Rationality in algebras with a series operation [1]

K. Lodaya
Institute of Mathematical Sciences
CIT Campus
Chennai 600 113 – India.
kamal@imsc.ernet.in

P. Weil
LaBRI
Université Bordeaux 1 and CNRS
351 cours de la Libération
33405 Talence Cedex – France.
Pascal.Weil@labri.u-bordeaux.fr

Résumé. Nous considérons la notion de partie rationnelle dans des algèbres possédant une opération binaire associative particulière, appelée mise en série ou produit séquentiel. Nous définissons des automates qui opèrent dans ces algèbres, des expressions rationnelles qui ont le même pouvoir d'expression qu'eux, et nous montrons que ce pouvoir d'expression est celui de la reconnaissabilité algébrique. Le modèle que nous considérons contient à la fois le semigroupe libre et l'algèbre des termes, et la restriction de nos résultats à ces domaines coïncide exactement avec les résultats classiques (Kleene, Myhill et Nerode pour les langages de mots, et Thatcher et Wright pour les langages de termes).

Nous considérons ensuite le comportement des automates et des expressions rationnelles que nous avons introduits lorsque l'on impose des conditions comme l'associativité ou la commutativité aux opérations de formation de termes. Nous caractérisons également algébriquement, syntaxiquement et par automates les langages dans lesquels on borne le nombre d'appels imbriqués à certaines opérations. Enfin, nous considérons l'application de nos résultats à l'étude des langages d'ensembles partiellement ordonnés série-parallèle.

Abstract. We consider the notion of rationality in algebras with a designated binary associative operation called the series operation, or the sequential product. We define automata operating in these algebras and rational expressions matching their expressive power, and we show that this expressive power equals that of algebraic recognizability. The framework which we consider encompasses both the free semigroup and term algebras, and the restriction of our results to these special cases coincides exactly with the classical results on recognizability (Kleene, Myhill and Nerode for word languages, and Thatcher and Wright for term languages).

Next we consider the behavior of the automata and the rational expression which we introduce when conditions such as associativity and commutativity are imposed on the term operations. We also characterize algebraically, syntactically and automata-theoretically the languages which have a bound on the number of nested occurrences of certain designated term operations. Finally, we consider the applications of our results to the languages of series-parallel labelled posets.

Automata form one of the most commonly used computing device, from their historical stronghold of compilation to the newer field of model checking. They were initially introduced to handle languages of words, where the work of Kleene, Myhill, Nerode and Büchi depicts an ideal situation: the computational power of finite automata (the *regularity* of a language) can be characterized in an algebraic setting (*recognizability*), in a logical setting (*definability* in MSO – monadic second order logic), as well as in a syntactic way (the notion of *rational expressions*).

Early on, Thatcher and Wright followed an idea of Büchi (as cited in [19]) and constructed automata – the *bottom-up tree automata* – operating on trees, or terms in a free algebra $T_\Sigma(A)$

[1] K. Lodaya's visit to Bordeaux in 2000 and P. Weil's visit to Chennai in 2001 were supported by the Indo-French project IFCPAR/CEFIPRA 2102-1. The authors also wish to thank R. Ramanujam for his helpful comments on an earlier version of this paper.

over a signature Σ with generators A. There too, automata characterize (term) languages which can also be characterized algebraically, logically and syntactically (by *generalized rational expressions*).

However, if one considers the free semigroup A^+ as a quotient of the term algebra $T_\Sigma(A)$ where Σ consists of a single binary operation, then the images of the generalized rational term languages form the class of context-free languages, which properly contains the rational languages.

In [6, Section 5], Courcelle raises the natural question: what could be the notion of a rational subset of an arbitrary algebra? Assuming a binary associative operation in the signature (which we call the *series* operation or the *sequential* product), its Kleene closure is too weak to match recognizability, while the projection of rationality in the free term algebra is too powerful, as is exemplified by the word case.

In this paper, we propose a definition of rational expressions for such algebras which matches recognizability. We also introduce an automaton model whose computational power is the same.

The framework in which we operate unifies words and terms, in the following strong sense: word and term languages are particular cases of the languages we handle, and when restricted to either situation, our definitions and results coincide with the classical definitions of recognizability and rationality, from Kleene's theory for words and from Thatcher and Wright's approach for terms.

To be more specific, our framework is the following. Given a signature Σ, we add to Σ a new binary associative series operation, and we consider the free algebra $S_\Sigma(A)$ on an alphabet A for this extension of Σ. Thus the free semigroup on A and the free Σ-algebra on A are subsets of $S_\Sigma(A)$, which we call the free *series* Σ*-algebra* (or $s\Sigma$*-algebra*) on A. We call $s\Sigma$*-languages* the subsets of $S_\Sigma(A)$.

Our initial building blocks are the *recognizable* $s\Sigma$-languages, that is, the $s\Sigma$-languages which saturate some finite index congruence on $S_\Sigma(A)$; and the *generalized rational* $s\Sigma$-languages inherited from Thatcher and Wright. Since every recognizable $s\Sigma$-language contained in A^+ is a recognizable word language, it is immediately clear that the recognizable $s\Sigma$-languages form a proper subclass of the generalized rational ones. Next we characterize recognizable $s\Sigma$-languages by *branching automata*, and by a simple restriction on the syntax of generalized rational expressions, to yield what we call *rational expressions.*

Next we consider the status of our results when we add conditions on the Σ operations such as associativity and commutativity. (An example of this situation is given by (A-labelled) series-parallel posets [20], which form the free algebra on A over the associative sequential product, and the associative and commutative (binary) parallel product [2].) Our rational expressions still coincide with automaton acceptance, but as soon as associativity is introduced, a chasm occurs again between algebraic and automaton recognizability. However, if we only assume that some of the operations from Σ are commutative, we preserve the equivalence between the different forms of recognizability, that is, we preserve a "Kleene-Myhill-Nerode" situation.

Finally, if Υ is a designated subset of Σ, we consider bounded Υ-depth $s\Sigma$-languages, where we put a cap on the number of nested uses of a product operation from Υ. In this restricted framework, we prove again a correspondence between an algebraic, an automata-theoretic and a syntactic characterization.

In the paper we cover one important application which motivated our work, namely the theory of languages of N-free or series-parallel posets mentioned above, which play an important role in modelling concurrent systems [11]. The results in this paper allow us to characterize by means of rational expressions the computing power of an automaton model introduced by the authors [15], and to show that every recognizable series-parallel language is regular. The bounded framework mentioned above generalizes the bounded width series-parallel languages [17].

Logical aspects are of interest too, and Kuske has shown [12] that the MSO-definable bounded width series-parallel languages do match our characterization. When extending his work to more general posets, Kuske uses a D-sum term operation [13] and it is interesting to see a more general algebraic setting emerging in his work as well.

Another interesting direction of research which we have not pursued concerns graphs of bounded treewidth. These graphs admit a term representation, which is put to use to show the coincidence of logical definability and recognizability [5, 14]. A "gluing" operation can be seen to serve as a series operation in this context [7, Section 6.4]. Our treatment can be seen as a first step towards automaton models and rationality on these graphs.

The paper is organized as follows. Section 1 lays the foundation for the study of $s\Sigma$-languages, and reviews the definitions of recognizability and generalized rationality. Section 2 is devoted to branching automata, and the equivalence between their accepting power and algebraic recognizability is proved in Section 3. Rationality is introduced in Section 4, where it is proved to be equivalent to recognizability. Recognizability and rationality in axiomatized $s\Sigma$-algebras are investigated in Section 5, and $s\Sigma$-languages of bounded depth are characterized in Section 6.

1 Series Σ-algebras and languages

Throughout this paper, we let Σ be a finite signature: Σ is a finite set of operations, and we let Σ_n be the set of operations in Σ with arity n. We assume for convenience that there are no constants, and we use an alphabet of variables (or generators). If A is an *alphabet*, that is, a finite non-empty set, we denote by $T_\Sigma(A)$ the free Σ-algebra over the alphabet A. Our work can be readily adapted in the presence of constants: they are to be treated everywhere like the elements of A (except naturally in the definition of morphisms between Σ-algebras).

A *series Σ-algebra* is a Σ-algebra equipped with an extra binary, associative operation $\circ$, written multiplicatively, and called the *sequential product*. If A is an alphabet, we denote by $S_\Sigma(A)$ the free series Σ-algebra over the alphabet A. The free Σ-algebra $T_\Sigma(A)$ and the free semigroup A^+ (under the sequential product) are both subsets of $S_\Sigma(A)$.

Let $\Sigma_{seq} = \Sigma \cup \{\circ\}$. Then $S_\Sigma(A)$ is a relatively free quotient of $T_{\Sigma_{seq}}(A)$ by the identity which makes $\circ$ associative.

Büchi first noticed the correspondence between the free monoid A^* and the free algebra $T_{U(A)}(\{1\})$. Here $U(A)$ is the signature with every letter of A as a unary operation. 1 is a constant which generates the algebra. We use a different correspondence: the free semigroup A^+ is exactly $S_\emptyset(A)$.

Let $x \in S_\Sigma(A)$. We say that x is a *sequential element* if it can be written as a sequential product $x = yz$, and a *Σ-element* otherwise. More precisely, we say that x is a *t-element*, or a *t-product*, or a *t-operation* ($t \in \Sigma$) if x is of the form $x = t(x_1, \ldots, x_n)$. Since $S_\Sigma(A)$ is freely generated by A, its elements can be written as expressions using only the letters of A, the sequential product and the operations of Σ: as a semigroup under the sequential product, $S_\Sigma(A)$ is freely generated by the Σ-elements, and as a Σ-algebra, it is freely generated by the

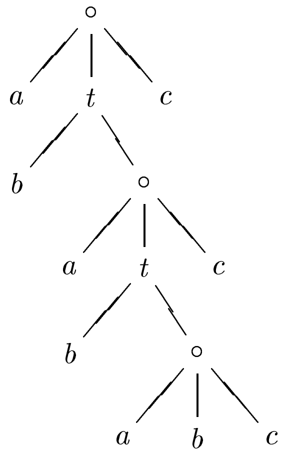


Figure 1: Tree representation of an element of $S_{\{t\}}(\{a,b,c\})$

letters and the sequential elements.

So if $x \in S_\Sigma(A)$ is a term, it admits an iterated factorization of sequential and Σ-factors, which can be conveniently represented by a tree whose leaves are labelled by letters of A and whose interior nodes are labelled either by $\circ$ or a letter of Σ. In addition, no sequential (resp. Σ) node is an immediate successor of a sequential (resp. Σ) node, and nodes labelled by a letter of Σ_n have exactly n immediate successors. Figure 1 shows the representation of the term $a\ t(b, a\ t(b, abc)\ c)\ c$.

In terms of this tree representation, the canonical morphism from $T_{\Sigma_{seq}}(A)$ to $S_\Sigma(A)$ flattens a hierarchy of $\circ$ nodes into successors of a single $\circ$ node.

The following notions will be useful in the sequel: we say that *an element z occurs in an element* $x \in S_\Sigma(A)$ if, either $x = z$, or x is of the form $x = y_1y_2$ or $x = t(y_1, \ldots, y_n)$ (with $t \in \Sigma_n$) and z occurs in one of the y_i.

The *length* of an element x of $S_\Sigma(A)$ is defined inductively as follows: if $x \in A$, then $|x| = 1$. If $x = yz$, then $|x| = |y| + |z|$, and if $x = t(x_1, \ldots, x_n)$ $(t \in \Sigma_n)$, then $|x| = 1 + |x_1| + \cdots + |x_n|$.

Let Υ be a non-empty subset of the signature Σ. The *Υ-depth* of an element $x \in S_\Sigma(A)$, $\mathrm{dp}_\Upsilon(x)$, is defined to be the maximum number of nodes labelled by an element of Υ along a path from a leaf to the root in the tree representation of x. For instance, the term shown in Figure 1 has $\{t\}$-depth two. In Section 6, we will be interested in *bounded Υ-depth* languages, subsets of $S_\Sigma(A)$ which have a uniform bound on the Υ-depth of all their elements. One verifies that the function dp_Υ can also be defined inductively by the following formulæ.

$$\begin{aligned}
\mathrm{dp}_\Upsilon(a) &= 0 \\
\mathrm{dp}_\Upsilon(xy) &= \max\{\mathrm{dp}_\Upsilon(x), \mathrm{dp}_\Upsilon(y)\} \\
\mathrm{dp}_\Upsilon(t(x_1, ..., x_n)) &= \begin{cases} 1 + \max\{\mathrm{dp}_\Upsilon(x_1), ..., \mathrm{dp}_\Upsilon(x_n)\} & \text{if } t \in \Upsilon \\ \max\{\mathrm{dp}_\Upsilon(x_1), ..., \mathrm{dp}_\Upsilon(x_n)\} & \text{if } t \notin \Upsilon \end{cases}
\end{aligned}$$

Finally, if ξ is a letter not in A, we say that an element $x \in S_\Sigma(A \cup \{\xi\})$ is a *ξ-term* if it has exactly one occurrence of ξ, and it is *a ξ-term of Υ-depth k* if there are k nodes labelled by an element of Υ on the path from the leaf labelled ξ to the root in the tree representation of x. If x

is a ξ-term of non-zero Υ-depth, let $f \in \Upsilon_n$ be the label of the Υ-labelled node between the root and ξ, which is closest to ξ. Let $c_1, \ldots, c_n$ be the elements of $S_\Sigma(A \cup \{\xi\})$ represented by the subtrees hanging under this node: then $f(c_1, \ldots, c_n)$ occurs in x, and we say that $f(c_1, \ldots, c_n)$ is the Υ-*context* of ξ in x. For instance, the $\{t\}$-context of ξ in $a\ t(b, a\xi c)\ c$ is $t(b, a\xi c)$.

1.1 Recognizability and generalized rationality

Subsets of $S_\Sigma(A)$ are called *series* Σ*-languages*, or $s\Sigma$*-languages*. Elementary properties of $s\Sigma$-languages were considered by Courcelle [4, Section 4].

A language L is said to be *recognizable* if there exists a morphism φ from $S_\Sigma(A)$ into a finite $s\Sigma$-algebra S such that $L = \varphi^{-1}\varphi(L)$. In that case, we say that L is *recognized* by the morphism φ, and by the $s\Sigma$-algebra S. Equivalently, L is recognizable if it is a union of classes of a finite-index congruence on $S_\Sigma(A)$.

