A Hierarchy Theorem for Regular Languages over Free Bisemigroups

Zoltán L. Németh
Dept. of Computer Science
University of Szeged
P.O.B. 652
6701 Szeged, Hungary
zlnemeth@inf.u-szeged.hu

June 10, 2004

Abstract

In this article a question left open in [2] is answered. In particular, we show that it is essential that in the definition of parenthesizing automata an arbitrary number of parentheses can be used. Moreover, we prove that the classes $\mathsf{Reg_m}$ of languages accepted by a parenthesizing automaton with at most m pairs of parentheses form a strict hierarchy. In fact, this hierarchy is proper for all alphabets.

1 Introduction

A bisemigroup is set equipped with two associative operations. In [2], the notion of parenthesizing automaton operating on elements of free bisemigroups was introduced. The elements of free bisemigroups can be represented by labelled series-parallel biposets, or sp-biposets, for short. Biposets are sets with two partial order relations and a labelling function defined on them. A biposet is series-parallel if it can be obtained from the singleton biposets by the two associative operations corresponding to the order relations, called the series product ($\bullet$) and the parallel product ($\circ$). Hence, subsets of free bisemigroups accepted by parenthesizing automata will be called regular sp-biposet languages here. This concept of regularity proved to be appropriate for the characterization of algebraic recognizability, which is a very general notion well established in a universal algebraic setting: recognizability means to be recognized by a homomorphism into a finite bisemigroup.

An important feature of parenthesizing automata is that it is allowed to use any finite number of pairs of parentheses. The question emerges naturally if this feature is really necessary, or the number of parentheses can be bounded. In other words, we want to know whether there is a number K such that each regular sp-biposet language can be accepted by a parenthesizing automaton with at most K pairs of parentheses. This article gives the answer to this question. We show that no such K exists. Moreover, if $\mathsf{Reg_m}$ denotes the class of all regular sp-biposet languages that can be accepted by an automaton with $m \geq 0$ pairs of parentheses, then the classes $\mathsf{Reg_0} \subsetneq \mathsf{Reg_1} \subsetneq \mathsf{Reg_2} \subsetneq \ldots$ form a strict hierarchy. Furthermore, we prove that this hierarchy is proper even when we consider languages over any fixed alphabet Σ.

This work can be seen principally as an addition to the paper of Ésik and the author on higher dimensional automata [2, 3]. (The latter is the journal version of the former.) Here we only briefly enumerate some related papers and refer to [3], where a whole section is devoted to a detailed comparison.

The theory of automata on biposets is closely related to the work of Lodaya and Weil [10, 11, 12] and Kuske [8, 9] on poset (partially ordered set) languages and on automata on (labelled posets). They studied the class of N-free (or in their terminology series-parallel) posets from the motivation of modelling concurrency. It is known that the N-free posets represent the elements of the free "semi-commutative bisemigroups", i.e., of the algebras equipped with an associative and an associative and commutative operation [4].

Automata and languages over free bisemigroups (more precisely, free bisemigroups with identity, called binoids or bimonoids) have also been studied in Hashiguchi et al., see [5, 6, 7]. But there the elements of the free binoid over Σ are represented by ordinary words in "standard form", which are the shortest expressions representing them over the extended alphabet $\Sigma \cup \{\bullet, \circ, (,)\}$. Ordinary finite automata are used to accept binoid languages. In [6] and [7], the notion of regular binoid expression is introduced defining the least class of binoid languages that contains the finite languages, and closed under the operations of union, the two product operations on languages, and the two iterations operations corresponding to the products. This class corresponds to the class of birational sp-biposet languages in [3]. The main result of [6] and [7] is that the binoid languages over Σ denoted by regular binoid expressions are those binoid languages, whose elements (in standard form) constitute regular word languages over the extended alphabet $\Sigma \cup \{\bullet, \circ, (,)\}$. This notion of regularity is less general than ours, e.g., the language of all biposets over an alphabet Σ cannot be represented by a regular binoid expression, but it can be accepted by a parenthesizing automaton with a single pair of parentheses. In fact, any language denoted by a regular binoid expression can be accepted with a single pair of parentheses.

