Efficient Algorithms for Membership in Boolean Hierarchies of Regular Languages

Christian Glasser, Heinz Schmitz, Victor Selivanov

Symposium on Theoretical Aspects of Computer Science 2008 (Bordeaux), pp. 337-348
www.stacs-conf.org

EFFICIENT ALGORITHMS FOR MEMBERSHIP IN BOOLEAN HIERARCHIES OF REGULAR LANGUAGES

C. GLASSER [1], H. SCHMITZ [2], AND V. SELIVANOV [3]

[1] Universität Würzburg, Germany.
E-mail address: glasser@informatik.uni-wuerzburg.de

[2] Fachhochschule Trier, Germany.
E-mail address: schmitz@informatik.fh-trier.de

[3] A. P. Ershov Institute of Informatics Systems, Russia.
E-mail address: vseliv@nspu.ru

Abstract. The purpose of this paper is to provide *efficient* algorithms that decide membership for classes of several Boolean hierarchies for which efficiency (or even decidability) were previously not known. We develop new forbidden-chain characterizations for the single levels of these hierarchies and obtain the following results:

- The classes of the Boolean hierarchy over level Σ_1 of the dot-depth hierarchy are decidable in NL (previously only the decidability was known). The same remains true if predicates mod d for fixed d are allowed.
- If modular predicates for arbitrary d are allowed, then the classes of the Boolean hierarchy over level Σ_1 are decidable.
- For the restricted case of a two-letter alphabet, the classes of the Boolean hierarchy over level Σ_2 of the Straubing-Thérien hierarchy are decidable in NL. This is the first decidability result for this hierarchy.
- The membership problems for all mentioned Boolean-hierarchy classes are logspace many-one hard for NL.
- The membership problems for quasi-aperiodic languages and for d-quasi-aperiodic languages are logspace many-one complete for PSPACE.

Introduction

The study of decidability and complexity questions for classes of regular languages is a central research topic in automata theory. Its importance stems from the fact that finite automata are fundamental to many branches of computer science, e.g., databases, operating systems, verification, hardware and software design.

There are many examples for decidable classes of regular languages (e.g., locally testable languages), while the decidability of other classes is still a challenging open question (e.g.,

Key words and phrases: automata and formal languages, computational complexity, dot-depth hierarchy, Boolean hierarchy, decidability, efficient algorithms.

This work was done during a stay of the third author at the University of Würzburg, supported by DFG Mercator program and by RFBR grant 07-01-00543a.

dot-depth two, generalized star-height). Moreover, among the decidable classes there is a broad range of complexity results. For some of them, e.g., the class of piecewise testable languages, efficient algorithms are known that work in nondeterministic logarithmic space (NL) and hence in polynomial time. For other classes, a membership test needs more resources, e.g., deciding the membership in the class of star-free languages is PSPACE-complete.

The purpose of this paper is to provide *efficient* algorithms that decide membership for classes of several Boolean hierarchies for which efficiency (or even decidability) were not previously known. Many of the known efficient decidability results for classes of regular languages are based on so-called forbidden-pattern characterizations. Here a language belongs to a class of regular languages if and only if its deterministic finite automaton does *not* have a certain subgraph (the forbidden pattern) in its transition graph. Usually, such a condition can be checked efficiently, e.g., in nondeterministic logarithmic space [Ste85a, CPP93, GS00a, GS00b].

However, for the Boolean hierarchies considered in this paper, the design of efficient algorithm is more involved, since here no forbidden-pattern characterizations are known. More precisely, wherever decidability is known, it is obtained from a characterization of the corresponding class in terms of *forbidden* alternating chains of word extensions. Though the latter also is a forbidden property, the known characterizations are not efficiently checkable in general. (Exceptions are the special 'local' cases $\Sigma_1^{\varrho}(n)$ and $\mathcal{C}_k^1(n)$ where decidability in NL is known [SW98, Sch01].) To overcome these difficulties, we first develop alternative forbidden-chain characterizations (they essentially ask only for certain reachability conditions in transition graphs). From our new characterizations we obtain efficient algorithms for membership tests in NL. For two of the considered Boolean hierarchies, these are the first decidable characterizations at all, i.e., for the classes $\Sigma_2^{\varrho}(n)$ for the alphabet $A = \{a, b\}$, and for the classes $\Sigma_1^{\tau}(n)$).

Definitions. We sketch the definitions of the Boolean hierarchies considered in this paper. Σ_1^{ϱ} denotes the class of languages definable by first-order Σ_1-sentences over the signature $\varrho = \{\leq, Q_a, \ldots\}$ where for every letter $a \in A$, $Q_a(i)$ is true if and only if the letter a appears at the i-th position in the word. Σ_1^{ϱ} equals level 1/2 of the Straubing-Thérien hierarchy (STH for short) [Str81, Thé81, Str85, PP86]. Σ_2^{ϱ} is the class of languages definable by similar first-order Σ_2-sentences; this class equals level 3/2 of the Straubing-Thérien hierarchy. Let σ be the signature obtained from ϱ by adding constants for the minimum and maximum positions in words and adding functions that compute the successor and the predecessor of positions. Σ_1^{σ} denotes the class of languages definable by first-order Σ_1-sentences of the signature σ; this class equals level 1/2 of the dot-depth hierarchy (DDH for short) [CB71, Tho82]. Let τ_d be the signature obtained from σ by adding the unary predicates $P_d^0, \ldots, P_d^{d-1}$ where $P_d^j(i)$ is true if and only if $i \equiv j \pmod d$. Let τ be the union of all τ_d. $\Sigma_1^{\tau_{\mathrm{d}}}$ (resp., Σ_1^{τ}) is the class of languages definable by first-order Σ_1-sentences of the signature τ_d (resp., τ). $\mathcal{C}_k^d$ is the generalization of Σ_1^{ϱ} where neighborhoods of $k+1$ consecutive letters and distances modulo d are expressible (Definition 1.2). For a class $\mathcal{D}$ (in our case one of the classes Σ_1^{ϱ}, Σ_1^{σ}, $\mathcal{C}_k^d$, $\Sigma_1^{\tau_{\mathrm{d}}}$, Σ_1^{τ}, and Σ_2^{ϱ} for $|A| = 2$), the *Boolean hierarchy over* $\mathcal{D}$ is the family of classes

$$\mathcal{D}(n) \stackrel{df}{=} \{L \,\big|\, L = L_1 - (L_2 - (\ldots - L_n)) \text{ where } L_1, \ldots, L_n \in \mathcal{D} \text{ and } L_1 \supseteq L_2 \supseteq \cdots \supseteq L_n\}.$$

The Boolean hierarchies considered in this paper are illustrated in Figure 1.