Another important notion is that of *rationality.* Our treatment follows Thatcher and Wright [19] (although they call it regularity; see also Gécseg and Steinby [10] and Büchi [3]).

Let K, L, L_1, ..., L_n be $s\Sigma$-languages, $t \in \Sigma_n$ and $\xi \in A$. We define the following operations on $s\Sigma$-languages, called respectively *sequential product*, *sequential iteration*, t*-operation* and ξ*-substitution.*

$$\begin{aligned}
KL &= \{xy \mid x \in K,\ y \in L\}, \\
& L^+ \text{ is the subsemigroup generated by } L \text{ under the sequential product.} \\
t(L_1, \ldots, L_n) &= \{t(x_1, \ldots, x_n) \mid x_1 \in L_1,\ \ldots,\ x_n \in L_n\}, \\
K \circ_\xi L & \text{ is the set of elements obtained from the elements of } L \text{ by replacing} \\
& \qquad (\textit{non-uniformly}) \text{ each occurrence of } \xi \text{ by an element of } K,
\end{aligned}$$

Repeated ξ-substitution on a $s\Sigma$-language L gives the operation of ξ*-exponentiation.* Since ξ-substitution is associative, the order in which substitutions are done does not matter.

$$L^{*\xi} = \bigcup_{i \geq 0} L_i^\xi, \text{ with } L_0^\xi = \{\xi\} \text{ and } L_{i+1}^\xi = L_i^\xi \circ_\xi L \cup L_i^\xi.$$

For example, the language $\{abc\} \cup \Big(b \circ_\xi (a\ t(b, a\xi c)^{*\xi} c)\Big)$ has elements like that in Figure 1, of every $\{t\}$-depth.

Remark. Our notation follows Gécseg and Steinby [10]. The language $K \circ_\xi L$ would be $L \cdot_\xi K$ in Thatcher and Wright's notation. What Büchi as well as Thatcher and Wright denote L^ξ does not necessarily contain ξ. It is what we might call $L^{+\xi} = L \circ_\xi L^{*\xi}$. Note that $L^+ = L \circ_\xi (\xi L)^{*\xi}$.

The *generalized rational expressions* over A (relative to Σ) can now be defined inductively.

(1) Every letter $a \in A$ is a generalized rational expression;

(2) if $t \in \Sigma_n$ and if $e_1, \ldots, e_n$ are generalized rational expressions, then so are $e_1 e_2$, $e_1 + e_2$ and $t(e_1, \ldots, e_n)$;

(3) if e_1 and e_2 are generalized rational expressions, and if $\xi \in A$, then $e_1 \circ_\xi e_2$ is a generalized rational expression;

(4) if e is a generalized rational expression, and if $\xi \in A$, then $e^{*\xi}$ is a generalized rational expression.

Next we associate with each generalized rational expression e a subset $L(e)$ of $S_\Sigma(A)$. The $s\Sigma$-languages so defined are said to be *generalized rational.*

(1) $L(a) = \{a\}$ for $a \in A$;
(2) $L(e_1e_2) = L(e_1)L(e_2)$, $L(e_1+e_2) = L(e_1) \cup L(e_2)$ and $L(t(e_1, \ldots, e_n)) = t(L(e_1), \ldots, L(e_n))$;
(3) $L(e_1 \circ_\xi e_2) = L(e_1) \circ_\xi L(e_2)$ and $L(e^{*\xi}) = L(e)^{*\xi}$.

We also introduce *multi-substitutions* and *equational* $s\Sigma$-languages: if $V = \{\xi_1, \ldots, \xi_n\}$ is a set of variables not in A and if $M \in S_\Sigma(A \cup V)$, we say that M is a *monomial in* V and we write $M(L_1, \ldots, L_n)$ for $L_1 \circ_{\xi_1} (\ldots (L_n \circ_{\xi_n} \{M\}) \ldots)$. A formal sum of monomials in V, say, $P = M_1 + \cdots + M_m$ is called a *polynomial in* V and we let $P(L_1, \ldots, L_n) = M_1(L_1, \ldots, L_n) \cup \ldots \cup M_m(L_1, \ldots, L_n)$.

A solution of a *polynomial system* $\mathcal{E} = \langle \xi_1 = P_1, \ldots, \xi_n = P_n \rangle$ is an n-tuple $(L_1, \ldots, L_n)$ of $s\Sigma$-languages such that $L_i = P_i(L_1, \ldots, L_n)$ for $i = 1, \ldots, n$. Such a system always has a least solution with respect to inclusion. An $s\Sigma$-language is said to be *equational* if it is a union of components of the least solution of a polynomial system. Equational sets were introduced by Mezei and Wright [18] (see also [4]). The notions of polynomial system and equational subset can be expressed in terms of context-free or algebraic grammars as well, see for instance [1].

1.2 Earlier work

Alternately, we could have defined generalized rational expressions and polynomial systems to describe subsets of the term algebra $T_{\Sigma_{seq}}(A)$. This is what is done for instance in [10, 18, 19, 3]. Then we would say that an $s\Sigma$-language L is generalized rational if it is the image of a generalized rational subset of $T_{\Sigma_{seq}}(A)$ in the canonical morphism onto $S_\Sigma(A)$.

The fundamental results on recognizability and generalized rationality in term algebras are summarized in the following statement [18, 19, 4].

Theorem 1.1 *Let M be an A-generated Σ-algebra and let $\pi: T_\Sigma(A) \to M$ be an onto morphism.*

(1) *A subset of $T_\Sigma(A)$ is recognizable if and only if it is generalized rational, if and only if it is equational.*
(2) *A subset L of M is recognizable if and only if $\pi^{-1}(L)$ is recognizable. In particular, every recognizable subset of M is generalized rational.*
(3) *The recognizable subsets of M form a Boolean algebra, and if $\varphi: M \to M'$ is a morphism, the inverse image by φ of a recognizable subset of M' is recognizable.*

Example 1.2 If Σ consists of a single, binary operation, then $T_\Sigma(A)$ is the set of finite binary trees, whose leaves are labelled with elements of A. The free semigroup A^+ is a quotient of $T_\Sigma(A)$, and it is known that the generalized rational subsets of A^+ are the context-free languages, a class which properly contains the recognizable languages (see for instance [1]). For instance, $(a\xi c)^{*\xi} = \{a^n\xi c^n \mid n \geq 0\}$ is generalized rational but not recognizable.

Applying Theorem 1.1 to $T_{\Sigma_{seq}}(A)$, we get the following results for its quotient $S_\Sigma(A)$.

Corollary 1.3 *Let $L \subseteq S_\Sigma(A)$ be an $s\Sigma$-language.*

(1) *If L is recognizable, then it is generalized rational. There are generalized rational $s\Sigma$-languages which are not recognizable.*
(2) *L is recognizable if and only if its inverse image in $T_{\Sigma_{seq}}(A)$ is recognizable.*
(3) *The recognizable $s\Sigma$-languages form a Boolean algebra, and are closed under inverse morphism.*
(4) *L is equational if and only if it is generalized rational.*

Remark. One can adapt the work in [18] to give a sufficient condition for a polynomial system to define recognizable $s\Sigma$-languages: every polynomial system is equivalent to an effectively computable system in a normalized form, called a *reduced deterministic* system, in such a way that the set of variables of this system is naturally equipped with a structure of Σ_{seq}-algebra recognizing the components of the minimal solution of $\mathcal{E}$. If this Σ_{seq}-algebra is an $s\Sigma$-algebra, that is, if the designated operation $\circ$ is associative, then these $s\Sigma$-languages are recognizable.

We cannot expect to have a decidable necessary and sufficient condition for recognizability, as the situation we are considering includes the case of binary tree languages and of context-free languages. If we were able to decide whether the components of the least solution of a given polynomial system are recognizable, then we would be able to decide whether a given context-free language is recognizable, and this problem is well known to be undecidable.

As we will see later in this paper (Corollary 4.8), the recognizable $s\Sigma$-languages are also closed under sequential product and iteration, Σ-products and ξ-substitution. It is only ξ-exponentiation which may take us outside recognizability (Example 1.2).

We will explore in this paper the question of defining a "rational" ξ-iteration. The first step in this direction is to define automata which work on terms of $S_\Sigma(A)$. This is done in the next section, after which we prove a Myhill-Nerode theorem. Then we define rational expressions and prove a Kleene theorem.

2 Branching automata and regular $s\Sigma$-languages

A *branching automaton* over the alphabet A consists of a finite set Q of *states*, designated subsets I and F of Q, respectively the sets of *initial* and *final* states, and sets of transitions.

Transitions are of several kind: there are the sequential transitions, the opening t-transitions and the closing t-transitions, for each $t \in \Sigma$. More precisely, the set of *sequential transitions* is a subset T_{seq} of $Q \times A \times Q$. An element $(p, a, q) \in T_{seq}$, usually written $p \xrightarrow{a} q$, is said to be an *a-labelled transition* from p to q. If $t \in \Sigma_n$ $(n \geq 2)$, the set of *opening t-transitions* is a subset $T_{t,0}$ of $Q \times Q^n$, whose elements are written in the form $p \to (p_1, \ldots, p_n)$. And the set of *closing t-transitions* is a subset $T_{t,1}$ of $Q^n \times Q$, whose elements are written in the form $(q_1, \ldots, q_n) \to q$. If $t \in \Sigma_1$, the set of opening (resp. closing) t-transitions is a subset of $Q \times Q^2$ (resp. $Q^2 \times Q$). The separate treatment of unary Σ-operations is for technical convenience.

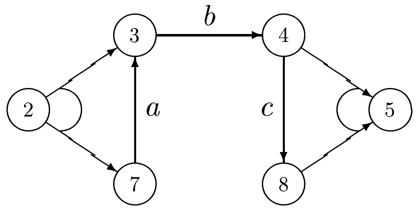
Let p, q be states of a branching automaton $\mathcal{A}$ and let $x \in S_\Sigma(A)$. The existence of a *run of $\mathcal{A}$ on x from p to q* is inductively defined as follows: there is a run on letter a from p to q if $p \xrightarrow{a} q$ is a sequential transition. If x can be written as a sequential product $x = yz$, there is a run on x from p to q if there exists a state r and there are runs on y from p to r and on z from r to q.

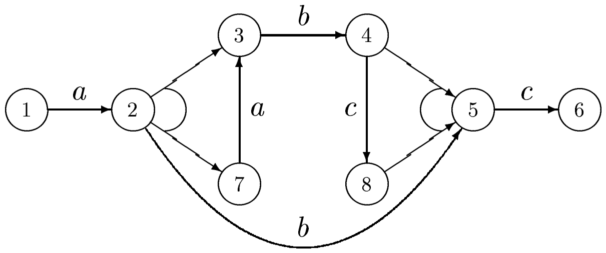
Suppose now that x is a Σ-element: then x can be written in a unique fashion in the form $x = t(x_1, \ldots, x_n)$, for some $n \geq 1$, $t \in \Sigma_n$ and $x_1, \ldots, x_n \in S_\Sigma(A)$. Let us first assume that $n \geq 2$: then there is a run on x from p to q if there is an opening t-transition $p \to (p_1, \ldots, p_n)$ and a closing transition $(q_1, \ldots, q_n) \to q$ such that $\mathcal{A}$ has runs on x_i from p_i to q_i for $i = 1, \ldots, n$. For unary t, there is a run on $x = t(x_1)$ from p to q if there is an opening transition $p \to (p_1, r)$ and a closing transition $(q_1, r) \to q$ such that $\mathcal{A}$ has a run on x_1 from p_1 to q_1. The state r serves as a dummy "waiting" state.

Finally, a run of $\mathcal{A}$ is *successful* if it starts in an initial state and it ends in a final state. The *$s\Sigma$-language accepted* by $\mathcal{A}$ is the set $L(\mathcal{A})$ of labels of successful runs. An $s\Sigma$-language is said to be *regular* if there is a branching automaton $\mathcal{A}$ such that $L = L(\mathcal{A})$. In that case, we say that *L is accepted by $\mathcal{A}$*.

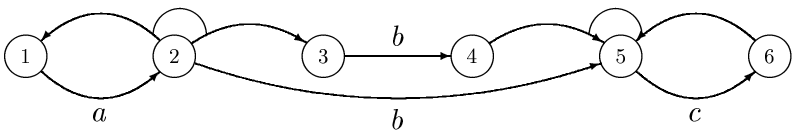
Example 2.1 Let $\Sigma = \{t\}$ consist of a single binary operation. In the following automata the opening t-transition is $2 \to (3, 7)$ and the closing t-transition is $(4, 8) \to 5$.

Letting 2 be the initial state and 5 be the final state, the first automaton accepts the singleton $\{t(b, abc)\}$. Letting 1 be the initial state and 6 be the final state, the second automaton accepts $\{abc, a\, t(b, abc)\, c\}$.





The next automaton, with opening t-transition $2 \to (3, 1)$, closing t-transition $(4, 6) \to 5$, initial state 1 and final state 6, accepts the language $\{abc\} \cup \Big(b \circ_\xi (a\ t(b, a\xi c)^{*\xi} c)\Big)$.



Remark. One can show directly that the regular $s\Sigma$-languages are generalized rational by the following remark. Let $\mathcal{A}$ be a branching automaton with state set Q. Let $V = \{\xi_{p,q} \mid p, q \in Q\}$ be a set in bijection with $Q \times Q$. For each $p, q \in Q$, we let $G_{p,q}$ be the set of monomials of the form $a \in A$ if (p, a, q) is a sequential transition of $\mathcal{A}$; $\xi_{p,r}\xi_{r,q}$ if $r \in Q$; and $t(\xi_{p_1,q_1}, \ldots, \xi_{p_s,q_s})$ if $t \in \Sigma$, $p \to (p_1, \ldots, p_s)$ is an opening t-transition and $(q_1, \ldots, q_s) \to q$ is a closing t-transition. Finally, we consider the polynomial system $\mathcal{E}$ given by

$$\xi_{p,q} = \sum_{g \in G_{p,q}} g \qquad (p, q \in Q).$$

Then the components of the least solution of $\mathcal{E}$ are the $s\Sigma$-languages $L_{p,q}$ such that $L_{p,q}$ is the set of labels of the runs of $\mathcal{A}$ from p to q. In particular, the language accepted by $\mathcal{A}$ is a union of components of this least solution, and hence it is equational.