2 Bisemigroups and biposets

In the sequel, n always denotes a positive integer and Σ a finite alphabet. We write $[n]$ for the set $\{1, 2, \ldots, n\}$. We use the notation Σ^n for the set of words over Σ of length n, and write Σ^* for the set of all words over Σ, as usual. Σ_m stands for any alphabet that has m letters. Let Ω denote a finite set of parentheses, its elements are usually written as $\langle_1, \rangle_1, \langle_2, \rangle_2, \ldots$. We assume that Ω is partitioned into the sets Ω_{op} of opening and Ω_{cl} of closing parentheses, which are in bijective correspondence.

We call a set equipped with n associative operations an *n-semigroup.* Automata operate on elements of some free algebra in general, on words in the classical case, i.e., on elements of free semigroups. Thus, if we want to generalize the notion of automata to higher dimensions, it is natural to investigate how they operate on elements of free n-semigroups. In the following sections we only consider the case when $n = 2$, but all of our results can be extended to any nonnegative integer n in a straightforward way. When $n = 2$, we call an n-semigroup a *bisemigroup.* So a bisemigroup is a finite set S with two associative binary operations on it, called the *horizontal product* and the *vertical product.* We denote the two products by $\bullet$ and $\circ$, respectively.

We have several possible ways to describe the elements of a bisemigroup, freely generated by some finite alphabet Σ. First, they can be represented as terms over the extended alphabet $\Sigma \cup \{\bullet, \circ, (,)\}$ modulo associativity, or we can consider the (finite ordered) tree representations of these terms. But an element of a free bisemigroup can also be represented by a finite sp-biposet defined below.

Definition 2.1 *A Σ-labelled* biposet *or* biposet*, for short, is a structure $(P, <_h^P, <_v^P, \lambda_P)$, where P is a finite nonempty set of vertices, $<_h^P$ and $<_v^P$ are (irreflexive) partial orders on P and $\lambda_P : P \to \Sigma$ is a labelling function.*

We say that two biposets are *isomorphic* if there is a bijective function on the vertices that preserves the partial orders and the labelling. Below we will identify isomorphic biposets.

Suppose that $P = (P, <_h^P, <_v^P, \lambda_P)$ and $Q = (Q, <_h^Q, <_v^Q, \lambda_Q)$ are Σ-labelled biposets. Without loss of generality, assume that P and Q are disjoint. We define their *horizontal product* as $P \bullet Q = (P \cup Q, <_h^{P\bullet Q}, <_v^{P\bullet Q}, \lambda_{P\bullet Q})$, and their *vertical product* as $P \circ Q = (P \cup Q, <_h^{P\circ Q}, <_v^{P\circ Q}, \lambda_{P\circ Q})$, where

$$\begin{array}{rclrcl} <_h^{P\bullet Q} & = & <_h^P \cup <_h^Q \cup (P \times Q), & <_h^{P\circ Q} & = & <_h^P \cup <_h^Q, \\ <_v^{P\bullet Q} & = & <_v^P \cup <_v^Q, & <_v^{P\circ Q} & = & <_v^P \cup <_v^Q \cup (P \times Q), \end{array}$$

and the labellings are $\lambda_{P\bullet Q} = \lambda_{P\circ Q} = \lambda_P \cup \lambda_Q$.

It is obvious that both product operations are associative. Each letter $\sigma \in \Sigma$ may be identified with the singleton biposet labelled σ. Let SPB(Σ) denote the collection of biposets that can be generated from the singletons corresponding to the letters in Σ by the two product operations. The biposets in SPB(Σ) are called *series-parallel biposets*, or *sp-biposets*, for short.

As we mentioned above, the sp-biposets SPB(Σ) may serve as a possible description of the free bisemigroup generated by Σ. This fact is formulated in the following theorem.

Proposition 2.2 [1] SPB(Σ) *is freely generated by Σ in the variety of bisemigroups.*

3 Parenthesizing Automata

In this section we define parenthesizing automata that process sp-biposets in a sequential manner. The class of sp-biposet languages accepted by parenthesizing automata will be called regular sp-biposet languages.

Definition 3.1 *A (nondeterministic)* parenthesizing automaton *is a 9-tuple* $\mathcal{A} = (S, H, V, \Sigma, \Omega, \delta, \gamma, I, F)$*, where* S *is the nonempty, finite set of* states*,* H *and* V *are the sets of* horizontal *and* vertical states*, which give a disjoint decomposition of* S*,* Σ *is the* input alphabet*,* Ω *is a finite set of parentheses, moreover,*

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

We say that a biposet P has a *horizontal decomposition* into the horizontal product of biposets $P_1, P_2, \ldots, P_n$, $n \geq 2$, if $P = P_1 \bullet P_2 \bullet \ldots \bullet P_n$. A horizontal decomposition is said to be *maximal* if no component P_i, $(1 \leq i \leq n)$ has a horizontal decomposition. Vertical decompositions and maximal vertical decompositions are defined similarly.