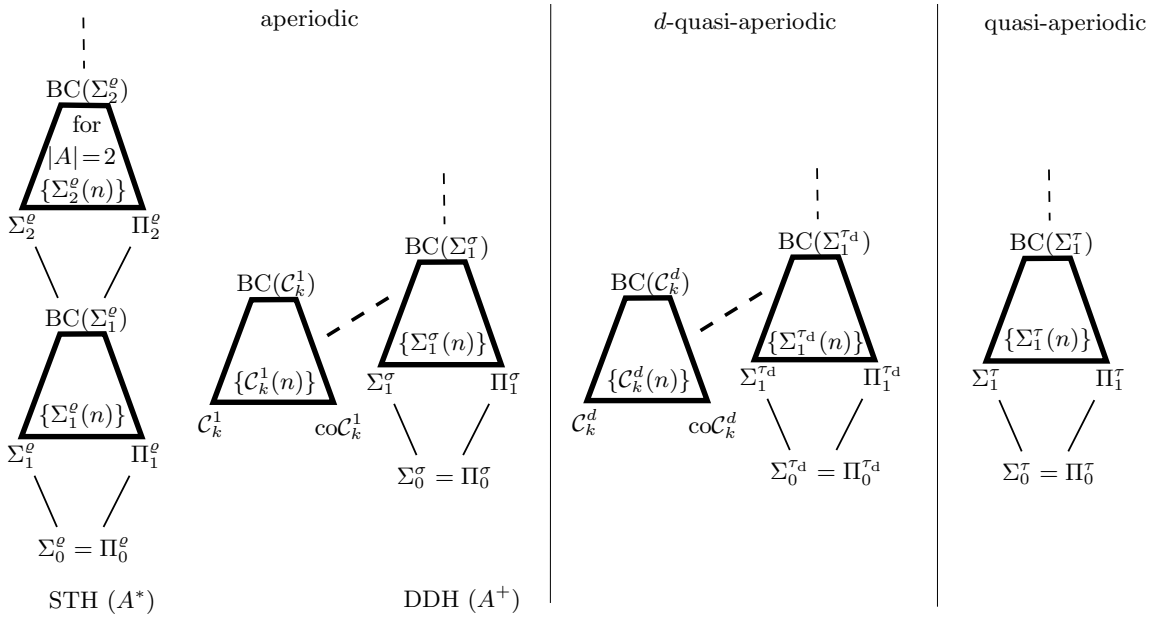

Figure 1: Boolean hierarchies considered in this paper.

Our Contribution. The paper contributes to the understanding of Boolean hierarchies of regular languages in two ways:

(1) For the classes $\Sigma_1^\sigma(n)$, $\Sigma_1^{\tau_d}(n)$, and $\Sigma_2^o(n)$ for the alphabet $A = \{a, b\}$ we prove new characterizations in terms of forbidden alternating chains. In case of $\Sigma_2^o(n)$ for the alphabet $A = \{a, b\}$, this is the first characterization of this class.

(2) For the classes $\Sigma_1^\sigma(n)$, $\mathcal{C}_k^d(n)$, $\Sigma_1^{\tau_d}(n)$, and $\Sigma_2^o(n)$ for the alphabet $A = \{a, b\}$ we construct the first efficient algorithms for testing membership in these classes. In particular, this yields the decidability of the classes $\Sigma_1^\tau(n)$, and of $\Sigma_2^o(n)$ for the alphabet $A = \{a, b\}$.

We also show that the membership problems for all mentioned Boolean-hierarchy classes are logspace many-one hard for NL. An overview of the obtained decidability and complexity results can be found in Table 1. Moreover, we prove that the membership problems for quasi-aperiodic languages and for d-quasi-aperiodic languages are logspace many-one complete for PSPACE.

Boolean hierarchies can also be seen as fine-grain measures for regular languages in terms of descriptional complexity. Note that the Boolean hierarchies considered in this paper do not collapse [Shu98, SS00, Sel04]. Moreover, all these hierarchies either are known or turn out to be decidable (see Table 1 for the attribution of these results). If in addition the Boolean closure of the base class is decidable, then we can even exactly compute the Boolean level of a given language. By known results (summarized in Theorem 1.1), one can do this exact computation of the level for the Boolean hierarchies over Σ_1^o, Σ_2^o (for alphabet $A = \{a, b\}$), $\mathcal{C}_k^1$, Σ_1^σ, and Σ_1^τ. To achieve the same for the Boolean hierarchies over $\mathcal{C}_k^d$ and $\Sigma_1^{\tau_d}$ we need the decidability of their Boolean closures which is not known.