2.1 A technical lemma

We will need the following lemma in later constructions (Section 4.1). It states that every regular $s\Sigma$-language is accepted by a branching automaton with certain "good" properties.

Lemma 2.2 *Let L be a regular $s\Sigma$-language. Then L is accepted by a branching automaton $\mathcal{A}$ satisfying the following properties:*

(1) *$\mathcal{A}$ has a unique initial state i, and i occurs only as the origin of sequential and of opening t-transitions of $\mathcal{A}$;*

(2) *$\mathcal{A}$ has a unique final state f, and f occurs only as the end of sequential and of closing t-transitions of $\mathcal{A}$;*

(3) *if $p \to (p_1, \ldots, p_n)$ is an opening t-transition of $\mathcal{A}$, there is no run of $\mathcal{A}$ from any of the p_i to f;*

(4) *if $(q_1, \ldots, q_n) \to q$ is a closing t-transition of $\mathcal{A}$, there is no run of $\mathcal{A}$ from i to any of the q_i;*

(5) *if p is a state of $\mathcal{A}$, there is a run of $\mathcal{A}$ from i to p if and only if there is a run of $\mathcal{A}$ from p to f.*

Proof. Let $\mathcal{B}$ be an automaton accepting L. We construct a new automaton $\mathcal{B}'$ as follows. Firstly $\mathcal{B}'$ contains all the states and transitions of $\mathcal{B}$. Secondly we add two new states i and f to $\mathcal{B}'$, and we add the following transitions: for each sequential or opening t-transition originating in an initial state of $\mathcal{B}$, we add a similar transition, with the same end state(s), originating from i; dually, for each sequential or closing t-transition ending in a final state of $\mathcal{B}$, we add a similar transition, with the same start state(s), ending at f. Finally, we make i the initial state and f the final state of $\mathcal{B}'$. It is immediately verified that $\mathcal{B}'$ accepts L as well, and that it satisfies Conditions (1) and (2). Thus we may now assume that $\mathcal{B}$ satisfies Conditions (1) and (2).

Next we construct an automaton $\mathcal{A}'$ as follows. First we take the disjoint union of copies of $\mathcal{B}$, denoted respectively $\mathcal{B}_0$ and $\mathcal{B}_{k,m}$ where k is an opening t-transition for some $t \in \Sigma$ and $m = 1, 2$. Next we delete from $\mathcal{B}_0$ all the opening and closing t-transitions, for all $t \in \Sigma$. For each opening t-transition $k = p \to (p_1, \ldots, p_n)$ of $\mathcal{B}$, we add to $\mathcal{A}'$ an opening t-transition $p \to (p_1, \ldots, p_n)$ where p is taken in $\mathcal{B}_0$, $p_1, \ldots, p_{n-1}$ are taken in $\mathcal{B}_{k,1}$ and p_n is taken in $\mathcal{B}_{k,2}$. Moreover, for each opening t-transition k and for each closing t-transition $j = (q_1, \ldots, q_n) \to q$ of $\mathcal{B}$, we add to $\mathcal{A}'$ a closing t-ransition $j_k = (q_1, \ldots, q_n) \to q$ simulating j: q is taken in $\mathcal{B}_0$, $q_1, \ldots, q_{n-1}$ are taken in $\mathcal{B}_{k,1}$ and q_n is taken in $\mathcal{B}_{k,2}$. Finally, the initial and final states of $\mathcal{A}'$ are the same as in $\mathcal{B}$, but taken in $\mathcal{B}_0$.

The only way into (resp. out of) one of the $\mathcal{B}_{k,m}$ is through one of the opening (resp. closing) t-transitions which we added, not all of whose branches end (resp. originate) in $\mathcal{B}_{k,m}$. Thus any run of $\mathcal{A}'$ that starts or ends in one of the $\mathcal{B}_{k,m}$, takes place entirely within $\mathcal{B}_{k,m}$. In particular, $\mathcal{A}'$ satisfies Conditions (3) and (4). In addition, any run of $\mathcal{A}'$ which starts (resp. ends) in $\mathcal{B}_0$, also ends (resp. starts) in $\mathcal{B}_0$.

Let us now verify that $\mathcal{A}'$ accepts L. We prove by induction on $|x|$ that $\mathcal{B}$ has a run on x from p to q, if and only if $\mathcal{A}'$ has a run on x from p to q, with p and q taken in $\mathcal{B}_0$.

This holds if $x \in A$ since $\mathcal{B}_0$ has the same sequential transitions as $\mathcal{B}$, and it is a routine verification if x is a sequential element, that is, $x = yz$ for some $y, z \in S_\Sigma(A)$.

Now suppose that x is a t-element, for some $t \in \Sigma$: then $w = t(x_1, \ldots, x_n)$ and $|x_j| < |x|$ for each i. We first assume that $n \geq 2$. If there is a run of $\mathcal{B}$ on x from p to q, then there exists an opening t-transition $k = (p \to (p_1, \ldots, p_n))$, a closing t-transition $(q_1, \ldots, q_n) \to q$, and runs on x_j from p_j to q_j for $j = 1, \ldots, n$. Now $\mathcal{A}'$ has an opening t-transition $p \to (p_1, \ldots, p_n)$ with p in $\mathcal{B}_0$, $p_1, \ldots, p_{n-1}$ in $\mathcal{B}_{k,1}$ and p_n in $\mathcal{B}_{k,2}$, and a closing t-transition $(q_1, \ldots, q_n) \to q$ with q in $\mathcal{B}_0$, $q_1, \ldots, q_{n-1}$ in $\mathcal{B}_{k,1}$ and q_n in $\mathcal{B}_{k,2}$. But $\mathcal{B}_{k,1}$ and $\mathcal{B}_{k,2}$ are copies of $\mathcal{B}$, so there are runs of $\mathcal{B}_{k,1}$ on x_j from p_j to q_j for $j = 1, \ldots, n-1$, and there is a run of $\mathcal{B}_{k,2}$ on x_n from p_n to q_n, so finally there is a run of $\mathcal{A}'$ on $x = t(x_1, \ldots, x_n)$ from p to q, with those two states taken in $\mathcal{B}_0$. Again, the converse implication is verified similarly. The case of a unary operation $t \in \Sigma_1$ is handled in the same fashion. This concludes the verification that $\mathcal{A}'$ accepts L.

We already observed that $\mathcal{A}'$ does not have any run from a state in $\mathcal{B}_0$ to a state in another component, nor does it have a run from a state outside $\mathcal{B}_0$ to one inside it. In particular, if there is a run on $x \in S_\Sigma(A)$ from i to some state p, or from p to f, then p lies in $\mathcal{B}_0$. The automaton $\mathcal{A}$ is obtained from $\mathcal{A}'$ by removing those states of $\mathcal{B}_0$ which are not both the origin of a run of $\mathcal{A}'$ to f and the end of a run of $\mathcal{A}'$ from i. Our observation implies that $\mathcal{A}$ also accepts L, and $\mathcal{A}$ now satisfies all 5 required conditions. $\square$

2.2 Deciding emptiness

In this section, we prove the following result.

Proposition 2.3 *Given a branching automaton $\mathcal{A}$, it is decidable whether $L(\mathcal{A})$ is empty or not. This decision can be reached in time polynomial in the description of $\mathcal{A}$.*

Proof. Let Q be the set of states of $\mathcal{A}$, and let P be the set of pairs $(p, q) \in Q \times Q$ such that there is a run of $\mathcal{A}$ from p to q. Clearly, it suffices to decide whether P contains a pair (i, f), with i an initial state and f a final state.

We compute P as follows. If R is a subset of $Q \times Q$, we denote by R^* the reflexive transitive closure of R (seen as a relation on Q).

Let S be the relation induced by the sequential transitions of $\mathcal{A}$, that is, the set of all (p, q) such that (p, a, q) is a sequential transition of $\mathcal{A}$, for some letter a. Let $P_0 = S^*$.

If P_k is constructed, let R_k be the set of all pairs (p, q) such that there is an operation $t \in \Sigma$, an opening t-transition $p \to (p_1, \ldots, p_n)$ and a closing t-transition $(q_1, \ldots, q_n) \to q$ for which all the pairs (p_i, q_i) are in P_k. Let $P_{k+1} = (P_k \cup R_k)^*$.

One verifies immediately, by induction on k, that P_k is exactly the set of pairs of states (p, q) such that there is a run from p to q on an element x of Σ-depth at most k. The definition of the P_k makes it clear that, if $P_k = P_{k+1}$, then $P_k = P_m$ for each $m > k$. So in that case, $P_k = P$, and hence P is computable. $\square$

2.3 Pumping lemmas

Pumping lemmas for regular $s\Sigma$-languages can be derived from the consideration of branching automata.

Proposition 2.4 *Let L be a recognizable subset of $S_\Sigma(A)$. There exists an integer N such that:*

(1) *if $x \in L$ is a sequential product of at least N factors, then x can be written as $x = uvw$, with $u, v, w \in S_\Sigma(A)$, in such a way that $uv^+w \subseteq L$;*

(2) *if $x \in L$ has Σ-depth at least N, then x can be written in the form $x = u \circ_\xi v \circ_\xi w$, with ξ a new letter not in A, $u \in S_\Sigma(A)$, $v, w \in S_\Sigma(A \cup \{\xi\})$, each with a single occurrence of ξ, and $u \circ_\xi v^{*\xi} \circ_\xi w \subseteq L$.*

Proof. Let $\mathcal{A}$ be a branching automaton accepting L, and let Q be the set of states of $\mathcal{A}$. The first statement is proved by the standard technique (standard for word languages), with $N = |Q|$.

For the second statement, we may take $N = |Q|^2$. (In fact it would suffice to take $N = |P|$ for the set P in the proof of Proposition 2.3.) If $x \in S_\Sigma(A)$ has Σ-depth n, then x can be written as $x = x_n \circ_\xi x_{n-1} \circ_\xi \cdots \circ_\xi x_0$, where ξ is a new letter not in A and $x_0, \ldots, x_{n-1}$ are ξ-terms of Σ-depth 1. If $x \in L$, there is a pair (p_0, q_0) of an initial and a final state such that x labels a run ρ_0 of $\mathcal{A}$ from p_0 to q_0. But $x = (x_n \circ_\xi x_{n-1} \circ_\xi \cdots \circ_\xi x_1) \circ_\xi x_0$, so the run ρ_0 contains a sub-run ρ_1, on $x_n \circ_\xi x_{n-1} \circ_\xi \cdots \circ_\xi x_1$, from a state p_1 to a state q_1. Iterating this analysis, we define a sequence of subruns $(\rho_i)_{0 \le i \le n}$, such that ρ_i is a run on $x_n \circ_\xi x_{n-1} \circ_\xi \cdots \circ_\xi x_i$, from state p_i to state q_i. If $n \ge N$, then there exist $0 \le i < j \le n$ such that $p_i = p_j$ and $q_i = q_j$. Now letting $u = x_n \circ_\xi \cdots \circ_\xi x_j$, $v = x_{j-1} \circ_\xi \cdots \circ_\xi x_i$ and $w = x_{i-1} \circ_\xi \cdots \circ_\xi x_0$, we verify that every element of $u \circ_\xi v^{*\xi} \circ_\xi w$ labels a run of $\mathcal{A}$ from p_0 to q_0, which concludes the proof. □

3 Regularity vs. recognizability for $s\Sigma$-languages

In this section, we prove a Myhill-Nerode-like theorem.

Theorem 3.1 *Let L be an $s\Sigma$-language. Then L is recognizable if and only if L is regular.*

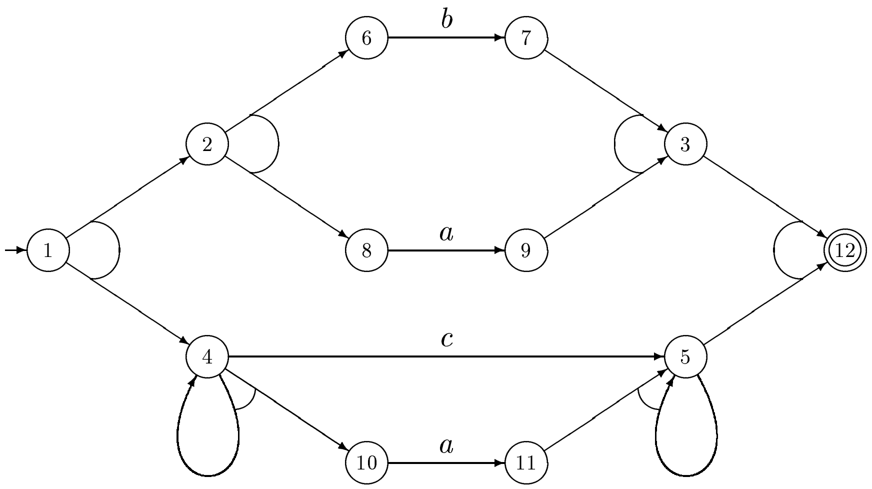
The two directions of the statement are proved in Propositions 3.2 and 3.5 below.

Proposition 3.2 *Every regular $s\Sigma$-language is recognizable.*

Proof. Let L be a regular $s\Sigma$-language, and let $\mathcal{A}$ be a branching automaton accepting L. Let $x, y \in S_\Sigma(A)$. We let $x \sim y$ if for all states p, q of $\mathcal{A}$, there is a run of $\mathcal{A}$ on x from p to q if and only if there is a run of $\mathcal{A}$ on y from p to q. This is trivially an equivalence relation on $S_\Sigma(A)$, and it has finite index, at most $2^{|Q|^2}$, where Q is the state set of $\mathcal{A}$.

It is a routine verification that $\sim$ is a congruence on $S_\Sigma(A)$. Now let S be the quotient $s\Sigma$-algebra, $S = S_\Sigma(A)/\sim$, and $\mu: S_\Sigma(A) \to S$ be the natural projection. By definition of a successful run, if $\mu(x) = \mu(y)$ and x is accepted by $\mathcal{A}$, then so is y. This means exactly that L is recognized by the morphism μ, and hence L is recognizable. □

Example 3.3 Let $A = \{a, b, c\}$, let $\Sigma = \{\|\}$ consist of a single, binary operation, and consider the following branching automaton, $\mathcal{A}$:



where the opening $\|$-transitions are $1 \to (2,4)$, $2 \to (6,8)$ and $4 \to (10,4)$, and the closing $\|$-transitions are $(7,9) \to 3$, $(11,5) \to 5$ and $(3,5) \to 12$. We have $c \sim a\|c \sim a\|(a\|c) \sim \ldots$, since $c, a\|c, a\|(a\|c), \ldots$, label runs from 4 to 5.