The operation of the parenthesizing automata is based on the notion of the run, defined as follows.

Definition 3.2 *Suppose that* $P \in \mathrm{SPB}(\Sigma)$ *and* $p, q \in S$*. We say that* $\mathcal{A} = (S, H, V, \Sigma, \Omega, \delta, \gamma, I, F)$ has a run on P from p to q*, denoted* $[p, P, q]_{\mathcal{A}}$ *if one of the following conditions holds.*

(Base) $P = \sigma \in \Sigma$ *and* $(p, \sigma, q) \in \delta$*.*

(HH) $p, q \in H$ *and* P *has maximal horizontal decomposition* $P = P_1 \bullet \ldots \bullet P_n$*, where* $n \geq 2$*, and there exist* $r_1, \ldots, r_{n-1} \in S$*,* $r_0 = p$*,* $r_n = q$ *such that* $[r_{i-1}, P_i, r_i]_{\mathcal{A}}$*, for all* $i \in [n]$*.*

(VV) $p, q \in V$ *and* P *has maximal vertical decomposition* $P = P_1 \circ \ldots \circ P_n$*, where* $n \geq 2$*, and there exist* $r_1, \ldots, r_{n-1} \in S$*,* $r_0 = p$*,* $r_n = q$ *such that* $[r_{i-1}, P_i, r_i]_{\mathcal{A}}$ *for all* $i \in [n]$*.*

(HV) $p, q \in H$ *and* P *has maximal vertical decomposition* $P = P_1 \circ \ldots \circ P_n$*, where* $n \geq 2$*, and there exist* $\langle_k, \rangle_k \in \Omega$*,* $p', q' \in V$ *and* $(p, \langle_k, p'), (q', \rangle_k, q) \in \gamma$ *such that* $[p', P, q']_{\mathcal{A}}$ *holds.*

(VH) $p, q \in V$ *and* P *has maximal horizontal decomposition* $P = P_1 \bullet \ldots \bullet P_n$*, where* $n \geq 2$*, and there exist* $\langle_k, \rangle_k \in \Omega$*,* $p', q' \in H$ *and* $(p, \langle_k, p'), (q', \rangle_k, q) \in \gamma$ *such that* $[p', P, q']_{\mathcal{A}}$ *holds.*

An *accepting run* is a run from an initial state to a final state. The sp-biposet *language* $L(\mathcal{A})$ *accepted by the automaton* $\mathcal{A}$ is defined as the set of all labels of the accepting runs. Formally,

$$L(\mathcal{A}) \;=\; \{P \in \mathrm{SPB}(\Sigma) \mid \exists i \in I,\ f \in F \;:\; [i, P, f]_{\mathcal{A}}\}.$$

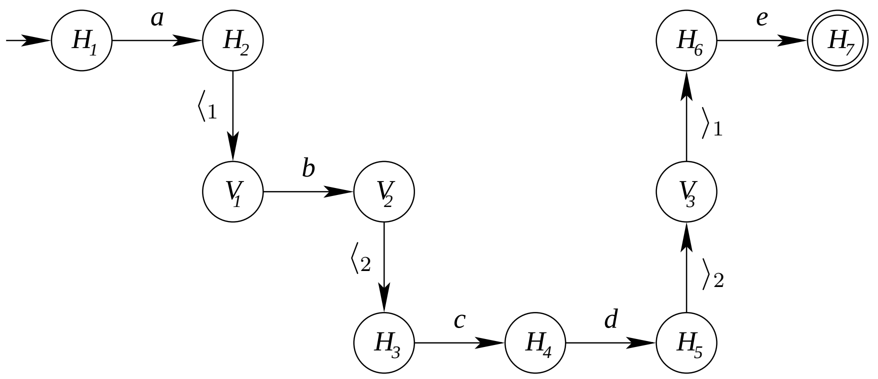

Figure 1: A parenthesizing automaton accepting $\{a \bullet (b \circ (c \bullet d)) \bullet e\}$.

