\{\Sigma_n^p\}$ was established: $Leaf(\mathcal{L}_n) = \Sigma_n^p$, for any $n > 0$.

In this paper, we consider the possibility of extending the last result to some natural refinements of the above–mentioned hierarchies (in the context of complexity theory, these refinements were introduced and studied in [Se93,

*Supported by the Alexander von Humboldt Foundation, by the German Research Foundation (DFG) and by the Russian Foundation for Basic Research Grant 00-01-00810.

†On a short leave from Novosibirsk Pedagogical University. Email: vseliv@nspu.ru.

Se94, Se95, Se99]). Note that for the important particular case of the boolean hiearachy over NP a result similar to ours was earlier established in [SW98, Theorem 6.3], and we actually use the idea of proof of that theorem. We make also an essential use of a result from [PW97] cited in Section 3.

In Section 2 we give the exact definitions of our hierarchies, in Section 3 we consider some relevant notions from language theory, in Sections 4—6 we present our main results, and further we give some examples and discussions.

2 Hierarchies

In different areas of mathematics, people consider a lot of hierarchies which are typically used to classify some objects according to their complexity. Here we define and discuss some hierarchies relevant to the topic of this paper.

We already mentioned the polynomial hierarchy $\{\Sigma_n^p\}$ which is one of the most popular objects of complexity theory. Note that classes (or levels) of the polynomial hierarchy are classes of languages over some finite alphabet X. In the context of complexity theory, the cardinality of X is not important (provided that it is at least 2), so it is often assumed that X is just the binary alphabet $\{0,1\}$. For detailed definition and properties of the polynomial hierarchy and other relevant notions see any standard textbook on complexity theory, say [BDG88, BDG90]. Sometimes it is convenient to use more compact notation for the polynomial hierarchy, namely PH; hence $PH = \{\Sigma_n^p\}$.

Let us define now two hierarchies which are rather popular in automata theory. A word $u = u_0 \dots u_n \in A^+$ (A^+ denotes the set of finite nonempty words over an alphabet A, while A^*—the set of all finite words over A, including the empty word ε) may be considered as a first–order structure $\mathbf{u} = (\{0,\dots,n\}; <, Q_a, \dots)$, where $<$ has its usual meaning and $Q_a (a \in A)$ are unary predicates on $\{0,\dots,n\}$ defined by $Q_a(i) \leftrightarrow u_i = a$. By Theorem of McNaughton and Papert [MP71], the so called star–free regular languages are exactly the sets of words u, for which $\mathbf{u}$ satisfies a first–order sentence of signature $\sigma_A = \{<, Q_a,\}_{a \in A}$.

For any $n > 0$, let $\mathcal{L}_n$ denote the class of languages defined by Σ_n–sentences of signature σ (i.e., sentences in prenex normal form starting with the existential quantifier and having $n-1$ quantifier alternations); the sequence $SH = \{\mathcal{L}_n\}$ is known as the Straubing (or Straubing–Thérien) hierarchy. In cases when it is important to specify the alphabet, we denote the n-th level as $A^+\mathcal{L}_n$, and the whole hierarchy as A^+SH. There is also a *-version of the Straubing hierarchy which will be denoted as $A^*SH = \{A^*\mathcal{L}_n\}$; the relationship between both versions is very easy: $A^*\mathcal{L}_n = A^+\mathcal{L}_n \cup A^+\mathcal{L}_n^\varepsilon$ (for any class $\mathcal{X}$ of subsets of A^+ we denote $\mathcal{X}^\varepsilon = \{X \cup \{\varepsilon\} | X \in \mathcal{X}\}$).

The Brzozowski hierarchy is defined in the same way, only in place of σ_A one takes the signature $\sigma'_A = \sigma_A \cup \{\bot, \top, s\}$, where $\bot$ and $\top$ are constant symbols and s is a unary function symbol ($\bot, \top$ are assumed to denote the least and the greatest element respectively, while s denotes the successor function). Brzozowski hierarchy will be denoted by $BH = \{\mathcal{B}_n\}$, with the corresponding variations in case when we need to mention the alphabet explicitely.

Note that in automata theory people usually define the Straubing and Brzozowski hierarchies by means of regular expressions; the equivalence of those definitions to definitions used here is known from [T82, PP86]. For more information on logical aspects of automata–theoretic hierarchies see also [Se01].

Next we would like to define some refinements of the introduced hierarchies. In order to do this in a uniform way, we need a technical notion of a base. Let $(B; \cup, \cap, \bar{}, 0, 1)$ be a boolean algebra (b.a.). Without loss of generality, one may think that this is the class of subsets of some set. By *a base in* B we mean any sequence $L = \{L_n\}_{n<\omega}$ of sublattices of $(B; \cup, \cap, 0, 1)$, satisfying the inclusions $L_n \cup co(L_n) \subseteq L_{n+1}$ (here $co(L_n)$ denotes the dual set $\{\bar{x} | x \in L_n\}$ for L_n). Note that levels of the hierarchies introduced above (as well as of many other popular hierarchies) are bases (take in place of L_n respectively Σ^p_{n+1}, $\mathcal{L}_{n+1}$ and $\mathcal{B}_{n+1}$).

With any base $L = \{L_n\}_{n<\omega}$ one can associate a sequence of new subsets of B as follows. Let T be the set of terms of the signature $(\cup, \cap, \bar{}, 0, 1)$ with variables $v^n_k (k, n < \omega)$. We call $v^n_k (k < \omega)$ *variables of type* n, and elements of T—*typed boolean terms*. Relate to any term $t \in T$ the set $t(L)$ of all its values when variables of type n range over L_n. We call the sequence $\{t(L)\}_t$ *the typed boolean hierarchy over* L.

Let us state some easy properties of the introduced classes.

Lemma 2.1 *(i) Typed boolean hierarchy is a refinement of* L, *i.e. any class* L_n *is among the classes* $t(L), t \in T$.

