A Survey on Small Fragments of First-Order Logic over Finite Words

Volker Diekert[1], Paul Gastin[2], Manfred Kufleitner[1,3]

[1] FMI, Universität Stuttgart
Universitätsstr. 38, D-70569 Stuttgart, Germany
{diekert,kufleitner}@fmi.uni-stuttgart.de

[2] LSV, ENS de Cachan & CNRS
61, Av. du Président Wilson, F-94235 Cachan Cedex, France
Paul.Gastin@lsv.ens-cachan.fr

[3] LaBRI, Université de Bordeaux & CNRS
351, Cours de la Libération, F-33405 Talence Cedex, France

Contents

Abstract

We consider fragments of first-order logic over finite words. In particular, we deal with first-order logic with a restricted number of variables and with the lower levels of the alternation hierarchy. We use the algebraic approach to show decidability of expressibility within these fragments. As a byproduct, we survey several characterizations of the respective fragments. We give complete proofs for all characterizations and we provide all necessary background. Some of the proofs seem to be new and simpler than those which can be found elsewhere. We also give a proof of Simon's theorem on factorization forests restricted to aperiodic monoids because this is simpler and sufficient for our purpose.

Keywords: First-order logic, monoids, factorization forests, piecewise-testable languages

Preamble

There are many brilliant surveys on formal language theory [36, 41, 48, 85, 86]. Quite many surveys cover first-order and monadic second-order definability. But there are also nuggets below. There are deep theorems on proper fragments of first-order definability. The most prominent fragment is FO^2; it is the class of languages which are defined by first-order sentences which do not use more than two names for variables. Although various characterizations are known for this class, there seems to be little knowledge in a broad community. A reason for this is that the proofs are spread over the literature and even in the survey [76] many proofs are referred to the original literature which in turn is sometimes quite difficult to read.

This is our starting point. We restrict our attention to fragments strictly below first-order definability. We concentrate on algebraic and formal language theoretic characterizations for those fragments where decidability results are known, although we do not discuss complexity issues here. We give a clear preference to full proofs rather than to state all results. In our proofs we tried to be minimalistic. All technical concepts which are introduced are also used in the proofs for the main results.

1 Introduction

Probably all courses on formal languages speak about regular languages and the basic transformations between finite automata and rational expressions. However, very often the connection to logic and algebra is completely ignored although highlights in formal language theory can be found here. The connection between automata and logic goes back to Büchi. He used monadic second-order logic (MSO) for describing properties of words. Hence, every sentence in MSO defines a language by specifying the set of all words having this property. He gave effective transformations of MSO sentences into finite automata and vice versa [12]. This shows that definability in MSO yields exactly the class of regular languages. This complements Kleene's characterization of regular languages using rational expressions [30] and Myhill's characterization in terms of finite monoids [40].

Many results on the interplay between regular languages seem to be less known than they deserve. We focus on the world below first-order definability. The focus is here, because we think that the least knowledge is here. When considering subclasses of regular languages, it turns out that finite monoids are a very advantageous point of view. For instance, Schützenberger has shown that a language is star-free if and only if it is recognized by some finite and aperiodic monoid [57]. Brzozowski and Simon as well as McNaughton have shown independently that it is decidable whether a regular language is locally testable by describing an algebraic counterpart [11, 38]. Simon has characterized piecewise-testable languages in terms of finite $\mathcal{J}$-trivial monoids [60]. Inspired by these results, Eilenberg has proposed a general framework for such correspondences between classes of regular languages and classes of finite monoids [21]. More recent presentations of the algebraic approach in the study of regular languages can be found in [1, 44] or the annex of [43].

McNaughton and Papert have considered definability in first-order logic for finite words and they showed that this coincides with the class of star-free languages [39]. Together with Schützenberger's Theorem this gives decidability of the problem whether a regular language is definable in first-order logic. Kamp has shown that every first-order sentence is equivalent to a formula in linear temporal logic [29]. Since linear temporal logic can be considered as a fragment of first-order logic, both mechanisms have the same expressive power. Every modality in linear temporal logic can be defined in first-order logic with at most three variables. Therefore, three variables are sufficient to express every first-order definable language. A survey of these results can be bound in [20].

Within first-order logic, one can restrict several resources. The first limitation we consider is the number of variables. Definability in first-order logic with only two variables yields a class of languages for which even more different characterizations are known than for star-free languages. The algebraic counterpart is the class **DA** of finite monoids. The first letter **D** stands for one of Green's relations [25] and the second letter comes from **A**periodic. Schützenberger has characterized **DA** by unambiguous polynomials which are particular regular languages [58]. Later, several other characterizations were added [23, 33, 53, 54, 59, 81, 90]. If we allow only one variable, the situation is rather trivial since all binary relation symbols are useless. Another resource of formulae is the number of quantifier alternations. Here, algebraic and language theoretic characterizations of the so-called alternation hierarchy are known [54, 84], but decidability is only known for a few small levels [3, 4, 46, 60]. Weis and Immerman have combined both restrictions, number of variables and alternation depth. They initiated the research on the alternation hierarchy within first-order logic with only two variables [90].

The aim of this survey is to present complete proofs for the decidability of the expressibility within the following first-order fragments:

- First-order logic with only one variable, denoted $\mathrm{FO}^1[<]$.
- First-order logic with two variables, denoted $\mathrm{FO}^2[<]$.
- Existential first-order logic, denoted $\Sigma_1[<]$.
- The Boolean closure of $\Sigma_1[<]$, denoted $\mathbb{B}\Sigma_1[<]$.

- Formulae with two blocks of quantifiers and starting with a block of existential quantifiers, denoted $\Sigma_2[<]$.

We stop here because we are not aware of any decidable fragment between $\Sigma_2[<]$ and full first-order logic. The class $\Pi_n[<]$ consists of negations of formulae in $\Sigma_n[<]$. The decidability of expressibility within $\Sigma_1[<]$ and within $\Sigma_2[<]$ yields decidability for $\Pi_1[<]$ and $\Pi_2[<]$, respectively. The usual way to obtain the above decidability results is to find algebraic characterizations in terms of properties of finite monoids. Deciding the expressibility then goes by verifying that property of a canonical finite monoid which is effectively computable. We also obtain some other (well-known) properties of the above fragments of first-order logic, such as Schützenberger's characterization of **DA** in terms of unambiguous polynomials, which in turn corresponds to $\mathrm{FO}^2[<]$. We try to use as little background in semigroup theory as possible. Some proofs are new and might be considered as simplifications of the existing ones. An exception is the characterization of $\mathbb{B}\Sigma_1[<]$ where we use the original proof of Straubing and Thérien [71]. We try to keep all sections as self-contained as possible. It should be possible to read the proof of every characterization without the need to read (the hard parts of) the characterizations of the other fragments. An important tool in the study of the alternation hierarchy is the existence of factorization forests of finite height for every homomorphism to a finite monoid [62].

2 Words, languages, logic, and finite monoids

Words and languages. A *word* means here a finite sequence over some finite alphabet Γ. A word of length n is usually written as a product $a_1 \cdots a_n$ with $a_i \in \Gamma$; it is also viewed as a labeled linear order over the set of *positions* $\{1, \ldots, n\}$ with the natural linear order and labels $\lambda(j) = a_j$ for $1 \leq j \leq n$. A position labeled by a is also called *a-position.* The *length* of a word w is denoted by $|w|$, and ε is the empty word. The *alphabet* $\mathrm{alph}(w)$ of a word $w = a_1 \cdots a_n$ is the set $\{a_1, \ldots, a_n\} \subseteq \Gamma$; a *subword* of $a_1 \cdots a_n$ is here a word of the form $a_{i_1} \cdots a_{i_t}$ where $1 \leq i_1 < \cdots < i_t \leq n$. A *factor* of w is a word u such that we can write $w = puq$. By Γ^* we denote the set of all words. It is the free monoid over Γ. Remember that a *semigroup* is a set equipped with a binary associative operation and a *monoid* is a semigroup that contains a neutral element. If N is a subset of a monoid M, then N^* denotes the submonoid of M generated by N. A *language* is a set of words, i.e., a subset of Γ^*. Essentially all languages in this survey will be regular. Thus, they can be specified by a (non-deterministic) finite automaton or equivalently by an MSO-formula.

Polynomials. An important class of regular languages in our setting are the polynomials. By definition it is the smallest family containing all finite subsets of Γ^*, all subsets of the form A^* for $A \subseteq \Gamma$, and which is closed under finite union and concatenation. In order to define also the degree of a polynomial we use the following. A *monomial* of *degree* k is a language of the form

$$A_0^* a_1 A_1^* \cdots a_k A_k^* \quad \text{with } a_i \in \Gamma \text{ and } A_i \subseteq \Gamma.$$

A *polynomial* is a finite union of monomials. Its degree is given by the minimal value which appears as a maximal degree of the monomials over all such descriptions. For example, $\Gamma^* ab \Gamma^*$ is a monomial of degree 2. If $\Gamma = \{a, b\}$, then the complement is a polynomial of degree 1 since it is given as $a^* \cup b^* ba^*$. But $\Gamma^* \setminus \Gamma^* ab \Gamma^*$ is not a polynomial as soon as Γ contains at least three letters. Indeed, consider $(acb)^*$. Assume this subset is contained in a polynomial of degree k, then at least one factor acb in $(acb)^{k+1}$ sits inside some factor of the form A_i^*, so we cannot insist to see the letter c; and therefore we cannot avoid a factor ab in some words of the polynomial.

As we will see later unambiguous polynomials form a proper subclass. A monomial is *unambiguous* if for all $w \in A_0^* a_1 A_1^* \cdots a_k A_k^*$ there exists exactly one factorization $w = w_0 a_1 w_1 \cdots a_k w_k$ with $w_i \in A_i^*$. A language is an *unambiguous polynomial* if it is a finite disjoint union of unambiguous monomials. For every alphabet, $\Gamma^* a_1 \Gamma^* \cdots a_k \Gamma^*$ is an unambiguous monomial. To see this let $A_i = \Gamma \setminus \{a_{i+1}\}$, then we have $\Gamma^* a_1 \Gamma^* \cdots a_k \Gamma^* = A_0^* a_1 A_1^* \cdots a_k \Gamma^*$. The language $\{a, b\}^* \, ab \, \{a, b\}^*$ is also an unambiguous monomial, because $\{a, b\}^* \, ab \, \{a, b\}^* = b^* \, a \, a^* \, b \, \{a, b\}^*$. However, for $\Gamma = \{a, b, c\}$ the monomial $\Gamma^* ab \Gamma^*$ is not unambiguous, since the complement is no polynomial, but the class of unambiguous polynomials is closed under complementation, as we will see below.

First-order logic. In this paper we are interested in subclasses of first-order definable languages. The syntax of *first-order logic* formulae FO[<] is built upon atomic formulae of type

$$\lambda(x) = a \quad \text{or} \quad x < y \quad \text{or} \quad \top.$$

Here, x and y are variables, $a \in \Gamma$ is a letter, and $\top$ is a constant which means *true*. If φ, ψ are first-order formulae, then

$$\neg\varphi \quad \text{and} \quad \varphi \vee \psi \quad \text{and} \quad \exists x \, \varphi$$

are first-order formulae, too. We use the usual shorthands as $\bot = \neg\top$ meaning *false*, $\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$, and $\forall x \, \varphi = \neg\exists x \, \neg\varphi$. By FO[<] we denote the set of all first-order formulae. The notation $\mathrm{FO}^m[<]$ means the set of formulae where at most m distinct variables occur. By definition, $\mathrm{FO}^m[<]$ is closed under Boolean operations.

Given $\varphi \in \mathrm{FO}[<]$, the semantics is defined as follows: The variables range over positions in words and hence, the atomic formulae $x < y$ and $\lambda(x) = a$ have a well-defined truth value. Boolean operations and quantification of variables are as usual. A variable which is not quantified is called *free*. A *sentence* is a formula in FO[<] without free variables. Let the free variables in φ be a subset of $\{x_1, \ldots, x_n\}$. If each x_i is associated with a position j_i of w, then (under this interpretation) φ has a well-defined truth value, which is written as

$$w, j_1, \ldots, j_n \models \varphi.$$

We identify formulae by semantic equivalence. Hence, if φ and ψ are formulae with free variables from $x_1, \ldots, x_n$, then we write $\varphi = \psi$ as soon as $w, j_1, \ldots, j_n \models \varphi \leftrightarrow \psi$ for all words w and all positions $j_1, \ldots, j_n$ of w. A first-order sentence φ yields the language

$$L(\varphi) = \{w \in \Sigma^* \mid w \models \varphi\}.$$

Languages of this form are called *first-order definable*.

The *quantifier depth* of a formula is defined inductively. For atomic formulae it is zero, for logical connectives it is the maximum over the subformulae, and adding a quantifier in front increases the quantifier depth by one. For example, the following formula in $\mathrm{FO}^2[<]$ has quantifier depth three:

$$\varphi_1 \;=\; \exists x\colon \Big(\lambda(x) = a \;\wedge\; \forall y\colon \big(x \le y \;\wedge\; \exists x\colon (y \le x \;\wedge\; \lambda(x) = b)\big)\Big)$$

Another important measure for the complexity of a formula is the number of quantifier alternations. Remember that every formula is equivalent to a formula in prenex normal form, i.e., to a formula where all quantifiers stand at the beginning of the formula. For example, the above formula φ_1 is equivalent to:

$$\varphi_2 \;=\; \exists x \forall y \exists z\colon \big(\lambda(x) = a \;\wedge\; x \le y \;\wedge\; y \le z \;\wedge\; \lambda(z) = b\big)$$

We can now count the number of blocks of different quantifiers (existential or universal). This yields the fragments $\Sigma_n[<]$ and $\Pi_n[<]$ of first-order formulae, in which we allow n blocks of quantifiers. The formulae in $\Sigma_n[<]$ start with a block of existential quantifiers whereas those in $\Pi_n[<]$ start with a block of universal quantifiers. Thus, $\varphi_2 \in \Sigma_3[<]$. It is possible that some of the quantifier blocks are empty. This yields the inclusion

$$\Sigma_n[<] \cup \Pi_n[<] \;\subseteq\; \Sigma_{n+1}[<] \cap \Pi_{n+1}[<].$$

According to our convention to identify equivalent formulae, it makes sense to write e.g.:

$$\varphi \in \Sigma_n[<] \;\Leftrightarrow\; \neg\varphi \in \Pi_n[<].$$

The fragments $\Sigma_n[<]$ and $\Pi_n[<]$ are both closed under conjunction and disjunction (but not under negation). The intersection $\Sigma_n[<] \cap \Pi_n[<]$ is denoted by $\Delta_n[<]$. The fragment $\Delta_n[<]$ is the largest class of formulae within $\Sigma_n[<]$ and within $\Pi_n[<]$ which is closed under Boolean operations. In the above example, we have $\varphi_1 = \varphi_2 \in \Sigma_3[<]$. On the other hand, φ_1 is also equivalent to

$$\varphi_3 \;=\; \exists x \exists z \forall y\colon \big(\lambda(x) = a \;\wedge\; x \le y \;\wedge\; y \le z \;\wedge\; \lambda(z) = b\big)$$

and therefore we have $\varphi_1 = \varphi_2 = \varphi_3 \in \mathrm{FO}^2[<] \cap \Sigma_2[<]$. Note that $L(\varphi_1) = a\Gamma^* b$.

Example 1 The language $\Gamma^* a_1 \Gamma^* \cdots a_k \Gamma^*$ is definable with only two variables x and y. This is best seen by induction. Define $\varphi_1(x) = \big(\lambda(x) = a_1\big)$ and for $1 < i \le k$ let

$$\varphi_i(x) \;=\; \big(\,\lambda(x) = a_i \;\wedge\; \exists y\colon y < x \wedge \varphi_{i-1}(y)\,\big).$$

We obtain $L\big(\exists x\colon \varphi_k(x)\big) = \Gamma^* a_1 \Gamma^* \cdots a_k \Gamma^*$. It is clear that $\exists x\colon \varphi_k(x)$ is a $\Sigma_1[<]$ sentence, but in prenex normal form we need more variables than two. $\diamond$

Finite monoids. A language $L \subseteq \Gamma^*$ is *recognized* by a monoid M if there exists a homomorphism $\mu : \Gamma^* \to M$ such that $L = \mu^{-1}(\mu(L))$. A language $L \subseteq \Gamma^*$ is called *recognizable*, if it is recognized by a finite monoid. It is well-known that recognizable languages are regular and vice versa.