Related Work. Due to the many characterizations of regular languages there are several approaches to attack decision problems on subclasses of regular languages: Among them there is the algebraic, the automata-theoretic, and the logical approach. In this paper

Boolean hierarchy classes	decidability	complexity
$\Sigma_1^{\varrho}(n)$	[SW98]	NL-complete [SW98]
$\mathcal{C}_k^1(n)$	[GS01a, Sel01]	NL-complete [Sch01]
$\Sigma_1^{\sigma}(n)$	[GS01a]	NL-complete [this paper]
$\mathcal{C}_k^d(n)$	[Sel04]	NL-complete [this paper]
$\Sigma_1^{\tau_{\mathrm{d}}}(n)$	[Sel04]	NL-complete [this paper]
$\Sigma_1^{\tau}(n)$	[this paper]	no efficient bound known (see Remark 4.3)
$\Sigma_2^{\varrho}(n)$ for $\lvert A\rvert = 2$	[this paper]	NL-complete [this paper]

Table 1: Overview of decidability and complexity results.

we mainly use the logical approach which has a long tradition starting with the early work of Trakhtenbrot [Tra58] and Büchi [Büc60]. Decidability questions for Boolean hierarchies over classes of concatenation hierarchies were previously studied by [SW98, Sch01, GS01a, Sel04]. Enrichments of the first-order logics related to the dot-depth hierarchy and the Straubing-Thérien hierarchy were considered in [BCST92, Str94, MPT00, Sel04, CPS06]. For more background on regular languages, starfree languages, concatenation hierarchies, and their decidability questions we refer to the survey articles [Brz76, Pin95, Pin96a, Pin96b, Yu96, PW02, Wei04].

Paper Outline. After the preliminaries, we explain the general idea of an efficient membership algorithm for the classes $\mathcal{C}_k^d(n)$ (section 2). This easy example shows how a suitable characterization of a Boolean hierarchy can be turned into an efficient membership test. The algorithms for the other Boolean hierarchies are similar, but more complicated. Section 3 provides new alternating-chain characterizations for the Boolean hierarchies over Σ_1^{σ}, $\Sigma_1^{\tau_{\mathrm{d}}}$, and Σ_2^{ϱ} for the alphabet $A = \{a, b\}$. In section 4 we exploit these characterizations and obtain efficient algorithms for testing the membership in these classes. In particular, we obtain the decidability of the classes $\Sigma_1^{\tau}(n)$ and $\Sigma_2^{\varrho}(n)$ for the alphabet $A = \{a, b\}$. Finally, section 5 provides lower bounds for the complexity of the considered decidability problems. As a consequence (with the exception of $\Sigma_1^{\tau}(n)$) the membership problems of all considered Boolean levels are logspace many-one complete for NL. In contrast, the membership problems of the general classes FO_{τ} and FO_{τ_d} are logspace many-one complete for PSPACE and hence are strictly more complex.

Detailed proofs are available in the technical report [GSS07].

1. Preliminaries

In this section we recall definitions and results that are needed later in the paper. If not stated otherwise, A denotes some finite alphabet with $|A| \geq 2$. Let A^* and A^+ be the sets of finite (resp., of finite non-empty) words over A. If not stated otherwise, variables range over the set of natural numbers. We use $[m, n]$ as abbreviation for the interval $\{m, m+1, \ldots, n\}$. For a deterministic finite automaton $M = (A, Z, \delta, s_0, F)$ (dfa for short), the number of states is denoted by $|M|$ and the accepted language is denoted by $L(M)$. Moreover, for words x and y we write $x \equiv_M y$ if and only if $\delta(s_0, x) = \delta(s_0, y)$. For a class of languages $\mathcal{C}$, $\mathrm{BC}(\mathcal{C})$ denotes the Boolean closure of $\mathcal{C}$, i.e., the closure under union, intersection, and complementation.

All hardness and completeness results in this paper are with respect to logspace many-one reductions, i.e., whenever we refer to NL-complete sets (resp., PSPACE-complete sets) then we mean sets that are logspace many-one complete for NL (resp., PSPACE).

1.1. The Logical Approach to Regular Languages

We relate to an arbitrary alphabet $A = \{a, \ldots\}$ the signatures $\varrho = \{\leq, Q_a, \ldots\}$ and $\sigma = \{\leq, Q_a, \ldots, \bot, \top, p, s\}$, where $\leq$ is a binary relation symbol, Q_a (for any $a \in A$) is a unary relation symbol, $\bot$ and $\top$ are constant symbols, and p, s are unary function symbols. A word $u = u_0 \ldots u_n \in A^+$ may be considered as a structure $\mathbf{u} = (\{0, \ldots, n\}; \leq, Q_a, \ldots)$ of signature σ, where $\leq$ has its usual meaning, $Q_a (a \in A)$ are unary predicates on $\{0, \ldots, n\}$ defined by $Q_a(i) \Leftrightarrow u_i = a$, the symbols $\bot$ and $\top$ denote the least and the greatest elements, while p and s are respectively the predecessor and successor functions on $\{0, \ldots, n\}$ (with $p(0) = 0$ and $s(n) = n$). Similarly, a word $v = v_1 \ldots v_n \in A^*$ may be considered as a structure $\mathbf{v} = (\{1, \ldots, n\}; \leq, Q_a, \ldots)$ of signature ϱ. For a sentence ϕ of σ (resp., ϱ), let $L_\phi = \{u \in A^+ | \mathbf{u} \models \phi\}$ (resp., $L_\phi = \{v \in A^* | \mathbf{v} \models \phi\}$). Sentences ϕ, ψ are treated as equivalent when $L_\phi = L_\psi$. A language is FO_σ-definable (resp., FO_ϱ-definable) if it is of the form L_ϕ, where ϕ ranges over first-order sentences of σ (resp., ϱ). We denote by Σ_k^σ (resp., Π_k^σ) the class of languages that can be defined by a sentence of σ having at most $k-1$ quantifier alternations, starting with an existential (resp., universal) quantifier. Σ_k^ϱ and Π_k^ϱ are defined analogously.

It is well-known that the class of FO_σ-definable languages (and FO_ϱ-definable languages) coincides with the class of *regular aperiodic languages* which are also known as the *star-free languages*. Moreover there is a levelwise correspondence to concatenation hierarchies: The classes Σ_k^ϱ, Π_k^ϱ, and $\mathrm{BC}(\Sigma_k^\varrho)$ coincide with the classes of the Straubing-Thérien hierarchy [PP86], while the classes Σ_k^σ, Π_k^σ, and $\mathrm{BC}(\Sigma_k^\sigma)$ coincide with the classes of the dot-depth hierarchy [Tho82].