Example 3.3 The automaton given on Figure 1 has a single initial state H_1 and a single final state H_7. The horizontal states are those labelled H_i and the vertical states those labelled V_j, for some i and j. This automaton accepts the singleton sp-biposet language consisting of the biposet $a \bullet (b \circ (c \bullet d)) \bullet e$. Indeed, there are runs $[H_1, a, H_2]_{\mathcal{A}}$, $[H_2, b \circ (c \bullet d), H_6]_{\mathcal{A}}$ and $[H_6, e, H_7]_{\mathcal{A}}$ corresponding to the horizontal decomposition of the biposet. Moreover, according to the case (HV) of Definition 3.2, the second run, $[H_2, b \circ (c \bullet d), H_6]_{\mathcal{A}}$ starts with a parenthesizing transition $(H_2, \langle_1, V_1)$ followed by a subrun $[V_1, b \circ (c \bullet d), V_3]_{\mathcal{A}}$ and ends with the transition $(V_3, \rangle_1, H_6)$. The existence of the subrun $[V_1, b \circ (c \bullet d), V_3]_{\mathcal{A}}$ comes from the existence of the runs $[V_1, b, V_2]_{\mathcal{A}}$ and $[V_2, c \bullet d, V_3]_{\mathcal{A}}$ that can be seen similarly. Several other examples of parenthesizing automata can be found in [3].

Recall from [3] that an sp-biposet language $L \subseteq \mathrm{SPB}(\Sigma)$ is called *recognizable* if there is a finite bisemigroup B, a homomorphism $h : \mathrm{SPB}(\Sigma) \to B$, and a set $F \subseteq B$ with $L = h^{-1}(F)$. Let $\mathsf{Rec}(\Sigma)$ and $\mathsf{Reg}(\Sigma)$ denote the set of recognizable and regular sp-biposet languages of $\mathrm{SPB}(\Sigma)$, respectively. Moreover, write Rec and Reg for the classes of all regular and recognizable sp-biposet languages, respectively. One of the main results of [3] shows that these two classes coincide.

Theorem 3.4 [3] $\mathsf{Rec} = \mathsf{Reg}$, *i.e., an sp-biposet language* $L \subseteq \mathrm{SPB}(\Sigma)$ *is recognizable iff* L *is regular.*

4 Hierarchy Theorems

In this session we present the main hierarchy results of the paper, formulated in Theorem 4.9 and 4.10. Let Reg_m denote the regular sp-biposet languages that can be accepted

