Higher Dimensional Automata

Abstract of the Ph.D. Thesis

Zoltán L. Németh

Supervisor:

Prof. Zoltán Ésik

University of Szeged
Department of Foundations of
Computer Science

2007

Introduction

Finite automata together with regular languages form one of the cornerstones of theoretical computer science. They are actively investigated mathematical objects which are of unquestionable importance both from a theoretical and practical point of view [RS97]. Their widespread use is mainly due to two facts. First, words can serve as models for a wide range of sequential systems, as they can simulate sequential behavior quite naturally. Second, the concept of regular word languages can be defined in several different, but equivalent ways. To establish the terms and notations of the concepts which we will work with, we will fix the following terminology:

- ***Regularity*** will mean acceptance by finite automata.
- ***Recognizability*** will mean algebraic recognizability by finite algebras or finite-index congruences.
- ***Rationality*** will mean expressibility by rational (also called regular) expressions.
- ***MSO-definability*** will mean definability by monadic second-order logical formulas.

In the following we will use these concepts not just for word languages, but also for languages of other structures. In addition, we shall employ the following notations for the corresponding language classes: Reg, Rec, Rat and MSO. The classical results of automata theory (due to Büchi, Kleene, Myhill and Nerode) demonstrate that the equalities $\mathsf{Reg} = \mathsf{Rec} = \mathsf{Rat} = \mathsf{MSO}$ hold for languages of finite words.

It should be emphasized here that these four concepts are not simply four different ways of defining the same class of word languages, but rather each of them contains the essence of this class from a different perspective. In certain situations one of them may have some advantage and be better suited than the other three.

Of course, there are many other computational models that have more complex structures than finite words. These include infinite words [PP04, Wil94], trees [GS84], traces [DR95], partially ordered sets (posets for short) [Pra86, LW98, LW00, Kus03a], message sequence charts[Kus03b] and graphs [Cou91, CW05]. These models were introduced and applied to capture other computational aspects like timing and concurrency.

When investigating these more complex models the natural question arises – which is of crucial importance – about what results of the classical theory of words can be generalized and how. In many important cases the above notions can be suitably defined and are known to be equivalent. But sometimes we are faced with serious problems. It is not always clear how to choose an appropriate algebraic or logical framework. And, for instance, for graphs, for posets, and even for sp-posets in general, the concept of an automaton that captures recognizability is not known. For a general overview of this topic, we refer to the paper by Weil [Wei04a] which surveys the concept of recognizability in computer science.

The subject of this thesis is about the generalization of the fundamental results of classical automata theory to higher dimensions. Both finite and infinite higher dimensional words and their languages will be defined and investigated.

Fortunately, we can restrict our studies to just the two-dimensional case, since both our concept and results can be readily generalized to any finite number of dimensions. For the generalization we adopted an algebraic approach, namely we considered languages over free *binoids* – a generalization of monoids, where two independent associative operations are defined and they have a common identity element.

Let us continue with a brief overview of the related literature. One of our starting points will be the concept of (m,n)-structures introduced by Ésik in [Ési00], where m and n are nonnegative integers. They will provide us with a description of the elements of the free binoids we will work with. This realization of binoid languages is essential for extending logical definability.

Our study was influenced to a great extent by the work of Lodaya and Weil [LW98, LW00, LW01] and Kuske [Kus03a] on automata operating on series-parallel posets (sp-posets for short). Sp-poset languages can be regarded as a two-dimensional generalization of the classical theory of words in which two independent associative operations are defined, but one of them is commutative as well. Also, sp-posets may be characterized as those posets that does not contain an induced subgraph isomorphic to the "N" directed graph [Gra81]. Moreover, sp-posets may serve as models of modularly constructed concurrent systems [Pra86].

Our investigation also owes much to the work of Hoogeboom and ten Pas [HtP96, HtP97] on text languages. In particular, we will use their result which establishes the equality $\mathsf{Rec} = \mathsf{MSO}$ for text languages in order to show that the same equality holds for binoid languages as well.

Automata and languages over free binoids have also been studied independently by Hashiguchi et al. [HIJ00, HWJ03, HSJ04]. However they employed a totally different approach, namely they used ordinary finite automata to define regular binoid languages. We make a detailed comparison between their concepts of regularity and ours.

A different two-dimensional generalization of the classical framework is provided by *picture languages* [GR97]. In [Dol05], Dolinka demonstrated that picture languages and binoid languages satisfy the same identities (for the operations of union, the two products, the two (Kleene) iterations of the two products and some constants). See [Dol07] too for more details about the axiomatization of the equational theory of binoid languages. Binoid languages are also closely related to visibly pushdown and nested word languages [AM04, AM06].

Results of the Thesis

Biwords and their Representations

It is generally agreed that automata models operate on elements of some free algebra. Thus if we want to generalize the notion of automata to higher dimensions, it is natural to ask how they can operate on the elements of the free binoids.