For every language L there exists a unique minimal monoid which recognizes L which is given by the *syntactic congruence* $\equiv_L$. For words $v, w \in \Gamma^*$ we write $v \equiv_L w$ if and only if

$$\forall p, q \in \Gamma^* :\ pvq \in L \Leftrightarrow pwq \in L.$$

The congruence classes of $\equiv_L$ constitute the *syntactic monoid* $M(L)$ which is $\Gamma^* / \equiv_L$. The syntactic monoid $M(L)$ recognizes the language L via the natural homomorphism mapping a word w to its class $[w]$. A simple observation shows that if a monoid M recognizes L, then $M(L)$ is a homomorphic image of a submonoid of M. In terms of semigroup theory this means that $M(L)$ is a *divisor* of every recognizing monoid for L. A language L is recognizable if and only if its syntactic monoid $M(L)$ is finite. Moreover, the syntactic monoid $M(L)$ is effectively computable as the transition monoid of the minimal automaton of L. All syntactic monoids under consideration will be finite.

Next, we recall some notations. A monoid M is *commutative* if $uv = vu$ for all $u, v \in M$. An element e is *idempotent* if $e^2 = e$ and a monoid is *idempotent* if all its elements are idempotent. For a finite monoid M with n elements let $\omega = n!$, then u^ω is idempotent for all $u \in M$; and a simple calculation shows that it is the unique idempotent positive power of u. In the following, we will use the notation u^ω to denote this idempotent power in every finite monoid M whatever the cardinality of M is. A monoid M is called *aperiodic*, if for all $u \in M$ there is some $n \in \mathbb{N}$ such that $u^{n+1} = u^n$. A direct consequence is that homomorphic images of aperiodic monoids are aperiodic. Note also that a finite monoid M is aperiodic if and only if $u^{\omega+1} = u^\omega$ for all $u \in M$.

Example 2 The syntactic monoid of the monomial $\Gamma^* ab \Gamma^*$ is aperiodic and can be represented by the following six elements: $\{1, a, b, c, ba, 0\}$ where 0 corresponds to ab, and c appears only if $|\Gamma| \geq 3$. The multiplication table is:

	1	a	b	c	ba	0
1	1	a	b	c	ba	0
a	a	a	0	c	0	0
b	b	ba	b	b	ba	0
c	c	a	c	c	a	0
ba	ba	ba	0	b	0	0
0	0	0	0	0	0	0

Hence, this monoid is neither commutative nor idempotent. Later we will reuse this example on several occasions. ◇

Some of our proofs use *Green's relations*. Let M be a monoid and let $u, v \in M$. Those of Green's relations which play a role here are defined by:

$$\begin{array}{rclcrcl}
u \mathrel{\mathcal{J}} v & \Leftrightarrow & MuM = MvM, & \qquad & u \leq_{\mathcal{J}} v & \Leftrightarrow & MuM \subseteq MvM, \\
u \mathrel{\mathcal{R}} v & \Leftrightarrow & uM = vM, & & u \leq_{\mathcal{R}} v & \Leftrightarrow & uM \subseteq vM, \\
u \mathrel{\mathcal{L}} v & \Leftrightarrow & Mu = Mv, & & u \leq_{\mathcal{L}} v & \Leftrightarrow & Mu \subseteq Mv.
\end{array}$$

The letter $\mathcal{J}$ refers to *ideal* whereas $\mathcal{R}$ and $\mathcal{L}$ refer to *right-* and *left-ideal*, respectively. Let $\mathcal{G} \in \{\mathcal{J}, \mathcal{R}, \mathcal{L}\}$ be one of Green's relations. We write $u <_{\mathcal{G}} v$ if $u \leq_{\mathcal{G}} v$ but not $u \mathrel{\mathcal{G}} v$. It is very common to interpret Green's relations in terms of *factors*, *prefixes*, or *suffixes*. For example, $u \leq_{\mathcal{R}} v$ if and only if there exists $x \in M$ such that $u = vx$, i.e., v is a prefix of u. Therefore, $u \mathrel{\mathcal{R}} v$ if and only if u and v are prefixes of one another.

Example 3 We consider the syntactic monoid of $\Gamma^* ab \Gamma^*$ as in Example 2. If $|\Gamma| \geq 3$, then $ba \mathrel{\mathcal{R}} b$ since $b = ba \cdot c$ and $ba = b \cdot a$. On the other hand, if $\Gamma = \{a, b\}$, then $ba <_{\mathcal{R}} b$ since there is no element x with $b = ba \cdot x$. ◇

3 One variable

The following theorem on the fragment $\mathrm{FO}^1[<]$ is treated as a warm-up. It serves as an archetype for most proofs of the characterizations of logical fragments given in this survey. The first step is to show that if a language satisfies some algebraic property, then it belongs to a particular class of languages; the second step is to show that all languages within this class can be expressed by a formula of the given fragment; and the third step consists in verifying the algebraic property of the logical fragment. Usually, the first step is the most difficult one. It often uses deep theorems from the theory of finite semigroups. In some cases the second step gives a non-trivial normal form for formulae within the logical fragment. Usually, the algebraic characterization has several benefits. The most important one in the considered cases is that it yields decidability of the membership problem for the logical fragments. The naïve approach to solve this problem is to compute the syntactic monoid and verify the algebraic properties. Another advantage of the algebraic characterization is that one obtains certain closure properties of the logical fragments for free, such as closure under inverse homomorphisms.

Theorem 1 *Let $L \subseteq \Gamma^*$. The following assertions are equivalent:*

1. *L is recognized by some finite, idempotent, and commutative monoid M.*
2. *L is a Boolean combination of languages of the form A^* with $A \subseteq \Gamma$.*
3. *L is definable in $\mathrm{FO}^1[<]$.*

Proof: *"1 ⇒ 2"*: Let $\mu : \Gamma^* \to M$ recognize L. If $\mathrm{alph}(u) = \mathrm{alph}(v) = \{a_1, \ldots, a_n\}$, then $\mu(u) = \mu(a_1 \cdots a_n) = \mu(v)$ since M is idempotent and commutative. Hence, $u \in L$ if and only if $v \in L$. This shows that L is a finite (disjoint) union of languages of the form

$$\{w \mid \mathrm{alph}(w) = A\} \;=\; A^* \setminus \left(\bigcup_{a \in A} (A \setminus \{a\})^* \right).$$

"2 ⇒ 3": Note that $\mathrm{FO}^1[<]$ is closed under Boolean operations. Thus, the claim follows because A^* for $A \subseteq \Gamma$ can be expressed by the formula

$$\forall x \colon \bigvee_{a \in A} \lambda(x) = a.$$

We decompose *"3 ⇒ 1"* into *"3 ⇒ 2 ⇒ 1"*. An alternative approach would be verifying idempotency and commutativity of the syntactic monoid.

"3 ⇒ 2": In the fragment $\mathrm{FO}^1[<]$ the binary predicate $<$ is useless since $x < x$ is always false. We can only use atomic formulae of the form $\lambda(x) = a$ or $\top$. Thus, we can express only Boolean combinations of alphabetic conditions.

"2 ⇒ 1": For every $A \subseteq \Gamma$ the language A^* is recognized by the two element monoid $\{1, 0\}$ which is idempotent and commutative. A Boolean combination of languages of type A^* is recognized by a direct product of $\{1, 0\}$, which is still idempotent and commutative. □

Note that $A^* = \Gamma^* \setminus \left(\bigcup_{b \notin A} \Gamma^* b \Gamma^*\right)$ and $\Gamma^* b \Gamma^* = \Gamma^* \setminus (\Gamma \setminus \{b\})^*$. An alternate proof of Theorem 1 using the emerging reformulation of *"2"* can be found in [78].

4 Two variables and the variety DA

In this section we consider the fragment $\mathrm{FO}^2[<]$ with only two different variables. We will show that $\mathrm{FO}^2[<]$ admits the class **DA** of finite monoids as an algebraic characterization. The notation **DA** goes back to Schützenberger. It means that *regular $\mathcal{D}$-classes are aperiodic semigroups.* We do not intend to explain this notation here, but we give an alternate algebraic definition in terms of equations. We say that a finite monoid M belongs to **DA**, if for all $u, v, w \in M$ we have

$$(uvw)^\omega \, v \, (uvw)^\omega = (uvw)^\omega. \tag{1}$$

Remember that u^ω denotes the idempotent element generated by u. Every monoid $M \in \mathbf{DA}$ is aperiodic since equation (1) with $u = w = 1$ implies $v^{\omega+1} = v^\omega$ for all $v \in M$.

Remark 1 *Sometimes* **DA** *is given by the identity $(uv)^\omega \, v \, (uv)^\omega = (uv)^\omega$. This definition is asymmetric, but in fact equivalent to equation (1). The reason is that if M satisfies one of the following equations, then it satisfies all three equations.*

1. $(uv)^\omega \, v \, (uv)^\omega = (uv)^\omega$.
2. $(uv)^\omega \, u \, (uv)^\omega = (uv)^\omega$.
3. $(uvw)^\omega \, v \, (uvw)^\omega = (uvw)^\omega$.

First, let us show that the first equation implies the second one. We may use that $w^{\omega+1} = w^\omega$ for all w. Hence we obtain

$$(uv)^\omega = u\,(vu)^\omega\, v = u\,(vu)^\omega\, u\,(vu)^\omega\, v = (uv)^\omega\, u\,(uv)^{\omega+1} = (uv)^\omega\, u\,(uv)^\omega.$$

Thus, by symmetry the first two equations are in fact equivalent. The third equation implies the first one with $w = 1$. For the converse, let $e = (uvw)^\omega$ and assume that M satisfies both the first and second equation. Then the first equation implies $e = e(vw)e = (ev)(we)$. Thus, ev is a prefix of e and the second equation now yields $e = e^\omega = \big((ev)(we)\big)^\omega = e^\omega(ev)e^\omega = e(ev)e = eve$.

Schützenberger showed the correspondence between **DA** and unambiguous polynomials [58]. As an intermediate step from **DA** to unambiguous polynomials we will use the fragment $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ of temporal logic. The syntax is as follows. The sole atomic formula is $\top$. We allow Boolean connectives and for each letter $a \in \Gamma$ we allow temporal operators X_a (neXt-a) and Y_a (Yesterday-a); hence if φ and ψ are formulae in $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ then so are $\neg\varphi$, $\varphi \vee \psi$, $\mathsf{X}_a\, \varphi$ and $\mathsf{Y}_a\, \varphi$. The *operator depth* of a formula $\varphi \in \mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ is the maximal number of nested (unary) temporal operators. The semantics is as follows: $\top$ is true at all positions, Boolean operations are as usual, and $\mathsf{X}_a\, \varphi$ is true at a position x if at the first a-position y after x the formula φ holds. More formally, $w, x \models \mathsf{X}_a\, \varphi$ is defined by:

$$\exists y \colon (w, y \models \varphi) \ \wedge \ x < y \ \wedge \ \lambda(y) = a \ \wedge \ \forall z \colon x < z < y \ \rightarrow \ \lambda(z) \neq a.$$

Note that $w, x \models \mathsf{X}_a\, \varphi$ does not hold, if there is no a-position after x. The operator Y_a is left-right symmetric to X_a: $w, x \models \mathsf{Y}_a\, \varphi$ is true if at the last a-position before x the formula φ holds. In order to define the truth value of $w \models \varphi$ for $\varphi \in \mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ we imagine that we start at a position outside the word w. Now, $w \models \mathsf{X}_a\, \varphi$ is true if at the first a-position of w the formula φ holds and symmetrically, $w \models \mathsf{Y}_a\, \varphi$ is true if at the last a-position of w the formula φ holds. A language L is definable in $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ if there exists $\varphi \in \mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ such that $L = L(\varphi) = \{w \in \Gamma^* \mid w \models \varphi\}$. For example, with $\varphi = \mathsf{X}_a \neg \mathsf{X}_a \top$ we have $L(\varphi) = \{w_0 a w_1 \mid a \notin \mathrm{alph}(w_0 w_1)\}$ and the operator depth of φ is 2.

Every sequence of the operators X_a and Y_a defines at most one position in a word. In [90] this concept is called a *ranker*. More formally, a ranker is a nonempty word over the alphabet $\{\mathsf{X}_a, \mathsf{Y}_a \mid a \in \Gamma\}$. A ranker r can be identified with the formula $r\top \in \mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. For instance, $r = \mathsf{Y}_a \mathsf{X}_b$ is identified with $r\top = \mathsf{Y}_a \mathsf{X}_b \top$. For a word u and a ranker r with $u \models r\top$ we define $r(u)$ as the position of u reached by the formula $r\top$. Therefore, $\mathsf{Y}_a \mathsf{X}_b(u)$ is the first b-position after the last a-position of u. If $u \not\models r\top$, then $r(u)$ is undefined. We let R_n be the set of rankers of length at most n. For $u \in \Gamma^*$, we define $R_n(u) = \{r(u) \mid r \in R_n,\ u \models r\top\}$ as the set of positions in u reachable by a ranker of length at most n. This induces the factorization $u = u_0 a_1 u_1 \cdots a_k u_k$ with $a_i \in \Gamma$ and $u_i \in \Gamma^*$ such that the a_i's correspond exactly to the positions in $R_n(u)$. We call $\mathrm{RW}_n(u) = a_1 \cdots a_k$ the *ranker word* of u of depth n. Note that $\mathrm{RW}_m(u) = \mathrm{RW}_m(\mathrm{RW}_n(u))$ for all $m \leq n$. The language $\{v \in \Gamma^* \mid \mathrm{RW}_n(v) = \mathrm{RW}_n(u)\}$ is called the *ranker class* of u of depth n. A *ranker language* is a finite union of ranker classes.

Another fragment of temporal logic is $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$ with two unary operators XF (neXt Future) and YP (Yesterday Past), also sometimes called *strict future* and *strict past*. The syntax is similar to $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ with the only difference that for every $a \in \Gamma$ we allow an atomic formula a. The semantics, apart from the classical Boolean connectives, is as follows:

$$\begin{aligned}
w, x &\models a \quad \text{for } a \in \Gamma && \Leftrightarrow \ \lambda(x) = a,\\
w, x &\models \mathsf{XF}\, \varphi && \Leftrightarrow \ \exists y \colon x < y \ \wedge \ w, y \models \varphi,\\
w, x &\models \mathsf{YP}\, \varphi && \Leftrightarrow \ \exists y \colon y < x \ \wedge \ w, y \models \varphi.
\end{aligned}$$

Next, we define when $w \models \varphi$ holds for $\varphi \in \mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. We again imagine that we start at a position outside the word w. For $a \in \Gamma$ the truth value of $w \models a$ is false and $w \models \mathsf{XF}\, \varphi$ is equivalent to $w \models \mathsf{YP}\, \varphi$ which holds if and only if there is a position x such that $w, x \models \varphi$. For example with $\varphi = \mathsf{XF}(a \wedge \neg \mathsf{XF}\, a \wedge \neg \mathsf{YP}\, a)$ we again have $L(\varphi) = \{w_0 a w_1 \mid a \notin \mathrm{alph}(w_0 w_1)\}$. Also, $\psi = \neg \mathsf{XF} \top$ defines

$L(\psi) = \{\varepsilon\}$. The operator depth of φ is 2 and the operator depth of ψ is 1. We are now ready to state the main theorem of this section. As advocated in [76] it is indeed a diamond in formal language theory.

Theorem 2 *Let $L \subseteq \Gamma^*$. The following assertions are equivalent:*

1. *L is recognized by a monoid in* **DA**.
2. *L is definable in* $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$.
3. *L is a ranker language.*
4. *L is an unambiguous polynomial.*
5. *L is definable in* $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$.
6. *L is definable in* $\mathrm{FO}^2[<]$.
7. *L is a polynomial and $\Gamma^* \setminus L$ is also a polynomial.*
8. *L is definable in* $\Delta_2[<]$.