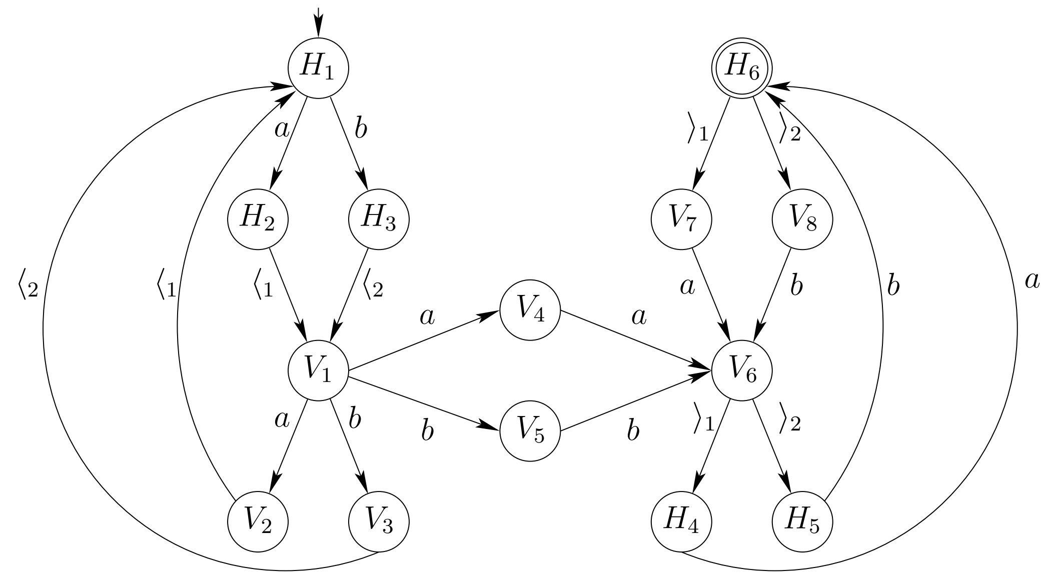

Figure 2: An automaton $\mathcal{A}_2$ accepting $\hat{L}$

by a parenthesizing automaton with at most $m \geq 0$ pairs of parentheses. We will show that $\mathsf{Reg_m} \subsetneq \mathsf{Reg_{m+1}}$ for all $m \geq 0$. Moreover, if $\mathsf{Reg_m}(\Sigma)$ is the set of all regular sp-biposet languages over any fixed alphabet Σ that can be accepted by an automaton with at most m pairs of parentheses, then $\mathsf{Reg_m}(\Sigma) \subsetneq \mathsf{Reg_{m+1}}(\Sigma)$, for all $m \geq 0$. Our first aim is to define a language $\hat{L}$ in $\mathsf{Reg_2} \backslash \mathsf{Reg_1}$.

Definition 4.1 *For any words $u = \sigma_1\sigma_2 \dots \sigma_n$ and $v = \rho_1\rho_2 \dots \rho_n$ of even length over any alphabet Σ, let $P_{uv^{-1}}$ denote the biposet*

$$P_{uv^{-1}} = \sigma_1 \bullet (\sigma_2 \circ (\sigma_3 \bullet (\sigma_4 \circ \dots (\sigma_{n-1} \bullet (\sigma_n \circ \rho_n) \bullet \rho_{n-1}) \dots \circ \rho_4) \bullet \rho_3) \circ \rho_2) \bullet \rho_1.$$

Definition 4.2 *Let Σ_2 denote the two-letter alphabet $\Sigma_2 = \{a, b\}$, and let*

$$\begin{aligned} \hat{L} &= \bigcup_{i=1}^{\infty} \hat{L}_{2i}, \textit{ where} \\ \hat{L}_2 &= \{\, \sigma \bullet (\sigma' \circ \sigma') \bullet \sigma \mid \sigma, \sigma' \in \Sigma_2 \,\}, \\ \hat{L}_{2i+2} &= \{\, \sigma \bullet (\sigma' \circ P \circ \sigma') \bullet \sigma \mid \sigma, \sigma' \in \Sigma_2 \textit{ and } P \in \hat{L}_{2i} \,\}, \textit{ for all } i \geq 1. \end{aligned}$$

Remark 4.3 Note that $\hat{L} = \{\, P_{ww^{-1}} \mid w \in \Sigma_2^*,\ w \text{ has even length} \,\}$.

Proposition 4.4 *$\hat{L}$ can be accepted by a parenthesizing automaton that has two pairs of parentheses, i.e., $\hat{L} \in \mathsf{Reg_2}(\Sigma_2)$.*

Proof. It is not hard to see that the automaton $\mathcal{A}_2$ in Figure 2 accepts $\hat{L}$. ☐

Our next aim is to show that $\hat{L}$ is not in $\mathsf{Reg}_1(\Sigma_2)$. In order to see this we must describe the runs of an automaton step by step. So we introduce the notion of generalized state of an automaton as follows.

Definition 4.5 *A* generalized state *of a parenthesizing automaton $\mathcal{A} = (S, H, V, \Sigma, \Omega, \delta, \gamma, I, F)$ is a pair (q, ω), where $q \in S$ and $\omega \in \Omega_{op}^*$.*

Definition 4.6 *We say that an automaton $\mathcal{A}$* has a transition *from a generalized state (q_1, ω_1) to a generalized state (q_2, ω_2) with respect to the symbol $\alpha \in \Sigma \cup \Omega$ if one of the following conditions holds:*

(i) $\alpha \in \Sigma$, $\omega_1 = \omega_2$ and $(q_1, \alpha, q_2) \in \delta$, or

(ii) $\alpha \in \Omega_{op}$, $\omega_2 = \omega_1\alpha$ and $(q_1, \alpha, q_2) \in \gamma$, or

(iii) $\alpha \in \Omega_{cl}$, $\omega_1 = \omega_2\bar{\alpha}$, where $\bar{\alpha}$ is the opening pair of the closing parenthesis α and $(q_1, \alpha, q_2) \in \gamma$.