(ii) If L *is a base in* B, L' *is a base in* B' *and* $f : B \to B'$ *is a homomorphism of boolean algebras such that* $f(L_n) \subseteq L'_n$ *for all* $n < \omega$, *then* $f(t(L)) \subseteq t(L')$ *for any* $t \in T$.

Proof. (i) Let $t = v^n_0$, then $t(L) = L_n$.

(ii) We have to deduce $f(a) \in t(L')$ from $a \in t(L)$. Let $t = t(x_0, \dots, x_k)$, where $x_0, \dots, x_k$ are some typed variables. By definition of $t(L)$, $a = t(a_0, \dots, a_k)$ for some $a_0, \dots, a_k \in B$ such that $a_j \in L_n$ whenever x_j is of type n. By a property of homomorphisms, $f(a) = t(f(a_0), \dots, f(a_k))$. From $f(L_n) \subseteq L'_n$ it follows that $f(a_j) \in L'_n$ whenever x_j is of type n. Hence, $f(a) \in t(L')$. This completes the proof.

Now we prove a more subtle property useful for some considerations below. Let L be a base in a b.a. B, and let $B' = B \times \{0, 1\}$ be the cartesian product of B and of the 2-element b.a. $\{0, 1\}$. Hence, $B' = \{(b, c) | b \in B, c \leq 1\}$ and the boolean operations in B' are componentwise. We will state a close relationship between the typed boolean hierarchies over L and over the base L' in B' defined by $L'_n = L_n \times \{0, 1\}$.

Let $B^c = B \times \{c\}$, then $B' = B^0 \cup B^1$ and B^c may be considered as a b.a. isomorpic to B (isomorphism is given of course by the function $b \mapsto (b, c)$; note that symbols $\cap, \cup$ have the same interpretations in B^c as they have in B', while the symbol $\bar{}$ has slightly different interpretations in B^0 and B^1). Let L^c be the base in B^c induced by L under this isomorphism, i.e. $L^c_n = L_n \times \{c\}$. Then $L'_n = L^0_n \cup L^1_n$.

We call a boolean term $t(x_0, \dots, x_k)$ *nontrivial*, if the function induced by this term on $\{0, 1\}$ is not constant, i.e. for any $c \leq 1$ there are $c_j \leq 1$ satisfying $t(c_0, \dots, c_k) = c$. As is well known, a term is nontrivial iff it is not equivalent to the constant boolean terms $0, 1$ in the theory of boolean algebras.

Lemma 2.2 *(i) For any* $t \in T$ *and* $c \leq 1$, $t(L^c) = t(L) \times \{c\}$.

(ii) For any nontrivial $t \in T$, $t(L') = t(L^0) \cup t(L^1)$.

Proof. (i) follows from Lemma 2.1.

(ii) Let $t = t(x_0, \ldots, x_k)$ for some typed variables x_j, and let $a \in t(L'), a = (b,c) \in B'$. Then $a = t((b_0,c_0), \ldots, (b_k,c_k))$ for some $(b_j,c_j) \in B'$ such that $(b_j,c_j) \in L'_n$ whenever x_j is of type n. We have $a = (t(b_0,\ldots,b_k), t(c_0,\ldots,c_k))$, and $b_j \in L_n$ whenever x_j is of type n. Hence, $b \in t(L)$ and $a \in t(L) \times \{c\} = t(L^c)$.

It remains to check the inclusion $t(L) \times \{c\} \subseteq t(L')$. By nontriviality of t, $c = t(c_0,\ldots,c_k)$ for some $c_j \leq 1$. Let $a = (b,c) \in t(L) \times \{c\}$, then $b = t(b_0,\ldots,b_k)$ for some $b_0,\ldots,b_k \in B$ such that $b_j \in L_n$ whenever x_j is of type n. We have $a = t((b_0,c_0),\ldots,(b_k,c_k))$ and $(b_j,c_j) \in L'_n$ whenever x_j is of type n. Hence, $a \in t(L')$ completing the proof.

Taking in place of L the base $PH = \{\Sigma^p_{n+1}\}_n$, we get the typed Boolean hierarchy $\{t(PH)\}$ over PH introduced and studied in [Se94, Se94a, Se95, Se99]. In particular, the following fact was established.

Lemma 2.3 *All classes of the typed boolean hierarchy over PH are closed downwards under the polynomial m−reducibility and contain polynomially m−complete sets.*

Taking in place of L the base SH (the base BH), we get the typed Boolean hierarchy $\{t(SH)\}$ (resp., $\{t(BH)\}$) over SH (resp., over BH). A study of these hierarchies, as well as of fine hierarchies over SH and BH defined below was initiated in [SS00]. If we want to specify the alphabet explicitly we again use notation like $t(A^+SH)$.

The three examples of the typed Boolean hierarchy introduced above will be the main objects of this paper. Let us state the following easy facts on the typed boolean hierarchies over SH and BH.

Lemma 2.4 *(i) For any $n > 0$ and any alphabet A, $A^+\mathcal{L}_n \subseteq A^*\mathcal{L}_n$ and $A^+\mathcal{L}_n \subseteq A^+\mathcal{B}_n$.*

(ii) For any $t \in T$ and any alphabet A, $t(A^+SH) \subseteq t(A^*SH)$ and $t(A^+SH) \subseteq t(A^+BH)$.

Proof. (i) is evident.

(ii) Follows from (i) and Lemma 2.1.(ii), by taking the identity function in place of f.

Now we establish a relationship between the *- and +-versions of the typed boolean hierarchy over SH which informally means the equivalence of the both versions for the purposes of this paper.

Lemma 2.5 *(i) For any nontrivial $t \in T$ and any alphabet A, $t(A^*SH) = t(A^+SH) \cup t(A^+SH)^\varepsilon$.*

(ii) For any $t \in T$ and any alphabet A, $Leaf(t(A^*SH)) = Leaf(t(A^+SH))$.