In order to prove the converse statement, namely that every recognizable $s\Sigma$-language is regular, we associate a branching automaton with each morphism into a finite $s\Sigma$-algebra.

Let S be a finite $s\Sigma$-algebra, and let $\varphi: S_\Sigma(A) \to S$ be a morphism. We construct a branching automaton $\mathcal{A}(S)$ as follows. The state set of $\mathcal{A}(S)$ is the union of $2 + |S|$ copies of $S \cup \{1\}$, denoted S_0 and S_s ($s \in S \cup \{1\}$), which are called the components of $\mathcal{A}(S)$. Here, 1 is a new element, not in S, which acts as a unit for the sequential product (other term building operations involving 1 are not defined). State 1 in component S_0 is chosen to be the initial state. The choice of a set of final states is postponed.

The sequential transitions of $\mathcal{A}(S)$ all take place within the components, and all the components have the same sequential transitions, namely the triples $(x, a, x\varphi(a))$, for all $x \in S \cup \{1\}$ and $a \in A$.

Let $t \in \Sigma_n$ with $n \geq 2$. The opening t-transitions of $\mathcal{A}(S)$ are the tuples of the form $p \to (p_1, \ldots, p_n)$, where $p \in S \cup \{1\}$ is taken in an arbitrary component T of $\mathcal{A}(S)$, $p_1 = p_2 = \ldots = p_{n-1} = 1$ are taken in the same component T, and $p_n = 1$ is taken in S_p. The closing t-transitions of $\mathcal{A}(S)$ are the tuples of the form $(q_1, \ldots, q_n) \to q$, where $q_1, \ldots, q_{n-1}$ and q are taken in a component T of $\mathcal{A}(S)$, q_n is taken in a component S_p, and $p\ t(q_1, \ldots, q_n) = q$ in S.

Finally, let us consider a unary operation $t \in \Sigma_1$: the opening t-transitions of $\mathcal{A}(S)$ are the tuples of the form $p \to (p_1, p_2)$, where p and p_1 are taken in a component T, p_2 is taken in S_p, and $p_1 = p_2 = 1$. The closing t-transitions are the tuples of the form $(q_1, q_2) \to q$, where q_1 and q are taken in a component T, q_2 is taken in a component of the form S_p, and we have $q_2 = 1$ and $q = p\ t(q_1)$ in S.

The automaton $\mathcal{A}(S)$ satisfies the following properties.

Lemma 3.4 *Let $x \in S_\Sigma(A)$. Then $\mathcal{A}(S)$ has a run on x from p to q if and only if the two following conditions hold:*

(1) *p and q are in the same component of $\mathcal{A}(S)$;*
(2) *$q = p\varphi(x)$ in S.*

Proof. The proof is by induction on $|x|$. If $x = a \in A$, the statement is trivial, as all the sequential transitions of $\mathcal{A}(S)$ take place within a component, and represent the right translations of S under the sequential product.

We now assume that the lemma holds for $y, z \in S_\Sigma(A)$, and we consider a run of $\mathcal{A}(S)$ on $x = yz$ from p to q. Then there exists a state r of $\mathcal{A}(S)$ such that there is a run on y from p to r, and a run on z from r to q. By induction, it follows that p, r and q are in the same component of $\mathcal{A}(S)$, and $q = r\varphi(z) = p\varphi(y)\varphi(z) = p\varphi(x)$.

Conversely, if p and q are in the same component and $q = p\varphi(x)$ in S, let $r = p\varphi(y)$. Then by induction, $\mathcal{A}$ has runs on y from p to r and on z from r to q, with r taken in the same component as p and q, so x labels a run of $\mathcal{A}$ from p to q, and x satisfies the required properties.

Next we consider a run of $\mathcal{A}(S)$ from p to q on $x = t(x_1, \ldots, x_n)$, with $t \in \Sigma_n$ and $n \geq 2$. Then $\mathcal{A}(S)$ has an opening t-transition $p \to (p_1, \ldots, p_n)$, a closing t-transition $(q_1, \ldots, q_n) \to q$, and runs on x_i from p_i to q_i for $i = 1, \ldots, n$. By induction, it follows that p_i and q_i are in the same component for each i, and in S, $p_i = 1$ and $q_i = \varphi(x_i)$ for each i.

Now, by definition of $\mathcal{A}(S)$, p and p_1 are in the same component, and q_1 and q are in the same component. So p and q are in the same component. In addition, p_n and q_n are in S_p and in S we have $q = p\ t(\varphi(x_1), \ldots, \varphi(x_n)) = p\ \varphi(x)$.

Conversely, suppose that p and q are in the same component, say T, and that in S $q = p\ t(\varphi(x_1), \ldots, \varphi(x_n))$. By definition of $\mathcal{A}(S)$, there exist an opening t-transition $p \to (p_1, \ldots, p_n)$ and a closing t-transition $(q_1, \ldots, q_n) \to q$ such that p, p_1, $\ldots$, p_{n-1}, q_1, $\ldots$, q_{n-1} and q are in T, and p_n and q_n are in S_p, and such that in S, $p_i = 1$ and $q_i = \varphi(x_i)$ for $i = 1, \ldots, n$. In addition, by induction, $\mathcal{A}(S)$ has a run on x_i from p_i to q_i for each i. Thus $\mathcal{A}(S)$ has a run on x from p to q.

The case of a run on a t-element where t has arity 1 is handled similarly. This concludes the proof of the lemma. □

The following result is now immediate, and it completes the proof of Theorem 3.1.

Proposition 3.5 *Let $\varphi\colon S_\Sigma(A) \to S$ be a morphism into a finite $s\Sigma$-algebra, let $F \subseteq S$ and let $L = \varphi^{-1}(F)$. Then L is accepted by the automaton $\mathcal{A}(S)$, where the final states are chosen to be the elements of F, taken in S_0.*

Remark. Automata of the form $\mathcal{A}(S)$ can be compared to the deterministic complete automata for words: for every $x \in S_\Sigma(A)$, $\mathcal{A}(S)$ has exactly one run on x starting at every state.

4 Rationality and the Kleene theorem for $s\Sigma$-languages

Now that we have a notion of automata which matches recognizability, we can look for a definition of rationality which matches both. We first prove some closure properties of regular languages, which leads to a definition of rationality. Because of closure, rational languages are regular. We round off the Kleene theorem with a proof that recognizable languages are rational.

4.1 Closure properties of regular languages

We first show that the class of regular $s\Sigma$-languages is closed under all the generalized rational operations except for ξ-exponentiation.

4.1.1 Union, sequential iteration, substitution and products

We prove the following closure properties of the family of $s\Sigma$-regular languages.

Proposition 4.1 *Let $t \in \Sigma_n$ and let $L_1, \ldots, L_n$ be regular $s\Sigma$-languages. Then $L_1 \cup L_2$, $L_1 L_2$, $t(L_1, \ldots, L_n)$ and L_1^+ are regular. If $\xi \in A$, then $L_2 \circ_\xi L_1$ is regular.*

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be branching automata accepting respectively L_1 and L_2. The proof that $L_1 \cup L_2$ is regular, is the same as in the classical theory of automata: $L_1 \cup L_2$ is accepted by the disjoint union of $\mathcal{A}_1$ and $\mathcal{A}_2$.

The other closure operations considered in this section, namely ξ-substitution, sequential product, sequential iteration and the construction of t-products, require the use of automata satisfying the conditions of Lemma 2.2. We refer the reader to [17, Section 4.3] for an illustration of the necessity of considering such restrictions on automata.

We first consider the substitution operation, as the other closure properties can be quickly deduced from it. We assume that $\mathcal{A}_2$ satisfies the conditions in Lemma 2.2.

We construct a new automaton $\mathcal{B}$ as follows. Let T_ξ be the set of pairs (p, q) of states of $\mathcal{A}_1$, such that there $\mathcal{A}_1$ has a ξ-transition from p to q. First we consider a copy of $\mathcal{A}_1$ and $|T_\xi|$ copies of $\mathcal{A}_2$, denoted by $\mathcal{B}_{p,q}$ ($(p, q) \in T_\xi$). We keep as initial (resp. final) states the initial (resp. final) states of $\mathcal{A}_1$.

Next, we add new transitions to $\mathcal{B}$. For each $(p, q) \in T_\xi$, we delete this transition in our copy of $\mathcal{A}_1$ and we add duplicates of all the sequential and opening t-transitions of $\mathcal{B}_{p,q}$ ($t \in \Sigma$) originating in the initial state, with p substituted for the initial state on the left-hand side. Dually, we add duplicates of all the sequential and closing t-transitions of $\mathcal{B}_{p,q}$ ending in the final state, with q substituted for the final state on the right-hand side. In addition, if there is a letter a such that there exists an a-labelled transition from the initial state of $\mathcal{A}_2$ to its final state, then we add an a-labelled transition from p to q.

We prove the following technical lemma.

Lemma 4.2 *Let $x \in S_\Sigma(A)$ and let p and q be states of $\mathcal{B}$, such that $\mathcal{B}$ has a run on x from p to q. Then the following holds:*

(1) *if p and q are in $\mathcal{A}_1$, then $x \in L_2 \circ_\xi x'$ for some $x' \in S_\Sigma(A)$ such that $\mathcal{A}_1$ has a run on x' from p to q;*

(2) *if p is in $\mathcal{B}_{r,s}$ and q is in $\mathcal{A}_1$, then $x = yz$ where y is the suffix of an element of L_2, y labels a run of $\mathcal{A}_2$ from p to f, the final state of $\mathcal{A}_2$, and z labels a run of $\mathcal{B}$ from s to q (z may be empty);*

(3) *if q is in $\mathcal{B}_{r,s}$ and p is in $\mathcal{A}_1$, then $x = zy$ where y is the prefix of an element of L_2, y labels a run of $\mathcal{A}_2$ from i, the initial state of $\mathcal{A}_2$ to q, and z labels a run of $\mathcal{B}$ from from p to r (z may be empty);*

(4) *if p is in $\mathcal{B}_{r,s}$ and q is in $\mathcal{B}_{r',s'}$, then either $(r, s) = (r', s')$ and the run of $\mathcal{B}$ on x takes place entirely within $\mathcal{B}_{r,s}$, or $x = zyz'$ where z is a suffix of L_2, z' is a prefix of L_2, z labels*

a run of $\mathcal{A}_2$ from p to f, z' labels a run of $\mathcal{A}_2$ from i to q, and y labels a run of $\mathcal{B}$ from s to r'.

Proof. We proceed by induction on $|x|$. The case where x is a letter is trivial, whether the run of $\mathcal{B}$ on that letter is a new transition or not. We now suppose that the lemma holds for $y, z \in S_\Sigma(A)$ and we consider a run of $\mathcal{B}$ on $x = yz$ from p to q. Then $\mathcal{B}$ has runs on y from p to r, and on z from r to q, for some state r, and the lemma is easily verified.

We now suppose that $x = t(x_1, \ldots, x_n)$, with $t \in \Sigma_n$ and $n \geq 2$. Then $\mathcal{B}$ has an opening t-transition $p \to (p_1, \ldots, p_n)$, a closing t-transition $(q_1, \ldots, q_n) \to q$, and runs on x_i from p_i to q_i for $i = 1, \ldots, n$. Let us suppose for instance that p lies in $\mathcal{B}_{r,s}$ and q lies in $\mathcal{A}_1$. By definition of $\mathcal{B}$, the p_i lie in $\mathcal{B}_{r,s}$ as well. If the run of $\mathcal{B}$ from p_i to q_i is not set entirely within $\mathcal{B}_{r,s}$, then by induction, $\mathcal{A}_2$ has a run from p_i to f, in contradiction with Condition (3) in Lemma 2.2. Thus x_i labels a run of $\mathcal{A}_2$ from p_i to q_i for each i. In addition, since q is in $\mathcal{A}_1$, then $\mathcal{A}_2$ has a closing t-transition $(q_1, \ldots, q_n) \to f$ and $q = s$. Thus the lemma holds in this case.

Let us now suppose that p and q lie in $\mathcal{A}_1$. By definition of $\mathcal{B}$, the p_i are simultaneously in $\mathcal{A}_1$ or out of $\mathcal{A}_1$, and the same holds for the q_i. If the p_i and the q_i are in $\mathcal{A}_1$, then by induction, for each i, $x_i \in L_2 \circ_\xi x'_i$ for some x'_i such that $\mathcal{A}_1$ has a run on x'_i from p_i to q_i. Letting $x' = t(x'_1, \ldots, x'_n)$, we see that $\mathcal{A}_1$ has a run on x' from p to q, and $x \in L_2 \circ_\xi x'$. If the p_i are not in $\mathcal{A}_1$, but the q_i are, then by induction, $\mathcal{A}_2$ has a run from p_i to f, in contradiction again with Condition (3) in Lemma 2.2. Dually, it is not possible to have the p_i in $\mathcal{A}_1$ and the q_i outside $\mathcal{A}_1$. Finally, if neither p_i nor q_i is in $\mathcal{A}_1$, then $\mathcal{A}_2$ has an opening t-transition $i \to (p_1, \ldots, p_n)$, a ξ-transition from p to, say, p', and the p_i lie in $\mathcal{B}_{p,p'}$. If the run of $\mathcal{B}$ on x_i is not set entirely within $\mathcal{B}_{p,p'}$, again we contradict Condition (3) in Lemma 2.2 by induction. Therefore we verify as above that $p' = q$, the q_i lie in $\mathcal{B}_{p,q}$, and $x = t(x_1, \ldots, x_n) \in L_2$. The lemma holds for x with $x' = \xi$.