Remark 4.7 Thus, every run $[p, P, q]_{\mathcal{A}}$ corresponds to a sequence of transitions between generalized states

$$(q_0, \varepsilon) \overset{\alpha_1}{\vdash} (q_1, \omega_1) \overset{\alpha_2}{\vdash} (q_2, \omega_2) \overset{\alpha_3}{\vdash} \dots \overset{\alpha_n}{\vdash} (q_n, \varepsilon),$$

where $q_0 = p, q_n = q$ and $(q_i, \alpha_i, q_{i+1}) \in \delta \cup \gamma$ for $i \in [n]$, and ω_i is the sequence of opened but not yet closed parentheses after the first i steps of the run.

Proposition 4.8 *There is no parenthesizing automaton with a single pair of parentheses accepting $\hat{L}$, i.e., $\hat{L} \notin \mathsf{Reg}_1(\Sigma_2)$.*

Proof. On the contrary, suppose that there is a parenthesizing automaton $\mathcal{A} = (S, H, V, \Sigma_2, \Omega_1, \gamma, \delta, I, F)$ accepting $\hat{L}$ with $\Omega_1 = \{\langle, \rangle\}$. Let n be an even integer greater than $|V|$. The number of all biposets $P_{ww^{-1}} \in \hat{L}$, where w is of length n, is 2^n. Thus, since $2n - 2 \leq 2^n$ holds for all n, either there are n biposets accepted between horizontal states, or there are n biposets accepted between vertical states. For simplicity, we assume the former case. This is not a real restriction, since in the other case our proof would be essentially the same, only an opening and a closing parenthesizing transition needs to be added before and after the runs.

Thus, let us take n distinct words $w_j \in \Sigma_2^n$, $j \in [n]$, and consider the biposets

$$P_{w_j w_j^{-1}} = \sigma_1^j \bullet (\sigma_2^j \circ (\sigma_3^j \bullet (\sigma_4^j \circ \dots (\sigma_{n-1}^j \bullet (\sigma_n^j \circ \sigma_n^j) \bullet \sigma_{n-1}^j) \dots \circ \sigma_4^j) \bullet \sigma_3^j) \circ \sigma_2^j) \bullet \sigma_1^j,$$

where $\sigma_1^j \sigma_2^j \dots \sigma_n^j = w_j$. Now, each $P_{w_j w_j^{-1}}$ is in $\hat{L}$, by definition, but, as we shall see, $\mathcal{A}$ must accept biposets that do not belong to $\hat{L}$.

Indeed, each accepting run of $\mathcal{A}$ on any $P_{w_j w_j^{-1}}$ must have the form

$$(q_0,\varepsilon) \overset{\sigma_1^j}{\vdash} (q_1,\varepsilon) \overset{\langle}{\vdash} (q_2,\langle) \overset{\sigma_2^j}{\vdash} (q_3,\langle) \overset{\langle}{\vdash} (q_4,\langle\langle) \overset{\sigma_3^j}{\vdash} \dots \overset{\sigma_{n-1}^j}{\vdash} (q_{2n-3},\langle^{n-2}) \overset{\langle}{\vdash} (q_{2n-2},\langle^{n-1}) \overset{\sigma_n^j}{\vdash}$$
$$(q_{2n-1},\langle^{n-1}) \overset{\sigma_n^j}{\vdash} (q_{2n},\langle^{n-1}) \overset{\rangle}{\vdash} (q_{2n+1},\langle^{n-2}) \overset{\sigma_{n-1}^j}{\vdash} (q_{2n+2},\langle^{n-2}) \overset{\rangle}{\vdash} \dots \overset{\sigma_2^j}{\vdash} (q_{4n-4},\langle) \overset{\rangle}{\vdash}$$
$$(q_{4n-3},\varepsilon) \overset{\sigma_1^j}{\vdash} (q_{4n-2},\varepsilon), \text{where } q_0 \in I \cap H \text{ and } q_{4n-2} \in F \cap H.$$

For our investigation the main point is that after processing the "first half" of $P_{w_j w_j^{-1}}$, (i.e., after $2n-1$ transitions) the automaton enters a generalized state $(q_{2n-1},\langle^{n-1})$, where q_{2n-1} is vertical and after an additional $2n-1$ transitions, the run ends in (q_{4n-2},ε).