Let Σ be an *alphabet* (i.e. a finite nonempty set). We can consider the *free binoid* over Σ, which we will be denoted by $\Sigma^*(\bullet, \circ)$. This is well-defined from universal algebraic considerations. The two product operations will be called the *horizontal product* ($\bullet$) and the *vertical product* ($\circ$). In the following the elements of $\Sigma^*(\bullet, \circ)$ will be called *biwords*, while the subsets of $\Sigma^*(\bullet, \circ)$ will be *binoid languages* (over Σ). The identity of $\Sigma^*(\bullet, \circ)$ will be called the *empty biword*, denoted by ε.

It is usual to describe biwords by *terms* using the letters of Σ, parentheses and two operation symbols, but we will also find that biwords can be represented in several other equivalent ways. First we consider perhaps the most intuitive one of them, which will be called the *two-dimensional word representation*.

To construct two-dimensional words from the letters of Σ, we need two independent concatenation operations. The first one will be called the *horizontal concatenation* (denoted by $\bullet$), while the second one will be called the *vertical concatenation* (denoted by $\circ$).

We will build two-dimensional words inductively from smaller elements called blocks. Initially we can use just the letters of Σ as blocks, then we can form more complex blocks by using the two concatenation operations. Naturally, the horizontal concatenation places some finite number of blocks to the left/right of each other, while the vertical concatenation places the blocks above/beneath each other. Now two-dimensional words are defined as those blocks that can be obtained from the elements of Σ by a finite number of applications of the two concatenations. We also have an empty two-dimensional word ε, which has no letters.

Another representation of biwords can be given by using biposets. A *biposet* is a relational structure of the form $(P, <_h, <_v)$, where $<_h$ and $<_v$ are arbitrary partial order relations on the set P. If we add a labeling function $\lambda : P \to \Sigma$, where Σ is an alphabet, then $(P, <_h, <_v, \lambda)$ is a *labeled biposet*. The relation $<_h$ is called the *horizontal order*, while $<_v$ is the *vertical order* relation.

The two partial order relations naturally induce two product operations on the set of (labeled) biposets. If we consider two biposets, their *horizontal product* (resp. *vertical product*) is defined by taking the disjoint union of them and letting all the elements of the first biposet be horizontally (resp. vertically) less than all the elements of the second biposet. Of course the original order relations remain unchanged inside the two operands. Now *sp-biposets*[1] are those that can be generated from the singletons by the two product operations.

It can be proved that both the algebra of two-dimensional words and the algebra of sp-biposets over an alphabet Σ are isomorphic to $\Sigma^*(\bullet, \circ)$. As usual, terms can be represented by ordered unranked trees, hence we obtain another representation, that of the *tree representation of biwords*. The above-mentioned representations of a biword are illustrated in the figure below.

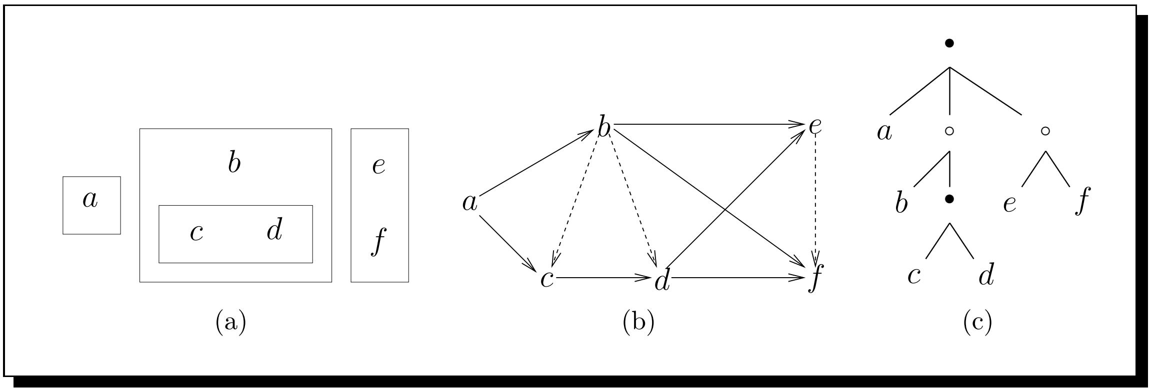

Figure 1: The two-dimensional word representation (a); the biposet representation (b); and the tree representation (c) of the biword $a \bullet (b \circ (c \bullet d)) \bullet (e \circ f)$.

Parenthesizing Automata

In the following we investigate the possibility of extending the four basic concepts (namely recognizability, logical definability, regularity and rationality) to binoid languages. Let us begin with regularity. To achieve an extension we introduce the concept of *parenthesizing automata*. Let Ω denote some finite *set of parentheses*. Of course, Ω and Σ are always disjoint, and elements of Ω are usually written as $\langle_1, \rangle_1, \langle_2, \rangle_2, \ldots$

[1]The horizontal/vertical product is also called the series/paralell product, hence the reason for the abbreviation.