All the remaining cases, and the situation where t has arity 1, are treated similarly. $\square$

We can now verify that $\mathcal{B}$ accepts exactly $L_2 \circ_\xi L_1$. Indeed, it is immediate that if $(p, q) \in T_\xi$, then $\mathcal{B}$ has a run on every element of L_2, from p to q. In particular, $L_2 \circ_\xi L_1$ is contained in the $s\Sigma$-language accepted by $\mathcal{B}$. Conversely, if x labels a successful run of $\mathcal{B}$, then $x \in L_2 \circ_\xi L_1$ by Lemma 4.2 (1). It follows that $L_2 \circ_\xi L_1$ is regular.

From the preservation of regularity under substitution, we derive the rest of the proof of Proposition 4.1. Indeed, $\xi_1, \ldots, \xi_n$ be new letters not in A. The $s\Sigma$-languages $\{\xi_1\xi_2\}$, $\{t(\xi_1, \ldots, \xi_n)\}$ and ξ_1^+ are easily seen to be regular. We conclude immediately, since $L_1L_2 = L_1 \circ_{\xi_1} (L_2 \circ_{\xi_2} \{\xi_1\xi_2\})$, $t(L_1, \ldots, L_n) = L_1 \circ_{\xi_1} (\ldots (L_n \circ_{\xi_n} \{t(\xi_1, \ldots, \xi_n)\}) \ldots)$ and $L_1^+ = L_1 \circ_{\xi_1} \xi_1^+$.

4.1.2 An exponentiation which preserves regularity

We now come to the matter of ξ-exponentiation. Since the troublesome example was $(a\xi c)^{*\xi}$ (generalized rational but not recognizable), we begin by banning exponentiation of sequential products.

Proposition 4.3 *Let L be a regular $s\Sigma$-language containing no sequential element, and let $\xi \in A$. Then $L^{*\xi}$ is regular.*

To prove Proposition 4.3, we consider an automaton $\mathcal{A}$ accepting L and satisfying the conditions of Lemma 2.2. The hypothesis on L implies that there are no sequential transitions out of the initial state i or into the final state f. In addition, no state other than i (resp. f) is the origin (resp. the end) of a path to f (resp. from i).

We construct an automaton $\mathcal{C}$ as follows. Let $T_\xi \subseteq Q \times Q$ be the set of ξ-transitions of $\mathcal{A}$, as in Section 4.1.1. First we consider $|T_\xi| + 1$ copies of $\mathcal{A}$, denoted by $\mathcal{A}_0$ and $\mathcal{A}_{p,q}$ ($(p,q) \in T_\xi$), which are called the components of $\mathcal{C}$. We make states i and f in $\mathcal{A}_0$, respectively the initial and the final state of $\mathcal{C}$.

Next, we add new transitions to $\mathcal{C}$, all of them opening and closing t-transitions. For each $(p,q) \in T_\xi$, for each component $\mathcal{B}$ of $\mathcal{C}$, for each $t \in \Sigma$ and for each opening t-transition $i \to (p_1, \dots, p_n)$ out of the initial state of $\mathcal{A}$, we add to $\mathcal{C}$ an opening t-transition $p \to (p_1, \dots, p_n)$, where p and $p_1, \dots, p_{n-1}$ are taken in $\mathcal{B}$ and p_n is taken in $\mathcal{A}_{p,q}$.

Also, for each $(p,q) \in T_\xi$, for each component $\mathcal{B}$ of $\mathcal{C}$, for each $t \in \Sigma$ and for each closing t-transition $(q_1, \dots, q_n) \to f$ into the final state of $\mathcal{A}$, we add to $\mathcal{C}$ a closing t-transition $(q_1, \dots, q_n) \to q$, where $q_1, \dots, q_{n-1}$ and q are taken in $\mathcal{B}$ and q_n is taken in $\mathcal{A}_{p,q}$.

Finally, we add a ξ-labelled transition from state i of $\mathcal{A}_0$ to state f of $\mathcal{A}_0$, and we remove states i and f (and the transitions attached to them) from all the components of $\mathcal{C}$ except $\mathcal{A}_0$. We first prove two technical lemmas.

Lemma 4.4 *Let $x \in L^{*\xi}$. Then $\mathcal{C}$ has a run on x from i to f. In addition, for each $(p,q) \in T_\xi$, $\mathcal{C}$ also has a run on x from p to q, with p and q taken in $\mathcal{A}_0$.*

Proof. The result is trivial if $x = \xi$. Next we verify the claim if $x \in L$, that is, when x is accepted by $\mathcal{A}$. As $\mathcal{A}_0$ is a copy of $\mathcal{A}$, x is also accepted by $\mathcal{C}$. By assumption on L, x is a t-element, for some $t \in \Sigma$. If t has arity $n \geq 2$, then $x = t(x_1, \dots, x_n)$, and $\mathcal{A}$ has an opening t-transition $i \to (p_1, \dots, p_n)$ and a closing t-transition $(q_1, \dots, q_n) \to f$ such that $\mathcal{A}$ has a run on x_j from p_j to q_j for each j. Moreover, by our assumption on $\mathcal{A}$, the run of $\mathcal{A}$ on x_j does not visit i or f. In particular, every component of $\mathcal{C}$ has a run on x_j from p_j to q_j. Now, let $(p,q) \in T_\xi$. Then $\mathcal{C}$ has an opening t-transition $p \to (p_1, \dots, p_n)$ and a closing t-transition $(q_1, \dots, q_n) \to q$ with $p, p_1, \dots, p_{n-1}, q_1, \dots, q_{n-1}, q$ taken in $\mathcal{A}_0$ and p_n, q_n taken in $\mathcal{A}_{p,q}$. Thus, $\mathcal{C}$ has a run on x from p to q, with p and q both taken in $\mathcal{A}_0$. The case where $t \in \Sigma_1$ is handled similarly.

Let now $x \in L^{*\xi}$. We proceed by induction on $|x|$. If $|x| = 1$, then $x = \xi$. Thus, if $|x| = 1$ or if $|x| > 1$ and $x \in L$, we already verified the required properties. We now assume that $x \notin L \cup \{\xi\}$. Then there exists $x' \in L$ such that x is obtained from x' by replacing each occurrence of ξ by an element of $L^{*\xi}$. Let us consider a successful run of $\mathcal{A}_0$ on x', an occurrence of ξ in x', and $(p,q) \in T_\xi$, the ξ-labelled transition of $\mathcal{A}_0$ used by this occurrence of ξ. Let z be the element of $L^{*\xi}$ substituted for this occurrence of ξ in the construction of x. By induction, $\mathcal{C}$ has a run on z from p to q, with p and q taken in $\mathcal{A}_0$. Considering each occurrence of ξ in y, this shows that $\mathcal{C}$ has a run on x from p to q with p, q taken in $\mathcal{A}_0$. $\square$

Lemma 4.5 *Let p, q be states of $\mathcal{C}$ and let $x \in S_\Sigma(A)$ such that $\mathcal{C}$ has a run on x from p to q. Then we have:*

(1) *p and q are in the same component of $\mathcal{C}$;*

(2) *there exists $x' \in S_\Sigma(A)$ such that $x \in L^{*\xi} \circ_\xi x'$ and $\mathcal{A}$ has a run on x' from p to q.*

Proof. The proof is by induction on $|x|$. If $x \in A$, Statement (1) is trivial, since the sequential transitions of $\mathcal{C}$ are all set within a component. In addition, Statement (2) holds with $x' = x$:

either $p = i$, $q = f$ and $x = \xi$, or $\mathcal{A}$ has a run on x from p to q in each of its components. In the latter case, there is in particular a run of $\mathcal{A}_0$ on x from p to q and this proves (2) with $x' = x$.

We now assume that the lemma holds for $y, z \in S_\Sigma(A)$, and we consider a run of $\mathcal{C}$ on $x = yz$ from p to q. Then there exists a state r of $\mathcal{C}$ such that there is a run on y from p to r, and a run on z from r to q. By induction, it follows that p, r and q are in the same component of $\mathcal{C}$, so Statement (1) holds for x. Also by induction, there exist elements $y', z' \in S_\Sigma(A)$ such that $y \in L^{*\xi} \circ_\xi y'$, $z \in L^{*\xi} \circ_\xi z'$, and there are runs of $\mathcal{A}$, from p to r on y' and from r to q on z'. Then Statement (2) holds for x, with $x' = y'z'$.

Next we assume that the lemma holds for $x_1, \ldots, x_n \in S_\Sigma(A)$, and we consider a run of $\mathcal{C}$ from p to q, on $x = t(x_1, \ldots, x_n)$ for some $t \in \Sigma_n$. We first assume that $n \geq 2$. Then $\mathcal{C}$ has an opening t-transition $p \to (p_1, \ldots, p_n)$, a closing t-transition $(q_1, \ldots, q_n) \to q$, and a run on x_j from p_j to q_j for $j = 1, \ldots, n$. By induction, it follows that for each j, p_j and q_j are in the same component and $x_j \in L^{*\xi} \circ_\xi x'_j$ for some $x'_j \in S_\Sigma(A)$ such that $\mathcal{A}$ has a run on x'_j from p_j to q_j.

If $p \to (p_1, \ldots, p_n)$ and $(q_1, \ldots, q_n) \to q$ are not new transitions, then p and the p_j are in the same component and q and the q_j are in the same component, so that Statement (1) holds for x. More over, these opening and closing t-transitions also exist in component $\mathcal{A}_0$. It is now easy to verify that Statement (2) holds for x, with $x' = t(x'_1, \ldots, x'_n)$.

If $p \to (p_1, \ldots, p_n)$ and $(q_1, \ldots, q_n) \to q$ both are new transitions, then by definition of $\mathcal{C}$, p and $p_1, \ldots, p_{n-1}$ lie in the same component, say $\mathcal{B}$, and p_n is in a component of the form $\mathcal{A}_{p,p'}$, such that $\mathcal{A}$ has a ξ-transition from p to p'. Similarly, q and $q_1, \ldots, q_{n-1}$ lie in the same component, which by induction must be $\mathcal{B}$, so that Statement (1) holds for x. Moreover q_n is in a component of the form $\mathcal{A}_{q',q}$, such that $\mathcal{A}$ has a ξ-transition from q' to q. By induction again, we have $p' = q$ and $p = q'$, so $\mathcal{A}$ has a ξ-transition from p to q. Using again the definition of $\mathcal{C}$, we see that $i \to (p_1, \ldots, p_n)$ is an opening t-transition of $\mathcal{A}$ and $(q_1, \ldots, q_n) \to f$ is a closing t-transition of $\mathcal{A}$, so that $\mathcal{A}$ has a successful run on $t(x'_1, \ldots, x'_n)$. In particular, $t(x'_1, \ldots, x'_n) \in L$, and as $x \in L^{*\xi} \circ_\xi t(x'_1, \ldots, x'_n)$, we have $x \in L^{*\xi}$. Thus Statement (2) holds for x with $x' = \xi$.

Let us now assume that $p \to (p_1, \ldots, p_n)$ is a new transition and $(q_1, \ldots, q_n) \to q$ is not. Then $i \to (p_1, \ldots, p_n)$ is an opening t-transition of $\mathcal{A}$ and $(q_1, \ldots, q_n) \to q$ is a closing t-transition of $\mathcal{A}$. By induction, $\mathcal{A}$ also has runs on x'_j from p_j to q_j for each j. Thus $\mathcal{A}$ has a run on $t(x'_1, \ldots, x'_n)$, from i to q. By Condition (5) in Lemma 2.2, it follows that $q = f$, and hence the run of $\mathcal{C}$ on x ends with a transition of $\mathcal{A}_0$ (since state f was removed from all the components of $\mathcal{C}$ but $\mathcal{A}_0$). In particular, q_n is in $\mathcal{A}_0$, and by induction, p_n also lies in $\mathcal{A}_0$, contradicting the assumption that $p \to (p_1, \ldots, p_n)$ is a new transition. The case where $(q_1, \ldots, q_n) \to q$ is a new transition and $p \to (p_1, \ldots, p_n)$ is not, is proved to be contradictory by the dual reasoning.

The case where x is a t-element with $t \in \Sigma_1$ is treated similarly, and this completes the proof of the Lemma. □

We can now prove Proposition 4.3. Lemma 4.4 shows that $L^{*\xi}$ is contained in the $s\Sigma$-language accepted by $\mathcal{C}$. Conversely, let $x \in S_\Sigma(A)$ be the label of a successful run of $\mathcal{C}$. By Lemma 4.5, there exists $x' \in S_\Sigma(A)$ such that x' labels a successful run of $\mathcal{A}$, and $x \in L^{*\xi} \circ_\xi x'$. Thus $x' \in L$, and $x \in L^{*\xi}$. Thus we have proved that $\mathcal{C}$ accepts exactly $L^{*\xi}$, and hence $L^{*\xi}$ is regular.

4.2 Rational $s\Sigma$-languages

The preceding proof has given us a sufficient condition for the preservation of regularity by exponentiation. We use this to define *rational expressions* and *rational $s\Sigma$-languages.*

The *rational expressions* over A are defined inductively as follows.

(1) Every letter $a \in A$ is a rational expression;

(2) if $t \in \Sigma_n$ and if $e_1, \ldots, e_n$ are rational expressions, then so are $e_1 + e_2$, $e_1 e_2$, e_1^+ and $t(e_1, \ldots, e_n)$;

(3) if e_1 and e_2 are rational expressions, and if $\xi \in A$, then $e_1 \circ_\xi e_2$ is a rational expression;

(4) if e is a rational expression which is a sum of Σ-operations, and if $\xi \in A$, then $e^{*\xi}$ is a rational expression.

In other words, a rational expression is a generalized rational expression in which we specifically allow the usage of the sequential iteration e^+, and in which other ξ-exponentiations occur only over sub-expressions which are sums of Σ-operations.

An $s\Sigma$-language is *rational* if it is described by a rational expression. Note that, with this definition, an $s\Sigma$-language entirely contained in A^+ (resp. $T_\Sigma(A)$) is rational if and only if it is generalized rational, and hence if and only if it is recognizable. As we will see (Theorem 4.7 below), the restriction on ξ-exponentiation in the definition of rational $s\Sigma$-languages allows us to maintain a link between rationality and recognizability for all $s\Sigma$-languages.

The following statement is an immediate consequence of the results in Section 4.1.