Proof. (i) Let $B = P(A^+)$ and $L = A^+SH$. Then b.a. $P(A^*)$ is naturally isomorphic to the b.a. $B' = B \times \{0,1\}$ (the isomorphism $h : B \times \{0,1\} \to P(A^*)$ is defined by $h(L,0) = L$ and $h(L,1) = L \cup \{\varepsilon\}$). Using notation introduced before Lemma 2.2, we have $h(L'_n) = A^*\mathcal{L}_n$. Now the assertion follows from Lemma 2.2.

(ii) If t is equivalent (in the theory of boolean algebras) to 0, then $t(A^*SH) = \{\emptyset\} = t(A^+SH)$, and the equation follows. If t is equivalent to 1, then $t(A^*SH) = \{A^*\}$ and $t(A^+SH) = \{A^+\}$. But $Leaf(A^*) = X^* = Leaf(A^+)$, because the leaf word is always nonempty. Finally, let t be nontrivial. Then the equation follows from (i), because $Leaf(t(A^+SH)^\varepsilon) = Leaf(t(A^+SH))$, again by the nonemptyness of the leaf word. This completes the proof of the lemma.

We will consider also another hierarchy called the *fine hierarchy* over L. It was introduced by the author in the context of recursion theory and then considered also in several other contexts. Let us briefly recall the definition of the fine hierarchy over L. Its classes (or levels) S_α are numbered by ordinals $\alpha < \varepsilon_0$, where $\varepsilon_0 = sup\{\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots\}$ (for more information about the well–known ordinal ε_0 and the ordinal arithmetic see e.g. [KM67]).

We define the classes S_α^n, where n is an auxiliary parameter, by induction on α as follows (simplifying notation we write in this definition ab in place of $a \cap b$):

$S_0^n = 0$; $S_{\omega^\gamma}^n = S_\gamma^{n+1}$ for $\gamma > 0$;

$S_{\alpha+1}^n = \{u_0x_0 \cup u_1x_1 | u_i \in L_n, x_0 \in S_\alpha^n, x_1 \in co(S_\alpha^n), u_0u_1x_0 = u_0u_1x_1\}$;

$S_{\delta+\omega^\gamma}^n = \{u_0x_0 \cup u_1x_1 \cup \bar{u}_0\bar{u}_1y | u_i \in L_n, x_0 \in S_\alpha^n, x_1 \in co(S_\alpha^n), y \in S_\delta^n, u_0u_1x_0 = u_0u_1x_1\}$ for $\delta = \omega^\gamma \cdot \delta' > 0, \gamma > 0$.

To see that this definition is correct note that every nonzero ordinal $\alpha < \varepsilon_0$ is uniquely representable in the form $\alpha = \omega^{\gamma_0} + \cdots + \omega^{\gamma_k}$ for a finite sequence $\gamma_0 \geq \cdots \geq \gamma_k$ of ordinals $< \alpha$. Applying the definition we subsequently get $S_{\omega^{\gamma_0}}^n, S_{\omega^{\gamma_0}+\omega^{\gamma_1}}^n, \ldots, S_\alpha^n$. Finally, let $S_\alpha = S_\alpha^0$.

Let us recall some simple properties of the fine hierarchy over any base L (for more information and for proofs see e.g. [Se95]).

Lemma 2.6 *(i) $S_\alpha \cup co(S_\alpha) \subseteq S_\beta$ for all $\alpha < \beta < \varepsilon_0$.*

(ii) Fine hierarchy is a refinement of L, i.e. any class L_n is among the classes $S_\alpha (\alpha < \varepsilon_0)$.

(iii) If L is a base in B, L' is a base in B' and $f : B \to B'$ is a homomorphism of boolean algebras such that $f(L_n) \subseteq L'_n$ for all $n < \omega$, then $f(S_\alpha) \subseteq S'_\alpha$ for all $\alpha < \varepsilon_0$ (here $\{S'_\alpha\}$ is of course the fine hierarchy over L').

We will consider the fine hierarchy over all the three bases PH, SH, BH introduced above, and we denote the corresponding hierarchies as $\{S_\alpha(PH)\}$, $\{S_\alpha(SH)\}$, and $\{S_\alpha(BH)\}$, respectively. Again, to mention the alphabets explicitly we use notation like $S_\alpha(A^+BH)$.

3 Families of Languages

By a *+-class of languages* [PW97] we mean a correspondence $\mathcal{C}$ which associates with each finite alphabeth A a set $A^+\mathcal{C} \subseteq P(A^+)$, where $P(A^+)$ is the set of all subsets of A^+. In this paper we need classes of languages with some minimal closure properties as specified in the following

Definition. By a *+-family of languages* we mean a +-class $\mathcal{C} = \{A^+\mathcal{C}\}_A$ such that

(1) for every semigroup morphism $\phi : A^+ \to B^+$, $L \in B^+\mathcal{C}$ implies $\phi^{-1}(L) \in A^+\mathcal{C}$;

(2) if $L \in A^+\mathcal{C}$ and $a \in A$, then $a^{-1}L = \{v \in A^+ | av \in L\}$ and $La^{-1} = \{v \in A^+ | va \in L\}$ are in $A^+\mathcal{C}$.

This notion is obtained from the notion of a positive +-variety introduced in [PW97] by omitting the condition that any $A^+\mathcal{C}$ is closed under finite union and intersection. The notion of a *-family of languages is obtained from the above definition by using * in place of + and monoid morphism in place of the semigroup morphism (as again in [PW97] for the notion of a positive *-variety).

There is a relationship of *-families of languages to a notion of reducibility considered in [B95]. For languages $L, K \subseteq A^*$, let $L \leq_{oh} K$ denote that for some words $y, z \in A^*$ and some monoid morphism $h : A^* \to A^*$ we have $L = \{x \in A^* | yh(x)z \in K\}$.