Definition 3.1[2] **([ÉN04])** *A (nondeterministic)* parenthesizing automaton, *PA for short, is a 9-tuple* $\mathcal{A} := (S, H, V, \Sigma, \Omega, \delta, \gamma, I, F)$*, where* S *is a nonempty, finite set of* states*;* H *and* V *are the sets of* horizontal *and* vertical states *which give a disjoint partition of* S*,* Σ *is the* input alphabet *and* Ω *is a finite set of parentheses. Furthermore*

- $\delta \subseteq (H \times \Sigma \times H) \cup (V \times \Sigma \times V)$ *is the* labeling transition relation*,*
- $\gamma \subseteq (H \times \Omega \times V) \cup (V \times \Omega \times H)$ *is the* parenthesizing transition relation*,and*
- $I, F \subseteq S$ *are the sets of* initial *and* final states*, respectively.*

Example 3.2 A simple illustration of a PA is given in Figure 2. The horizontal states are those labeled by H_i and the vertical states are those labeled by V_j, for some i and j. There is a single initial state H_1, and a single final state H_7. After defining the notion of a run, we see that this automaton has a single run from H_1 to H_7, hence the automaton just accepts the biword $a \bullet (b \circ (c \bullet d)) \bullet e$. Of course, if the automaton had cycles, the accepted binoid language would be more complicated than in our example.

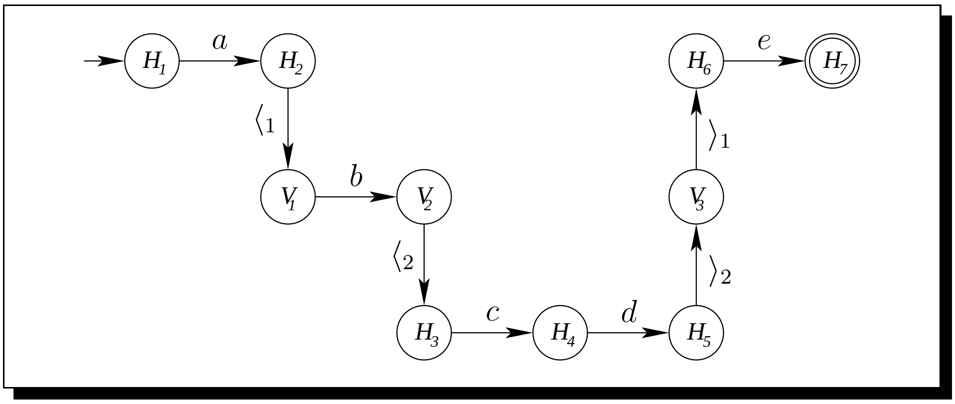

Figure 2: A PA accepting $\{ a \bullet (b \circ (c \bullet d)) \bullet e \}$.

Our next goal is to define the operation of parenthesizing automata formally. Let $\mathcal{A} = (S, H, V, \Sigma, \Omega, \delta, \gamma, I, F)$ be a PA. If $t = (p, x, q)$ is a labeling or parenthesizing transition of $\mathcal{A}$, i.e. $t \in \delta \cup \gamma$, then the *starting* and the *ending state* of t will be denoted by $\mathrm{start}(t) := p$ and $\mathrm{end}(t) := q$, respectively. Two transitions t_1 and t_2 are *adjacent* (in this order) if $\mathrm{end}(t_1) = \mathrm{start}(t_2)$. The words from $(\delta \cup \gamma)^*$ will be called *transition sequences*, but we will demand that in any transition sequence the consecutive transitions be adjacent. The concatenation of two transition sequences $\mathbf{r_1}$ and $\mathbf{r_2}$ will be denoted by $\mathbf{r_1 r_2}$, as usual. If $\mathbf{r} = t_1 t_2 \ldots t_n \in (\delta \cup \gamma)^*$ is a transition sequence, then let $\mathrm{start}(\mathbf{r}) := \mathrm{start}(t_1)$ and $\mathrm{end}(\mathbf{r}) := \mathrm{end}(t_n)$. Here we say that two

[2]The numbering of the definitions and theorems in this abstract follows that of the thesis.

parenthesizing transitions $t_1 = (p, \omega_1, q)$ and $t_2 = (s, \omega_2, t) \in \gamma$ form a *parenthesizing transition pair* if ω_1 is an opening parenthesis and ω_2 is the corresponding closing parenthesis.