The equivalence of **DA** and $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ has been shown in [32, 33] in the more general context of Mazurkiewicz traces. A slightly modified result can be found in [59] where so-called *turtle automata* are used to describe languages whose syntactic monoid is in **DA**. A refinement of this characterization is used to relate the number of quantifier alternations within $\mathrm{FO}^2[<]$ with the number of changes in the direction, see [90]. This yields the characterization of ranker languages. The connection between **DA** and unambiguous polynomials is due to Schützenberger [58]. An algebraic counterpart for the language operation of taking finite unions of unambiguous products over more general classes of languages can be found in [53]. In [81] the relationship between **DA** and $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$ has been stated. The equivalence of $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$ and $\mathrm{FO}^2[<]$ does not only hold on the language level. In [23], a syntactic conversion of $\mathrm{FO}^2[<]$-formulae with at most one free variable into equivalent $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$-formulae has been given. The characterizations by polynomials whose complement is also a polynomial and by $\Delta_2[<]$ follow from a characterization of the fragment $\Sigma_2[<]$ which we present in Section 9, see Theorem 9. These two characterizations rely on [3, 4, 54, 84]. A survey dedicated to the class **DA** and its numerous appearances can be found in [76]. See [73, 77] for FO^2 with modular quantifiers, or [7, 31] for applications of FO^2 in circuit complexity, or [9] for FO^2 over words with data. A decomposition technique in terms of so-called block products for monoids in **DA** has been introduced in [72].

Example 4 Consider the language $L = \Gamma^* ab \Gamma^* \cap \Gamma^* b$. If $|\Gamma| \geq 3$, then the syntactic monoid of L is not in **DA** since for all $n \geq 1$ we have $(bacb)^n a (bacb)^n \in L$ whereas $(bacb)^n \notin L$. If $\Gamma = \{a, b\}$ then we have $L = \Gamma^* a \Gamma^* b$ which is easily definable in $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ by $\mathsf{X}_a \top \wedge \mathsf{Y}_b \neg \mathsf{X}_a \top$. Using Theorem 2, we see that L is definable in $\mathrm{FO}^2[<]$ if and only if $\Gamma = \{a, b\}$. $\diamond$

To handle the class **DA** we need a little algebraic background which we provide now. The following lemma states an important property of aperiodic monoids.

Lemma 1 *Let M be an aperiodic monoid and let $u, v \in M$. If $u \leq_{\mathcal{L}} v$ and $v \leq_{\mathcal{R}} u$, then $u = v$.*

Proof: Since $u \in Mu \subseteq Mv$, there exists $x \in M$ such that $u = xv$. Similarly, there exists $y \in M$ such that $v = uy$. We have $u = xv = xuy = x^\omega u y^\omega = x^\omega u y^{\omega+1} = uy = v$. □

The crucial properties for monoids in **DA** are aperiodicity (as used in the lemma just above) and the property as given in the next lemma. Of course, there is also a symmetric statement using Green's relation $\mathcal{L}$.

Lemma 2 *Let $u, v, a \in M \in$ **DA**. If $u \mathrel{\mathcal{R}} uv$ and $v \in MaM$, then $u \mathrel{\mathcal{R}} uva$.*

Proof: We have $uw \leq_{\mathcal{R}} u$ for all $u, w \in M$. Therefore, it suffices to show $u \leq_{\mathcal{R}} uva$. Let $x, y, z \in M$ be such that $v = xay$ and $u = uvz$. Then

$$\begin{aligned} uv &= uv \cdot zv = uv \cdot zxay \\ &= uv \cdot (zxay)^\omega \\ &= uv \cdot (zxay)^\omega a (zxay)^\omega && \text{since } M \in \mathbf{DA} \\ &= uva \cdot (zxay)^\omega \in uvaM. \end{aligned}$$

This shows $uv \leq_{\mathcal{R}} uva$ and together with $u \mathrel{\mathcal{R}} uv$ we conclude $u \leq_{\mathcal{R}} uva$. □

Proposition 1 *Let $L \subseteq \Gamma^*$. If L is recognized by a monoid in* **DA**, *then L is definable in* $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$.

Proof: Let $M \in$ **DA** and let $\mu : \Gamma^* \to M$ be a homomorphism with $\mu^{-1}\mu(L) = L$. We define an equivalence relation on words, called *operator-depth-equivalence*. For $n \geq 0$ and $u, v \in \Gamma^*$ we have $u \equiv_n v$ if u and v satisfy exactly the same formulae in $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ of operator depth at most n. Let $n > 2\,|M| \cdot |\Gamma|$ and $u, v \in \Gamma^*$ with $u \equiv_n v$. We show that this implies $\mu(u) = \mu(v)$. From $n \geq 1$ we conclude that $\mathrm{alph}(u) = \mathrm{alph}(v)$. If $\mathrm{alph}(u) = \emptyset$, then $u = \varepsilon = v$ and hence $\mu(u) = \mu(v)$. Thus, we may assume $\mathrm{alph}(u) \neq \emptyset$ and we perform an induction on the size of this alphabet.

We can write $u = u_0 a_1 u_1 \cdots a_k u_k$ where $u_i \in \Gamma^*$ and $a_i \in \Gamma$ such that:

- $1_M \mathrel{\mathcal{R}} \mu(u_0)$,
- $\mu(u_0 a_1 u_1 \cdots a_i) \mathrel{\mathcal{R}} \mu(u_0 a_1 u_1 \cdots a_i u_i)$ for all $1 \leq i \leq k$,
- $\mu(u_0 a_1 u_1 \cdots a_i u_i) >_{\mathcal{R}} \mu(u_0 a_1 u_1 \cdots a_i u_i a_{i+1})$ for all $0 \leq i < k$.

The idea is that exactly the letters a_i are reducing the level of the $\mathcal{R}$-classes. Since there are at most $|M|$ many $\mathcal{R}$-classes, we have $k < |M|$. By Lemma 2 we see $\mu(u_{i-1}) \notin M\mu(a_i)M$ for all $1 \leq i \leq k$ and hence $a_i \notin \mathrm{alph}(u_{i-1})$. It follows

$$u \models \mathsf{X}_{a_1} \mathsf{X}_{a_2} \ldots \mathsf{X}_{a_k} \top$$

and the operator depth of this formula is $k < |M| < n$. From $u \equiv_n v$ we conclude that $v = v_0 a_1 v_1 \cdots a_k v_k$ with $a_i \notin \mathrm{alph}(v_{i-1})$ for $1 \leq i \leq k$. A position x of u is within the factor u_i for $i < k$ if and only if $u, x \models \varphi_i$ with

$$\varphi_i = \big(\mathsf{Y}_{a_i} \mathsf{Y}_{a_{i-1}} \ldots \mathsf{Y}_{a_1} \top\big) \wedge \big(\mathsf{X}_{a_{i+1}} \neg \mathsf{Y}_{a_{i+1}} \mathsf{Y}_{a_i} \ldots \mathsf{Y}_{a_1} \top\big)$$

using the convention that for $i = 0$ the subformula $\mathsf{Y}_{a_0} \ldots \mathsf{Y}_{a_1} \top$ is true. Similarly, $v, y \models \varphi_i$ if and only if y is a position within the factor v_i of v. The operator depth of the formulae φ_i is at most $|M|$. As above, we can reach the positions of a_i and a_{i+1} with formulae of depth at most $< |M|$, and we can use the formulae φ_i to ensure that we stay inside the factor u_i (or v_i, respectively). With this relativization technique, $u \equiv_n v$ implies $u_i \equiv_{n-2|M|} v_i$ for all $0 \leq i < k$. By induction on the size of the alphabet we obtain $\mu(u_i) = \mu(v_i)$ for all $0 \leq i < k$ (we cannot use the induction hypothesis for $i = k$ since we may have $\mathrm{alph}(u_k) = \Gamma$). Thus

$$\mu(v) \leq_{\mathcal{R}} \mu(v_0 a_1 \cdots v_{k-1} a_k) = \mu(u_0 a_1 \cdots u_{k-1} a_k) \mathrel{\mathcal{R}} \mu(u).$$

This means $\mu(v) \leq_{\mathcal{R}} \mu(u)$. Symmetrically, by starting with a factorization of v with respect to Green's $\mathcal{L}$-relation we see that $\mu(u) \leq_{\mathcal{L}} \mu(v)$. From Lemma 1 we conclude $\mu(u) = \mu(v)$.

Up to equivalence there are only finitely many formulae of operator depth $\leq n$. By specifying which of them hold and which of them do not hold we see that each $\equiv_n$-class can be expressed by a $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ formula. Now, we have seen that $u \equiv_n v$ implies $\mu(u) = \mu(v)$. Hence, for all $p \in M$ the language $\mu^{-1}(p)$ is a finite union of $\equiv_n$-classes. We deduce that $\mu^{-1}(p)$ can be expressed by a $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ formula. The proposition follows since $L = \bigcup_{p \in \mu(L)} \mu^{-1}(p)$. □

Extending the definition of unambiguous monomials, we say that a product $L = L_0 a_1 L_1 \cdots a_k L_k$ with $a_i \in \Gamma$ and $L_i \subseteq \Gamma^*$ is *unambiguous* if for all $w \in L$ there exists a unique factorization $w = w_0 a_1 w_1 \cdots a_k w_k$ with $w_i \in L_i$.

Lemma 3

1. *Let $L = L_0 a_1 L_1 \cdots a_k L_k$ be an unambiguous product of unambiguous polynomials $L_0, \ldots, L_k$. Then L is also an unambiguous polynomial.*
2. *Let $A, B \subseteq \Gamma$. Then $[A, B] = \{w \mid A \subseteq \mathrm{alph}(w) \subseteq B\}$ is an unambiguous polynomial.*

Proof: "1": If each L_i is an unambiguous monomial, then L is also an unambiguous monomial. For the general case, we can write each $L_i = \bigcup_j L_{i,j}$ as a finite disjoint union of unambiguous monomials $L_{i,j}$. Then,

$$L = \bigcup L_{0,j_0} a_1 \cdots a_k L_{k,j_k}.$$

Since the product $L_0 a_1 L_1 \cdots a_k L_k$ is unambiguous, we see that this union is in fact a disjoint union. Moreover, each product $L_{0,j_0} a_1 \cdots a_k L_{k,j_k}$ is unambiguous, hence it is an unambiguous monomial.

"2": For $A = \emptyset$ we obtain $[A, B] = B^*$. Otherwise we can write

$$[A, B] = \bigcup_{a \in A} (B \setminus A)^* a [A \setminus \{a\}, B].$$

This is a disjoint union. Every language $(B \setminus A)^* a [A \setminus \{a\}, B]$ is an unambiguous polynomial using *"1"* and induction on the size of A. □

Proposition 2 *Every ranker class is an unambiguous polynomial.*

Proof: We fix $n \in \mathbb{N}$. Let $u \in \Gamma^*$ and consider the factorization $u = u_0 a_1 u_1 \cdots a_k u_k$ with $a_i \in \Gamma$ and $u_i \in \Gamma^*$ such that the a_i's correspond exactly to the positions in u reachable by a ranker of length at most n. Let $L(u) = L_0 a_1 L_1 \cdots a_k L_k$ with $L_i = \{w \mid \mathrm{alph}(w) = \mathrm{alph}(u_i)\}$.

Let $v \in L(u)$ and write $v = v_0 a_1 v_1 \cdots a_k v_k$ with $\mathrm{alph}(v_i) = \mathrm{alph}(u_i)$. By induction on the length of a ranker, we see that $\mathrm{RW}_n(v) = \mathrm{RW}_n(u)$. In particular, the above factorization of v is unique and hence, the product $L(u) = L_0 a_1 L_1 \cdots a_k L_k$ is unambiguous. Furthermore, it follows that $L(u) = L(v)$. Therefore, the ranker class of u of depth n is the finite disjoint union of languages of the form $L(v)$ with $\mathrm{RW}_n(v) = \mathrm{RW}_n(u)$. By Lemma 3 it follows that the ranker class of u of depth n is an unambiguous polynomial as desired. □

Proposition 3 *Every unambiguous monomial $L = A_0^* a_1 A_1^* \cdots a_k A_k^*$ is definable in* $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$.

Proof: We perform an induction on k. For $k = 0$ the result is obvious, hence we may assume $k \geq 1$. Since the product $L = A_0^* a_1 A_1^* \cdots a_k A_k^*$ is unambiguous, we cannot have all letters $a_1, \ldots, a_k$ contained in $A_0 \cap A_k$, because otherwise $(a_1 \cdots a_k)^2$ would admit two different factorizations. Thus, by symmetry we may assume that for some i we have $a_i \notin A_0$. Every word $w \in L$ can be written as $w = w' a_i w''$ with $a_i \notin \mathrm{alph}(w')$. There are two cases. The first one is that the left-most a_i in w is one of the a_j's, i.e., for some $1 \leq j \leq i$ we have:

$$w' \in A_0^* a_1 A_1^* \cdots a_{j-1} A_{j-1}^*, \quad a_i = a_j, \quad w'' \in A_j^* a_{j+1} A_{j+1}^* \cdots a_k A_k^*.$$

The other case is if the first a_i in w is not one of the a_j's, i.e., for some $0 \leq j < i$ we have:

$$w' \in A_0^* a_1 A_1^* \cdots a_j A_j^*, \quad a_i \in A_j, \quad w'' \in A_j^* a_{j+1} A_{j+1}^* \cdots a_k A_k^*.$$

Since $a_i \notin A_0$, we even have $1 \leq j < i \leq k$ in the second case. Note that the four expressions are unambiguous, because L is unambiguous. Thus, all four products above have shorter unambiguous expressions than L, and, by induction, we have formulae in $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$ describing them. Obviously, we can also express intersections with languages of type B^* for $B \subseteq \Gamma$. So there is a finite list of formulae in $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$ such that for each $w \in L$ there are φ and ψ from the list and a letter a with $w \in L(\varphi) a L(\psi) \subseteq L$ and $L(\varphi) \subseteq (\Gamma \setminus \{a\})^*$. Now, the first a-position in each $w \in L(\varphi) a L(\psi)$ is uniquely defined by $\xi_a = a \wedge \neg \mathsf{YP}\, a$. Using relativization techniques, we now define formulae $\varphi_{<a}, \psi_{>a} \in \mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$ such that

$$L(\varphi)\, a\, L(\psi) = L(\varphi_{<a} \wedge \mathsf{XF}\, a \wedge \psi_{>a}).$$

We give the inductive construction for $\varphi_{<a}$. The other one for $\psi_{>a}$ is symmetric.

$$\begin{aligned} b_{<a} &= b, & (\alpha \vee \beta)_{<a} &= \alpha_{<a} \vee \beta_{<a}, & (\mathsf{XF}\, \alpha)_{<a} &= \mathsf{XF}(\alpha_{<a} \wedge \mathsf{XF}\, \xi_a), \\ \top_{<a} &= \top, & (\neg \alpha)_{<a} &= \neg(\alpha_{<a}), & (\mathsf{YP}\, \alpha)_{<a} &= \mathsf{YP}(\alpha_{<a} \wedge \mathsf{XF}\, \xi_a). \end{aligned}$$

The formula for L becomes a disjunction of the formulae $\varphi_{<a} \wedge \mathsf{XF}\, a \wedge \psi_{>a}$. □

Proposition 4 *Every* $\mathrm{FO}^2[<]$*-definable language is recognized by a monoid in* **DA**.