Proposition 4.6 *Every rational $s\Sigma$-language is regular.*

As an example, recall the language of terms like that in Figure 1, of every depth ≥ 0, described by the rational expression $abc + b \circ_\xi a\ t(b, a\xi c)^{*\xi} c$. The above language could also be described as $b \circ_\xi (a\ t(b,\xi)\ c)^{*\xi}$, which is not rational. We suspect that expressions $e^{*\xi}$, such as $(a\ t(b,\xi)\ c)^{*\xi}$, where every occurrence of ξ in e is inside a Σ-operation, can be transformed to rational expressions, but we leave the formulation of a complete set of transformations for future work.

4.3 Recognizable $s\Sigma$-languages are rational

The final link in our chain is to prove that the recognizable languages can be described by rational expressions. Our proof is an extension of the McNaughton-Yamada construction, inspired by Büchi's ideas [3, Section 6.6, Exercise 4]. It is a slight variant of the proof of [16, Theorem 1], where it was used to get a less general result.

Let L be a recognizable $s\Sigma$-language, and let S be a finite $s\Sigma$-algebra recognizing L by the morphism $\varphi\colon S_\Sigma(A) \to S$. The rational expressions we will construct, will be over the alphabet $B = A \cup \{\xi_s \mid s \in S\}$, with extra variables for each element of the algebra. The morphism φ is extended to $S_\Sigma(B)$ by setting $\varphi(\xi_s) = s$ for each $s \in S$.

Since L is the union of the $\varphi^{-1}(s)$, when s runs over $\varphi(L)$, it suffices to find rational expressions for the $s\Sigma$-languages of the form $\varphi^{-1}(s)$ $(s \in S)$.

For each set Z of extra variables, $Z \subseteq B \setminus A$, for each subset $R \subseteq S$, and for each $s \in S$, let $L(Z, R, s)$ be the set of elements $x \in S_\Sigma(A \cup Z)$ such that $\varphi(x) = s$, and for every Σ-element y occurring in x, $\varphi(y)$ is in R. Since $\varphi^{-1}(s) = L(\emptyset, S, s)$, it suffices to construct rational expressions for the $s\Sigma$-languages $L(Z, R, s)$. In fact, it suffices to do so when Z and R are such that, for each $u \in R$, $\xi_u \notin Z$. We proceed by induction on $|R|$.

If $|R| = 0$, the elements of $L(Z, \emptyset, s)$ do not use any of the operations of Σ, so they are in fact words over the alphabet $A \cup Z$, and $L(Z, \emptyset, s)$ is a word language accepted by the morphism φ from $(A \cup Z)^+$ into the semigroup S equipped with the sequential product. By the classical Kleene theorem, $L(Z, \emptyset, s)$ can be expressed using the operations of union, sequential product and sequential iteration, so it has a rational expression.

We now assume that R is such that $L(Y, R, s)$ has a rational expression for any value of $s \in S$ and $Y \subseteq B \setminus A$ such that $\xi_u \notin Y$ whenever $u \in R$. Let $s \in S$, let $r \in S \setminus R$ and let $Z \subseteq B \setminus A$ such that $\xi_u \notin Z$ for all $u \in R \cup \{r\}$. In order to conclude our proof by induction, we need to provide a rational expression for $L(Z, R \cup \{r\}, s)$. This is done by proving Equality (1) below. As the notation gets a bit cumbersome, we fix some abbreviations. If $s \in S$, we let:

$$\begin{aligned}
L_s &= L(Z, R, s) \\
\widehat{L}_s &= L(Z \cup \{\xi_r\}, R, s) \\
\widetilde{L}_s &= L(Z, R \cup \{r\}, s) \\
\overline{L}_s &= L(Z \cup \{\xi_r\}, R \cup \{r\}, s)
\end{aligned}$$

The equality we want to prove is the following:

$$\widetilde{L}_s = L_r \circ_{\xi_r} \big(\bigcup t(\widehat{L}_{p_1}, \dots, \widehat{L}_{p_n})\ \big)^{*\xi_r} \circ_{\xi_r} \widehat{L}_s, \tag{1}$$

where the union runs over all the operations $t \in \Sigma$ (n is the arity of t) and over all the elements $p_1, \dots, p_n \in S$ such that $t(p_1, \dots, p_n) = r$. We abbreviate this union by $\widehat{L}_\Sigma$. Since the substitutions and exponentiations all concern ξ_r, we also write $\circ_r$ and ^{*r} instead of $\circ_{\xi_r}$ and ${}^{*\xi_r}$. Equality (1) to be proved is now written

$$\widetilde{L}_s = L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_s. \tag{2}$$

Since $\circ_r$ is associative, the above equation can be parenthesized any which way.

We first show that the right hand side of Equation (2) is included in the left. This part of the proof does not depend on the induction hypothesis. If $p_1, \dots, p_n \in S_\Sigma(A)$ and $t(p_1, \dots, p_n) = r$, then we have $t(\widehat{L}_{p_1}, \dots, \widehat{L}_{p_n}) \subseteq \overline{L}_r$. Indeed, the φ-image of elements of $t(\widehat{L}_{p_1}, \dots, \widehat{L}_{p_n})$ is $t(p_1, \dots, p_n) = r$; the alphabet of $t(\widehat{L}_{p_1}, \dots, \widehat{L}_{p_n})$ is that of $\widehat{L}_{p_1}, \dots, \widehat{L}_{p_n}$, namely $Z \cup \{\xi_r\}$; and the Σ-elements occurring in $t(\widehat{L}_{p_1}, \dots, \widehat{L}_{p_n})$ either occur in one of the $\widehat{L}_{p_i}$, or they are of the form $t(y_{p_1}, \dots, y_{p_n})$ with $y_i \in \widehat{L}_{p_i}$ for each i: in any case, their φ-image is in $R \cup \{r\}$.

It follows that $\widehat{L}_\Sigma \subseteq \overline{L}_r$. Now it is not difficult to verify that $\overline{L}_r^{*r} = \overline{L}_r$. So $\widehat{L}_\Sigma^{*r} \subseteq \overline{L}_r$, and $L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_s \subseteq L_r \circ_r \overline{L}_r \circ_r \widehat{L}_s$.

Next, we verify that $L_r \circ_r \overline{L}_r \subseteq \widetilde{L}_r$. If x is an element of $\overline{L}_r$ and every ξ_r in x is replaced with an element of L_r, then the resulting element does not have any occurrence of ξ_r, since $\xi_r \notin Z$ does not occur in L_r, and hence it lies in $\widetilde{L}_r$.

It follows that $L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_s \subseteq \widetilde{L}_r \circ_r \widehat{L}_s$. The verification that $\widetilde{L}_r \circ_r \widehat{L}_s$ is contained in $\widetilde{L}_s$ is elementary. Thus we have proved the following inclusion:

$$L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_s \subseteq \widetilde{L}_s.$$

We now prove the converse inclusion for all values of s simultaneously. Let $x \in \widetilde{L}_s$. We show that $x \in L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_s$ by induction on the number n of Σ-elements occurring in x, whose φ-image is equal to r.

If $n = 0$, then $x \in L_s$, which is trivially contained in $L_r \circ_r \widehat{L}_s$. In addition, since $\xi_r \in \widehat{L}_\Sigma^{*r}$, $\widehat{L}_s$ is contained in $\widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_s$, so $x \in L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_s$.

We now assume that $n > 0$, and we consider all the Σ-elements of $S_\Sigma(A)$ with φ-image equal to r, which occur in x, but do not occur in another Σ-element occurring in x (in a tree representation of x, these are the minimal Σ-elements with φ-image equal to r, in terms of the order induced by the tree representation of x, where the root is the minimum element). Replacing each of these Σ-elements by ξ_r yields an element of $\widehat{L}_s$: indeed, we added occurrences of ξ_r, but we removed every occurrence of a t-product ($t \in \Sigma$) whose φ-image is equal to r.

Thus it suffices to show that each of these Σ-elements $t(y_1, \ldots, y_n)$ lies in $L_r \circ_r \widehat{L}_\Sigma^{*r}$. If $\varphi(y_i) = p_i$ for each i, then $t(p_1, \ldots, p_n) = r$ and $y_i \in \widetilde{L}_{p_i}$ for each i. Also, we note that the number of Σ-elements occurring in each y_i, whose φ-image is r, is at most $n - 1$. So $y_i \in L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_{p_i}$ by induction. Therefore

$$t(y_1, \ldots, y_n) \in L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r t(\widehat{L}_{p_1}, \ldots, \widehat{L}_{p_n}) \subseteq L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_\Sigma \subseteq L_r \circ_r \widehat{L}_\Sigma^{*r}.$$

Thus we have proved that x lies in $L_r \circ_r \widehat{L}_\Sigma^{*r} \circ_r \widehat{L}_s$, which concludes the proof.

Using Theorem 3.1 and Proposition 4.1, we now have a complete proof of a Kleene-Myhill-Nerode theorem.

Theorem 4.7 *Let L be an $s\Sigma$-language. Then the following are equivalent:*

(1) *L is rational;*
(2) *L is regular;*
(3) *L is recognizable.*

In view of Corollary 1.3, the following result is an immediate consequence of Theorem 4.7.

Corollary 4.8 *The class of recognizable $s\Sigma$-languages is closed under Boolean operations, sequential product and iteration, Σ-operations, ξ-substitution and rational ξ-exponentiation, as well as under direct and inverse morphisms.*

5 Axiomatized $s\Sigma$-algebras

In this section, we investigate the connection between recognizability, regularity and rationality in certain relatively free quotients of the free $s\Sigma$-algebra $S_\Sigma(A)$. We use the concept of a *theory* (see [8, 4]), restricted to $s\Sigma$-algebras. More precisely, let a theory be a pair $\mathcal{T} = (\Sigma, E)$ where E is a set of pairs of elements of a free $s\Sigma$-algebra, considered as identities for $s\Sigma$-algebras. An *$s\mathcal{T}$-algebra* is an $s\Sigma$-algebra satisfying all the identities in E.

If $\mathcal{T}$ is a theory, the free $s\mathcal{T}$-algebra over the alphabet A is denoted by $S_\mathcal{T}(A)$: there exists a canonical morphism $\pi: S_\Sigma(A) \to S_\mathcal{T}(A)$, mapping each letter of A onto itself.

Let us say that a subset L of $S_\mathcal{T}(A)$ is *rational* if it is the π-image of a rational subset of $S_\Sigma(A)$. Similarly, let us say that a branching automaton $\mathcal{A}$ has *a run on an element* $x \in S_\mathcal{T}(A)$

from state p to state q if $\mathcal{A}$ has a run from p to q on some element in $\pi^{-1}(x)$. Finally, we say that a subset $L \subseteq S_{\mathcal{T}}(A)$ is *regular* if it is accepted by a branching automaton.

With these definitions, the following is immediate, in view of Theorems 1.1 and 4.7.

Proposition 5.1 *Let $\mathcal{T}$ be a theory, and let L be a subset of $S_{\mathcal{T}}(A)$.*

(1) *L is rational if and only if L is regular, if and only if L is the image under π of a recognizable subset of $S_{\Sigma}(A)$.*
(2) *L is recognizable if and only if $\pi^{-1}(L)$ is recognizable.*
(3) *If L is recognizable, then L is rational.*

5.1 Commutativity

Say that a theory $\mathcal{T} = (\Sigma, E)$ is *commutative* if all the elements of E are of the form

$$t(x_1, \ldots, x_n) = t(x_{\sigma(1)}, \ldots, x_{\sigma(n)}), \qquad (E_{t,\sigma})$$

for some $t \in \Sigma_n$ with $n \geq 2$ and for some permutation σ on $\{1, \ldots, n\}$.

Observe that E associates with each operation $t \in \Sigma_n$ $(n \geq 2)$ a subgroup H_t of the group of permutations on n elements: the subgroup generated by the permutations σ such that $E_{t,\sigma}$ is in E.

The simple structure of the elements of E also leads to the following characterization of runs of a branching automaton on elements of $S_{\mathcal{T}}(A)$.

Lemma 5.2 *Let $\mathcal{A}$ be a branching automaton. There is a run on letter a from p to q if (p, a, q) is a sequential transition of $\mathcal{A}$. If x can be written as a sequential product $x = yz$, there is a run on x from p to q if and only if there exists a state r and there are runs on y from p to r and on z from r to q. If $t \in \Sigma_1$ and $x = t(x_1)$, there is a run from p to q on x if and only if there is an opening t-transition $p \to (p_1, r)$ and a closing t-transition $(q_1, r) \to q$ such that $\mathcal{A}$ has a run on x_1 from p_1 to q_1. Finally, if $x = t(x_1, \ldots, x_n)$ for some $n \geq 2$ and $t \in \Sigma_n$, there is a run on x from p to q if there is an opening t-transition $p \to (p_1, \ldots, p_n)$, a closing t-transition $(q_1, \ldots, q_n) \to q$ and a permutation $\sigma \in H_t$ such that $\mathcal{A}$ has runs on $x_{\sigma(i)}$ from p_i to q_i for $i = 1, \ldots, n$.*

In this situation, we retain the good properties of $s\Sigma$-languages.

Theorem 5.3 *Let $\mathcal{T}$ be a commutative theory and L a subset of $S_{\mathcal{T}}(A)$. Then L is rational if and only if L is regular, if and only if L is recognizable.*

Proof. In view of Proposition 5.1, it suffices to show that if L is a regular subset of $S_{\mathcal{T}}(A)$, then L is recognizable. So let $\mathcal{A}$ be a branching automaton accepting L, and let $\sim$ be the relation on $S_{\mathcal{T}}(A)$ defined by $x \sim y$ if for all states p, q of $\mathcal{A}$, there is a run of $\mathcal{A}$ on x from p to q if and only if there is a run of $\mathcal{A}$ on y from p to q. As in the proof of Proposition 3.2, it is a routine verification that $\sim$ is a finite-index congruence on $S_{\mathcal{T}}(A)$, and that this can be used to conclude that L is recognizable. □

Corollary 5.4 *Let $\mathcal{T}$ be a commutative theory. The class of recognizable subsets of $S_{\mathcal{T}}(A)$ is closed under Boolean operations, sequential product and iteration, Σ-operations, ξ-substitution and rational ξ-exponentiation, as well as under direct and inverse morphisms.*

5.2 Associativity

The situation is not as favourable if we introduce associativity conditions rather than commutativity. Let Σ have a single binary operation $\|$, which we will call *parallel product*, and let $\mathcal{T} = (\Sigma, E)$ state that $\|$ is associative. Note that the runs of a branching automaton on elements of $S_{\mathcal{T}}(A)$ are characterized inductively as follows: there is a run on letter a from p to q if (p, a, q) is a sequential transition of $\mathcal{A}$; if x can be written as a sequential product $x = yz$, there is a run on x from p to q if and only if there exists a state r and there are runs on y from p to r and on z from r to q; if x can be written as a parallel product, there is a run on x from p to q if and only if there is an opening $\|$-transition $p \to (p_1, p_2)$ and a closing $\|$-transition $(q_1, q_2) \to q$, and there is a factorization of x in the form $x = x_1 \| x_2$ such that $\mathcal{A}$ has a run on x_1 from p_1 to q_1 and a run on x_2 from p_2 to q_2.