Definition 3.7 ([Ném07]) *Let $\mathcal{A}$ be a PA. The set of its* runs, $\mathrm{Runs}(\mathcal{A})$*, is the least set of transition sequences that contains*

(i) *the* singleton runs: (p, σ, q)*, for all* $(p, \sigma, q) \in \delta$*;*

(ii) *the* direct runs: $\mathbf{r_1 r_2}$*, for every* $\mathbf{r_1}, \mathbf{r_2} \in \mathrm{Runs}(\mathcal{A})$ *with* $\mathrm{end}(\mathbf{r_1}) = \mathrm{start}(\mathbf{r_2})$*;*

(iii) *the* indirect runs: $t_1 \mathbf{r}\, t_2$*, for every direct run* $\mathbf{r} \in \mathrm{Runs}(\mathcal{A})$*, and parenthesizing transition pair* t_1*,* t_2 *with* $\mathrm{end}(t_1) = \mathrm{start}(\mathbf{r})$ *and* $\mathrm{end}(\mathbf{r}) = \mathrm{start}(t_2)$*.*

Definition 3.8 ([Ném07]) *Suppose that $\mathcal{A}$ is a PA and $\mathbf{r} \in \mathrm{Runs}(\mathcal{A})$. Then the* label *of $\mathbf{r}$ is a biword from $\Sigma^*(\bullet, \circ)$ defined inductively as follows:*

(i) *If* $\mathbf{r} = (p, \sigma, q)$*, then* $\mathrm{Label}(\mathbf{r}) := \sigma$*.*

(ii) *If $\mathbf{r}$ is a direct run, and $\mathbf{r} = \mathbf{r_1 r_2}$ for some $\mathbf{r_1}, \mathbf{r_2} \in \mathrm{Runs}(\mathcal{A})$, then*

- *if* $\mathrm{end}(\mathbf{r_1}) \in H$*, then* $\mathrm{Label}(\mathbf{r}) := \mathrm{Label}(\mathbf{r_1}) \bullet \mathrm{Label}(\mathbf{r_2})$*;*
- *if* $\mathrm{end}(\mathbf{r_1}) \in V$*, then* $\mathrm{Label}(\mathbf{r}) := \mathrm{Label}(\mathbf{r_1}) \circ \mathrm{Label}(\mathbf{r_2})$*.*

(iii) *If $\mathbf{r}$ is an indirect run $\mathbf{r} = t_1 \mathbf{r}'\, t_2$, then* $\mathrm{Label}(\mathbf{r}) := \mathrm{Label}(\mathbf{r}')$*.*

Since $\bullet$ and $\circ$ are associative, the definition of $\mathrm{Label}(\mathbf{r})$ does not depend on the choice of factorization in case (ii) above. The binoid language accepted by a PA is defined as the labels of runs from an initial to a final state, as usual. Moreover, if there is state that belongs to both the initial and final states, the empty biword also belongs to the language of the PA. Of course a binoid language is regular if it can be accepted by a PA.

In the thesis we prove that every PA is equivalent to one in normal form, i.e. with a single initial and a single final state. These two states can be chosen to be two horizontal as well as two vertical states.

The horizontal and vertical products can be naturally extended from biwords to binoid languages, and one can define the Kleene-iterations of the two products in the usual way, as well. Now assume that $L_1, L_2 \subseteq \Sigma^*(\bullet, \circ)$ and let $\xi \in \Sigma$, the ξ-*substitution* of L_2 into L_1 be denoted by $L_1[L_2/\xi]$. It is obtained by non-uniformly substituting biwords in L_2 for ξ in the members of L_1 (cf. [GS84]).

Theorem 3.25 ([ÉN04]) *The class* Reg *of regular binoid languages is (effectively) closed under ξ-substitution, i.e. if $L_1 \subseteq (\Sigma \cup \{\, \xi \,\})^*(\bullet, \circ)$, $L_2 \subseteq \Sigma^*(\bullet, \circ)$, then $L_1, L_2 \in$* Reg *implies* $L_1[L_2/\xi] \in \mathsf{Reg}$*.*

Using the above theorem we can immediately derive some further closure properties of Reg.

Corollary 3.26 ([ÉN04]) *The class* Reg *of regular binoid languages is (effectively) closed under horizontal and vertical products, horizontal and vertical iterations, and homomorphisms.*

An important feature of parenthesizing automata that an automata may possess any finite number of parenthesis pairs. Let Reg_i stand for the class of those regular binoid languages which can be accepted by a PA which has at most i pairs of parenthesis symbols ($i \geq 0$).

Theorem 3.32 ([Ném04]) *The classes* $\mathsf{Reg}_0 \subsetneq \mathsf{Reg}_1 \subsetneq \mathsf{Reg}_2 \subsetneq \dots$ *form a strict hierarchy of regular (i.e. recognizable) binoid languages.*

Recognizability, MSO-definability and Rationality

The concept of *recognizable binoid languages*, i.e. recognizable subsets of $\Sigma^*(\bullet, \circ)$ can be derived from standard general notions of universal algebra (cf. [GS84]).