Lemma 3.1 *For any *-family of languages $\mathcal{C}$ and any alphabet A, the class $A^*\mathcal{C}$ is closed downwards under $\leq_{oh}$.*

Proof. Let $K \in A^*\mathcal{C}$ and $L \leq_{oh} K$, then $L = \{x \in A^* | yh(x)z \in K\}$. In other words, $L = h^{-1}(y^{-1}(Kz^{-1}))$, where $Kz^{-1} = \{u \in A^* | uz \in K\}$, and similarly for $y^{-1}K$. By definition of the *-family, $L \in A^*\mathcal{C}$ completing the proof.

The following evident fact will be of some use in the next section.

Lemma 3.2 *Let $\mathcal{C}$ be a *-family of languages and A, B be any alphabets of the same cardinality. Then $Leaf(A^*\mathcal{C}) = Leaf(B^*\mathcal{C})$.*

Proof. By symmetry, it suffices to check the inclusion in one direction, say $Leaf(A^*\mathcal{C}) \subseteq Leaf(B^*\mathcal{C})$. Let $K \in Leaf(A^*\mathcal{C})$, then $K = M^{-1}(L)$ for an $L \in A^*\mathcal{C}$ and a suitable machine M. Let $\phi : A \to B$ be a one-one correspondence between A and B, and $\phi_1 : B^* \to A^*$ be the monoid morphism induced by ϕ^{-1}. Then $L_1 = \phi_1^{-1}(L) \in B^*\mathcal{C}$ and $K = M_1^{-1}(L_1)$, where M_1 is a machine behaving just as M with the only difference that it prints $\phi(a)$ whenever M prints a. Hence, $K \in Leaf(B^*\mathcal{C})$ completing the proof.

From results in [PW97] we easily deduce the following facts about classes of hierarchies introduced in Section 2.

Lemma 3.3 *(i) For any $n > 0$, $\{A^+\mathcal{L}_n\}_A$ and $\{A^+\mathcal{B}_n\}_A$ are positive +-varieties, while $\{A^*\mathcal{L}_n\}_A$ is a positive *-variety.*

*(ii) For any typed Boolean term t, $\{t(A^+SH)\}_A$ and $\{t(A^+BH)\}_A$ are +-families of languages while $\{t(A^*SH)\}_A$ is a *-family of languages.*

*(iii) For any $\alpha < \varepsilon$, $\{S_\alpha(A^+SH)\}_A$ and $\{S_\alpha(A^+BH)\}_A$ are +-families of languages while $\{S_\alpha(A^+SH)\}_A$ is a *-family of languages.*

Proof. (i) is proved in [PW97] and plays a principal role for our paper.

(ii) Let $\phi : A^+ \to B^+$ be a semigroup morphism and let $L \in t(B^+\mathcal{L})$. By (i), the preimage map ϕ^{-1} satisfies conditions of Lemma 2.2.(ii). Hence, $\phi^{-1}(L) \in t(A^+\mathcal{L})$. Property (2) from definition of the family of languages, as well as the remaining assertions from (ii), are checked in the same way.

(iii) follows from Lemma 2.6.(iii) and the fact that the operations $\phi^{-1}(L)$, $a^{-1}L$ and La^{-1} respect (according to (i)) all the classes $A^+\mathcal{L}_n, A^+\mathcal{B}_n, A^*\mathcal{L}_n$. This completes the proof.

4 Typed Boolean Hierarchy over SH

In this section we relate some hierarchies introduced in Section 2 via the leaf language approach. First we consider languages from classes of the typed boolean hierarchy over SH as leaf languages.

Theorem 4.1 *For any typed boolean term t, $\cup_A Leaf(t(A^*SH)) = t(PH) = \cup_A Leaf(t(A^+SH))$.*

Proof. By Lemma 2.5, it suffices to prove the equality $\cup_A Leaf(t(A^*SH)) = t(PH)$. First let us note that the result from [BV98] cited in the Introduction is exactly formulated as the equality $\cup_A Leaf(A^*\mathcal{L}_n) = \Sigma_n^p$, for any $n > 0$.

Now let us check the inclusion $\cup_A Leaf(t(A^*SH)) \subseteq t(PH)$. Let $K \in Leaf(t(A^*SH))$, then $K = M^{-1}(L)$ for some polynomially bounded nondeterministic Turing machine M and some $L \in t(A^*SH)$. The map $M^{-1} : P(A^*) \to P(X^*)$ is a homomorphism of boolean algebas satisfying (by the Theorem of Burtschick and Vollmer) the inclusions $M^{-1}(A^*\mathcal{L}_n) \subseteq \Sigma_n^p$. By Lemma 2.1.(ii), $K = M^{-1}(L) \in t(PH)$, as desired.

For the converse inclusion, choose any K in $t(PH)$ and let $t = t(x_0, \dots, x_k)$, where x_j are typed variables. Then $K = t(K_0, \dots, K_k)$ for some $K_0, \dots, K_k \subseteq X^*$ such that $K_j \in \Sigma_{n+1}^p$ whenever x_j is of type n. By the Theorem of Burtschik and Vollmer, there exist alphabets $A_0, \dots, A_k$ and languages $L_j \subseteq A_j^*$ such that $K_j \in Leaf(L_j)$ and $L_j \in \mathcal{L}_{n+1}$ whenever x_j is of type n. By Lemma 3.2, the alphabets $A_0, \dots, A_k$ may be without loss of generality assumed pairwise disjoint. Let $A = A_0 \cup \dots \cup A_k$. Now it suffices to show that $K \in Leaf(t(A^*SH))$.

Let $M_0, \dots, M_k$ be nondeterministic polynomyal time Turing machines satisfying $K_j = M_j^{-1}(L_j)$. Consider the nondeterministic polynomial time Turing machine M which behaves as follows: on input $x \in X^*$, it branches nondeterministically into $k+1$ computation paths, and on the j-th (from left to right) path just mimicks completely the behavior of the machine M_j. Note that the leaf string $M(x)$ will be the concatenation of the leaf strings $M_j(x)$, i.e. $M(x) = M_0(x) \cdots M_k(x)$.