Given $P_{w_j w_j^{-1}}$, let i_j, v_j and f_j denote the states q_0, q_{2n-1}, and q_{4n-2}, respectively, which appear in the above accepting run of $\mathcal{A}$ on the biposet $P_{w_j w_j^{-1}}$. Moreover, let us abbreviate the transition sequences determined by the first $2n-1$ and the second $2n-1$ transitions by

$$(i_j,\varepsilon) \overset{P_{w_j *}}{\vdash} (v_j,\langle^{n-1}) \text{ and } (v_j,\langle^{n-1}) \overset{P_{*w_j^{-1}}}{\vdash} (f_j,\varepsilon),$$

respectively.

Now we have n vertical states $v_1, v_2, \dots, v_n$, but we have chosen $n > |V|$, so there must be two indices $k \neq l$ such that $v_k = v_l$. Hence, the transition sequence

$$(i_k,\varepsilon) \overset{P_{w_k *}}{\vdash} (v_k,\langle^{n-1}) = (v_l,\langle^{n-1}) \overset{P_{*w_l^{-1}}}{\vdash} (f_l,\varepsilon)$$

corresponds to a valid run of $\mathcal{A}$, showing that $P_{w_k w_l^{-1}}$ is accepted by $\mathcal{A}$. But $P_{w_k w_l^{-1}} \notin \hat{L}$, a contradiction. Thus, no $\mathcal{A}$ with a single pair of parentheses can accept $\hat{L}$, so $\hat{L} \notin \mathsf{Reg}_1(\Sigma_2)$. $\square$

The previous theorem can be extended to any $m \geq 1$ as follows.

THEOREM 4.9 *For all $m \geq 1$ there exists a language $\hat{L}(\Sigma_m)$ that can be accepted by an automaton with m but not with $m-1$ pairs of parentheses. Thus, the classes $\mathsf{Reg}_0 \subsetneq \mathsf{Reg}_1 \subsetneq \mathsf{Reg}_2 \subsetneq \dots$ form a strict hierarchy of regular (i.e., recognizable) sp-biposet languages.*

Proof. Our claim is trivial for $m = 1$, and $\mathsf{Reg}_1 \subsetneq \mathsf{Reg}_2$ was shown in Proposition 4.4 and Proposition 4.8. As for $m \geq 3$, we show how the proofs of these two propositions can be generalized. Let

$$\hat{L}(\Sigma_m) = \{\, P_{ww^{-1}} \mid w \in \Sigma_m^*,\ w \text{ has even length}\}.$$

Note that $\hat{L} = \hat{L}(\Sigma_2)$. In Proposition 4.4, the automaton $\mathcal{A}_2$ accepting $\hat{L}$ can easily be generalized to an automaton $\mathcal{A}_m$ accepting $\hat{L}(\Sigma_m)$, by using m different parentheses corresponding to the m letters.

In order to see that $\hat{L}(\Sigma_m) \notin \mathsf{Reg}_{\mathsf{m-1}}$, suppose on the contrary that $\hat{L}(\Sigma_m) = L(\mathcal{A})$ for some automaton $\mathcal{A} = (S, H, V, \Sigma_m, \Omega_{m-1}, \gamma, \delta, I, F)$, where $\Omega_{m-1} = \{ \langle_1, \rangle_1, \langle_2, \rangle_2, \ldots, \langle_{m-1}, \rangle_{m-1} \}$. We choose pairwise different words $w_1, w_2, \ldots, w_n$ in Σ_m^n as before, but this time we give the value of n later.

We notice that after reading the first half of $P_{w_j w_j^{-1}}$, where $P_{w_j w_j^{-1}}$ is defined as in the proof of Theorem 4.8, automaton $\mathcal{A}$ is necessarily in a generalized state

$$(v_j, \langle_{i_1} \langle_{i_2} \ldots \langle_{i_{n-1}}), \qquad (*)$$

where $v_j \in V$ and $\langle_{i_k} \in \Omega_{m-1}$ for all $k = 1, 2, \ldots, n-1$.

But the number of this type of generalized states, $|V| \cdot (m-1)^{n-1}$ is asymptotically less than m^n, the number of words w_j of length n, which is the number of all biposets of the form $P_{w_j w_j^{-1}}$. Thus, in the same way as above, n can always be chosen such that $\mathcal{A}$ accepts at least one biposet $P_{w_j w_k^{-1}}$ for some $w_j \neq w_k$. $\square$

In the proof of Theorem 4.9 we used the m-letter alphabet Σ_m to show that $\mathsf{Reg}_{\mathsf{m-1}}(\Sigma_\mathsf{m}) \subsetneq \mathsf{Reg}_\mathsf{m}(\Sigma_\mathsf{m})$. However this proper inclusion holds for every Σ.