Definition 3.34 *A binoid language* $L \subseteq \Sigma^*(\bullet, \circ)$ *is* recognizable *if there is a finite binoid* B*, a homomorphism* $h : \Sigma^*(\bullet, \circ) \to B$*, and a set* $F \subseteq B$ *with* $L = h^{-1}(F)$.

Formulating the concept of *logical definability* is not as straightforward as recognizability, but it can be done with the help the sp-biposet representation. Indeed biposets, and also their special cases – sp-biposets – are relation structures, and this allows us to interpret logical formulas on them, as in biposets the horizontal and vertical order relations are explicitly present.

Next, we consider several *rational* classes of binoid languages, whose definitions depend on what operations are allowed from the following list: Boolean operations (union, intersection, complementation), horizontal product ($\bullet$), vertical product ($\circ$), horizontal iteration ($*_\bullet$) and vertical iteration ($*_\circ$). Let $\mathsf{Fin}[\mathrm{op}_1, \dots, \mathrm{op}_n]$ denote the class of those binoid languages that can be generated from the finite binoid languages by a finite number of applications of the operations op_1,…,op_n. In the thesis the following classes are defined

- $\mathsf{HRat} = \mathsf{Fin}[\cup, \bullet, *_\bullet, \circ]$ the *horizontal rational languages*,
- $\mathsf{VRat} = \mathsf{Fin}[\cup, \circ, *_\circ, \bullet]$ the *vertical rational languages*,
- $\mathsf{BRat} = \mathsf{Fin}[\cup, \bullet, *_\bullet, \circ, *_\circ]$, the *birational languages*,
- $\mathsf{GRat} = \mathsf{Fin}[\cup, \bullet, *_\bullet, \circ, *_\circ, {}^-]$, the *generalized birational languages*, where $^-$, is the operation of taking the complement.

As usual, a binoid language is called *finite* if it contains a finite number of biwords. Similarly, a binoid language $L \subseteq \Sigma^*(\bullet, \circ)$ is *cofinite* if its complementer with respect to $\Sigma^*(\bullet, \circ)$ is finite. We denote the class of *finite languages* by Fin.

A Comparison of the Basic Classes

Our main results for binoid languages are the following.

Theorem 3.35 ([ÉN04]) $\mathsf{Rec} = \mathsf{Reg}$*, i.e. a binoid language* $L \subseteq \Sigma^*(\bullet, \circ)$ *is recognizable if and only if* L *is regular.*

Theorem 3.70 ([ÉN04]) $\mathsf{Rec} = \mathsf{MSO}$*, i.e. a binoid language* $L \subseteq \Sigma^*(\bullet, \circ)$ *is recognizable if and only if* L *is* MSO*-definable.*

Theorem 3.59 ([ÉN04]) *It is decidable whether a regular binoid language is finite, cofinite, birational, horizontal rational or vertical rational.*

It is usual, and sometimes even necessary, to apply some restrictions on the structures under study. These restrictions sometimes naturally arise due to practical limitations – e.g. the finite number of the available processors. Here we will study three such restricted classes of binoid languages: HB – the class of horizontally bounded languages, VB – the class of vertically bounded languages, and BD – the class of bounded (alternation) depth languages.

As for the definitions, the easiest way to define horizontally and vertically bounded binoid languages is through their sp-biposet representations. Recall that a *chain* in a poset is a subset in which each pair of elements is comparable, i.e. a totally ordered subset. The *height* of a poset is the cardinality of a longest (maximum cardinality) chain. If $(P, <_h, <_v, \lambda)$ is a biposet, let its *horizontal height* be the height of the poset $(P, <_h)$. Similarly, let its *vertical height* be the height of the poset $(P, <_v)$. Now a binoid language is *horizontally* (resp. *vertically*) *bounded* if there is an upper bound for the horizontal (resp. vertical) height of the sp-biposet representations of its elements.

We say that a binoid language L has a *bounded depth* if there is an integer K such that, for every biword $w \in L$, the maximal depth of nested parenthesization in the term representation of w is at most K. Let BD denote the class of binoid languages that have a bounded depth.

We established the inclusion relations among the considered classes. These can be summarized in Figure 3. Moreover, we proved that all inclusions suggested by the figure are strict.