Proof: It suffices to show that the syntactic monoid of an $\mathrm{FO}^2[<]$-definable language satisfies the defining equation $(uvw)^\omega v (uvw)^\omega = (uvw)^\omega$ of **DA**. Since $\mathrm{alph}(v) \subseteq \mathrm{alph}(uvw)$, it is enough to prove Lemma 4. □

Lemma 4 *Let $n \geq 1$ and φ be a sentence in $\mathrm{FO}^2[<]$ of quantifier depth at most n. Let $p, q, u, v \in \Gamma^*$ and $\mathrm{alph}(v) \subseteq \mathrm{alph}(u)$. Then we have $pu^n v u^n q \models \varphi$ if and only if $pu^{2n} q \models \varphi$.*

Proof: The assertion is trivial for $v = \varepsilon$. In particular, we may assume $u \neq \varepsilon$. In the proof we identify positions of $w' = pu^{2n}q$ with a subset of the positions of $w = pu^n v u^n q$ in a natural way. The positions of w' cover inside w the common prefix pu^n and the common suffix $u^n q$. Hence, we consider positions x, y, x', y' in w only, but x', y' are never taken from the middle factor v. It follows that x' and y' are positions in w'. We say that a tuple (x, y, x', y') is *legal* if

$$\begin{gathered}\text{neither } x' \text{ nor } y' \text{ lies in } v,\\ x = y \;\Leftrightarrow\; x' = y',\\ x < y \;\Leftrightarrow\; x' < y',\\ \lambda(x) = \lambda(x') \;\text{ and }\; \lambda(y) = \lambda(y').\end{gathered}$$

For $n \geq k \geq 0$ we let B_k be a ball around v. More precisely, we define B_k by set of positions of the middle factor $u^k v u^k$ of w. Below, we depict the positions in B_k and B_{k+1}:

$$w \;=\; pu^{n-k-1}\; \underbrace{u\;\overbrace{u^k\,v\,u^k}^{B_k}\;u}_{B_{k+1}}\; u^{n-k-1}q.$$

We say that (x, y, x', y') is *k-close*, if it is legal and if in addition the following conditions hold:

$$\begin{gathered}x = x' \;\text{ or }\; x, x' \in B_k,\\ y = y' \;\text{ or }\; y, y' \in B_k.\end{gathered}$$

For example, let z be the first position in w. Then (z, z, z, z) is 0-close, because it is legal due to $n \geq 1$ (otherwise pu^n could be empty).

We are going to prove the following claim by induction on k: If (x, y, x', y') is k-close and if $\varphi(x, y)$ is an $\mathrm{FO}^2[<]$-formula with quantifier depth at most $n-k$ then

$$w, x, y \models \varphi(x, y) \;\Leftrightarrow\; w', x', y' \models \varphi(x', y').$$

For $k = n$ the claim holds since (x, y, x', y') is legal and $\varphi(x, y)$ is a Boolean combination of atomic formulae. Let now $k < n$. Without restriction we can assume that $\varphi(x, y) = \exists x\, \psi(x, y)$. Suppose $w, x, y \models \varphi(x, y)$. Then there exists a position x_1 in w such that $w, x_1, y \models \psi(x_1, y)$. If $x_1 = y$, then we can choose $x_1' = y'$ and obtain a $(k+1)$-close tuple (x_1, y, x_1', y') and by induction, $w', x_1', y' \models \psi(x_1', y')$. Therefore, $w', x', y' \models \varphi(x', y')$.

If $x_1 < y$ and $x_1 \notin B_k$, then we can choose $x_1' = x_1$ so that $x_1' < y'$ and we obtain a $(k+1)$-close tuple (x_1, y, x_1', y'). If $x_1 < y$ and $x_1 \in B_k$, then we can choose x_1' to be the first position in B_{k+1} with label $\lambda(x_1)$. This is possible since $\mathrm{alph}(v) \subseteq \mathrm{alph}(u)$. Note that $x'_1 \in B_{k+1} \setminus B_k$, because u is nonempty. Hence, $x'_1 < y'$ and again, (x_1, y, x'_1, y') is $(k+1)$-close. As before, we see that $w', x', y' \models \varphi(x', y')$. The case $x_1 > y$ is symmetric to the case $x_1 < y$. Therefore, the implication $w, x, y \models \varphi(x, y) \Rightarrow w', x', y' \models \varphi(x', y')$ holds. The converse $w', x', y' \models \varphi(x', y') \Rightarrow w, x, y \models \varphi(x, y)$ is similar, but slightly easier since it does not rely on $\mathrm{alph}(v) \subseteq \mathrm{alph}(u)$.

This proves the lemma, since we can apply the above claim where $k = 0$ and $x = y = x' = y'$ is the first position of w. □

Proof (Theorem 2): The implication *"1 ⇒ 2"* is Proposition 1. The implication *"2 ⇒ 3"* follows from the following equivalences of formulae in $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$: The formula $\mathsf{X}_a \neg\varphi$ is equivalent to $\mathsf{X}_a \top \wedge \neg \mathsf{X}_a \varphi$. Also, $\mathsf{X}_a(\varphi \vee \psi)$ and $\mathsf{X}_a(\varphi \wedge \psi)$ are equivalent to $\mathsf{X}_a \varphi \vee \mathsf{X}_a \psi$ and $\mathsf{X}_a \varphi \wedge \mathsf{X}_a \psi$, respectively. Symmetric equivalences hold for Y_a. Hence, any formula $\varphi \in \mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ with operator depth at most n is equivalent to a Boolean combination of rankers of length at most n. Therefore, $\mathrm{RW}_n(u) = \mathrm{RW}_n(v)$ implies that u and v satisfy the same formulae with operator depth at most n. Hence, $L(\varphi)$ is a union of ranker classes of depth n. The direction *"3 ⇒ 4"* follows from Proposition 2, since by increasing the depth of the rankers, we can assume that every ranker language is a disjoint union of ranker classes. The direction *"4 ⇒ 5"* follows from Proposition 3 and the implication *"5 ⇒ 6"* is trivial since every $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$-formula can be immediately translated into an $\mathrm{FO}^2[<]$-formula. Finally, *"6 ⇒ 1"* is Proposition 4. The characterizations *"7"* and *"8"* are postponed to Theorem 9. □

5 Ordered monoids

If a monoid M recognizes a language $L \subseteq \Gamma^*$, then M recognizes its complement $\Gamma^* \setminus L$, too. Therefore, this notion of recognition by finite monoids is inadequate for language classes which are not closed under complementation. Such a language class is given for example by polynomials. The idea of Pin [46] to cope with classes which are not closed under complementation is to consider ordered monoids instead; and to refine the notion of recognizability. A pair $(M, \leq)$ forms an *ordered monoid* if M is a monoid and $\leq$ is a partial order on M such that for all $u, v, v', w \in M$ we have

$$v \leq v' \quad \text{implies} \quad uvw \leq uv'w,$$

i.e., $\leq$ is compatible with multiplication. In particular, $(M, =)$ is an ordered monoid. An ordered monoid comes with a natural closure operator. A subset $D \subseteq M$ is called *downward closed*, if $q \leq p \in D$ implies $q \in D$. For $D \subseteq M$ we denote by $\downarrow\! D$ the smallest downward closed subset containing D. We say that a language $L \subseteq \Gamma^*$ is recognized by an ordered monoid $(M, \leq)$ via a homomorphism $\mu : \Gamma^* \to M$, if L is the inverse image of a downward closed subset, i.e.,

$$L = \mu^{-1}(\downarrow\! \mu(L)).$$

Requiring that L is the inverse image of a downward closed subset, restricts the class of languages which can be recognized by an ordered monoid.

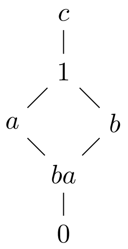
The notion is very natural in our context, because the syntactic monoid of a language L has a canonical order. For words u and v one defines $u \leq_L v$ by:

$$\forall p, q \in \Gamma^* \colon pvq \in L \;\Rightarrow\; puq \in L.$$

The preorder $\leq_L$ over Γ^* induces a partial order over the syntactic monoid $M(L)$ such that $(M(L), \leq_L)$ forms an ordered monoid called the *syntactic ordered monoid* of L. Note that the image of L under its syntactic homomorphism is downward closed in $(M(L), \leq_L)$. The syntactic ordered monoid is the smallest ordered monoid which recognizes L. Actually, if L is recognized by any ordered monoid $(M, \leq)$ via a homomorphism μ, then μ induces a monotone surjective homomorphism from $(\mu(\Gamma^*), \leq)$ onto $(M(L), \leq_L)$. Finally note that if $(M, \leq)$ recognizes L, then $(M, \geq)$ recognizes $\Gamma^* \setminus L$.

Let us consider again a polynomial L of degree k and words u, v with $\mathrm{alph}(v) \subseteq \mathrm{alph}(u)$. Then for all p, q we have that $pu^{k+1}q \in L$ implies that $pu^k v u^k q \in L$, too. Thus, the syntactic ordered monoid of L satisfies an equation of type $u^\omega v u^\omega \leq_L u^\omega$ as soon as some alphabetic constraints are satisfied. We will make this precise later, but we can see it on an example.

Example 5 As we have seen in Example 2, the syntactic monoid of $\Gamma^* ab \Gamma^*$ can be represented by the elements $\{1, a, b, c, ba, 0\}$. The order relation of the syntactic ordered monoid is depicted in the following diagram:



The element 0 is the minimal element and c is the maximal element. ◇

6 Existential first-order logic

A language $L \subseteq \Gamma^*$ is called a *simple polynomial*, if it is a finite union of languages of the form $\Gamma^* a_1 \Gamma^* \cdots a_k \Gamma^*$. The aim of this section is to prove the following characterization of the fragment $\Sigma_1[<]$.

Theorem 3 *Let $L \subseteq \Gamma^*$. Then the following assertions are equivalent:*

1. *L is recognized by a finite ordered monoid $(M, \leq)$ which satisfies $u \leq 1$ for all $u \in M$.*
2. *L is regular and for all $p, q, w \in \Gamma^*$ with $pq \in L$ we have $pwq \in L$, too.*
3. *L is a simple polynomial.*
4. *L is definable in the existential first-order logic $\Sigma_1[<]$.*

Property *"2"* is a reformulation of *"1"* in terms of the syntactic ordered monoid. The equivalence of *"1"* and *"3"* is due to Pin [46] and the correspondence of simple polynomials and $\Sigma_1[<]$ is due to Thomas [84]. In fact, they have given a more general connection between the fragments $\Sigma_n[<]$ and a language operation called the *polynomial closure*, see [17, 37, 66, 67, 79, 87] for hierarchies involving this language operation. A profound relation between the polynomial closure and an algebraic operation called *Mal'cev product* has been shown by Pin and Weil [54]. The equivalence of *"1"* and *"3"* can be seen as a special case of this relation. From Theorem 3 together with Example 1 it follows that every $\Sigma_1[<]$-definable language is also $\mathrm{FO}^2[<]$-definable.

Proof (Theorem 3): The equivalence *"1 ⇔ 2"* is trivial. To see *"2 ⇒ 3"*, fix any automaton which accepts the regular language L. Assume that it has n states, and let $a_1 \cdots a_k$ be the label of a path that visits every state at most once from some initial to a final state. We have $k \leq n$, and it is clear that the simple monomial $\Gamma^* a_1 \Gamma^* \cdots a_k \Gamma^*$ is a subset of L. Hence L is a finite union of simple monomials of degree at most n. For *"3 ⇒ 4"* note that $\Sigma_1[<]$ is closed under disjunction. It therefore suffices to show that every monomial of the form $\Gamma^* a_1 \Gamma^* \cdots a_k \Gamma^*$ is definable in $\Sigma_1[<]$, which is obvious. For *"4 ⇒ 2"* let φ be a propositional formula with free variables $x_1, \ldots, x_k$. Suppose $pq \models \exists x_1 \ldots \exists x_k\, \varphi$. We have to show that $pwq \models \exists x_1 \ldots \exists x_k\, \varphi$ for all $w \in \Gamma^*$, but this is trivial again, because we can choose the k positions inside p and q. □

Remark 2 *The property that L is regular in assertion "2" of Theorem 3 is redundant. Indeed, let L be any subset of Γ^*, then Higman's Lemma [27] says that there is a finite list of words $a_1 \cdots a_k \in L$ such that every other word $w \in L$ contains one of these $a_1 \cdots a_k$ as a subword. Therefore, if $pq \in L$ implies $pwq \in L$ for all $p, q, w \in \Gamma^*$, then L is a simple polynomial.*

Phrased differently, Higman's Lemma says that the subword ordering is a well-quasi-ordering*. A short proof of this fact can be found in [56].*

7 The Boolean closure of existential first-order logic

The first-order fragment $\mathbb{B}\Sigma_n[<]$ consists of all Boolean combinations of $\Sigma_n[<]$-formulae. In particular, $\mathbb{B}\Sigma_n[<]$ is closed under complementation. It follows that $\mathbb{B}\Sigma_n[<] = \mathbb{B}\Pi_n[<]$ where $\mathbb{B}\Pi_n[<]$ contains all Boolean combinations of $\Pi_n[<]$-formulae. In this section $u \sim_k v$ means for words u and v that u and v have exactly the same subwords of length up to k. Note that $\sim_k$ is a congruence of finite index (for every finite alphabet). A language is called *piecewise-testable*, if it is a union of $\sim_k$-classes for some $k \in \mathbb{N}$. An easy reflection shows that piecewise-testable languages can be defined in the first-order fragment $\mathbb{B}\Sigma_1[<]$.

It turns out that there is also an algebraic characterization by so-called $\mathcal{J}$-trivial monoids, known as Simon's Theorem [60]. A monoid M is called *$\mathcal{J}$-trivial*, if $MuM = MvM$ (i.e., $u \mathrel{\mathcal{J}} v$ in terms of Green's relation) implies $u = v$ for all $u, v \in M$. The aim of this section is to provide the following characterization of the Boolean closure of existentially first-order definable languages, i.e., of $\mathbb{B}\Sigma_1[<]$.

Theorem 4 (Simon) *Let $L \subseteq \Gamma^*$. The following assertions are equivalent:*

1. *L is recognized by some finite and $\mathcal{J}$-trivial monoid.*
2. *L is piecewise-testable.*
3. *L is definable in $\mathbb{B}\Sigma_1[<]$.*

The direction from *"2"* to *"3"* is trivial. We will see that the direction from *"3"* to *"1"* is actually a corollary of Theorem 3. Hence the key point in Simon's Theorem is the direction from *"1"* to *"2"*. We concentrate on this. First, we consider $\mathcal{J}$-trivial monoids in order to prove a result of Straubing and Thérien [71]. Our proof follows the original proof in [71].

Example 6 In Example 2 we have considered the syntactic monoids $\{1, a, b, ba, 0\}$ and $\{1, a, b, c, ba, 0\}$ of $L = \Gamma^* ab \Gamma^*$. A direct calculation shows that the first one is $\mathcal{J}$-trivial, whereas the larger one is not, because a, b, ba, c are $\mathcal{J}$-equivalent. Therefore, L is definable in $\mathbb{B}\Sigma_1[<]$ if and only if $|\Gamma| = \{a, b\}$. ◇

All finite $\mathcal{J}$-trivial monoids are aperiodic. Actually, they are in **DA**. In every finite monoid we have:

$$(uv)^\omega = (uv)^\omega (uv)^\omega \leq_{\mathcal{J}} (uv)^\omega u \leq_{\mathcal{J}} (uv)^\omega.$$

Therefore, $(uv)^\omega u$ and $(uv)^\omega$ are $\mathcal{J}$-equivalent (in fact, they are $\mathcal{R}$-equivalent). Now, if M is a finite $\mathcal{J}$-trivial monoid, we have $(uv)^\omega u = (uv)^\omega$ for all $u, v \in M$. Using Remark 1, we conclude $M \in$ **DA**.

The infinite monoid $\mathbb{N} = (\mathbb{N}, +, 0)$ is $\mathcal{J}$-trivial, whereas its quotient $\mathbb{Z}/2\mathbb{Z}$ is not $\mathcal{J}$-trivial since it is a non-trivial group. Hence, a homomorphic image of a $\mathcal{J}$-trivial monoid needs not to be $\mathcal{J}$-trivial. Again, finiteness is crucial. See the next lemma for some basic properties of $\mathcal{J}$-trivial monoids.

Lemma 5 *Let M be a finite $\mathcal{J}$-trivial monoid.*

1. *The product over all elements of M is a zero-element in M.*
2. *Let $\mu : M \to N$ be a surjective homomorphism onto a monoid N. Then N is $\mathcal{J}$-trivial.*

Proof: *"1"*: Let u be a product over all elements of M. Then $u \leq_{\mathcal{J}} y$ for all $y \in M$. In particular, for all $x \in M$ we have $u \leq_{\mathcal{J}} xu$ and we deduce that $u \mathrel{\mathcal{J}} xu$, which implies $u = xu$ since M is $\mathcal{J}$-trivial. Similarly, $u = ux$ for all $x \in M$ and we have shown that u is a zero.