THEOREM 4.10 *For all alphabets Σ the classes $\mathsf{Reg}_0(\Sigma) \subsetneq \mathsf{Reg}_1(\Sigma) \subsetneq \mathsf{Reg}_2(\Sigma) \ldots$ form a strict hierarchy in $\mathsf{Reg}(\Sigma)$.*

Proof. It is sufficient to prove this claim for a one-letter alphabet. So assume that $\Sigma = \Sigma_1 = \{a\}$ and suppose that $m \geq 1$. Let h denote the bisemigroup homomorphism $\mathrm{SPB}(\Sigma_m) \rightarrow \mathrm{SPB}(\Sigma_1)$ determined by the assignment

$$a_i \mapsto \underbrace{a \bullet a \bullet \ldots \bullet a}_{i \text{ times}}, \text{for all } a_i \in \Sigma_m.$$

We claim that the language $h(\hat{L}(\Sigma_m))$ is in $\mathsf{Reg}_\mathsf{m}(\Sigma_1) \setminus \mathsf{Reg}_{\mathsf{m-1}}(\Sigma_1)$.

Indeed, it is not hard to modify the automaton $\mathcal{A}_m$ in the proof of Theorem 4.9 to accept $h(\hat{L}(\Sigma_m))$ instead of $\hat{L}(\Sigma_m)$. On the other hand, after reading the "first half" of a biposet $h(P_{w_j w_j^{-1}})$, any parenthesizing automaton with $m-1$ pairs of parentheses must be in a generalized state (*) as before. Thus, the same cardinality argument can be applied to show that $h(\hat{L}(\Sigma_m))$ is not in $\mathsf{Reg}_{\mathsf{m-1}}(\Sigma_1)$. $\square$

5 Conclusions and further work

In this paper we dealt with the descriptional complexity of regular sp-biposet languages measured in terms of the least number of parentheses that an automaton needs to accept them. We have shown that with more pairs of parentheses strictly larger classes of regular sp-biposet languages can be accepted.

Recall from [3] that BRat denotes the class of birational languages which is the least class of sp-biposet languages containing the finite languages and closed under union, horizontal and vertical product and horizontal and vertical iteration. We can prove that $\mathsf{BRat} \subseteq \mathsf{Reg}_1$. The proof relies on the fact that every birational language has bounded alternation depth, i.e., for each birational sp-biposet language L there is a bound $K \geq 0$ such that every element in L has at most K pairs of nested parentheses. BD is the class of bounded alternation depth languages. Thus, if a parenthesizing automaton $\mathcal{A}$ accepts a birational language, then the successful runs of $\mathcal{A}$ must have a bounded number of opened but not yet closed parentheses at any time. So one can construct an equivalent automaton $\mathcal{A}'$ with a single pair of parentheses by storing the information about all such opened parentheses in the (inner) states. This fact together with the equation $\mathsf{BRat} = \mathsf{Reg} \cap \mathsf{BD}$ (see [3]) leads to the characterization of BRat as $\mathsf{BRat} = \mathsf{Reg}_1 \cap \mathsf{BD}$.

An open problem that seems to be difficult to solve is the decidability status of the question whether a given regular language appears in a certain level of the hierarchy. It would also be interesting to find algebraic or logical characterizations of the levels of our hierarchy.

Acknowledgement The author is indebted to Zoltán Ésik for many valuable comments and suggestions.

References

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

[2] Z. Ésik and Z. L. Németh. Automata on series-parallel biposets. In: proc. *DLT'01*, LNCS 2295, Springer, 2002, 217–227.

[3] Z. Ésik and Z. L. Németh. Higher dimensional automata, *J. of Automata, Languages and Combinatorics*, 9(2004), 3–29.

[4] J. L. Gischer. The equational theory of pomsets. *Theoret. Comput. Sci.*, 61(1988), 199–224.

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

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

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

[8] D. Kuske. Infinite series-parallel posets: logic and languages. In: proc. *ICALP 2000*, LNCS 1853, Springer, 2001, 648–662.

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

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

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

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