We will consider also some enrichments of the signature σ. Namely, for any positive integer d let τ_d be the signature $\sigma \cup \{P_d^0, \ldots, P_d^{d-1}\}$, where P_d^r is the unary predicate true on the positions of a word which are equivalent to r modulo d. By FO_{τ_d}-definable language we mean any language of the form L_ϕ, where ϕ is a first-order sentence of signature τ_d. Note that signature τ_1 is essentially the same as σ because P_1^0 is the valid predicate. In contrast, for $d > 1$ the FO_{τ_d}-definable languages need not to be aperiodic. E.g., the sentence $P_2^1(\top)$ defines the language L consisting of all words of even length which is known to be non-aperiodic. We are also interested in the signature $\tau = \bigcup_d \tau_d$. Barrington et al. [BCST92, Str94] defined quasi-aperiodic languages and showed that this class coincides with the class of FO_τ-definable languages. With the same proof we obtain the equality of the class of d-quasi-aperiodic languages and the class of FO_{τ_d}-definable languages [Sel04]. It was observed in the same paper that $\Sigma_n^\tau = \bigcup_d \Sigma_n^{\tau_\mathrm{d}}$ for each $n > 0$, where Σ_n with an upper index denotes the class of regular languages defined by Σ_n-sentences of the corresponding signature in the upper index.

Theorem 1.1. *For the following classes $\mathcal{D}$ it is decidable whether a given dfa M accepts a language in $\mathcal{D}$:* $\mathrm{BC}(\Sigma_1^\varrho)$ [Sim75], $\mathrm{BC}(\Sigma_2^\varrho)$ *for* $|A| = 2$ [Str88], $\mathrm{BC}(\Sigma_1^\sigma)$ [Kna83], $\mathrm{BC}(\Sigma_1^\tau)$ [MPT00].

We do not know the decidability of $\mathrm{BC}(\Sigma_1^{\tau_\mathrm{d}})$. However, it is likely to be a generalization of Knast's proof [Kna83].

1.2. Preliminaries on the Classes $\mathcal{C}_k^d(n)$

We will also refer to 'local' versions of the BH's over Σ_1^σ and $\Sigma_1^{\tau_d}$ [Ste85a, GS01a, Sel01, Sel04]. For any $k \geq 0$ the following partial order on Σ^+ was studied in [Ste85a, GS01a, Sel01]: $u \leq_k v$, if $u = v \in A^{\leq k}$ or $u, v \in A^{>k}$, $p_k(u) = p_k(v)$, $s_k(u) = s_k(v)$, and there is a k-embedding $f : u \to v$. Here $p_k(u)$ (resp., $s_k(u)$) is the prefix (resp., suffix) of u of length k, and the k-embedding f is a monotone injective function from $\{0.\ldots, |u|-1\}$ to $\{0.\ldots, |v|-1\}$ such that $u(i)\cdots u(i+k) = v(f(i))\cdots v(f(i)+k)$ for all $i < |u| - k$. Note that $\leq_0$ is the subword relation.

Definition 1.2 ([Sel04])**.** Let $k \geq 0$ and $d > 0$.

(1) We say that a k-embedding $f : u \to v$ is a (k,d)-embedding, if $P_d^r(i)$ implies $P_d^r(f(i))$ for all $i < |u|$ and $r < d$.
(2) For all $u, v \in A^+$, let $u \leq_k^d v$ mean that $u = v \in A^{\leq k}$ or $u, v \in A^{>k}$, $p_k(u) = p_k(v)$, $s_k(u) = s_k(v)$, and there is a (k,d)-embedding $f : u \to v$.
(3) With $\mathcal{C}_k^d$ we denote the class of all upper sets in $(A^+; \leq_k^d)$.

Note that for $d = 1$ the order $\leq_k^d$ coincides with $\leq_k$. By an *alternating $\leq_k^d$-chain* of length n for a set L we mean a sequence $(x_0, \ldots, x_n)$ such that $x_0 \leq_k^d \cdots \leq_k^d x_n$ and $x_i \in L \Leftrightarrow x_{i+1} \notin L$ for every $i < n$. The chain is called *1-alternating* if $x_0 \in L$, otherwise it is called *0-alternating*.

Proposition 1.3 ([GS01a, Sel01, Sel04])**.** *For all $L \subseteq A^+$ and $n \geq 1$, $L \in \mathcal{C}_k^d(n)$ if and only if L has no 1-alternating chain of length n in $(A^+; \leq_k^d)$.*

Moreover, $(A^+; \leq_k^d)$ is a well partial order, $\Sigma_1^{\tau_d} = \bigcup_k \mathcal{C}_k^d$, and $\Sigma_1^\tau = \bigcup_{k,d} \mathcal{C}_k^d$ [GS01a, Sel01, Sel04].

Theorem 1.4 ([Ste85a])**.** *It is decidable whether a given dfa accepts a language in* $\mathrm{BC}(\mathcal{C}_k^1)$.

For $d > 1$ it is not known whether $\mathrm{BC}(\mathcal{C}_k^d)$ is decidable. However, we expect that this can be shown by generalizing the proof in [Ste85a].

2. Efficient Algorithms for $\mathcal{C}_k^d(n)$

The main objective of this paper is the design of *efficient* algorithms deciding membership for particular Boolean hierarchies. For this, two things are needed: first, we need to prove suitable characterizations for the single levels of these hierarchies. This gives us certain criteria that can be used for testing membership. Second, we need to construct algorithms that efficiently apply these criteria. If both steps are successful, then we obtain an efficient membership test.