"2": Let $u, v \in M$ and suppose $\mu(u)$ and $\mu(v)$ are $\mathcal{J}$-equivalent in N. We have to show $\mu(u) = \mu(v)$. There exist $x, y, \overline{x}, \overline{y} \in M$ with $\mu(xuy) = \mu(v)$ and $\mu(\overline{x} v \overline{v}) = \mu(u)$. Define $u' = (\overline{x}x)^\omega u (y\overline{y})^\omega$ and $v' = xu'y$. Now, u' and v' are $\mathcal{J}$-equivalent in M. Since M is $\mathcal{J}$-trivial, we conclude $u' = v'$. By construction, $\mu(u) = \mu(\overline{x}xuy\overline{y}) = \mu(u')$ and therefore $\mu(u) = \mu(u') = \mu(v') = \mu(v)$. □

We obtain a family of examples of $\mathcal{J}$-trivial monoids by the family of ordered monoids $(M, \leq)$ where 1 is the greatest element. These monoids are always $\mathcal{J}$-trivial, because $pvq \leq 1 \cdot v \cdot 1 = v$ shows $u \leq v$ if $u \leq_{\mathcal{J}} v$. Hence, $u \mathrel{\mathcal{J}} v$ implies $u = v$. The following theorem clarifies the relation between $\mathcal{J}$-trivial monoids and ordered monoids of the above type.

Theorem 5 (Straubing and Thérien) *A finite monoid M is $\mathcal{J}$-trivial if and only if it is a homomorphic image of a finite ordered monoid $(K, \leq)$ satisfying $u \leq 1$ for all $u \in K$.*

Proof: Suppose $(K, \leq)$ satisfies $u \leq 1$ for all $u \in K$, then K is $\mathcal{J}$-trivial as explained above. By Lemma 5, all homomorphic images of a $\mathcal{J}$-trivial monoid are $\mathcal{J}$-trivial. Therefore we have to show that a finite $\mathcal{J}$-trivial monoid is a quotient monoid (i.e., homomorphic image) of a finite ordered monoid $(K, \leq)$ satisfying $u \leq 1$.

Thus, let M be finite and $\mathcal{J}$-trivial. We construct the finite ordered monoid $(K, \leq)$ inductively on the size of M. Consider the natural homomorphism

$$\eta : M \to \prod_{x \neq 0} M/MxM$$

where $M/MxM = M \setminus MxM \cup \{MxM\}$ denotes the Rees-quotient of M by the ideal MxM, i.e., all elements in MxM are identified with the new *element* MxM. Note that if $|MxM| = 1$ then $x = 0$, hence $|M/MxM| < |M|$ if $x \neq 0$. If η is injective, we are done by induction. So assume, there are $m, n \in M$ with $\eta(m) = \eta(n)$ and $m \neq n$. We may assume $n \neq 0$. We show that this implies $m = 0$. Indeed, by contradiction assume $m \neq 0$ as well. Then $m \in MnM$ and $n \in MmM$ because $\eta(m) = \eta(n)$. But M is $\mathcal{J}$-trivial, hence $m = n$, which contradicts the assumption. Hence $m = 0$ and thus $n \in \bigcap_{x \neq 0} MxM$. In fact, we obtain:

$$\{0, n\} = \bigcap_{x \neq 0} MxM.$$

So the ideal $N = \bigcap_{x \neq 0} MxM$ has exactly two elements.

Next, assume $n^2 = n$. We show that this implies that $M \setminus \{0\}$ is a submonoid. Let $x, y \in M \setminus \{0\}$, we have to show that the product xy is still in $M \setminus \{0\}$. Since $n \in MxM \cap MyM$, we have $n = pxq = syt$ for some $p, q, s, t \in M$ and hence

$$n = n^2 = pxqpxq \leq_{\mathcal{J}} pxqpx \leq_{\mathcal{J}} pxq = n.$$

Since M is $\mathcal{J}$-trivial, we obtain $n = pxqpx \in Mx$. By symmetry we also have $n = ytsyt \in yM$. But this implies $n = n^2 \in MxyM$ and therefore $xy \neq 0$. Now, $M \setminus \{0\}$ is a submonoid, and by induction, it is a quotient of some finite ordered monoid $(K, \leq)$ satisfying $u \leq 1$. We may add a new zero element to K as a least element and we are done.

Hence from now on, $N = \{0, n\} = \bigcap_{x \neq 0} MxM$ with $0 \neq n$ and $n^2 = 0$ since $n^2 \in N \setminus \{n\}$. In particular, 0, n, and 1 are three different elements. We choose a finite set Γ and a surjective homomorphism

$$\mu : \Gamma^* \to M.$$

We let $\widetilde{M} = M/N$ be the quotient monoid and

$$\widetilde{\mu} : \Gamma^* \to M \to \widetilde{M}.$$

Remember that we realize $\widetilde{M}$ as $(M \setminus N) \cup \{N\}$. Thus, $\widetilde{\mu}(u) = \widetilde{\mu}(v) \neq N$ implies $\mu(u) = \mu(v)$ for all words $u, v \in \Gamma^*$.

By induction, $\widetilde{M}$ is a quotient of some finite ordered monoid $(\widetilde{K}, \preceq)$ satisfying $u \preceq 1$. Let $\kappa : \Gamma^* \to \widetilde{K}$ be a lifting of $\widetilde{\mu}$. Thus, $\kappa(u) = \kappa(v)$ implies $\widetilde{\mu}(u) = \widetilde{\mu}(v)$ for all words $u, v \in \Gamma^*$. Moreover, we may assume that κ is a surjective homomorphism. We let $\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$. Thus, $\Gamma_\varepsilon \subseteq \Gamma^*$. For $a, b \in \Gamma_\varepsilon$ we write $a \preceq b$, if either $a = b$ or $b = \varepsilon$.

The crucial step comes now. We are considering the Birget-Rhodes construction. For each word $u \in \Gamma^*$ we define

$$\varphi(u) = \left\{ (\kappa(p), a, \kappa(q)) \in \widetilde{K} \times \Gamma_\varepsilon \times \widetilde{K} \;\middle|\; u = paq \right\}.$$

Note that $\varphi(u)$ is never empty and that there are only finitely many different values for $\varphi(u)$. We introduce a natural preorder $\leq$ on the finite set $\{\varphi(u) \mid u \in \Gamma^*\}$ such that $\varphi(u) \leq \varphi(u')$ if and only if

$$\forall (x, a, y) \in \varphi(u) \, \exists (x', a', y') \in \varphi(u') \colon \; x \preceq x' \;\wedge\; a \preceq a' \;\wedge\; y \preceq y'.$$

We have the following facts for all words u, u', v and w:

1. $\varphi(u) \leq \varphi(\varepsilon)$,
2. $\varphi(u) \leq \varphi(u')$ implies $\kappa(u) \preceq \kappa(u')$,
3. $\varphi(u) \leq \varphi(u')$ implies $\varphi(vuw) \leq \varphi(vu'w)$.

Define a relation $\approx \subseteq \Gamma^* \times \Gamma^*$ such that $u \approx u'$ if and only if both $\varphi(u) \leq \varphi(u')$ and $\varphi(u') \leq \varphi(u)$. Then $\approx$ is a congruence due to 3. The preorder $\leq$ induces an order on the finite quotient monoid

$$K = \Gamma^* / \approx .$$

Thus, $(K, \leq)$ is an ordered monoid satisfying $u \leq 1$ for all $u \in K$ due to 1. The homomorphism $\kappa : \Gamma^* \to \widetilde{K}$ induces a surjective homomorphism from K to $\widetilde{K}$ which respects the ordering due to 2. It is clear that $\widetilde{\mu} : \Gamma^* \to \widetilde{M}$ factorizes through K. We want to show that $\mu : \Gamma^* \to M$ factorizes through K. We fix words $u, u' \in \Gamma^*$ such that $u \approx u'$. We have to show that $\mu(u) = \mu(u')$. We know $\widetilde{\mu}(u) = \widetilde{\mu}(u')$. Hence, if $\widetilde{\mu}(u') \neq N$ we are done. If $\mu(u) = 0$ and $\mu(u') = 0$, then we are done, too. Hence we may assume that $\mu(u) = n$ and we have to show that $\mu(u') = n$, too.

Consider the factorization $u = paq$ with $\mu(p) \notin N$ and $\mu(pa) = n$, where $a \in \Gamma$ is a letter. We cannot have $\mu(q) \in N$, because this would mean $\mu(u) = n^2 = 0$, but $n \neq 0$. Hence there is $(\kappa(p), a, \kappa(q)) \in \varphi(u)$ with $\mu(p) \neq n \neq \mu(q)$. Choose some maximal triple $(x, b, y) \in \varphi(u)$ such that $\kappa(p) \preceq x$ and $a \preceq b$ and $\kappa(q) \preceq y$. Because $\varphi(u) \leq \varphi(u') \leq \varphi(u)$ there are triples $(x', b', y') \in \varphi(u')$ and $(x'', b'', y'') \in \varphi(u)$ with

$$x \preceq x' \preceq x'' \;\wedge\; b \preceq b' \preceq b'' \;\wedge\; y \preceq y' \preceq y''.$$

Due to the maximality of (x, b, y) we obtain

$$x = x' \;\wedge\; b = b' \;\wedge\; y = y'.$$

We choose factorizations $u = rbs$ and $u' = r'bs'$ with $\kappa(r) = \kappa(r') = x$ and $\kappa(s) = \kappa(s') = y$. Note that this implies $\widetilde{\mu}(r) = \widetilde{\mu}(r')$ and $\widetilde{\mu}(s) = \widetilde{\mu}(s')$. Since $\mu(u) = \mu(r)\mu(b)\mu(s)$ and $\mu(u') = \mu(r')\mu(b)\mu(s')$, it is enough to show that both $\widetilde{\mu}(r) \neq N$ and $\widetilde{\mu}(s) \neq N$. The equalities $\mu(r) = \mu(r')$ and $\mu(s) = \mu(s')$ then follow.

By symmetry it is enough to show $\widetilde{\mu}(r) \neq N$. There are two cases, either r is a prefix of p or vice versa. If r is a prefix of p, then we cannot have $\mu(r) \in N$, because $\mu(p) \notin N = NM$. Thus, we can assume that p is a prefix of r. This implies $\kappa(r) \preceq \kappa(p)$, but $\kappa(p) \preceq x = \kappa(r)$ and hence $\kappa(r) = \kappa(p)$. Now, $\mu(p) \notin N$ implies $\widetilde{\mu}(p) \neq N$ and therefore $\widetilde{\mu}(r) \neq N$ as desired. □

Theorem 6 (Simon) *Every language that is recognized by some finite $\mathcal{J}$-trivial monoid is piecewise-testable.*

Proof: Let $\mu : \Gamma^* \to M$ be a homomorphism such that M is a finite and $\mathcal{J}$-trivial monoid. By Theorem 5 we can assume that there exists a partial order relation $\leq$ on M such that $(M, \leq)$ is an ordered monoid satisfying $p \leq 1$ for all $p \in M$. By Theorem 3 we know that $\mu^{-1}(\downarrow p)$ is a simple polynomial, so it is piecewise-testable. Thus, $\mu^{-1}(p) = \mu^{-1}(\downarrow p) \setminus \bigcup_{q<p} \mu^{-1}(\downarrow q)$ is piecewise-testable, too. □

Proof (Theorem 4): The direction *"1 ⇒ 2"* is exactly Theorem 6.

"2 ⇒ 3": Every $\sim_k$-class is uniquely described by the set of words of length $\leq k$ that occur as a subword and by those that do not occur as a subword. Since $\mathbb{B}\Sigma_1[<]$ is by definition closed under Boolean operations, it suffices to show that there exists a $\Sigma_1[<]$ formula that describes the set of words which contain the subword $a_1 \cdots a_k$, which is obvious and has also been used in the proof of Theorem 3.

"3 ⇒ 1": By Theorem 3, every language definable in $\Sigma_1[<]$ is recognized by a finite ordered monoid $(M, \leq)$ satisfying $u \leq 1$. By Theorem 5, M is $\mathcal{J}$-trivial. Hence, any Boolean combination of $\Sigma_1[<]$ languages can be recognized by a direct product of $\mathcal{J}$-trivial monoids, which is $\mathcal{J}$-trivial itself. □

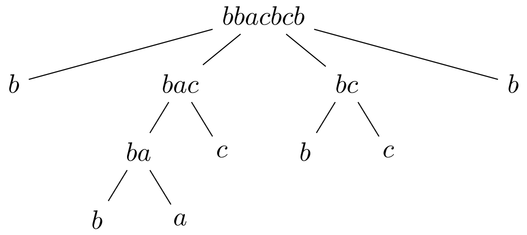
8 Factorization forests and polynomials

Let M be a finite monoid. A *factorization forest* of a homomorphism $\mu : \Gamma^* \to M$ is a function d which maps every word w with length $|w| \geq 2$ to a factorization $d(w) = (w_1, \ldots, w_n)$ of $w = w_1 \cdots w_n$ with $|w_i| < |w|$ for all i and such that $n \geq 3$ implies both $\mu(w_1) = \cdots = \mu(w_n)$ and $\mu(w_1)$ is idempotent in M. The *height* of a word w is defined as

$$h(w) = \begin{cases} 0 & \text{if } |w| \leq 1, \\ 1 + \max\{h(w_1), \ldots, h(w_n)\} & \text{if } d(w) = (w_1, \ldots, w_n). \end{cases}$$

We can think of the words of length ≤ 1 as the trees of height 0. If $d(w) = (w_1, \ldots, w_n)$ and if all w_i's are roots of trees, then w is the root of a tree with children $w_1, \ldots, w_n$. Now, w is the root of the tree defined by the "branching" d and $h(w)$ is the height of this tree.

Example 7 Let $\Gamma = \{a, b, c\}$ and let $\mu : \Gamma^* \to M$ be the syntactic homomorphism of $\Gamma^* ab \Gamma^*$ as in Example 2. Then $w = bbacbcb$ has some factorization tree of height 3 and it is easy to check that this height is minimal for w:



◇

Since we consider all words simultaneously, we obtain a forest. The *height* of the factorization forest d is therefore finite if and only if $h(w) \leq n$ for all $w \in \Gamma^*$ and some $n \in \mathbb{N}$. A famous theorem of Simon, Theorem 7, says that every homomorphism $\mu : \Gamma^* \to M$ has a factorization forest of finite height, see [61, 62, 63] for the original papers.

Before we go into details of the proof of Theorem 7 let us see how we can apply this concept. The idea is that every factorization forest of height h comes with a canonical finite set of monomials of degree at most 2^h. Let d be a factorization forest of height h. We can assume that all factors w_i in a factorization $d(w) = (w_1, \ldots, w_n)$ are nonempty. For each word w we define a monomial $P_d(w)$ as follows. For $|w| \leq 1$ we let $P_d(w) = \{w\}$. If $d(w) = (w_1, w_2)$, we let $P_d(w) = P_d(w_1)\, P_d(w_2)$. Finally, if $d(w) = (w_1, \ldots, w_n)$ with $n \geq 3$, then we let $P_d(w) = P_d(w_1) \cdot \mathrm{alph}(w)^* \cdot P_d(w_n)$. Note that the degree of $P_d(w)$ is indeed bounded by $2^{h(w)}$. This bound holds, because $P_d(w)$ begins and ends with a letter, if w is nonempty. Using these monomials, Simon's result yields the following application.