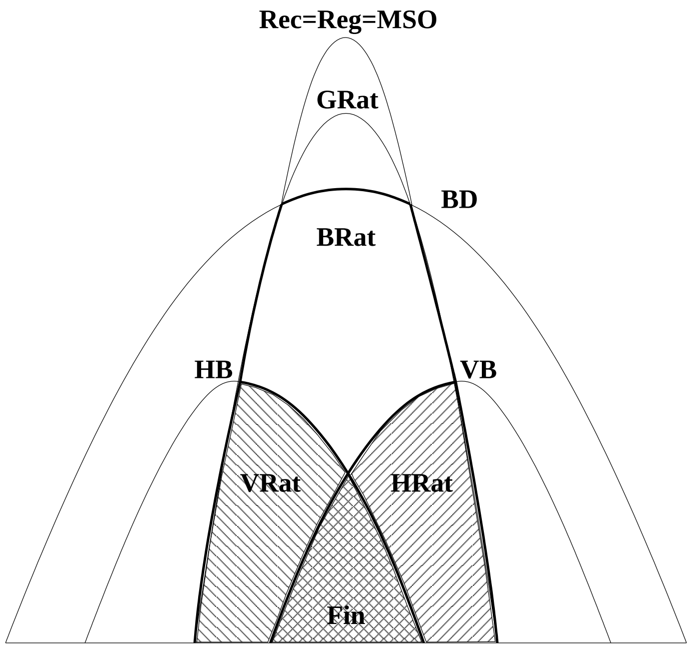

Figure 3: A comparison of language classes of finite biwords.

Therefore for binoid languages of bounded depth the equivalence of regularity, recognizability and MSO-definability can be extended with two additional characterizations of rationality.

Corollary 3.71 ([ÉN04]) *The following conditions are equivalent for a language $L \subseteq \Sigma^*(\bullet, \circ)$ of bounded depth:*

1. *L is recognizable.*
2. *L is regular.*
3. *L is birational.*
4. *L is generalized birational.*
5. *L is* MSO*-definable.*

When L is vertically bounded, the above conditions are also equivalent to the condition that L is horizontal rational.

In the thesis we relate our notion of automata and regularity to that of Hashiguchi et al. We find that their notion of regularity is less general than ours. Moreover, we are able to extend their monoid approach to our broader class of regular binoid languages. This means that with appropriate definitions ordinary finite automata are also capable of capturing the same concept of regularity. This gives a fourth equivalent characterization of the class Reg in the general unbounded depth case.

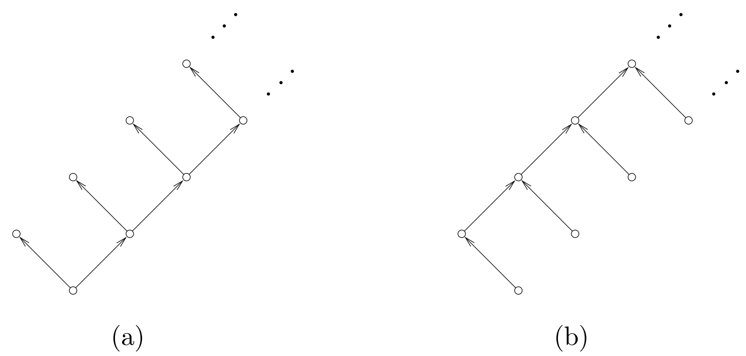

Figure 4: An upward comb (a) and a downward comb (b).

Languages of Infinite Biwords

In Chapter 4 of the thesis we extend our investigations to infinite biwords. First we define ω-bisemigroups in the pattern of ω-semigroups related to infinite words [PP04]. Now ω-biwords as abstract objects are just the elements of the free ω-bisemigroups.

Similarly to the finite case, we can represent ω-biwords by certain infinite biposets. For this notice that the products $-\bullet$ and $\circ-$ of two finite biposets can obviously be extended to the product of a finite biposet with an infinite one. Moreover, the two product operations also give rise to two ω-ary product operations. This means that we can define both the horizontal and the vertical product of a countably infinite number of finite biposets. We call a Σ-labeled biposet *constructible* if it can be generated from the singleton Σ-labeled biposets by the binary and the ω-ary product operations. In the thesis we show that the elements of the free ω-bisemigroups can be described by constructible ω-biposets. Then we present a graph-theoretic characterization of infinite constructible ω-biposets. This can be stated as follows:

Theorem 4.3 ([ÉN05]) *An infinite biposet $(P, <_h, <_v, \lambda)$ is a constructible biposet if and only if P is complete, and both posets $(P, <_h)$ and $(P, <_v)$*

(i) *are N-free,*

(ii) *are free of "upward combs"*[3]*,*

(iii) *are free of "downward combs", and*

(iv) *all of their principal ideals*[4] *are finite.*

Afterwards, we examine the tree and term representations of ω-biwords. It is followed by the extension of recognizability, MSO-definability and regularity to ω-binoid languages. To extend regularity we also need to define the concept and the

[3]"Upward comb" and "downward comb" are certain infinite posets which are depicted in Figure 4.

[4]A principle ideal is a collection of all elements which are less than or equal to a given element.

operation of parenthesizing Büchi-automata. The main result for ω-binoid languages is the generalization of the equivalences from the finite case.