Based on known ideas for membership tests for $\mathcal{C}_0^1(n)$ [SW98][1] and $\mathcal{C}_k^1(n)$ [Sch01], in this section we explain the construction of a nondeterministic, logarithmic-space membership algorithm for the classes $\mathcal{C}_k^d(n)$. This is the first efficient membership test for this general case. Our explanation has an exemplary character, since it shows how a suitable characterization of a Boolean hierarchy can be turned into an efficient membership test. Our results in later sections use similar, but more complicated constructions.

[1]For all n, the classes $\mathcal{C}_0^1(n)$ and $\Sigma_1^o(n)$ coincide up to the empty word, i.e., $\mathcal{C}_0^1(n) = \{L \cap A^+ \mid L \in \Sigma_1^o(n)\}$.

We start with the easiest case $k = 0$ and $d = 1$, i.e., with the classes $\mathcal{C}_0^1(n)$. By Proposition 1.3,

$$L \notin \mathcal{C}_0^1(n) \quad \Leftrightarrow \quad L \text{ has a 1-alternating } \leq_0\text{-chain of length } n. \tag{2.1}$$

We argue that for a given L, represented by a finite automaton M, the condition on the right-hand side can be verified in nondeterministic logarithmic space. So we have to test whether there exists a chain $w_0 \leq_0 \cdots \leq_0 w_n$ such that $w_i \in L$ if and only if i is even. This is done by the following algorithm.

```
0  // On input of a deterministic, finite automaton M = (A, Z, δ, z_0, F)
      the algorithm tests whether L(M) ∈ C_0^1(n).
1  let s_0 = ··· = s_n = z_0
2  do
3     nondeterministically choose a ∈ A and j ∈ [0,n]
4     for i = j to n
5        s_i = δ(s_i, a)        // stands for the imaginary command w_i := w_i a
6     next i
7  until ∀i,[s_i ∈ F ⇔ i is even]
8  accept
```

The algorithm guesses the words $w_0, \ldots, w_n$ in parallel. However, instead of constructing these words in the memory, it guesses the words letter by letter and stores only the states $s_i = \delta(z_0, w_i)$. More precisely, in each pass of the loop we choose a letter a and a number j, and we interpret this choice as appending a to the words $w_j, \ldots, w_n$. Simultaneously, we update the states $s_j, \ldots, s_n$ appropriately. By doing so, we guess all possible chains $w_0 \leq_0 \cdots \leq_0 w_n$ in such a way that we know the states $s_i = \delta(z_0, w_i)$. This allows us to easily verify the right-hand side of (2.1) in line 7. Hence, testing non-membership in $\mathcal{C}_0^1(n)$ is in NL. By NL = coNL [Imm88, Sze87], the membership test also belongs to NL.

The algorithm can be modified such that it works for $\mathcal{C}_0^d(n)$ where d is arbitrary: For this we have to make sure that the guessed $\leq_0$-chain is even a $\leq_0^d$-chain, i.e., the word extensions must be such that the lengths of single insertions are divisible by d. This is done by (i) introducing new variables l_i that count the current length of w_i modulo d and (ii) by making sure that $l_i = l_{i+1}$ whenever $j \leq i < n$ (i.e., letters that appear in both words, w_i and w_{i+1}, must appear at equivalent positions modulo d). So also the membership test for $\mathcal{C}_0^d(n)$ belongs to NL.

Finally, we adapt the algorithm to make it work for $\mathcal{C}_k^d(n)$ where d and k are arbitrary. So we have to make sure that the guessed $\leq_0^d$-chain is even a $\leq_k^d$-chain. For this, let us consider an extension $u \leq_k^d w$ where $u, w \in A^{>k}$. The (k,d)-embedding f that is used in the definition of $u \leq_k^d w$ ensures that for all i it holds that in u at position i there are the same $k+1$ letters as in w at position $f(i)$. Therefore, a word extension $u \leq_k^d w$ can be split into a series of elementary extensions of the form $u_1u_2 \leq_0^d u_1vu_2$ such that the length k prefixes of u_2 and vu_2 are equal. The latter is called the *prefix condition*. Moreover, we can always make sure that the positions in u at which the elementary extensions occur form a strictly increasing sequence. This allows us to guess the words in the $\leq_k^d$-chain letter by letter. Now the algorithm can test the prefix condition by introducing new variables v_i that contain a guessed preview of the next k letters in w_i. Each time a letter is appended to w_i, (i) we verify that this letter is consistent with the preview v_i and (ii) we update v_i by removing the first letter and by appending a new guessed letter. In this way the modified algorithm carries the length k previews of the w_i with it and it makes sure that guessed

letters are consistent with these previews. Moreover, we modify the algorithm such that whenever $j \leq i < n$, then the condition $v_i = v_{i+1}$ is tested. The latter makes sure that elementary extensions $u_1u_2 \leq_0^d u_1vu_2$ satisfy the prefix condition and hence the involved words are even in $\leq_k^d$ relation. This modified algorithm shows the following.

Theorem 2.1. $\{M \mid M$ *is a det. finite automaton and* $L(M) \in \mathcal{C}_k^d(n)\} \in$ NL *for* $k \geq 0$, $d \geq 1$.

We now explain why the above idea does not immediately lead to a nondeterministic, logarithmic-space membership algorithm for the classes $\Sigma_1^\sigma(n)$, although an alternating chain characterization for $\Sigma_1^\sigma(n)$ is known from [GS01a]. Note that the described algorithm for $\mathcal{C}_k^d(n)$ stores the following types of variables in logarithmic space.

(1) variables s_i that contain states of M
(2) variables l_i that contain numbers from $[0, d-1]$
(3) variables v_i that contain words of length k

However, the characterization of the classes $\Sigma_1^\sigma(n)$ [GS01a] is unsuitable for our algorithm: In order to verify the forbidden-chain condition, we have to guess a chain of so-called structured words and have to make sure that certain parts u in these words are M-idempotent (i.e., $\delta(s,u) = \delta(s,uu)$ for all states s). Again we would try to guess the words letter by letter, but now we have to make sure that (larger) parts u of these words are M-idempotent. We do not know how to verify the latter condition in logarithmic space.