Proposition 5 *Let $\mu : \Gamma^* \to (M, \leq)$ be a homomorphism to a finite ordered monoid. If each idempotent $e \in M$ is the greatest element in the subsemigroup $e\,\{s \mid e \in MsM\}^*\,e$, then $\mu^{-1}(\downarrow p)$ is a polynomial of degree at most $2^{3|M|}$ for all $p \in M$.*

Proof: We can assume that μ is surjective. The algebraic property of M applied with $s = \mu(u)$ and $e = s^\omega$ implies $\mu(u)^{\omega+1} = \mu(u)^\omega \mu(u) \mu(u)^\omega \leq \mu(u)^\omega$. It follows

$$\mu(u)^\omega \geq \mu(u)^{\omega+1} \geq \mu(u)^{\omega+2} \geq \mu(u)^{\omega+3} \geq \cdots \geq \mu(u)^{2\omega} = \mu(u)^\omega,$$

i.e., M is aperiodic. By Theorem 7 below there exists a factorization forest d whose height is bounded by $3\,|M|$ for the homomorphism μ. For a word w, consider the canonical monomial $P_d(w)$ defined above. We show that

$$\mu^{-1}(\downarrow p) = \bigcup_{\mu(w) \leq p} P_d(w).$$

Since the height of d is finite, there are only finitely many monomials of the form $P_d(w)$ and hence this union is finite. The assertion $w \in P_d(w)$ is trivial. All we have to show is that $v \in P_d(w)$ implies $\mu(v) \leq \mu(w)$. This is trivial if $|w| \leq 1$ or if $d(w) = (w_1, w_2)$ by induction. Now, let $d(w) = (w_1, \ldots, w_n)$ with $n \geq 3$. In particular, $\mu(w_1) = \mu(w_n) = \mu(w)$ and this element is idempotent. If $v \in P_d(w) = P_d(w_1) \cdot \mathrm{alph}(w)^* \cdot P_d(w_n)$, then $v = v_1 u v_n$ with $v_1 \in P_d(w_1)$, $\mathrm{alph}(u) \subseteq \mathrm{alph}(w)$, and $v_n \in P_d(w_n)$. We have

$$\mu(v) = \mu(v_1 u v_n) \leq \mu(w_1 u w_n) = \mu(wuw) \leq \mu(w)$$

where the first inequality follows by induction and the last inequality from the assumption on M, since $\mathrm{alph}(u) \subseteq \mathrm{alph}(w)$ implies $\mu(u) \in \{s \mid \mu(w) \in MsM\}^*$. □

Example 8 Let $\Gamma = \{a, b, c\}$ and let $(M, \leq)$ be the ordered syntactic monoid of $\Gamma^* ab \Gamma^*$ as in Example 2, i.e., $M = \{1, a, b, c, ba, 0\}$ with the ordering depicted in Example 5. Let us see whether the condition in Proposition 5 holds.

The idempotent c is on the top and $e\{s \mid e \in MsM\}^* e \subseteq \{0, e\}$ for the other four idempotents, so the condition is valid. Note that the condition is violated for $(M, \geq)$, which is the syntactic ordered monoid of $\Gamma^* \setminus \Gamma^* ab \Gamma^*$. If, on the other hand, $\Gamma = \{a, b\}$ and c is missing, then it is enough to consider $M' = \{1, a, b, ba, 0\}$. Now, we always have $e\{s \mid e \in M'sM'\}^* e = \{e\}$, so the condition holds for all orderings of M'. ◇

It remains to show the following theorem which we state in its general form. With a weaker bound, it is due to Simon [62].

Theorem 7 *Let M be a finite monoid. Every homomorphism $\mu : \Gamma^* \to M$ has a factorization forest of height $\leq 3|M|$.*

The proof is far from trivial and rather technical. In order to avoid too much machinery, we give the proof for aperiodic monoids, only. This is sufficient for our application in Proposition 5. A concise proof for the general case can be found in [13] and we use similar techniques here. The main difference is that we give an improved bound for the height. A full proof with the bound above is in [18] or [34]. Our proof (as many others) is based on Green's relations $\mathcal{J}$, $\mathcal{R}$, and $\mathcal{L}$, as defined in Section 2. We start with two auxiliary results.

Lemma 6 *Let M be a finite monoid. If $u \mathrel{\mathcal{J}} v$, then there exists $w \in M$ with $u \mathrel{\mathcal{R}} w$ and $w \mathrel{\mathcal{L}} v$.*

Proof: Since $u \mathrel{\mathcal{J}} v$, there exist $x, \overline{x}, y, \overline{y} \in M$ such that $v = xuy$ and $u = \overline{x}v\overline{y}$. Let $w = uy$. We have

$$u = \overline{x}x \cdot u \cdot y\overline{y} = (\overline{x}x)^\omega u (y\overline{y})^\omega = (\overline{x}x)^\omega u (y\overline{y})^\omega (y\overline{y})^\omega = u(y\overline{y})^\omega \in uyM.$$

It follows that $u \mathrel{\mathcal{R}} uy = w$. A symmetric reasoning as above shows $u = (\overline{x}x)^\omega u$. Therefore, $uy = (\overline{x}x)^\omega uy \in Mxuy = Mv$ and hence $w = uy \mathrel{\mathcal{L}} v$. □

Lemma 7 *Let M be a finite monoid. If $u \mathrel{\mathcal{J}} v$ and $u \leq_{\mathcal{R}} v$ (resp. $u \leq_{\mathcal{L}} v$), then $u \mathrel{\mathcal{R}} v$ (resp. $u \mathrel{\mathcal{L}} v$).*

Proof: There exist $x, y, z \in M$ with $u = vx$ and $v = zuy$. We have

$$v = zuy = zvxy = z^\omega v (xy)^\omega = z^\omega v (xy)^\omega (xy)^\omega = v(xy)^\omega \in vxM = uM.$$

This shows $v \leq_{\mathcal{R}} u$ and hence $u \mathrel{\mathcal{R}} v$. The case for $u \leq_{\mathcal{L}} v$ is symmetric. □

Proof (of Theorem 7 for aperiodic monoids): Let $[w]$ denote $\mu(w)$. In the proof we use the following notion. A *relaxed* factorization tree for a word w is obtained from a usual factorization tree by replacing each subtree whose root is labeled with a word v which is not in the $\mathcal{J}$-class of w by a leaf labeled v. In the usual definition, leaves are letters. Here we are more flexible and we allow leaves from $\{\varepsilon\} \cup \Gamma \cup \{v \mid [w] <_{\mathcal{J}} [v]\}$. Thus, a relaxed factorization tree for w is constructed top-down using the rules of usual factorization trees until each leaf has a label which is either the empty word, or a letter, or belongs to another $\mathcal{J}$-class which is greater in the ordering $<_{\mathcal{J}}$.

A simple reflection shows that it is enough to construct for each word w a relaxed factorization tree of height at most $3|\{x \in M \mid [w] \mathrel{\mathcal{J}} x\}|$. Indeed, the resulting (usual) factorization forest has height at most 3 times the length of the longest chain $(x_1, \ldots, x_\ell)$ of pairwise different elements in M with $x_i \leq_{\mathcal{J}} x_j$ for $i \leq j$.

Consider $w \in \Gamma^+$. The word w has a unique factorization

$$w = v_0 w_1 \cdots w_m$$

with $m \geq 1$, $w_i = a_i v_i$, $a_i \in \Gamma$ and $v_i \in \{\varepsilon\} \cup \{v \mid [w] <_{\mathcal{J}} [v]\}$ satisfying $[w] \mathcal{J} [w_i]$ for all $1 \leq i \leq m$: We successively choose $w_i = a_i v_i \in \Gamma^+$ from right to left to be the shortest nonempty word such that $[a_i v_i] \mathcal{J} [w]$. We are going to construct a relaxed factorization tree for w of height at most $3 \left|\{x \in M \mid [w] \mathcal{J} x\}\right|$ and where the leaves are $v_0, a_1, v_1, \ldots, a_m, v_m$. For each $1 \leq i < m$ we define a pair (L_i, R_i) where L_i is the $\mathcal{L}$-class of $[w_i]$ and R_i is the $\mathcal{R}$-class of $[w_{i+1}]$. The intersection $L_i \cap R_j$ is never empty by Lemma 6. If $x, y \in L_i \cap R_j$ then $x \leq_{\mathcal{L}} y \leq_{\mathcal{R}} x$ and by Lemma 1 we obtain $x = y$. Hence, $L_i \cap R_j$ contains exactly one element. We use the abbreviation $w_{ij} = w_{i+1} \cdots w_j$ for $0 \leq i < j \leq m$. We have

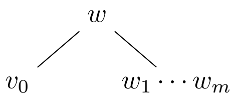
- $[w_{ij}] \leq_{\mathcal{R}} [w_{i+1}]$,
- $[w_{ij}] \leq_{\mathcal{L}} [w_j]$, and
- $[w] \leq_{\mathcal{J}} [w_{ij}] \leq_{\mathcal{J}} [w_{i+1}] \mathcal{J} [w_j] \mathcal{J} [w]$.

Together with Lemma 7 we conclude $[w_{ij}] \mathcal{R} [w_{i+1}]$ and $[w_{ij}] \mathcal{L} [w_j]$. If L_m is the $\mathcal{L}$-class of $[w_m]$, then $[w_{ij}]$ is the unique element in $R_i \cap L_j$ for all $1 \leq i < j \leq m$.

We use some book keeping for counting the levels in the tree with respect to some cost function. For the balance between the money and the cost, we use a bank account. Initially we put $S(w) = 3 \left|\{x \in M \mid [w] \mathcal{J} x\}\right|$ euros to the bank account of w as a start. For every level in the factorization tree of w we are willing to pay 1 euro from its bank account. If a node has s euros on its account, then it hands down $s - 1$ euros to each and every one of its children. So the amount of money has multiplied, but each child has 1 euro less than its parent.

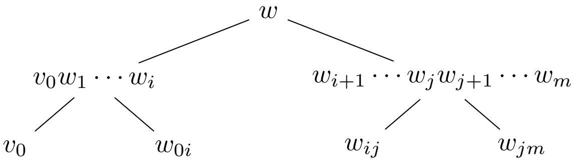
We define the cost $C(w) = 3 \left|\{(L_i, R_i) \mid 1 \leq i < m\}\right|$. Every pair (L_i, R_i) can be represented by the unique element in $L_i \cap R_i$, and if $x \in L_i \cap R_i$ then L_i is the $\mathcal{L}$-class of x and R_i is the $\mathcal{R}$-class of x. Since the representative of (L_i, R_i) is in the $\mathcal{J}$-class of $[w]$, we have $C(w) \leq S(w)$. The problem is that for a final step we need 1 euro more. As we will see, it is possible to save it in the first step of the factorization. We say that a word w is *rich* if there are more than $C(w)$ euros on its account and if $v_0 = \varepsilon$.

Let us see what happens, if for some element $u \in M$ with $u \mathcal{J} [w]$ the pair (L, R) consisting of the $\mathcal{L}$-class L and the $\mathcal{R}$-class R of u occurs at most twice, i.e., $(L, R) = (L_i, R_i) = (L_j, R_j)$ for at most two indices $i, j \in \{1, \ldots, m-1\}$. If there is no occurrence of the pair (L, R), we consider the following tree:



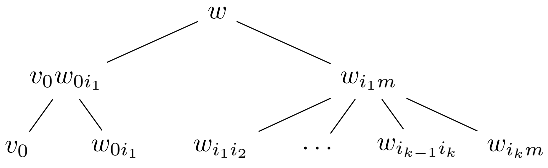
Note that v_0 is a leaf and that $w_1 \cdots w_m$ became rich, since $S(w)$ is at least $C(w) + 3$. Also note that this case includes $m = 1$ since indices must be in

$\{1, \ldots, m-1\}$. Now, we treat the case that there is at least one index but at most two indices with $(L, R) = (L_i, R_i) = (L_j, R_j)$ and $i \leq j$. Consider the following tree of height 2 where $w_{ij} = \varepsilon$ if $i = j$:



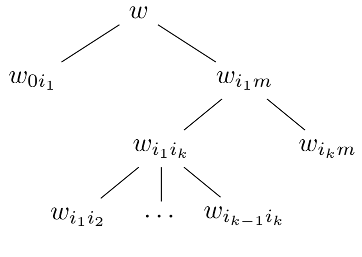
This reduces the cost of each nonempty word among w_{0i}, w_{ij}, and w_{jm} at least by 3, but the bank account of each word has decreased only by 2. Hence, each nonempty word among w_{0i}, w_{ij} and w_{jm} is rich.

Now, suppose that all possible pairs of $\mathcal{L}$- and $\mathcal{R}$-classes within the $\mathcal{J}$-class of $[w]$ occur at least 3 times. Define L as the $\mathcal{L}$-class of $[w_m]$ and R as the $\mathcal{R}$-class of $[w_m]$. Let $i_1, \ldots, i_k$ with $1 \leq i_1 < \cdots < i_k < m$ be the sequence of all positions with $(L, R) = (L_{i_j}, R_{i_j})$. By the observation above, we have $\big\{[w_{i_j i_\ell}]\big\} = R_{i_j} \cap L_{i_\ell} = R_{i_k} \cap L = \{[w_{i_k m}]\}$ for all $i_j < i_\ell$. Hence, $[w_{i_j i_\ell}] = [w_{i_k m}]$ for all $i_j < i_\ell$ and since $k \geq 3$, this element is idempotent. We obtain the following tree of height 2 for w:



Again, all remaining factors of the form w_{ij} are rich, once they have inherited the money from w.

After this preprocessing, we can assume that w is rich. We consider the set $\{(L_i, R_i) \mid 1 \leq i < m\}$ as before. If there exists a pair (L, R) in the list which occurs at most twice, we can use a tree of height 2 as before. If there exists some pair (L, R) which occurs at least three times, we can use the occurrences of the pair to obtain a tree of height 3: Let $i_1, \ldots, i_k$ with $1 \leq i_1 < \cdots < i_k < m$ be the sequence of all positions with $(L, R) = (L_{i_j}, R_{i_j})$. As before, we see that $[w_{i_j i_\ell}] = [w_{i_1 i_2}]$ for all $i_j < i_\ell$ and this element is idempotent. We cannot guarantee $[w_{i_1 i_2}] = [w_{i_k m}]$, and hence, we are not able to use the tree of height 2 as above. Instead, we consider the following tree of height 3:



Since w was rich, all descendants of the form w_{ij} are still rich. The process continues and ends in a situation where $\{(L_i, R_i) \mid 1 \leq i < m\}$ is empty, i.e., $m = 1$. But now $w = w_1$ is rich, which means there is at least 1 euro left. We can spend the last euro for the factorization $d(w_1) = (a_1, v_1)$. □

9 Two blocks of quantifiers

This section deals with the fragments $\Sigma_2[<]$ and $\Delta_2[<]$. Note that every characterization of $\Sigma_2[<]$ yields a symmetric characterization of $\Pi_2[<]$.

Theorem 8 *Let $L \subseteq \Gamma^*$. Then the following assertions are equivalent:*

1. *The syntactic ordered monoid $(M(L), \leq_L)$ is finite and the syntactic homomorphism $\Gamma^* \to M(L) : w \mapsto [w]$ has the following property: For all $e, s \in \Gamma^*$ we have*

$$[e] = [e]^2 \textit{ and } \operatorname{alph}(s) \subseteq \operatorname{alph}(e) \textit{ imply } [ese] \leq_L [e].$$

2. *L is recognized by a finite ordered monoid $(M, \leq)$ where each idempotent $e \in M$ is the greatest element in the subsemigroup $e \{s \mid e \in MsM\}^* e$.*
3. *L is a polynomial.*
4. *L is definable in $\Sigma_2[<]$.*

The subsemigroup $e \{s \mid e \in MsM\}^* e = e \{s \mid e \leq_{\mathcal{J}} s\}^* e$ is also called the *$\mathcal{J}$-localization at e*. It is a monoid where e is the identity. We can phrase the condition in *"2"* differently: $es_1 \cdots s_k e \leq e$ holds as soon as $e \in Ms_iM$ for all i. The equivalence of *"1"* (in a slight reformulation) and *"3"* is due to Arfi [3, 4] and the connection between polynomials and $\Sigma_2[<]$ is a result of Thomas [84]. In a more unified framework these results can be found in [54].