Theorem 4.25 ([Ném06]) *Let L be an ω-binoid language. Then L is recognizable if and only if L is regular if and only if L is* MSO*-definable.*

Publications of the Results

Much of the material of this thesis is based on the following publications:

[ÉN04] Z. Ésik and Z. L. Németh, Higher dimensional automata. *J. of Autom. Lang. Comb.* **9** (2004), 3–29.

[ÉN05] Z. Ésik and Z. L. Németh, Algebraic and graph-theoretic properties of infinite n-posets. *Theoret. Informatics Appl.* **39** (2005), 305–322.

[Ném04] Z. L. Németh, A hierarchy theorem for regular languages over free bisemigroups. *Acta Cybern.* **16** (2004), 567–577.

[Ném06] Z. L. Németh, Automata on infinite biposets. *Acta Cybern.* **18** (2006), 765–797.

[Ném07] Z. L. Németh, On the regularity of binoid languages: a comparative approach. In: preproc. *1st Int. Conf. on Language and Automata Theory and Appl., LATA'07*, March 29 – April 4, 2007, Tarragona, Spain.

Chapter 2 contains several ideas taken from the introductory parts of three papers [ÉN04, Ném06, Ném07]. The primary source of Chapter 3 is [ÉN04], but two sections of it, namely Section 3.8 and 3.12 present the results given in [Ném04] and [Ném07], respectively. Finally, Chapter 4 is based on the concepts and results given in [ÉN05] and [Ném06].

The thesis seeks to provide more than just the enumeration of the results of the above papers. It attempts to offer a precise account of the subject of regular binoid languages, with more detailed proofs and examples, along with justifications of the new concepts and conclusions. It also offers a new outlook on solved and unsolved problems, and suggests possible future directions of research.

Conclusions

In this thesis we laid the foundations for a two-dimensional theory of automata and languages. For the generalization from the one-dimensional case of words we adopted an algebraic approach, namely we considered languages over free binoids. It is a generalization of monoids where two independent associative operations are defined and they share a common identity element. We managed to generalize the equivalence of regularity, recognizability and MSO-definability from word languages to binoid languages and to ω-binoid languages as well. We also introduced various concepts of rational binoid languages and examined their relationships. All the results can be generalized to higher dimensions, i.e. to free algebras where three or more independent associative operations are present.

For the concept of regularity we introduced a new automata model called parenthesizing automata. This model is one of the main contributions of the thesis. The equivalence of regularity with recognizability and MSO-definability can be interpreted as a justification of the point that our concept of PA captures an essential and robust class of binoid languages. From this equivalence some closure properties of recognizable languages can be readily derived. Moreover, with the help of the new automata model we gave a more refined classification of regular binoid languages, since – by our hierarchy theorem (Theorem 3.32) – the minimal number of parentheses in automata needed to accept a given binoid language provides a complexity measure on the class of regular binoid languages.

We cannot deny that the results of this thesis are really just the first steps in the investigations of binoid languages. Not surprisingly, several problems remain open. We managed to generalize the equivalence of regularity, recognizability and MSO-definability from word languages to binoid languages, but we only succeeded in defining an equivalent concept of rationality in the bounded depth case. But what are the operations on binoid languages that capture the behavior of PA? We did not deal with first-order definable binoid languages. Their decidability and algebraic characterization are open problems as well. Two fundamental algorithms of classical automata theory are the determinization and minimization of automata. Can they be extended to parenthesizing automata?

In the thesis we mostly concentrate on the general theory of regularity, but we believe that the concept of binoid languages is sufficiently general to have some practical applications as well. The reader can peruse the study by Hashiguchi et al. on bicodes [HKJ02] and on a modified RSA cryptosystem based on bicodes [HHJ03]. In the future biwords may also be used in modeling the behaviors of concurrent systems, like sp-posets, which often serve as models for the behavior of modularly constructed concurrent systems (cf. [Pra86]). It would also be good to look for other concrete applications of our theory. Since the special feature of biwords and their n-dimensional generalizations is that they are naturally equipped with some *nested structures*, it seems obvious to look for applications where some nestedness (of arbitrary depth) is present, e.g. in XML databases and in modeling recursive function calls.

References

[AM04] R. Alur, P. Madhusudan, Visibly pushdown languages. In: proc *STOC 2004*, Chicago, IL, USA, June 13-16, 2004, 202–211.

[AM06] R. Alur, P. Madhusudan, Adding nesting structure to words. In: proc *DLT'06*, LNCS **4036**, 1-13.

[Cou91] B. Courcelle, The monadic second-order logic on graphs. V. On closing the gap between definability and recognizability. *Theoret. Comput. Sci.* **80** (1991), 153–202.

[CW05] B. Courcelle, P. Weil, The recognizability of sets of graphs is a robust property. *Theoret. Comput. Sci.* **342** (2005), 173-228.