In a similar way one observes that the known characterization of the classes $\Sigma_1^{\tau_d}(n)$ [Sel04] cannot be used for the construction of an efficient membership test. So new characterizations of $\Sigma_1^\sigma(n)$ and $\Sigma_1^{\tau_d}(n)$ are needed in order to obtain efficient membership algorithms.

3. New Characterizations of Boolean-Hierarchy Classes

In this section we develop new alternating-chain characterizations that allow the construction of efficient algorithms deciding membership for the Boolean hierarchies over Σ_1^σ, $\Sigma_1^{\tau_d}$, and Σ_2^o for $|A| = 2$. We begin with the introduction of *marked words* and related partial orders which turn out to be crucial for the design of efficient algorithms.

3.1. Marked Words

For a fixed finite alphabet A, let $\mathcal{A} \stackrel{df}{=} \{\, [a,u] \mid a \in A, u \in A^* \,\}$ be the corresponding marked alphabet. Words over $\mathcal{A}$ are called marked words. For $w \in \mathcal{A}^*$ with $w = [a_1,u_1]\cdots[a_m,u_m]$ let $\overline{w} \stackrel{df}{=} a_1 \cdots a_m \in A^*$ be the corresponding unmarked word. Sometimes we use the functional notation $f_i(w) = a_1u_1^i \cdots a_mu_m^i$, i.e., $f_0(w) = \overline{w}$. Clearly, $f_0 : \mathcal{A}^* \to A^*$ is a surjection. For $x = x_1 \cdots x_m \in A^+$ and $u \in A^*$ we define $[x,u] \stackrel{df}{=} [x_1,\varepsilon]\cdots[x_{m-1},\varepsilon][x_m,u]$.

Next we define a relation on marked words. For $w, w' \in \mathcal{A}^*$ we write $w \preceq w'$ if and only if there exist $m \geq 0$, marked words $x_i, z_i \in \mathcal{A}^*$, and marked letters $b_i = [a_i,u_i] \in \mathcal{A}$ where $u_i \in A^+$ s.t.

$$\begin{array}{rcccccccccc} w &=& x_0b_1 & & x_1b_2 & & x_2 & \cdots & b_m & & x_m, \text{ and} \\ w' &=& x_0b_1 & z_1b_1 & x_1b_2 & z_2b_2 & x_2 & \cdots & b_m & z_mb_m & x_m. \end{array}$$

We call b_i the context letter of the insertion z_ib_i. We write $w \preceq^d w'$ if $w \preceq w'$ and $|f_0(z_ib_i)| \equiv 0 \pmod d$ for all i. Note that $\preceq^1$ coincides with $\preceq$ and observe that $\preceq^d$ is a transitive relation.

For a dfa $M = (A, Z, \delta, s_0, F)$ and $s, t \in Z$ we write $s \xrightarrow[M]{w} t$, if $\delta(s, \overline{w}) = t$ and for all i, $\delta(s, a_1 \cdots a_i) = \delta(s, a_1 \cdots a_i u_i)$. So $s \xrightarrow[M]{w} t$ means that the marked word w leads from s to t in a way such that the labels of w are consistent with loops in M. We say that w is M-consistent, if for some $t \in Z$, $s_0 \xrightarrow[M]{w} t$ and denote by $\mathcal{B}_M$ the set of marked words that are M-consistent. Every M-consistent word has the following nice property.

Proposition 3.1. *For $w = [c_1, u_1] \cdots [c_m, u_m] \in \mathcal{B}_M$ and all $j \geq 0$, $f_0(w) \equiv_M c_1 u_1^j \cdots c_m u_m^j$.*

3.2. New Characterization of the Classes $\boldsymbol{\Sigma_1^\sigma(n)}$ and $\boldsymbol{\Sigma_1^{\tau_{\mathrm{d}}}(n)}$

We extend the known characterization of the classes $\Sigma_1^{\tau_{\mathrm{d}}}(n)$ [Sel04] and add a characterization in terms of alternating chains on M-consistent marked words. Because we can also restrict the length of the labels u_i, we denote by $\mathcal{B}_M^c$ for any $c > 0$ the set of marked words $[a_0, u_0] \cdots [a_n, u_n]$ that are M-consistent and satisfy $|u_i| \leq c$ for all $i \leq n$.

Theorem 3.2. *The following is equivalent for $d, n \geq 1$, a dfa M, $c = |M|^{|M|}$, and $L = L(M) \subseteq A^+$.*

(1) $L \in \Sigma_1^{\tau_{\mathrm{d}}}(n)$
(2) $f_0^{-1}(L)$ *has no 1-alternating chain of length n in* $(\mathcal{B}_M; \preceq^d)$
(3) $f_0^{-1}(L)$ *has no 1-alternating chain of length n in* $(\mathcal{B}_M^c; \preceq^d)$

The case $d = 1$ is an alternative to the known characterization of the classes $\Sigma_1^\sigma(n)$ [GS01a].

Theorem 3.3. *Let M be a dfa, $L = L(M) \subseteq A^+$ and $n \geq 1$. Then $L \in \Sigma_1^\sigma(n)$ if and only if $f_0^{-1}(L)$ has no 1-alternating chain of length n in $(\mathcal{B}_M; \preceq)$.*

We can give an upper bound on d for languages in $\Sigma_1^\tau(n)$.

Theorem 3.4. *For every dfa M, $c = |M|^{|M|}$, and $d = c!$, $L(M) \in \Sigma_1^\tau(n) \Rightarrow L(M) \in \Sigma_1^{\tau_{\mathrm{d}}}(n)$.*

3.3. Characterization of the Classes $\boldsymbol{\Sigma_2^\varrho(n)}$ for $\boldsymbol{|A| = 2}$

We obtain an alternating-chain characterization for the classes of the Boolean hierarchy over Σ_2^ϱ for the case $|A| = 2$. This allows us to prove the first decidability result for this hierarchy. Note that only in case $|A| = 2$ decidability of $\mathrm{BC}(\Sigma_2^\varrho)$ [Str88] and Σ_3^ϱ [GS01b] is known.