For any $j \leq k$, let $\phi_j : A^* \to A_j^*$ be the morphism erasing all letters not in A_j. Then, by Lemma 3.3, $\phi_j^{-1}(L_j) \in \mathcal{L}_{n+1}$ whenever x_j is of type n. Hence, the language $P = t(\phi_0^{-1}(L_0), \dots, \phi_k^{-1}(L_k))$ is in $t(A^*SH)$. Hence, it suffices to check that $K = M^{-1}(P) = t(M^{-1}(\phi_0^{-1}(L_0)), \dots, M^{-1}(\phi_k^{-1}(L_k))$. But $\phi_j(M(x)) = M_j(x)$, hence $M^{-1}(\phi_j^{-1}(L_j)) = M_j^{-1}(L_j)$ and the desired equality follows immediately from the equality $K = t(K_0, \dots, K_k) = t(M_0^{-1}(L_0), \dots, M_k^{-1}(L_k))$. This concludes the proof of the theorem.

In [BV98] the result was proved in a more exact form than it was formulated above. They proved also that for any $n > 0$ there is an alphabet A and a language $L \in A^+\mathcal{L}_n$ such that $Leaf(L) = \Sigma_n^p$. This is also generalizable to the typed boolean hierarchy.

Theorem 4.2 *For any $t \in T$ there exist an alphabet A and a language $L \in t(A^+SH)$ such that $Leaf(L) = t(PH)$.*

Proof. By Lemma 2.3, there exists a language $K \subseteq X^*$ polynomially m-complete in $t(PH)$. By Theorem 4.1, there exist an alphabet A and a language $L \in t(A^+SH)$ such that $K \in Leaf(L) \subseteq t(PH)$. It is well–known [BCS92] that the class $Leaf(L)$ is closed downwards under the polynomial m-reducibility. Hence, $t(PH) \subseteq Leaf(L)$ completing the proof.

5 Typed Boolean Hieararchy over BH

The next result is an analog of Theorem 4.1 for the Brzozowski hierarchy.

Theorem 5.1 *For any $t \in T$, $t(PH) = \cup_A Leaf(t(A^+BH))$.*

Proof. The inclusion from left to right follows from Theorem 4.1 and Lemma 2.1. For the converse inclusion, one straightforwardly checks that the proof in [BV98] of the inclusion $Leaf(A^*\mathcal{L}_{n+1}) \subseteq \Sigma^p_{n+1}$ is easily modified to the proof of inclusion $Leaf(A^*\mathcal{B}_{n+1}) \subseteq \Sigma^p_{n+1}$ (see e.g. [BKS99] for the proof for the first level). From the last inclusion the desired inclusion $\cup_A Leaf(t(A^+BH)) \subseteq t(PH)$ follows just in the same way as above for the Straubing hierarchy. This completes the proof.

The relationships between automata–theoretic hierarchies and the complexity–theoretic ones established in Theorems 4.1 and 5.1 look dependent on the alphabet. It seems that for the Straubing case the dependence is really essential (though we have yet no formal proof of this). Our next result shows that for the Brzozowski case one can get an alphabet–independent version of Theorem 5.1.

Theorem 5.2 *For any $t \in T$ and any alphabet A having at least two symbols, $Leaf(t(A^+BH)) = t(PH)$.*

The idea of proof is evident: to code symbols of a bigger alphabet by sequences of symbols of a smaller alphabet using the presence of the successor function in the signature σ'_A from Section 2. In the next few lemmas we collect observations needed for realization of this idea. For technical convenience, we will assume in these lemmas that the alphabet A is a finite subset of ω.

Define a function $f : \omega \to \{0,1\}^+$ by $f(n) = 01\dots10$, where the sequence of 1's is of length $n+1$. With any alphabet $A \subseteq \omega$ we associate a semigroup morphism $f = f_A : A^+ \to \{0,1\}^+$ induced by the restriction of f to A. E.g., for $A = \{0,1,2\}$ and $w = 0212$ we get $f(w) = 01001110011001110$. In general, if $w = a_0 \cdots a_k$ for $a_j \in A$ then $f(w)$ is the superposition $f(a_0) \cdots f(a_k)$. For $i \leq k$, let i' denote the position of the first letter of $f(a_j)$ (this letter is of course 0) in the word $f(w)$. As usual, the length of a word v is denoted by $|v|$, and for $i \leq |v|$ the i-th letter in v is denoted by v_i. The following assertion is evident.

Lemma 5.3 *(i) For all $i, j \leq |w|$, $i < j$ iff $i' < j'$.*

(ii) For any $l \leq |f(w)|$, $l \in \{i' | i \leq |w|\}$ iff $(f(w))_l = 0$ and $(f(w))_{l+1} = 1$.

Let $\sigma_A = \{<, Q_a\}_{a \in A}$ and $\sigma' = \sigma'_{\{0,1\}} = \{<, Q_0, Q_1, \bot, \top, s\}$ be the signatures discussed in Section 2. Relate to any formula ϕ of σ_A a formula ϕ' of σ' by the following induction:

if ϕ is $x = y$ or $x < y$ then ϕ' is ϕ;
if ϕ is $Q_a(x)$ then ϕ' is $Q_0(x) \wedge Q_1(s(x)) \wedge \dots \wedge Q_1(s^{a+1}(x)) \wedge Q_0(s^{a+2}(x))$;
if ϕ is $\phi_1 \wedge \phi_2$ then ϕ' is $\phi'_1 \wedge \phi'_2$;
if ϕ is $\neg\phi_1$ then ϕ' is $\neg\phi'_1$;
if ϕ is $\forall x \phi_1$ then ϕ' is $\forall x (Q_0(x) \wedge Q_1(s(x)) \to \phi'_1)$.

The other connectives $\vee, \to, \exists$ are expressed through $\wedge, \neg, \forall$ in the usual way. Below we discuss structures of the form $\mathbf{w}$ related to words w as specified in Section 2. When we want to stress the signature in which we consider the structure, we use notation like $(\mathbf{w}; \sigma_A)$. For $i_0, \dots, i_k \leq |w|$, $(\mathbf{w}; \sigma_A, i_0, \dots, i_k)$ denotes the usual enrichment of the structure $(\mathbf{w}; \sigma_A)$ by constants i_j.