[DR95] V. Diekert, G. Rozenberg (eds.), *The book of traces*. World Scientific Publishing Co., Inc., River Edge, NJ, 1995.

[Dol05] I. Dolinka, A note on identities of two-dimensional languages. *Disc. Appl. Math.* **146** (2005), 43–50.

[Dol07] I. Dolinka, Axiomatizing the identities of two-dimensional languages. *Theoret. Comput. Sci.* **372** (2007), 1–14.

[Ési00] Z. Ésik, Free algebras for generalized automata and language theory. *RIMS Surikaisekikenkyusho Kokyuroku*, Kyoto University, Kyoto, **1166** (2000), 52–58.

[ÉN04] Z. Ésik, Z. L. Németh, Higher dimensional automata. *J. of Autom. Lang. Comb.* **9** (2004), 3–29.

[ÉN05] Z. Ésik, Z. L. Németh, Algebraic and graph-theoretic properties of infinite n-posets. *Theoret. Informatics Appl.* **39** (2005), 305–322.

[GS84] F. Gécseg, M. Steinby, *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.

[GR97] D. Giammarresi, A. Restivo, Two-dimensional languages. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of formal languages*, Vol. 3, Springer-Verlag, Berlin, 1997, 215–267.

[Gra81] J. Grabowski, On partial languages. *Fund. Inform.* **4** (1981), 427–498.

[HHJ03] K. Hashiguchi, K. Hashimoto, S. Jimbo. Modified RSA cryptosystems over bicodes. *Advances in algebra,* World. Sci. Publ., NJ, 2003, 377–389.

[HIJ00] K. Hashiguchi, S. Ichihara, S. Jimbo, Formal languages over free binoids. *J. Autom. Lang. Comb.* **5** (2000), 219–234.

[HKJ02] K. Hashiguchi, T. Kunai, S. Jimbo, Finite codes over free binoids. *J. Autom. Lang. Comb.* **7** (2002), 505–518.

[HSJ04] K. Hashiguchi, Y. Sakakibara, S. Jimbo, Equivalence of regular binoid expressions and regular expressions denoting binoid languages over free binoids. *Theoret. Comput. Sci.* **312** (2004), 251–266.

[HWJ03] K. Hashiguchi, Y. Wada, S. Jimbo, Regular binoid expressions and regular binoid languages. *Theoret. Comput. Sci.* **304** (2003), 291–313.

[HtP96] H. J. Hoogeboom, P. ten Pas, Text languages in an algebraic framework. *Fund. Inform.* **25** (1996), 353–380.

[HtP97] H. J. Hoogeboom, P. ten Pas, Monadic second-order definable text languages. *Theory Comput. Syst.* **30** (1997), 335–354.

[Kus03a] D. Kuske, Towards a language theory for infinite N-free pomsets. *Theoret. Comput. Sci.* **299** (2003), 347–386.

[Kus03b] D. Kuske, Regular sets of infinite message sequence charts. *Inform. and Comput.* **187** (2003), 80–109.

[LW98] K. Lodaya, P. Weil, Kleene iteration for parallelism. In: proc. *FST & TCS'98*. LNCS **1530**, Springer-Verlag, 1998, 355–366.

[LW00] K. Lodaya, P. Weil, Series-parallel languages and the bounded-width property. *Theoret. Comput. Sci.* **237** (2000), 347–380.

[LW01] K. Lodaya, P. Weil, Rationality in algebras with a series operation. *Inform. and Comput.* **171** (2001), 269–293.

[Ném04] Z. L. Németh, A hierarchy theorem for regular languages over free bisemigroups. *Acta Cybern.* **16** (2004), 567–577.

[Ném06] Z. L. Németh, Automata on infinite biposets. *Acta Cybern.* **18** (2006), 765–797.

[Ném07] Z. L. Németh, On the regularity of binoid languages: a comparative approach. In: preproc. *1st International Conference on Language and Automata Theory and Applications, LATA'07*, March 29 – April 4, 2007, Tarragona, Spain.

[PP04] D. Perrin, J.-E. Pin, *Infinite words.* Pure and Applied Mathematics, Vol. **141**, Elsevier, 2004.

[Pra86] V. R. Pratt, Modeling concurrency with partial orders. *International Journal of Parallel Programming* **15** (1986), 33–71.

[RS97] G. Rozenberg, A. Salomaa (eds.), *Handbook of formal languages.* Vols. 1–3., Springer-Verlag, Berlin, 1997.

[Wei04a] P. Weil, Algebraic recognizability of languages. In: proc. *MFCS'04*, LNCS **3153**, Springer-Verlag, 2004, 149–175.

[Wil94] Th. Wilke, An algebraic theory for regular languages of finite and infinite words. *Internat. J. Algebra Comput.* **3** (1993), 447–489.