In these conditions, the proof of Theorem 5.3 does not carry over.

Example 5.5 Let us consider the automaton $\mathcal{A}$ in Example 3.3. The relation $\sim$ on $S_\Sigma(A)$ discussed in that Example, is now considered as a relation on $S_{\mathcal{T}}(A)$ (like in the proof of Theorem 5.3). We observed in Example 3.3 that c is $\sim$-equivalent to $a\|c$. However $b\|c$ is not $\sim$-equivalent to $b\|a\|c$, since $b\|a\|c = (b\|a)\|c$ labels a run from 1 to 12, whereas $b\|c$ does not label any run of $\mathcal{A}$. Thus $\sim$ is not a congruence on $S_{\mathcal{T}}(A)$.

In fact, the class of recognizable subsets is properly contained in the class of rational subsets. For instance, if ξ is a new letter not in A, then $(a\|\xi\|b)^{*\xi}$ is a rational set which is not recognizable, whereas the finite set $\{a\|\xi\|b\}$ is certainly recognizable.

The class of recognizable subsets of $S_{\mathcal{T}}(A)$ enjoys however some nice closure properties. A result of Courcelle [4, Corollary 3] shows that it is closed under letter-to-letter morphisms (relabellings).

Proposition 5.6 *The class of recognizable subsets of $S_{\mathcal{T}}(A)$ is closed under Boolean operations, sequential and parallel product, sequential and parallel iteration, as well as under inverse morphisms.*

Proof. Closure under Boolean operations and inverse morphism is a consequence of Theorem 1.1. We refer the reader to [17, Lemma 2.9] and [17, Lemma 2.11] for the proof of the closure of recognizable sets under sequential product and iteration. In [17], the corresponding results were stated and proved under the assumption that the parallel product is commutative as well as associative (that is, for series-parallel posets, see Section 5.3 below), but the proof can be used without any change in the present setting. As the two products have the same properties, and hence have symmetrical roles as far as recognizability is concerned, the same techniques can be used to show that recognizable subsets are closed under parallel product and its iteration. $\square$

Remark. In the situation described, where we consider the free A-generated algebra F on two associative operations, let us call them temporarily op_1 and op_2, we have three competing definitions of a rational subset: one uses op_1 as the sequential product, and lets $\Sigma = \{\mathrm{op}_2\}$; the second one inverts the roles of the two operations, and the third lets $\Sigma = \{\mathrm{op}_1, \mathrm{op}_2\}$ (with a third, unnamed operation as the sequential product in the $s\Sigma$-algebra). In the last configuration, rationality and generalized rationality are equivalent for the subsets of F. In the first two configurations, certain generalized rational sets are not rational, typically the set $(a \text{ op}_1\ \xi \text{ op}_1\ b)^{*\xi}$ or the set $(a \text{ op}_2\ \xi \text{ op}_2\ b)^{*\xi}$. In neither case does the class of rational subsets coincide with that of recognizable subsets. It would be interesting to characterize recognizability in terms of restrictions imposed on generalized rational expressions in this setting, or more generally in the setting of "multi-$s\Sigma$-algebras", algebras whose axioms assert the associativity of several designated binary operations. (While this paper was under revision, the authors came to know of Ésik's [9] which describes the free algebras in signatures with n associative and m associative commutative binary operations.)

5.3 Series-parallel languages

The study of series-parallel languages, or *sp*-languages, from the point of view of rationality and regularity was initiated in [15]. Series-parallel languages play an important role in theoretical computer science, and we refer the reader to [17] and [20] for more details on this connection. In this section, we tie together the results of this paper with those in [15, 16, 17]. As in Section 5.2, we consider $\Sigma = \{\|\}$ and a theory $\mathcal{T} = (\Sigma, E)$, but E now asserts the associativity *and* commutativity of the parallel product. The free $\mathcal{T}$-algebra over A is denoted $SP(A)$ and *sp*-languages are, by definition, subsets of $SP(A)$.

As in Section 5.2, the runs of a branching automaton on elements of $SP(A)$ are characterized inductively as follows: there is a run on letter a from p to q if (p, a, q) is a sequential transition of $\mathcal{A}$; if x can be written as a sequential product $x = yz$, there is a run on x from p to q if and only if there exists a state r and there are runs on y from p to r and on z from r to q; if x can be written as a parallel product, there is a run on x from p to q if and only if there is an opening $\|$-transition $p \to (p_1, p_2)$ and a closing $\|$-transition $(q_1, q_2) \to q$, and there is a factorization of x in the form $x = x_1 \| x_2$ such that $\mathcal{A}$ has a run on x_1 from p_1 to q_1 and a run on x_2 from p_2 to q_2. In fact, the automata introduced by the authors in [15, 17] were slightly different, but they define the same class of *sp*-languages as we shall verify in Appendix A below.

Previous work by the authors [17] can now be completed with a characterization of regular *sp*-languages, as a special case of Proposition 5.1. Let π be the canonical morphism from $S_\Sigma(A)$ onto $SP(A)$.

Theorem 5.7 *Let L be an sp-language. L is rational if and only if L is regular, if and only if L is the image under π of a recognizable subset of $S_\Sigma(A)$.*

Remark. In [16], the authors attempted to give such a characterization. In that paper, the definition of rational *sp*-languages was a bit different, and allowed ξ-exponentiation in more cases than the present definition. Unfortunately, part of the proof presented in [16], namely Lemma 4.5, was too sketchy, and we were unable to complete that proof. The present paper thus offers a complete proof of the correct statement we had been wishing for earlier.

It was shown in [17] that the class of recognizable *sp*-languages is closed under Boolean operations, sequential and parallel product, sequential iteration, as well as under inverse morphisms. It is also closed under letter-to-letter morphisms, by [4, Corollary 3]. On the other hand, recognizable *sp*-languages are not closed under parallel iteration or direct homomorphic images. A counterexample [17] is provided by the *sp*-language $(a\|b)^{\oplus}$, the parallel iteration of $\{a\|b\}$, which is not recognizable, even though it is the parallel iteration of a finite, and hence recognizable *sp*-language, and it is the image of $a^{\oplus}$ (a recognizable *sp*-language) under the morphism mapping letter a to $a\|b$.

6 $s\Sigma$-languages of bounded depth

A special case of $s\Sigma$-languages is when boundedly many operations are allowed in a term from Υ, a designated non-empty subset of the signature Σ. We will fix Υ for this section. For instance, in the case of series-parallel languages (Section 5.3 above), it makes sense to consider languages which allow a bounded amount of parallelism [15, 17]. Here too, we relate an algebraic characterization, an automata-theoretic characterization, and one in terms of rational expressions. That is, we obtain a Kleene-Myhill-Nerode theorem, Theorem 6.10 below.

6.1 The Υ-nesting relation

Let S be an $s\Sigma$-algebra, and let $\pi\colon S_\Sigma(S) \to S$ be the natural projection. Let $s, t \in S$. We say that $s \prec_\Upsilon t$ if there exists a letter $\xi \notin S$ and a ξ-term u of Υ-depth 1 such that $s = \pi(t \circ_\xi u)$ and $\pi(t) \neq \pi(t \circ_\xi f(u_1, \ldots, u_n))$, where $f(u_1, \ldots, u_n)$ is the Υ-context of ξ in u (see Section 1). Note that if S is A-generated, then $S_\Sigma(S)$ can be replaced with $S_\Sigma(A)$ in this definition. If π is understood, we omit writing it.

The relation $<_\Upsilon$ is defined to be the transitive closure of $\prec_\Upsilon$, and it is called the Υ-*nesting relation*. The relation $\leq_\Upsilon$ is defined to be the reflexive closure of $<_\Upsilon$. By construction, $\leq_\Upsilon$ is a quasi-order on S.

Note the following lemma.

Lemma 6.1 *Let $\varphi\colon S \to T$ be an onto morphism between $s\Sigma$-algebras. Let $s \in S$ and $t, t' \in T$ be such that $\varphi(s) = t$ and $t' \prec_\Upsilon t$. Then there exists $s' \in S$ such that $\varphi(s') = t'$ and $s' \prec_\Upsilon s$.*

Proof. For this proof we consider (implicitly) an onto projection morphism π from a free $s\Sigma$-algebra $S_\Sigma(A)$ onto S; the ξ-terms considered are picked in $S_\Sigma(A \cup \{\xi\})$. By assumption, there exists a ξ-term u of Υ-depth 1 such that $t' = t \circ_\xi u$. In addition, if $f(u_1, \ldots, u_n)$ is the Υ-context of ξ in u, then $t \neq t \circ_\xi f(u_1, \ldots, u_n)$. Let $s' = s \circ_\xi u$. Taking the φ-images, we get $\varphi(s') = \varphi(s) \circ_\xi u = t \circ_\xi u = t'$, and $\varphi(s \circ_\xi f(u_1, \ldots, u_n)) = \varphi(s) \circ_\xi f(u_1, \ldots, u_n) = t \circ_\xi f(u_1, \ldots, u_n)$. Thus $\varphi(s \circ_\xi f(u_1, \ldots, u_n)) \neq t = \varphi(s)$, and hence $s \circ_\xi f(u_1, \ldots, u_n)) \neq s$ so that $s' \prec_\Upsilon s$ as required. $\square$

It is immediate that, if $s, t \in S_\Sigma(A)$ and u is a ξ-term such that $u \neq \xi$, then $s = t \circ_\xi u$ implies $|s| > |t|$. In particular, $s \prec_\Upsilon t$ if and only if $s = t \circ_\xi u$ for some ξ-term u of Υ-depth 1, and the relation $<_\Upsilon$ is irreflexive (and hence a strict partial order). In fact, the maximum length of an ascending $\prec_\Upsilon$-chain starting at s is exactly $\mathrm{dp}_\Upsilon(s)$.

6.2 $s\Sigma$-algebras with bounded Υ-depth

We say that an $s\Sigma$-algebra S has *bounded* Υ-*depth* if the relation $<_\Upsilon$ is irreflexive (as in $S_\Sigma(A)$), and if there is a uniform bound on the length of $\prec_\Upsilon$-chains in S (unlike $S_\Sigma(A)$). An $s\Sigma$-algebra with bounded Υ-depth always has a $\leq_\Upsilon$-minimal element.

Lemma 6.2 *Let S be an $s\Sigma$-algebra with bounded Υ-depth, and let z be a $\leq_\Upsilon$-minimal element of S. Then z is $\leq_\Upsilon$-minimum, and it is absorbing for the sequential product and for each of the operations in Σ.*

Proof. Let $f \in \Upsilon$ be an operation with arity n, let $t_1, \ldots, t_{n-1} \in S$ and let u be a ξ-term with Υ-depth 0. For each $s \in S$, we have

$$f(t_1, \ldots, t_{j-1}, s \circ_\xi u, t_j, \ldots, t_{n-1}) \prec_\Upsilon s \text{ or } f(t_1, \ldots, t_{j-1}, s \circ_\xi u, t_j, \ldots, t_{n-1}) = s.$$

Since z is $\leq_\Upsilon$-minimal, this yields $f(t_1, \ldots, t_{j-1}, z \circ_\xi u, t_j, \ldots, t_{n-1}) = z$, and in particular (for $u = \xi$), z is absorbing for every operation of Υ.

Let $x \neq z \in S$. The above observation also implies (using $u = \xi x$) that $z = f(zx, x, \ldots, x) \prec_\Upsilon x$, so z is $\leq_\Upsilon$-minimum in S.

Furthermore, $zx = f(zx, x, \ldots, x)x = (zx) \circ_\xi (f(\xi, x, \ldots, x)x)$. The Υ-context of ξ in $f(\xi, x, \ldots, x)x$ is $f(\xi, x, \ldots, x)$, so if $zx \neq z$, then we get $zx \prec_\Upsilon zx$, in contradiction with the hypothesis that $<_\Upsilon$ is irreflexive in S. So $zx = z$ for each $x \in S$. The same reasoning shows, *mutatis mutandis*, that $xz = z$ and, more generally, z is absorbing for every operation in Σ. □

Let S be an $s\Sigma$-algebra having bounded Υ-depth, and let 0 be its $\leq_\Upsilon$-minimum element. We say that S is Υ-*nilpotent* if for all $f \in \Upsilon$ of arity n and for all $x_1, \ldots, x_n \in S \setminus \{0\}$, then $f(x_1, \ldots, x_n)$ is not equal to any of the x_i. We now give two technical results which will be useful to characterize $s\Sigma$-languages with bounded Υ-depth.

Lemma 6.3 *Let S be an $s\Sigma$-algebra with bounded Υ-depth. Then S is Υ-nilpotent if and only if, for any $s, t \in S$ and for any ξ-term u of Υ-depth 1,*

$$s \neq 0 \text{ and } s = t \circ_\xi u \text{ implies } s \prec_\Upsilon t.$$

Proof. First let us assume that whenever $s \neq 0$ and $s = t \circ_\xi u$ for some ξ-term u of Υ-depth 1, then $s \prec_\Upsilon t$. Let $f \in \Upsilon$ be an operation of arity n and let $x_1, \ldots, x_n \in S \setminus \{0\}$. Note that $f_i = f(x_1, \ldots, x_{i-1}, \xi, x_{i+1}, \ldots, x_n)$ is a ξ-term of Υ-depth 1. Thus we have $f(x_1, \ldots, x_n) = x_i \circ_\xi f_i = 0$, or $f(x_1, \ldots, x_n) \prec_\Upsilon x_i$. In either case, $f(x_1, \ldots, x_n) \neq x_i$. Thus S is Υ-nilpotent.