Lemma 5.4 *(i) Let $\phi = \phi(x_0, \ldots, x_k)$ be a formula of signature σ_A, $w \in A^+$ and $i_0, \ldots, i_k \leq |w|$. Then we have $(\mathbf{w}; \sigma_a, i_0, \ldots, i_k) \models \phi(x_0, \ldots, x_k)$ iff $(\mathbf{f(w)}; \sigma', i'_0, \ldots, i'_k) \models \phi'(x_0, \ldots, x_k)$.*

(ii) For any $w \in A^+$ and any sentence ϕ of σ_A, we have $(\mathbf{w}; \sigma_A) \models \phi$ iff $(\mathbf{f(w)}; \sigma') \models \phi'$.

Proof. (i) is checked by a strightforward induction on ϕ; one should take into account Lemma 5.3 and the definition of f.

(ii) is a particular case of (i). This completes the proof.

Let Φ be the set of all sentences of signature σ_A. For $\phi, \psi \in \Phi$, let $L_\phi = \{w \in A^+ | \mathbf{w} \models \phi\}$ and let $\phi \equiv \psi$ iff $L_\phi = L_\psi$. Let B be the quotient of the structure $(\Phi; \wedge, \vee, \neg, true, false)$ under the congruence relation $\equiv$. As is well–known, B is a boolean algebra; abusing (and simplifying) notation, we denote elements of B as corresponding sentences. For $n < \omega$, let L_n be the subset of B corresponding to the set of Σ_{n+1}-sentences; then $L = \{L_n\}$ is a base in B. When we want to stress the alphabet, we denote this base by L_A.

The construction from preceding paragraph applies also to sentences of σ'_A in place of σ_A (as well as to sentences of any other signature). We denote the corresponding base as $L' = L'_A$.

Lemma 5.5 *Let $t \in T$ and A be any alphabet, $A \subseteq \omega$.*

(i) $L \in t(A^+SH)$ iff $L = L_\phi$ for some $\phi \in t(L_A)$, and similarly for A^+BH and for the base L'_A.

(ii) If $\phi \in t(L_A)$ then $\phi' \in t(L'_{\{0,1\}})$.

Proof. (i) By Theorem of McNaughton and Papert cited in Section 2, the map $\phi \mapsto L_\phi$ induces an isomorphism between b.a.'s B and $\{L \subseteq A^+ | L$ is star-free$\}$, and $\phi \in L_n$ iff $L_\phi \in A^+\mathcal{L}_{n+1}$. By Lemma 2.1.(ii), the assertion (i) follows. The same proof works also for the Brzozowski case.

(ii) By definition of the map $\phi \mapsto \phi'$, it induces a homomorphism $h : B \to B'$ satisfying $h(L_n) \subseteq L'_n$. By Lemma 2.1.(ii), the assertion follows. This completes the proof.

Now we are able to give a proof of the theorem.

Proof of Theorem 5.2. The inclusion $Leaf(t(A^+BH)) \subseteq t(PH)$ was already noticed above. For the converse, it clearly suffices to check that $t(PH) \subseteq Leaf(t(\{0,1\}^+BH))$. By Theorem 4.1 and Lemma 3.2, it suffices to check the inclusion $Leaf(t(A^+SH)) \subseteq Leaf(t(\{0,1\}^+BH))$, for any alphabet $A \subseteq \omega$. So assume that $K \in Leaf(t(A^+SH))$. Then $K = M^{-1}(L)$ for a suitable machine M and for some $L \in t(A^+SH)$. By Lemma 5.5, $L = L_\phi$ for some $\phi \in t(L_A)$. Again by Lemma 5.5, $\phi' \in t(L'_{\{0,1\}})$ and $L_{\phi'} \in t(\{0,1\}^+BH)$. Hence, it remains to check that $M^{-1}(L_\phi) \in Leaf(L_{\phi'})$. In other words, it remains to find a suitable machine M_1 satisfying $M^{-1}(L_\phi) = M^{-1}(L_{\phi'})$.

Let M_1 on the input x behaves exactly as M on input x, with the only exception that at the end of any computation path, whenever M prints $a \in A$, the machine M_1 "prints" the word $f(a) \in \{0,1\}^+$ (more exactly, M_1 branches nondeterministically to $|f(a)|$ paths and on these paths prints subsequently bits of the word $f(a)$). In this way we get $M_1(x) = f(M(x))$. By Lemma 5.4, $M(x) \models \phi$ iff $M_1(x) \models \phi'$. In other words, $x \in M^{-1}(L_\phi)$ iff $x \in M_1^{-1}(L_{\phi'})$, for any $x \in X^*$. Hence, $M^{-1}(L_\phi) = M^{-1}(L_{\phi'})$ completing the proof of the theorem.

Remark. Our proof makes no use of the constant symbols in the signature σ', hence Theorem 5.2 is true also for the version of the Brzozowski hierarchy obtained by using the signature $\{<, Q_0, Q_1, s\}$ in place of σ'.

As in Section 4, we automatically get

Corollary 5.6 *For any $t \in T$ there exists a language $L \in t((\{0,1\}^*BH))$ satisfying $Leaf(L) = t(PH)$.*

6 Fine Hierarchy

In two preceeding sections we have described the situation with the typed boolean hierarchy rather comprehensively. Here we discuss similar questions for the fine hierarchy. The general question is to understand the relationships between classes $Leaf(S_\alpha(A^+SH))$, $Leaf(S_\alpha(A^+BH))$ and $S_\alpha(PH)$, for any $\alpha < \varepsilon_0$. We give some partial information relevant to this question.