For $u \in A^*$ let $\alpha(u)$ be the set of letters in u. We say that a marked word $w = [c_1, u_1] \cdots [c_m, u_m]$ satisfies the *alphabet condition* if for all $u_i \neq \epsilon$ it holds that $\alpha(u_i) = A$.

Theorem 3.5. *Let $A = \{a, b\}$, $n \geq 1$ and let $L(M) \subseteq A^*$ for some dfa M such that $L = L(M)$ is a star-free language. Then $L \in \Sigma_2^\varrho(n)$ if and only if $f_0^{-1}(L)$ has no 1-alternating chain $(w_0, \ldots, w_n)$ in $(\mathcal{B}_M; \preceq)$ such that all w_i satisfy the alphabet condition.*

4. Decidability and Complexity

The alternating-chain characterizations from the last sections can be used for the construction of efficient algorithms for testing the membership in these classes. As corollaries we obtain new decidability results: the classes $\Sigma_1^\tau(n)$ and $\Sigma_2^\varrho(n)$ for $|A| = 2$ are decidable.

The characterizations given in Theorems 3.2 and 3.3 allow the construction of nondeterministic, logarithmic-space membership tests for $\Sigma_1^\sigma(n)$ and $\Sigma_1^{\tau_{\mathrm{d}}}(n)$.

Theorem 4.1. *For all* $n \geq 1$, $\{M \,\big|\, M$ *is a det. finite automaton and* $L(M) \in \Sigma_1^\sigma(n)\} \in$ NL.

Theorem 4.2. *For all* $n \geq 1$, $\{M \,\big|\, M$ *is a det. finite automaton and* $L(M) \in \Sigma_1^{\tau_{\mathrm{d}}}(n)\} \in$ NL.

Remark 4.3. Unfortunately, we do not obtain NL-decidability for the classes $\Sigma_1^\tau(n)$. The reason is that the d in Theorem 3.4 is extremely big, i.e., we only know the upper bound $d \leq (m^m)!$ where m is the size of the automaton. We leave the question for an improved bound open. Note that if d can be bounded polynomially in the size of the automaton, then $\Sigma_1^\tau(n)$ is decidable in NL. Although d is very large, it is still computable from the automaton M which implies the decidability of all levels $\Sigma_1^\tau(n)$. This settles a question left open in [Sel04].

Theorem 4.4. *For all* $n \geq 1$, $\{M \,\big|\, M$ *is a det. finite automaton and* $L(M) \in \Sigma_1^\tau(n)\}$ *is decidable.*

Theorem 4.5. *For all* $n \geq 1$,
$\{M \,\big|\, M$ *is a det. finite automaton over the alphabet* $\{a, b\}$ *and* $L(M) \in \Sigma_2^\varrho(n)\} \in$ NL.

5. Exact Complexity Estimations

With the exception of $\Sigma_1^\tau(n)$, the membership problems of all classes of Boolean hierarchies considered in this paper are NL-complete. In contrast, the membership problems of the general classes FO_τ and FO_{τ_d} are PSPACE-complete and hence are strictly more complex.

Proposition 5.1. *Let* $\mathcal{C}$ *be any class of regular quasi-aperiodic languages over* A *with* $|A| \geq 2$ *and* $\emptyset \in \mathcal{C}$. *Then it is* NL*-hard to decide whether a given dfa* M *accepts a language in* $\mathcal{C}$.

Together with the upper bounds established in the previous sections this immediately implies the following exact complexity estimations.

Theorem 5.2. *Let* $k \geq 0$, $n \geq 1$, $d \geq 1$ *and* $\mathcal{C}$ *is one of the classes* $\mathcal{C}_k^d(n)$, $\Sigma_1^\sigma(n)$, $\Sigma_1^{\tau_{\mathrm{d}}}(n)$, *or* $\Sigma_2^\varrho(n)$ *for* $|A| = 2$. *Then* $\{M \,\big|\, M$ *is a det. finite automaton and* $L(M) \in \mathcal{C}\}$ *is* NL*-complete.*

We conclude this section with a corollary of the PSPACE-completeness of deciding FO_σ which was established by Stern [Ste85b] and by Cho and Huynh [CH91]. It shows that the complexity of deciding the classes FO_τ and FO_{τ_d} is strictly higher than the complexity of deciding the classes mentioned in Theorem 5.2. (Note that NL is closed under logspace many-one reductions, NL $\subseteq$ DSPACE$(\log^2 n)$ [Sav70] and DSPACE$(\log^2 n) \subsetneq$ PSPACE [HS65]. Hence the classes FO_τ and FO_{τ_d} can not be decided in NL.)

Theorem 5.3. *The classes* FO_τ *and* FO_{τ_d} *are* PSPACE*-complete.*

6. Conclusions

The results of this paper (as well as several previous facts that appeared in the literature) show that more and more decidable levels of hierarchies turn out to be decidable in NL. One is tempted to strengthen the well-known challenging conjecture of decidability of the dot-depth hierarchy to the conjecture that all levels of reasonable hierarchies of first-order definable regular languages are decidable in NL. At least, it seems instructive to ask this question about any level of such a hierarchy known to be decidable.

In this paper we considered the complexity of classes of regular languages only w.r.t. the representation of regular languages by dfa's. Similar questions are probably open for other natural representations of regular languages, like nondeterministic finite automata and propositions of monadic second order, first order or temporal logics.

Acknowledgements

We are grateful to Klaus W. Wagner for many helpful discussions. We would like to thank the anonymous referees for their valuable comments.