Proof (Theorem 8): For *"1 ⇒ 2"* we show that $(M(L), \leq_L)$ satisfies the condition in *"2"*. Let $[e] = [e]^2$. We can assume that $\operatorname{alph}(e) = \bigcup_{[e]=[f]} \operatorname{alph}(f)$, i.e., the alphabet of $e \in \Gamma^*$ is maximal. If $[u]$ is contained in the set $[e] \cdot \{s \mid [e] \leq_{\mathcal{J}} s\}^* \cdot [e]$, then $[u] = [es_1 \cdots s_k e]$ with $[e] \leq_{\mathcal{J}} [s_i]$ for all $1 \leq i \leq k$. Since the alphabet of e is maximal we deduce that $\operatorname{alph}(s_i) \subseteq \operatorname{alph}(e)$. Therefore, $\operatorname{alph}(s_1 \cdots s_k) \subseteq \operatorname{alph}(e)$ and *"1"* implies

$$[u] = [es_1 \cdots s_k e] \leq_L [e].$$

Therefore, the syntactic ordered monoid of L satisfies the property in *"2"*.

The direction *"2 ⇒ 3"* follows from Proposition 5.

For the step *"3 ⇒ 4"* note that $\Sigma_2[<]$ is closed under disjunction. It therefore suffices to show that every monomial

$$A_0^* a_1 A_1^* \cdots a_n A_n^*$$

is definable in $\Sigma_2[<]$. This can be done by the following $\Sigma_2[<]$ sentence, where $\lambda(y) \in A_i$ is a macro for $\bigvee_{a \in A_i} \lambda(y) = a$.

$$\begin{aligned} \exists x_1 \ldots \exists x_n \forall y \colon & \left(\bigwedge_{1 \leq i < n} x_i < x_{i+1} \right) \wedge \left(\bigwedge_{1 \leq i \leq n} \lambda(x_i) = a_i \right) \wedge \\ & (y < x_1 \to \lambda(y) \in A_0) \wedge (x_n < y \to \lambda(y) \in A_n) \wedge \\ & \left(\bigwedge_{1 \leq i < n} x_i < y < x_{i+1} \to \lambda(y) \in A_i \right). \end{aligned}$$

For "4 ⇒ 1" let $\varphi = \exists \overline{x} \forall \overline{y} \colon \psi(\overline{x}, \overline{y}) \in \Sigma_2[<]$ where $\overline{x} = (x_1, \ldots, x_m)$, $\overline{y} = (y_1, \ldots, y_m)$, and ψ is a propositional formula. Let $p, q, s, t \in \Gamma^*$ and assume $\mathrm{alph}(s) \subseteq \mathrm{alph}(t)$. We show that for all $k \geq m^2 + m$ we have

$$pt^kq \models \varphi \;\Rightarrow\; pt^kst^kq \models \varphi. \tag{2}$$

If $u = pt^kq$ models φ, then there exist positions $j_1, \ldots, j_m$ in the word u such that

$$u, \overline{j} \models \forall \overline{y} \colon \psi(\overline{x}, \overline{y}) \tag{3}$$

where $\overline{j} = (j_1, \ldots, j_m)$. We refer to the k copies of the factor t in u as *blocks* numbered by 1 to k from left to right. By choice of k there exist m consecutive blocks such that no j_i is a position within these blocks, i.e.,

$$u = pt^{k_1} \cdot t^m \cdot t^{k_2}q$$

and all j_i's are positions either in the prefix pt^{k_1} or in the suffix $t^{k_2}q$ of u. Consider the following factorization:

$$v = pt^kst^kq = pt^{k_1} \cdot t^{m+k_2}st^{k_1+m} \cdot t^{k_2}q.$$

Since the prefix and suffix in this factorization are equal to that in the factorization of u and since all j_i's are positions in these parts of u, we can choose the corresponding positions $j'_1, \ldots, j'_m$ in the identical parts of v. We claim that for $\overline{j'} = (j'_1, \ldots, j'_m)$ we have

$$v, \overline{j'} \models \forall \overline{y} \colon \psi(\overline{x}, \overline{y}).$$

Let $\ell'_1, \ldots, \ell'_m$ be positions in v and let $\overline{\ell'} = (\ell'_1, \ldots, \ell'_m)$. If ℓ'_i is a position in the prefix pt^{k_1} or in the suffix $t^{k_2}q$ of v, we can choose an analogous position ℓ_i in u. We order the remaining positions $\ell'_{i_1} \leq \ell'_{i_2} \leq \cdots \leq \ell'_{i_n}$ with $n \leq m$. We let ℓ_{i_1} be some position labeled by $\lambda(\ell'_{i_1})$ in the block $k_1 + 1$ of u. For $1 \leq j < n$, we let $\ell_{i_{j+1}} = \ell_{i_j}$ if $\ell'_{i_{j+1}} = \ell'_{i_j}$ and otherwise we let $\ell_{i_{j+1}}$ be some position labeled by $\lambda(\ell'_{i_{j+1}})$ in the block $k_1 + j + 1$ of u. This is possible since $\mathrm{alph}(s) \subseteq \mathrm{alph}(t)$. Now, all positions $\ell_{i_1}, \ldots, \ell_{i_n}$ are in the middle factor t^m of u. By construction, the structures $(u, \overline{j}, \overline{\ell})$ and $(v, \overline{j'}, \overline{\ell'})$ satisfy the same propositional formulae. By (3) we have $u, \overline{j}, \overline{\ell} \models \psi(\overline{x}, \overline{y})$ hence we get $v, \overline{j'}, \overline{\ell'} \models \psi(\overline{x}, \overline{y})$. This proves (2). For $L = L(\varphi)$ it follows that $[t^kst^k] \leq_L [t^k]$ holds in the syntactic ordered monoid $(M(L), \leq_L)$ of L. Property "1" now follows since $[t^k] = [t]$ if $[t] = [e]$ is idempotent. □

Remember that $\Delta_2[<]$ denotes those formulae in $\Sigma_2[<]$ which have an equivalent formula in $\Pi_2[<]$. This is often written as $\Delta_2[<] = \Sigma_2[<] \cap \Pi_2[<]$. We can now establish the following characterizations of languages whose syntactic monoid is in **DA**. These characterizations were already stated in Theorem 2.

Theorem 9 *Let $L \subseteq \Gamma^*$. Then the following are equivalent:*

1. *L is recognized by a monoid in* **DA**.
2. *L is a polynomial and $\Gamma^* \setminus L$ is also a polynomial.*
3. *L is definable in $\Delta_2[<]$.*

The main part of the theorem is a corollary of Theorem 8. It remains to show that the algebraic characterization of *"2"* and *"3"* which results from Theorem 8 yields exactly **DA**. A direct proof of this algebraic correspondence is given in [53]. Our proof uses the characterization of **DA** in terms of unambiguous polynomials in Theorem 2 and actually does not rely on Proposition 5 and the Factorization Forest Theorem 7.

Proof (Theorem 9): The equivalence of *"2"* and *"3"* follows from Theorem 8. For *"3⇒1"* let L be definable in $\Delta_2[<]$ and let $\Gamma^* \to (M(L), \leq_L) : w \mapsto [w]$ be its syntactic homomorphism onto its syntactic ordered monoid. Then $(M(L), \geq_L)$ is the syntactic ordered monoid of $\Gamma^* \setminus L$. From Theorem 8 we get the following property of $M(L)$: if $[e] = [e]^2$ and $\mathrm{alph}(s) \subseteq \mathrm{alph}(e)$ then $[ese] \leq_L [e]$ and $[ese] \geq_L [e]$. Hence

$$[e] = [e]^2 \text{ and } \mathrm{alph}(s) \subseteq \mathrm{alph}(e) \text{ imply } [ese] = [e].$$

If we apply this property to $[e] = [uvw]^\omega$ and $s = v$, we get

$$[uvw]^\omega [v] [uvw]^\omega = [uvw]^\omega.$$

This shows $M(L) \in \mathbf{DA}$. For *"1⇒2"* let $M \in \mathbf{DA}$ be the syntactic monoid of L. The complement $\Gamma^* \setminus L$ has the same syntactic monoid M. By Theorem 2 both L and its complement are (unambiguous) polynomials. □

10 Summary

The following picture repeats the various relations between logics, languages, and monoids. The expressive power of the logical fragments is strictly increasing top down with the sole exception that $\mathrm{FO}^1[<]$ and $\Sigma_1[<]$ are incomparable. We use the following notation: Pol denotes the language class of polynomials, co-Pol contains all languages whose complement is a polynomial, and UPol is the class of unambiguous polynomials.

Logic	**Languages**	**Algebra**	
$\mathrm{FO}^1[<]$	$\mathbb{B}\{A^* \mid A \subseteq \Gamma\}$	commutative and idempotent	Thm. 1
$\Sigma_1[<]$	simple polynomials	$u \leq 1$	Thm. 3
$\mathbb{B}\Sigma_1[<]$	piecewise testable	$\mathcal{J}$-trivial	Thm. 4
$\mathrm{FO}^2[<]$ $= \Delta_2[<]$ $= \mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$ $= \mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$	UPol = Pol ∩ co-Pol = ranker languages	**DA**	Thm. 2 Thm. 9
$\Sigma_2[<]$	Pol	$e\{s \mid e \leq_{\mathcal{J}} s\}^* e \leq e$	Thm. 8
FO $= \mathrm{FO}^3[<]$ $= \mathrm{TL}[\mathsf{X}, \mathsf{U}]$	star-free	aperiodic	see e.g. [20] for a survey

11 Related topics

Alternation hierarchies. For the fragments obtained by restricting the number of variables we have given an exhaustive characterization in Sections 3 and 4. This has mainly been possible because this hierarchy is not strict: three variables are already sufficient to express arbitrary first-order properties [28, 29]. The situation is different if we instead restrict the number of quantifier alternations. Here, the hierarchy is strict [10, 84], and decidability is only known for $\Sigma_1[<]$, the Boolean closure of $\Sigma_1[<]$ and for $\Sigma_2[<]$ (and hence also for $\Pi_1[<]$ and $\Pi_2[<]$). For example, about the membership problem for $\Sigma_n[<]$ for $n \geq 3$ very little is known. Concerning the Boolean closure of $\Sigma_2[<]$, decidability is only known for a few special cases, such as two-letter alphabets [68] or inverse monoids [19]; see also [45, 51, 55, 74, 88] for necessary or sufficient conditions for definability within $\mathbb{B}\Sigma_2[<]$.

In [42, 84], it has been shown that $\mathbb{B}\Sigma_n[<]$ corresponds to the n-th level of the Straubing-Thérien hierarchy, see e.g. [49] for definitions. A similar result holds for $\mathbb{B}\Sigma_n[<, \mathrm{suc}]$ and the n-th level of the so-called dot-depth hierarchy [84]. Here, suc denotes the successor predicate.

The class of first-order formulae with n variables and at most $m-1$ quantifier alternations (on every path in their parse tree) is denoted by $\mathrm{FO}^n_m[<]$. Recently, Weis and Immerman have initiated the study of the classes $\mathrm{FO}^2_m[<]$, see [90]. They related the number of alternations with the number of changes of direction in rankers.

Quantifier depth. Another measure for first-order formulae is the quantifier depth. Tesson and Thérien have presented an algebraic characterization of the quantifier depth hierarchy [78]. In the same article they have also given an algebraic counterpart of the quantifier depth hierarchy within $\mathrm{FO}^2[<]$. The membership problem for hierarchies defined by a bounded quantifier depth is trivially decidable since over a fixed alphabet, up to equivalence, there is only a finite number of first-order sentences of a given quantifier depth.

The successor relation. The successor predicate $\mathrm{suc}(x, y)$ can be expressed by the $\mathrm{FO}^3[<]$-formula $x < y \wedge \forall z\colon z \leq x \vee y \leq z$. This requires quantification over a new variable z. Hence, although we have $\mathrm{FO}^3[<, \mathrm{suc}] = \mathrm{FO}^3[<] = \mathrm{FO}[<] = \mathrm{FO}[<, \mathrm{suc}]$, additionally allowing the successor predicate increases the expressive power of fragments of first-order logic. For example, $\mathrm{FO}^2[<, \mathrm{suc}]$ has strictly more expressive power than $\mathrm{FO}^2[<]$. Thérien and Wilke have given an algebraic counterpart for $\mathrm{FO}^2[<, \mathrm{suc}]$, see [81], and Almeida has shown that the membership problem for this class of monoids is decidable [2]. Etessami, Vardi, and Wilke have given a transformation of $\mathrm{FO}^2[<, \mathrm{suc}]$-formulae into equivalent ones in unary temporal logic [23].

If we forbid $<$ and only allow suc as a binary relation symbol, we obtain the fragment $\mathrm{FO}[\mathrm{suc}]$. It was shown by Thomas that this fragment corresponds to the class of so-called *locally threshold testable* languages [84]. This is an instance of a more general theorem of Hanf about first-order logic [26]. In [5, 6], Beauquier and Pin observed that locally threshold testable languages have an algebraic characterization, for which decidability follows from a proof by Thérien and Weiss [80]. In particular, the membership problem for $\mathrm{FO}[\mathrm{suc}]$ is decidable;

a complete proof of this fact can also be found in Straubing's textbook [69]. An efficient algorithm for the membership problem is given by Pin [50] and an easily accessible (though inefficient) algorithm based on Presburger arithmetic has been presented by Bojańczyk [8]. One can consider the alternation hierarchy within FO[suc]. Thomas has shown that this hierarchy collapses at level 2 [84], i.e., $\mathrm{FO}[\mathrm{suc}] = \Sigma_2[\mathrm{suc}]$ and since FO[suc] is closed under Boolean operations, we indeed have $\mathrm{FO}[\mathrm{suc}] = \Delta_2[\mathrm{suc}]$ where $\Delta_2[\mathrm{suc}] = \Sigma_2[\mathrm{suc}] \cap \Pi_2[\mathrm{suc}]$. The Boolean closure of $\Sigma_1[\mathrm{suc}]$ is a strict subclass of FO[suc] and decidability of the membership problem for this subclass has been shown by Pin [50].

More logics. Kamp has shown that first-order logic and linear temporal logic have the same expressive power [29], see also [16]. The main advantage of temporal logic is that several computational problems can be solved more efficiently than in the case of using first-order formulae [64, 65], see also [22, 24]. A huge variety of fragments defined in terms of temporal logic has been researched. The most remarkable results are due to Thérien and Wilke who showed the decidability of the membership problem of a hierarchies defined in terms of nesting the until- and the since-operator [82, 83]. Further fragments of temporal logic are considered in Wilke's survey [91]. An overview of the relation between several hierarchies can be found in the PhD thesis of Strejček [75].

If we additionally allow quantification over sets of positions, we obtain a generalization of first-order logic called *monadic second-order* logic (MSO). A famous result of Büchi says that expressibility in MSO characterizes exactly the class of regular languages [12]. We note that in MSO it is possible to express the order relation $x < y$ in terms of the successor relation $\mathrm{suc}(x, y)$ and therefore $\mathrm{MSO}[<] = \mathrm{MSO}[\mathrm{suc}]$, see e.g. [86]. We also refer to [47] for a survey on fragments of second-order logic. If we additionally allow so-called *modular quantifiers* in first-order logic, we obtain an extension which lies strictly between first-order logic and MSO. We refer to Straubing's textbook for more information on this subject [69]. Another extension of first-order logic can be obtained by additionally allowing numerical predicates such as $x + y = z$. This generalization exceeds the class of regular languages, but one gets connections between (circuit) complexity classes and certain classes of numerical predicates. Again, an introduction can be found in [69].

We have already seen that language classes which are not closed under complementation can be captured by using ordered monoids instead of monoids [46], but still these descriptions imply some non-trivial closure properties which makes this approach useless for classes of languages which do not have those closure properties. For example, stutter-invariant languages are not closed under inverse homomorphisms. A language L is stutter-invariant, if $pa^2q \in L \Leftrightarrow paq \in L$ for all $p, q \in \Gamma^*$ and all $a \in \Gamma$. Intuitively, the syntactic monoid of a stutter-invariant language satisfies $a^2 = a$ for letters a but it does not need to satisfy $x^2 = x$ for arbitrary words x. This kind of restriction is formalized in classes of homomorphisms called $\mathcal{C}$-varieties [70]; see also [14, 35, 52]. An application of $\mathcal{C}$-varieties to first-order logic can be found in [15].