Proposition 6.1 *For any $\alpha < \varepsilon_0$ and any alphabet A, $Leaf(S_\alpha(A^+SH)) \subseteq Leaf(S_\alpha(A^+BH)) \subseteq S_\alpha(PH)$.*

Proof. Obviously, $A^+\mathcal{L}_n \subseteq A^+\mathcal{B}_n$ for any $n > 0$. By Lemma 3.1, this implies $S_\alpha(A^+SH) \subseteq S_\alpha(A^+BH)$, for any $\alpha < \varepsilon_0$. This proves the left inclusion of the theorem.

Now let $K \in Leaf(S_\alpha(A^+BH))$, i.e. $K = M^{-1}(L)$ for a suitable machine M and some $L \in S_\alpha(A^+BH)$. As noted in the proof of Theorem 5.1, M^{-1} maps $A^+\mathcal{B}_n$ into $A^+\Sigma_n^p$, for any $n > 0$. By Lemma 2.6, M^{-1} maps also $S_\alpha(A^+BH)$ into $S_\alpha(PH)$. Hence, $K = M^{-1}(L) \in S_\alpha(PH)$ completing the proof.

Unfortunately, till now we were unable to prove the remaining inclusion $S_\alpha(PH) \subseteq Leaf(\cup_A A^+S_\alpha(SH))$. The problem is that the fine hierarchy is defined in terms of values of boolean terms, values of whose variables satisfy some constraints (see Section 2). It is not clear how to preserve those constraints under transfer to another hierarchy.

The next result, which is reminscent of Theorem 5.2, shows that for the Brzozowski case we again may "reduce" alphabets.

Proposition 6.2 *For any $\alpha < \varepsilon_0$ and any alphabet A, $Leaf(S_\alpha(A^+SH)) \subseteq Leaf(S_\alpha(\{0,1\}^+BH))$.*

Proof is parallel to the proof of Theorem 5.2. We again assume $A \subseteq \omega$ and work with f, B, B' and other objects from the previous section. Similar to Lemma 5.5, we first prove that $L \in S_\alpha(A^+SH)$ iff $L = L_\phi$ for some $\phi \in S_\alpha(L_A)$, that the same holds for the BH and σ'_A, and that $\phi \in S_\alpha(L_A)$ implies $\phi' \in S_\alpha(L'_{\{0,1\}})$.

Now let $K \in Leaf(S_\alpha(A^+SH))$, then $K = M^{-1}(L)$ for a suitable machine M and for some $L \in S_\alpha(A^+SH)$. Choose $\phi \in S_\alpha(L_A)$ satisfying $L = L_\phi$. Then $\phi' \in S_\alpha(L'_{\{0,1\}})$ and a fortiori $L_{\phi'} \in S_\alpha(\{0,1\}^+BH)$. The machine M_1 from the proof of Theorem 5.2 again satisfies $M^{-1}(L_\phi) = M_1^{-1}(L_{\phi'})$. This completes the proof.

7 Examples and Discussion

The typed boolean hierarchy and the fine hierarchy are rather abstract and rich structures. In this section we formulate and discuss some interesting particular cases.

Let again $B = (B; \cup, \cap, \bar{}, 0, 1)$ be a b.a. Define an operation of addition of classes $X, Y \subseteq B$ by the equality $X + Y = \{x \triangle y | x \in X, y \in Y\}$, where $x \triangle y$ is the symmetric difference of x and y. This operation is induced by the operation of addition modulo 2, hence it is associative and commutative and we may a fortiori freely use expressions like $X_0 + \cdots + X_n$.

Let L be a sublattice of $(B; \cup, \cap, 0, 1)$. For any $k > 0$, let $D_k = L + \cdots + L$ (k summonds in the righthand side). In [KSW86] it was shown that the sequence $\{D_k(L)\}$ coincides with the well–known boolean (or difference) hierarchy over L.

Taking now NP in place of L, one gets the boolean hiearachy over NP, a rather popular object in complexity theory introduced in [WW85]. More generally, one could consider the boolean hierarchy $\{D_k(\Sigma^p_n)\}$ over the n-th level of the polynomial hierarchy. It is natural to ask: is there a natural description of these classes in terms of leaf languages? To answer the question, one have only to note that for any base L in B the boolean hierarchy over any class L_n is a fragment of the typed boolean hierarchy (as well as of the fine hierarchy), see [Se94]. E.g., we could consider the boolean hierarchy over any class $\mathcal{B}_n = \{0,1\}^+\mathcal{B}_n$ of the Brzozowski hierarchy and immediately get

Corollary 7.1 *For all $n, k > 0$, $Leaf(D_k(\mathcal{B}_n)) = D_k(\Sigma^p_n)$.*

For the case of the boolean hierarchy over NP and the boolean hierarchy over SH the corresponding result was earlier obtained in [SW98].

Another interesting example is the plus–hierarchy introduced implicitly in [Se94, Se95] and explicitely in [Se94a, Se99]. The levels of the plus–hierarchy over any base L are obtained when one applies the operation $+$ introduced above to the levels L_n, for all $n < \omega$. Any finite nonempty string $\sigma = (n_0, \ldots, n_k)$ of natural numbers satisfying $n_0 \geq \cdots \geq n_k$ defines the level $P_\sigma(L) = L_{n_0} + \cdots + L_{n_k}$ of the plus–hierarchy over L. One easily checks that in this way we get actually all the levels of the plus–hierarchy, that the finite sequences specified above are ordered lexicographically with the order type ω^ω, and that $P_\sigma \cup co(P_\sigma) \subseteq P_\tau$ whenever $\sigma < \tau$.