References

[BCST92] D. A. Mix Barrington, K. Compton, H. Straubing, and D. Thérien. Regular languages in NC^1. *J. Computer and System Sciences*, 44:478–499, 1992.

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

[Büc60] J. R. Büchi. Weak second-order arithmetic and finite automata. In *Z. Math. Logik und grundl. Math.*, 6:66-92, 1960.

[CH91] S. Cho and D. T. Huynh. Finite-automaton aperiodicity is PSPACE-complete. *Theoret. Computer Science*, 88:99–116, 1991.

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

[CPP93] J. Cohen, D. Perrin, and J. E. Pin. On the expressive power of temporal logic. *J. Computer and System Sciences*, 46:271–294, 1993.

[CPS06] L. Chaubard, J. E. Pin, and H. Straubing. First order formulas with modular predicates. In *Proc. 21th IEEE Symposium on Logic in Computer Science*, pp. 211–220. IEEE Comp. Society, 2006.

[GS00a] C. Glaßer and H. Schmitz. Languages of dot-depth 3/2. In *Proc. 17th STACS*, Lect. Notes in Comp. Science **1770**, pp. 555–566. Springer, 2000.

[GS00b] C. Glaßer and H. Schmitz. Decidable hierarchies of starfree languages. In *Proc. 20th FST-TCS*, Lect. Notes in Comp. Science **1974**, pp. 503–515. Springer, 2000.

[GS01a] C. Glaßer and H. Schmitz. The boolean structure of dot-depth one. *Journal of Automata, Languages and Combinatorics*, 6(4):437–452, 2001.

[GS01b] C. Glaßer and H. Schmitz. Level 5/2 of the Straubing-Thérien hierarchy for two-letter alphabets. In *Preproceedings 5th Conference on Developments in Language Theory*, pp. 254–265, 2001.

[GSS07] C. Glaßer, H. Schmitz and V. Selivanov. Efficient algorithms for membership in Boolean hierarchies of regular languages. ECCC Report TR07-094, October 2007.

[HS65] J. Hartmanis and R. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[Imm88] N. Immerman. Nondeterministic space is closed under complementation. *SIAM J. Computing*, 17:935–938, 1988.

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

[MPT00] A. Maciel, P. Péladeau, and D. Thérien. Programs over semigroups of dot–depth one. *Theoret. Computer Science*, 245:135–148, 2000.

[Pin95] J. E. Pin. Finite semigroups and recognizable languages: an introduction. In J. Fountain, editor, *NATO Advanced Study Institute: Semigroups, Formal Languages and Groups*, pp. 1–32. Kluwer Academic Publishers, 1995.

[Pin96a] J. E. Pin. Logic, semigroups and automata on words. *Annals of Mathematics and Artificial Intelligence*, 16:343–384, 1996.

[Pin96b] J. E. Pin. Syntactic semigroups. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*, volume I, pp. 679–746. Springer, 1996.

[PW02] J. E. Pin and P. Weil. The wreath product principle for ordered semigroups. *Communications in Algebra*, 30:5677–5713, 2002.

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

[Sav70] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

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

[Sel01] V. L. Selivanov. A logical approach to decidability of hierarchies of regular star-free languages. In *Proc. 18th STACS*, Lect. Notes in Comp. Science **2010**, pp. 539–550. Springer, 2001.

[Sel04] V. L. Selivanov. Some hierarchies and reducibilities on regular languages. Technical Report 349, Inst. für Informatik, Univ. Würzburg, 2004.

[Shu98] A.G. Shukin. Difference hierarchies of regular languages (in russian). *Computing Systems (Novosibirsk, Institute of Mathematics)*, (161):141–155, 1998.

[Sim75] I. Simon. Piecewise testable events. In *Proceedings 2nd GI Conference*, Lect. Notes in Comp. Science **33**, pp. 214–222. Springer, 1975.

[SS00] V. L. Selivanov and A. G. Shukin. On hierarchies of regular star-free languages. Technical Report Preprint 69, A. P. Ershov Institute of Informatics Systems, Novosibirsk, 2000.

[Ste85a] J. Stern. Characterizations of some classes of regular events. *Theoret. Computer Science*, 35:17–42, 1985.

[Ste85b] J. Stern. Complexity of some problems from the theory of automata. *Information and Control*, 66:163–176, 1985.

[Str81] H. Straubing. A generalization of the Schützenberger product of finite monoids. *Theoret. Computer Science*, 13:137–150, 1981.

[Str85] H. Straubing. Finite semigroup varieties of the form $\mathbf{V} * \mathbf{D}$. *J.Pure Appl.Algebra*, 36:53–94, 1985.

[Str88] H. Straubing. Semigroups and languages of dot-depth two. *Theoret. Computer Science*, 58:361–378, 1988.

[Str94] H. Straubing. *Finite automata, formal logic and circuit complexity*. Birkhäuser, Boston, 1994.

[SW98] H. Schmitz and K. W. Wagner. The Boolean hierarchy over level 1/2 of the Straubing-Thérien hierarchy. `http://arxiv.org/abs/cs.CC/9809118`.

[Sze87] R. Szelepcsényi. The method of forcing for nondeterministic automata. *Bull. of the EATCS*, 33:96–100, 1987.

[Thé81] D. Thérien. Classification of finite monoids: the language approach. *Theoret. Computer Science*, 14:195–208, 1981.

[Tho82] W. Thomas. Classifying regular events in symbolic logic. *J. Comp. and System Sciences*, 25:360–376, 1982.

[Tra58] B. A. Trakhtenbrot. Synthesis of logic networks whose operators are described by means of single-place predicate calculus. *Doklady Akad. Nauk SSSR*, 118:646–649, 1958.

[Wei04] P. Weil. Algebraic recognizability of languages. In *Proc. 29th Mathematical Foundations of Computer Science*, *Lect. Notes in Comp. Science* **3153**, pp. 149–175. Springer, 2004.

[Yu96] S. Yu. Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*, volume I, pp. 41–110. Springer, 1996.