A good starting point concerning logics over other structures such as infinite words, trees, or Mazurkiewicz traces is Weil's survey [89].

Acknowledgements. We thank the referee for his very detailed report which gave us many helpful suggestions and for pointing out some inaccuracies in a preliminary version of this paper.

References

[1] J. Almeida. *Finite Semigroups and Universal Algebra.* World Scientific, Singapore, 1994.

[2] J. Almeida. A syntactical proof of locality of *DA*. *International Journal of Algebra and Computation*, 6(2):165–177, 1996.

[3] M. Arfi. Polynomial operations on rational languages. In F.-J. Brandenburg, G. Vidal-Naquet, and M. Wirsing, editors, *4th Annual Symposium on Theoretical Aspects of Computer Science (STACS), Passau, Germany, February 19-21, 1987, Proceedings*, volume 247 of *Lecture Notes in Computer Science*, pages 198–206. Springer-Verlag, 1987.

[4] M. Arfi. Opérations polynomiales et hiérarchies de concaténation. *Theoretical Computer Science*, 91(1):71–84, 1991.

[5] D. Beauquier and J.-É. Pin. Factors of words. In G. Ausiello, M. Dezani-Ciancaglini, and S. R. D. Rocca, editors, *Proceedings of the 16th International Colloquium on Automata, Languages and Programming, ICALP'89 (Stresa, Italy, July 11-15, 1989)*, volume 372 of *Lecture Notes in Computer Science*, pages 63–79. Springer-Verlag, 1989.

[6] D. Beauquier and J.-É. Pin. Languages and scanners. In *Words, languages and combinatorics (Kyoto, 1990)*, pages 16–29. World Sci. Publ., River Edge, NJ, 1992.

[7] C. Behle, A. Krebs, and M. Mercer. Linear circuits, two-variable logic and weakly blocked monoids. In L. Kučera and A. Kučera, editors, *Mathematical Foundations of Computer Science 2007, 32nd International Symposium, MFCS 2007, Český Krumlov, Czech Republic, August 26-31, 2007, Proceedings*, volume 4708 of *Lecture Notes in Computer Science*, pages 147–158. Springer-Verlag, 2007.

[8] M. Bojańczyk. A new algorithm for testing if a regular language is locally threshold testable. *Information Processing Letters*, 104(3):91–94, 2007.

[9] M. Bojańczyk, A. Muscholl, Th. Schwentick, L. Segoufin, and C. David. Two-variable logic on words with data. In *LICS '06: Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science*, pages 7–16, Washington, DC, USA, 2006. IEEE Computer Society.

[10] J. A. Brzozowski and R. Knast. The dot-depth hierarchy of star-free languages is infinite. *Journal of Computer and System Sciences*, 16(1):37–55, 1978.

[11] J. A. Brzozowski and I. Simon. Characterizations of locally testable events. *Discrete Mathematics*, 4:243–271, 1973.

[12] J. R. Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[13] J. Chalopin and H. Leung. On factorization forests of finite height. *Theoretical Computer Science*, 310(1-3):489–499, 2004.

[14] L. Chaubard, J.-É. Pin, and H. Straubing. Actions, wreath products of $\mathcal{C}$-varieties and concatenation product. *Theoretical Computer Science*, 356:73–89, 2006.

[15] L. Chaubard, J.-É. Pin, and H. Straubing. First order formulas with modular predicates. In *LICS '06: Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science*, pages 211–220, Washington, DC, USA, 2006. IEEE Computer Society.

[16] J. Cohen, D. Perrin, and J.-É. Pin. On the expressive power of temporal logic. *Journal of Computer and System Sciences*, 46(3):271–294, June 1993.

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

[18] T. Colcombet. Factorisation forests for infinite words. In E. Csuhaj-Varjú and Z. Ésik, editors, *Fundamentals of Computation Theory, 16th International Symposium, FCT 2007, Budapest, Hungary, August 27-30, 2007, Proceedings*, volume 4639 of *Lecture Notes in Computer Science*, pages 226–237. Springer-Verlag, 2007.

[19] D. Cowan. Inverse monoids of dot-depth two. *International Journal of Algebra and Computation*, 3:411–424, 1993.

[20] V. Diekert and P. Gastin. First-order definable languages. In J. Flum, E. Grädel, and Th. Wilke, editors, *Logic and Automata: History and Perspectives*, Texts in Logic and Games, pages 261–306. Amsterdam University Press, 2008.

[21] S. Eilenberg. *Automata, Languages, and Machines*, volume B. Academic Press, New York and London, 1976.

[22] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 16, pages 995–1072. Elsevier Science Publisher B. V., 1990.

[23] K. Etessami, M. Y. Vardi, and Th. Wilke. First-order logic with two variables and unary temporal logic. *Information and Computation*, 179(2):279–295, 2002.

[24] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Clarendon Press, Oxford, 1994.

[25] J. A. Green. On the structure of semigroups. *Annals of Mathematics. Second Series*, 54:163–172, 1951.

[26] W. P. Hanf. Model-theoretic methods in the study of elementary logic. In J. W. Addison, L. Henkin, and A. Tarski, editors, *The Theory of Models*, pages 132–145. North-Holland, Amsterdam, 1965.

[27] G. Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society. Third Series*, 2:326–336, 1952.

[28] N. Immerman and D. Kozen. Definability with bounded number of bound variables. *Information and Computation*, 83(2):121–139, Nov. 1989.

[29] J. A. W. Kamp. *Tense Logic and the Theory of Linear Order.* PhD thesis, University of California, Los Angeles (California), 1968.

[30] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, number 34 in Annals of Mathematics Studies, pages 3–40. Princeton University Press, 1956.

[31] M. Koucký, C. Lautemann, S. Poloczek, and D. Thérien. Circuit lower bounds via Ehrenfeucht-Fraïssé games. In *Annual IEEE Conference on Computational Complexity*, volume 21, pages 190–201, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[32] M. Kufleitner. *Logical Fragments for Mazurkiewicz Traces: Expressive Power and Algebraic Characterizations.* Dissertation, Institut für Formale Methoden der Informatik, Universität Stuttgart, 2006.

[33] M. Kufleitner. Polynomials, fragments of temporal logic and the variety DA over traces. *Theoretical Computer Science*, 376:89–100, 2007. Special issue DLT 2006.

[34] M. Kufleitner. A proof of the factorization forest theorem. Technical report Nr. 2007/05, Formale Methoden der Informatik, Universität Stuttgart, Germany, October 2007.

[35] M. Kunc. Equational description of pseudovarieties of homomorphisms. *Theoret. Informatics Appl.*, 37:243–254, 2003.

[36] M. Lothaire. *Combinatorics on Words*, volume 17 of *Encyclopedia of Mathematics and its Applications.* Addison-Wesley, Reading, MA, 1983. Reprinted by *Cambridge University Press*, 1997.

[37] S. W. Margolis and J.-É. Pin. Products of group languages. In *Fundamentals of Computation Theory (Cottbus, 1985)*, volume 199 of *Lecture Notes in Computer Science*, pages 285–299. Springer-Verlag, Berlin, 1985.

[38] R. McNaughton. Algebraic decision procedures for local testability. *Mathematical Systems Theory*, 8(1):60–76, 1974.

[39] R. McNaughton and S. Papert. *Counter-Free Automata.* The MIT Press, Cambridge, Mass., 1971.

[40] J. R. Myhill. Finite automata and the representation of events. Technical Report 57-624, Wright Airport Development Command, 1957.

[41] D. Perrin. Finite automata. In *Handbook of Theoretical Computer Science, Vol. B*, pages 1–57. Elsevier, Amsterdam, 1990.

[42] D. Perrin and J.-É. Pin. First-order logic and star-free sets. *Journal of Computer and System Sciences*, 32(3):393–406, 1986.

[43] D. Perrin and J.-É. Pin. *Infinite words*, volume 141 of *Pure and Applied Mathematics*. Elsevier, Amsterdam, 2004.

[44] J.-É. Pin. *Varieties of Formal Languages*. North Oxford Academic, London, 1986.

[45] J.-É. Pin. A property of the Schützenberger product. *Semigroup Forum*, 35:53–62, 1987.

[46] J.-É. Pin. A variety theorem without complementation. In *Russian Mathematics (Izvestija vuzov.Matematika)*, volume 39, pages 80–90, 1995.

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

[48] J.-É. Pin. Syntactic semigroups. In *Handbook of Formal Languages, Vol. 1*, pages 679–746. Springer-Verlag, Berlin, 1997.

[49] J.-É. Pin. Algebraic tools for the concatenation product. *Theoretical Computer Science*, 292:317–342, 2003.

[50] J.-É. Pin. Expressive power of existential first-order sentences of Büchi's sequential calculus. *Discrete Mathematics*, 291(1-3):155–174, 2005.

[51] J.-É. Pin and H. Straubing. Monoids of upper triangular matrices. In *Colloquia Mathematica Societatis Janos Bolyal*, pages 259–272, 1981.

[52] J.-É. Pin and H. Straubing. Some results on $\mathcal{C}$-varieties. *Theoret. Informatics Appl.*, 39:239–262, 2005.

[53] J.-É. Pin, H. Straubing, and D. Thérien. Locally trivial categories and unambiguous concatenation. *Journal of Pure and Applied Algebra*, 52:297–311, 1988.

[54] J.-É. Pin and P. Weil. Polynomial closure and unambiguous product. *Theory of Computing Systems*, 30(4):383–422, 1997.

[55] J.-É. Pin and P. Weil. A conjecture on the concatenation product. *Informatique Théorique et Applications*, 35(6):597–618, 2001.

[56] J. Sakarovitch and I. Simon. Subwords. In *M. Lothaire: Combinatorics on Words*, volume 17 of *Encyclopedia of Mathematics and its Applications*, chapter 6, pages 105–144. Addison-Wesley, Reading, MA, 1983.

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

[58] M. P. Schützenberger. Sur le produit de concaténation non ambigu. *Semigroup Forum*, 13:47–75, 1976.

[59] T. Schwentick, D. Thérien, and H. Vollmer. Partially-ordered two-way automata: A new characterization of DA. In W. Kuich, G. Rozenberg, and A. Salomaa, editors, *Proc. of the 5th Int. Conf. on Developments in Language Theory (DLT)*, volume 2295 of *Lecture Notes in Computer Science*, pages 239–250. Springer-Verlag, 2001.

[60] I. Simon. Piecewise testable events. In H. Barkhage, editor, *Automata Theory and Formal Languages, 2nd GI Conference, Kaiserslautern, May 22–23, 1975*, volume 33 of *Lecture Notes in Computer Science*, pages 214–222. Springer-Verlag, 1975.

[61] I. Simon. Properties of factorization forests. In J.-É. Pin, editor, *Formal Properties of Finite Automata and Applications: LITP Spring School on Theoretical Computer Science*, volume 386 of *Lecture Notes in Computer Science*, pages 65–72. Springer-Verlag, 1988.

[62] I. Simon. Factorization forests of finite height. *Theoretical Computer Science*, 72(1):65–94, 1990.

[63] I. Simon. A short proof of the factorization forest theorem. In M. Nivat and A. Podelski, editors, *Tree Automata and Languages*, pages 433–438. Elsevier, 1992.

[64] A. P. Sistla and E. Clarke. The complexity of propositional linear time logic. *Journal of the Association for Computing Machinery*, 32:733–749, 1985.

[65] L. Stockmeyer. The complexity of decision problems in automata theory and logic. PhD thesis, TR 133, M.I.T., Cambridge, 1974.

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

[67] H. Straubing. Finite semigroup varieties of the form $V * D$. *Journal of Pure and Applied Algebra*, 36(1):53–94, 1985.

[68] H. Straubing. Semigroups and languages of dot-depth two. *Theoretical Computer Science*, 58(1-3):361–378, 1988. Thirteenth International Colloquium on Automata, Languages and Programming (Rennes, 1986).

[69] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity.* Birkhäuser, Boston, Basel and Berlin, 1994.

[70] H. Straubing. On the logical descriptions of regular languages. In S. Rajsbaum, editor, *Proc. of the 5th Latin American Theoretical Informatics Conference (LATIN'02), Cancun*, number 2286 in Lecture Notes in Computer Science, pages 528–538. Springer-Verlag, 2002.

[71] H. Straubing and D. Thérien. Partially ordered finite monoids and a theorem of I. Simon. *Journal of Algebra*, 119:393–399, 1988.

[72] H. Straubing and D. Thérien. Weakly iterated block products of finite monoids. In S. Rajsbaum, editor, *LATIN 2002: Theoretical Informatics, 5th Latin American Symposium, Cancun, Mexico, April 3-6, 2002, Proceedings*, volume 2286 of *Lecture Notes in Computer Science*, pages 91–104. Springer-Verlag, 2002.

[73] H. Straubing and D. Thérien. Regular languages defined by generalized first-order formulas with a bounded number of bound variables. *Theory of Computing Systems*, 36(1):29–69, 2003.

[74] H. Straubing and P. Weil. On a conjecture concerning dot-depth two languages. *Theoretical Computer Science*, 104(2):161–183, 1992.

[75] J. Strejček. *Linear Temporal Logic: Expressiveness and Model Checking.* PhD thesis, Faculty of Informatics, Masaryk University Brno, 2004.

[76] P. Tesson and D. Thérien. Diamonds are Forever: The Variety DA. In G. M. dos Gomes Moreira da Cunha, P. V. A. da Silva, and J.-É. Pin, editors, *Semigroups, Algorithms, Automata and Languages, Coimbra (Portugal) 2001*, pages 475–500. World Scientific, 2002.

[77] P. Tesson and D. Thérien. Restricted two-variable sentences, circuits and communication complexity. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, volume 3580 of *Lecture Notes in Computer Science*, pages 526–538. Springer-Verlag, 2005.

[78] P. Tesson and D. Thérien. Logic meets algebra: the case of regular languages. *Logical Methods in Computer Science*, 3(1):1–37, 2007.

[79] D. Thérien. Classification of finite monoids: the language approach. *Theoretical Computer Science*, 14(2):195–208, 1981.

[80] D. Thérien and A. Weiss. Graph congruences and wreath products. *Journal of Pure and Applied Algebra*, 36(2):205–215, 1985.

[81] D. Thérien and Th. Wilke. Over words, two variables are as powerful as one quantifier alternation. In *STOC*, pages 234–240, 1998.

[82] D. Thérien and Th. Wilke. Temporal logic and semidirect products: An effective characterization of the until hierarchy. *SIAM Journal on Computing*, 31(3):777–798, 2001.

[83] D. Thérien and Th. Wilke. Nesting until and since in linear temporal logic. *Theory of Computing Systems*, 37(1):111–131, 2004.

[84] W. Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25:360–376, 1982.

[85] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 4, pages 133–191. Elsevier Science Publishers B. V., 1990.

[86] W. Thomas. Languages, automata and logic. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, Beyond Words. Springer, Berlin, 1997.

[87] B. Tilson. Categories as algebras: An essential ingrediant in the theory of monoids. *Journal of Pure and Applied Algebra*, 48:83–198, 1987.

[88] P. Weil. Some results on the dot-depth hierarchy. *Semigroup Forum*, 46:352–370, 1993.

[89] P. Weil. Algebraic recognizability of languages. In *Mathematical Foundations of Computer Science 2004*, volume 3153 of *Lecture Notes in Computer Science*, pages 149–174. Springer, Berlin, 2004.

[90] Ph. Weis and N. Immerman. Structure theorem and strict alternation hierarchy for FO^2 on words. In J. Duparc and T. A. Henzinger, editors, *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, volume 4646 of *Lecture Notes in Computer Science*, pages 343–357. Springer-Verlag, 2007.

[91] Th. Wilke. Linear temporal logic and finite semigroups. In J. Sgall, A. Pultr, and P. Kolman, editors, *Mathematical Foundations of Computer Science 2001, 26th International Symposium, MFCS 2001 Marianske Lazne, Czech Republic, August 27-31, 2001, Proceedings*, volume 2136 of *Lecture Notes in Computer Science*, pages 96–110. Springer-Verlag, 2001.