Taking PH in place of L, we get the plus–hierarchy over PH. Though not so important as the boolean hierarchy over NP, this hierarchy seems also potentially useful (e.g., in [RW98, HHH98] it was implicitly used to estimate exactly the collapse of the PH from the collapse of the boolean hierarchy over NP). Hence, one may like to look at the description of the levels of this hierarchy in terms of leaf languages. Again, such descriptions are contained in the results of Sections 4 and 5. Note that the classes P_σ are again among the classes of the typed boolean hierarchy (as well as of the fine hierarchy) over L, see [Se94]. Taking now e.g. the plus–hierarchy over $\{0,1\}^+BH$, we get

Corollary 7.2 *For any sequence σ as above, $Leaf(P_\sigma(\{0,1\}^+BH)) = P_\sigma(PH)$.*

What is the aim of proving results of this type? In our opinion, the existence of nontrivial connections between automata–theoretic and complexity–theoretic hierarchies is interesting in its own right and is somewhat unexpected. Maybe, some time results of this type may be even of use. E.g., assume for a moment that the Brzozowski hierarchy collapses. By the Theorem of Burtschik and Vollmer, the polynomial hierarchy would then collapse too. This is of course unlikely, hence the Brzozowski hierarchy should not collapse. And this is actually a proven fact of the automata theory [BK78]. From [Ka85] we know that the boolean hierarchy over any Σ_n^p does not collapse, provided that PH does not collapse. Hence, the boolean hierarchy over any level of BH also should not collapse. And this was indeed proved in [Sh98,SS00], though the proofs are rather involved.

From [Se94, Se94a, Se99] we know that the plus–hierarchy over PH does not collapse, provided that the PH does not collapse. Hence, the plus–hierarchy over BH should also not collapse. This result is not yet published but hopefully we have a proof of this fact (as well as of the fact that the fine hierarchy over BH does not collapse). But this is another story.

Acknowledgements

This work was started at RWTH Aachen in spring of 1999 and finished 2 years later at the University of Würzburg. I am grateful to Wolfgang Thomas and Klaus Wagner for hospitality and for making those visits possible. I thank also both of them, as well as Heribert Vollmer, for helpfull discussions.

References

[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.

[B95] B. Borchert. On the acceptance power of regular languages. *Theoret. Comp. Sci*, 148 (1995), 207—225.

[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.

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

[BK78] J. A. Brzozowski and R Knast. The dot-depth hierarchy of star–free languages is infinite. *J. Comp. Systems Sci.*, 16 (1978), 37—55.

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

[HHH98] E. Hemaspaandra, L. Hemaspaandra and H. Hempel. What's up with downward collapse: Using the easy-hard technique to link boolean and polynomial hierarchy collapses. Compl. Theory Column 21, *ACM–SIGACT Newsletter*, 29(3, 1998, 10—22.)

[HL+93] 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.

[HVW95] U. Hertrampf, H. Vollmer and K. W. Wagner. On the power of number–theoretic operations with respect to counting, *Proc. 10th Structure in Complexity Theory*, 1995, 299—314.

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

[JMT96] B. Jenner, P. McKenzie and D. Therien. Logspace and logtime leaf languages, *Information and Computation*, 129 (1996), 21—33.

[KM67] K. Kuratowski and A. Mostowski. *Set Theory*. North Holland, 1967.

[KSW86] J. Köbler, U. Shöning and K.W. Wagner: The difference and truth-table hierarchies for NP. Dep. of Informatics, Koblenz. Preprint 7 (1986).

[MP71] R. McNaughton and S. Papert. *Counter–free automata*. MIT Press, Cambridge, Massachusets, 1971.

[PP86] D. Perrin and J.-E. Pin. First order logic and star–free sets. *J. Comp. and Syst. Sci.*, 32 (1986), 393—406.

[PW97] J.-E. Pin and P. Weil. Polynomial closure and unambiguous product. *Theory of Computing Systems*, 30 (1997), 383—422.

[RW98] S. Reith and K. W. Wagner. On boolean lowness and boolean highness. In: Proc. 4-th Ann. Int. Computing and Combinatorics Conf., *Lecture Notes in Computer Science*, v. 1449. Springer: Berlin 1998, p. 147—156.

[Se94] V. L. Selivanov. Two refinements of the polynomial hierarchy. In: Proc. of Symposium on Theor. Aspects of Computer Science STACS–94, *Lecture Notes in Computer Science*, v. 775. Springer: Berlin 1994, p. 439—448.

[Se94] V. L. Selivanov. Refining the polynomial hierarchy, Preprint No 9, the University of Heidelberg, Chair of Mathematical Logic, 1994, 20 p.

[Se95] V. L. Selivanov. Fine hierarchies and Boolean terms. *Journal of Symbolic Logic*, 60 (1995), p. 289—317.

[Se99] V. L. Selivanov. Refining the polynomial hierarchy. *Algebra and Logic*, 38 (1999), 456—475 (Russian, there is an English translation).

[Se01] V. L. Selivanov. A logical approach to decidability of hierarchies of regular star-free languages. In: Proc. of 18-th Int. Symposium on Theor. Aspects of Computer Science STACS-2001 in Dresden, Germany, *Lecture Notes in Computer Science*, v. 2010. Berlin: Springer, 2001, 539—550

[SS00] V. L. Selivanov and A. G. Shukin. On hierarchies of regular star-free languages (in Russian). Preprint 69 of A.P. Ershov Institute of Informatics Systems, 2000, 28 p.

[Sh98] A. G. Shukin. Difference hierarchies of regular languages. *Comp. Systems*, Novosibirsk, 161 (1998), 141—155 (in Russian).

[SW98] H. Schmitz and K. W Wagner. The Boolean hierarchy over level 1/2 of the Sraubing–Therien hierarchy, Technical Report 201, Inst. für Informatik, Univ. Würzburg (available at http://www.informatik.uni-wuerzburg.de).

[T82] W. Thomas. Classifying regular events in symbolic logic. *J. Comp. and Syst. Sci.*,25 (1982), 360—376.

[V93] N. K. Vereshchagin. Relativizable and non-relativizable theorems in the polynomial theory of algorithms. *Izvestiya Rossiiskoi Akademii Nauk*, 57 (1993), 51—90 (in Russian).

[WW85] G. Wechsung and K. Wagner. On the Boolean closure of NP. In: *Proceedings of the 1985 Int. Conf. on Fundamentals of Computation theory*, v.199 of *Lecture Notes in Computer Science*, p.485—493. Springer–Verlag, 1985.