Conversely, let us assume that S is Υ-nilpotent, let u be a ξ-term of Υ-depth 1, and let $s, t \in S$ be such that $s = t \circ_\xi u$. Let $f(u_1, \ldots, u_n)$ be the Υ-context of ξ in u and let j be the (only) index such that u_j is a ξ-term (of Υ-depth 0). If $t \neq t \circ_\xi f(u_1, \ldots, u_n)$, then $s \prec_\Upsilon t$ and we are done. Now let us assume that

$$t = t \circ_\xi f(u_1, \ldots, u_n) = f(u_1, \ldots, u_{j-1}, t \circ_\xi u_j, u_{j+1}, \ldots, u_n).$$

Then $t \circ_\xi u_j = f(u_1, \ldots, u_{j-1}, t \circ_\xi u_j, u_{j+1}, \ldots, u_n) \circ_\xi u_j = (t \circ_\xi u_j) \circ_\xi v$, with

$$v = f(u_1, \ldots, u_{j-1}, \xi, u_{j+1}, \ldots, u_n) \circ_\xi u_j.$$

Now v is a ξ-term of Υ-depth 1, and the Υ-context of ξ in v is $f(u_1, \ldots, u_{j-1}, \xi, u_{j+1}, \ldots, u_n)$. If $t \circ_\xi u_j \neq f(u_1, \ldots, u_{j-1}, t \circ_\xi u_j, u_{j+1}, \ldots, u_n)$, then $t \circ_\xi u_j \prec_\Upsilon t \circ_\xi u_j$, in contradiction with the fact that $<_\Upsilon$ is irreflexive. Therefore

$$t \circ_\xi u_j = f(u_1, \ldots, u_{j-1}, t \circ_\xi u_j, u_{j+1}, \ldots, u_n),$$

and since S is Υ-nilpotent, we have $f(u_1, \ldots, u_{j-1}, t \circ_\xi u_j, u_{j+1}, \ldots, u_n) = 0$. This implies $s = 0$ and we are done in this case as well. This concludes the proof. □

Corollary 6.4 *Let $\varphi: T \to S$ be a morphism between $s\Sigma$-algebras, and assume that S is Υ-nilpotent. If $t, t' \in T$ and $t \prec_\Upsilon t'$, then either $\varphi(t) = 0$ or $\varphi(t) \prec_\Upsilon \varphi(t')$.*

Proof. Note that φ induces a morphism from $S_\Sigma(T \cup \{\xi\})$ into $S_\Sigma(S \cup \{\xi\})$. If u is a ξ-term in $S_\Sigma(T \cup \{\xi\})$, then u and $\varphi(u)$ have the same Υ-depth. By assumption, $t = t' \circ_\xi u$ for some ξ-term u with Υ-depth 1. Taking the φ-images, we get $\varphi(t) = \varphi(t') \circ_\xi \varphi(u)$. By Lemma 6.3, this suffices to ensure that $\varphi(t) \prec_\Upsilon \varphi(t')$, unless $\varphi(t) = 0$. $\square$

6.3 Bounded Υ-depth and Υ-nilpotency

A first connection between the recognition of $s\Sigma$-languages with bounded Υ-depth and $s\Sigma$-algebras with bounded Υ-depth, is provided by the following lemma.

Lemma 6.5 *Let $\varphi: S_\Sigma(A) \to S$ be an onto morphism and let us assume that $s \in S$ and $s <_\Upsilon s$. Then $\varphi^{-1}(s)$ does not have bounded Υ-depth.*

Proof. By definition, there exist $s_1, \ldots, s_n \in S$ such that $s \prec_\Upsilon s_1 \prec_\Upsilon \ldots \prec_\Upsilon s_n \prec_\Upsilon s$. By Lemma 6.1, this implies that for each $x \in \varphi^{-1}(s)$, there exist elements $x_1, \ldots, x_n, y \in S_\Sigma(A)$ such that $\varphi(y) = s$ and $y \prec_\Upsilon x_1 \prec_\Upsilon \ldots \prec_\Upsilon x_n \prec_\Upsilon x$. In particular, $\mathrm{dp}_\Upsilon(y) > \mathrm{dp}_\Upsilon(x)$ and hence, $\varphi^{-1}(s)$ does not have bounded Υ-depth. $\square$

As in other algebraic structures, every subset of a free $s\Sigma$-algebra admits a least recognizing $s\Sigma$-algebra, called its syntactic $s\Sigma$-algebra. More precisely, if $L \subseteq S_\Sigma(A)$ and $x, y \in S_\Sigma(A)$, we let $x \sim_L y$ if

$$x \circ_\xi u \in L \iff y \circ_\xi u \in L \qquad \text{for each } \xi\text{-term } u.$$

One verifies (see for instance [17, Section 2.2]) that $\sim_L$ is a congruence on $S_\Sigma(A)$, called the *syntactic congruence* of L, and if $\mathrm{Synt}(L) = S_\Sigma(A)/\sim_L$ and $\mu: S_\Sigma(A) \to \mathrm{Synt}(L)$ is the natural projection (respectively the *syntactic $s\Sigma$-algebra* and the *syntactic morphism* of L), then L is recognized by a morphism $\varphi: S_\Sigma(A) \to S$ if and only if φ factors through μ. In particular, L is recognizable if and only if its syntactic congruence has finite index. The syntactic $s\Sigma$-algebra of a recognizable $s\Sigma$-language is effectively computable (see for instance [17, Section 2.3]).

Theorem 6.6 *Let $L \subseteq S_\Sigma(A)$ be an $s\Sigma$-language. The following properties are equivalent.*

(1) *L has bounded Υ-depth;*

(2) $\mathrm{Synt}(L)$ *is Υ-nilpotent and $0 \notin \mu(L)$, where μ is the syntactic morphism of L;*

(3) *L is recognized by a morphism $\varphi: S_\Sigma(A) \to S$ into an Υ-nilpotent $s\Sigma$-algebra S, such that $0 \notin \varphi(L)$.*

Proof. Let $\varphi: S_\Sigma(A) \to S$ into an Υ-nilpotent $s\Sigma$-algebra S. Let k be the maximum length of a $\prec_\Upsilon$-chain in S. It follows immediately from Corollary 6.4 that if $s \neq 0$, then the elements of $\varphi^{-1}(s)$ have Υ-depth at most k. Thus (3) implies (1).

Let us now assume that L has bounded Υ-depth, let k be the maximal Υ-depth of an element of L and let μ be the syntactic morphism of L. If $x \in S_\Sigma(A)$ and $\mathrm{dp}_\Upsilon(x) > k$, then for each ξ-term u we have $\mathrm{dp}_\Upsilon(x \circ_\xi u) > k$, and hence $x \circ_\xi u \notin L$. Thus all the elements of $S_\Sigma(A)$ of Υ-depth greater than k have the same μ-image, and that image is an absorbing element of $\mathrm{Synt}(L)$, which we denote by 0. It follows that $\mu^{-1}(s)$ has bounded Υ-depth for each $s \neq 0$ in $\mathrm{Synt}(L)$ and hence, by Lemma 6.5, $s <_\Upsilon s$ may occur in $\mathrm{Synt}(L)$ only for $s = 0$. It is however trivially observed that $0 <_\Upsilon 0$ does not hold since 0 is absorbing. Thus $<_\Upsilon$ is irreflexive in $\mathrm{Synt}(L)$. In addition, every $\prec_\Upsilon$-chain of $\mathrm{Synt}(L)$ has length at most k by Lemma 6.1. So $\mathrm{Synt}(L)$ has bounded Υ-depth and $0 \notin \mu(L)$ and (1) implies (2).

The remaining implication ((2) implies (3)) is immediate, since any $s\Sigma$-language is recognized by its syntactic morphism. □

Note that Theorem 6.6 holds even for non-recognizable $s\Sigma$-languages: it does not require the finiteness of the recognizing $s\Sigma$-algebra. The characterization of bounded depth recognizable $s\Sigma$-languages will be completed in the next section.

6.4 Υ-bounded rational expressions and Υ-acyclic automata

Υ-bounded rational expressions are defined by a restriction of the definition of rational expressions (Section 4.2). The operations we allow are the union $e + e'$, the operations of Σ, the sequential product ee' and the sequential iteration e^+ and the ξ-substitutions $e \circ_\xi e'$. Finally, ξ-exponentiation $e^{*\xi}$ is allowed provided e is a rational expression relative to $\Sigma \setminus \Upsilon$, which is a sum of $(\Sigma \setminus \Upsilon)$-operations: that is, e is a rational expression such that $e^{*\xi}$ is rational as well, and e does not use any operation from Υ.

An $s\Sigma$-language is said to be *Υ-bounded rational* if it can be described by an Υ-bounded rational expression.

Lemma 6.7 *Let L be an $s\Sigma$-language. Then L is Υ-bounded rational if and only if L is rational and it has bounded Υ-depth.*

Proof. The forward direction is immediate. For the converse, let us consider a rational expression e such that $L(e)$ has bounded Υ-depth. Let $g^{*\xi}$ be an occurrence of a ξ-exponentiation in e. Then $L(g^{*\xi})$ has bounded Υ-depth as well. By definition of a rational expression, g is a sum of Σ-operations. Suppose that $g = h + k$ where h is a (possibly empty) sum of $(\Sigma \setminus \Upsilon)$-operations and k is a non-empty sum of Υ-operations. If ξ has an occurrence in k, then $L(k^{*\xi})$ has unbounded Υ-depth, and hence so does $L(g^{*\xi})$, a contradiction. So ξ does not occur in k and we have $L(g^{*\xi}) = L((\xi + k) \circ_\xi h^{*\xi})$. Replacing $g^{*\xi}$ by $(\xi + k) \circ_\xi h^{*\xi}$, we obtain an equivalent rational expression with one less occurrence of a ξ-exponentiation of a sum involving Υ-operations. Iterating this manipulation, we transform e into an equivalent Υ-bounded rational expression. □

Example 6.8 By definition, if $\Upsilon = \Sigma$, then the restriction on the usage of ξ-exponentiation in Υ-bounded rational expressions is such that ξ-exponentiation is simply banned in Σ-bounded rational expressions. It is not difficult to verify that, in these conditions, one can also dispense with ξ-substitutions. Thus we retrieve the notion of series-rationality introduced in [17] for sp-languages.

Let us now turn to branching automata. Let t be an operation in Σ, let $k = (p \to (p_1, \ldots, p_n))$ be an opening t-transition and let $j = ((q_1, \ldots, q_n) \to q)$ be a closing t-transition. We say that the pair (k, j) of t-transition occurs as a *matched pair of t-transitions* in a run ρ on some element $x \in S_\Sigma(A)$ if x has a subterm of the form $y = t(x_1, \ldots, x_n)$ and the sub-run of ρ on y consists of the opening t-transition k, the closing t-transition j and runs ρ_i on x_i from p_i to q_i for each $1 \leq i \leq n$. The runs ρ_i and all the transitions and matched pairs occurring in them, are said to be *nested within the matched pair* (k, j).

We say that a run is Υ-*acyclic* if in this run, no matched pair of t-transitions with $t \in \Upsilon$ occurs nested within itself. We say that a branching automaton is Υ-*acyclic* if every one of its runs is Υ-acyclic. The decidability of this condition is verified in Section 6.5.

Lemma 6.9 *Let L be an $s\Sigma$-language. Then L is accepted by an Υ-acyclic branching automaton if and only if L is regular and it has bounded Υ-depth.*

Proof. If L is accepted by anΥ-acyclic branching automaton, then the Υ-depth of the elements of L is bounded above by the number of pairs (k, j) of an opening and a closing t-transition, with $t \in \Upsilon$.

Conversely, let us assume that L is recognizable and it has bounded Υ-depth. By Theorem 6.6, L is recognized by a morphism $\varphi \colon S_\Sigma(A) \longrightarrow S$ onto a finite Υ-nilpotent $s\Sigma$-algebra (say, with $S = \mathrm{Synt}(L)$), and we consider the branching automaton $\mathcal{A}(S)$ constructed to prove Proposition 3.5: $\mathcal{A}(S)$ accepts L. Next we construct a branching automaton $\mathcal{B}(S)$ by erasing state 0 in every component of $\mathcal{A}(S)$. As $0 \notin \varphi(L)$ and in view of Lemma 3.4, $\mathcal{B}(S)$ accepts L as well. We now verify that $\mathcal{B}(S)$ is Υ-acyclic.

Let $t \in \Upsilon$ be an operation with arity n, let $k = (p \longrightarrow (p_1, \ldots, p_n))$ and $j = ((q_1, \ldots, q_n) \longrightarrow q)$ be an opening and a closing t-transition of $\mathcal{B}(S)$, and let $x = t(x_1, \ldots, x_n) \in S_\Sigma(A)$. If there is a run of $\mathcal{B}(S)$ on x starting with transition k and ending with transition t, then by Lemma 3.4 $p_i = 1$ and $q_i = \varphi(x_i)$ for each i and $q = p\ t(q_1, \ldots, q_n)$. In particular $\mathrm{dp}_\Upsilon(q) > \mathrm{dp}_\Upsilon(q_i)$ for each i. By Lemma 3.4 again, every state r which is visited along the run from p_i to q_i labelled x_i satisfies $\mathrm{dp}_\Upsilon(q_i) \geq \mathrm{dp}_\Upsilon(r)$, and hence $\mathrm{dp}_\Upsilon(q) > \mathrm{dp}_\Upsilon(r)$. In particular, the matched pair (k, j) cannot occur in any of those runs, that is, it cannot occur nested within itself, and the automaton is Υ-acyclic. $\square$

Bringing together Theorem 3.1,Theorem 6.6 and the two above lemmas, we get the following statement.

Theorem 6.10 *Let L be an $s\Sigma$-language. The following are equivalent.*

(1) *L is recognizable and it has bounded Υ-depth;*
(2) $\mathrm{Synt}(L)$ *is finite, Υ-nilpotent and $0 \notin \mu(L)$, where μ is the syntactic morphism of L;*
(3) *L is recognized by a morphism $\varphi \colon SP(A) \to S$ into a finite Υ-nilpotent $s\Sigma$-algebra S, such that $0 \notin \varphi(L)$;*
(4) *L is accepted by an Υ-acyclic branching automaton;*
(5) *L is Υ-bounded rational.*

Remark. As in [17] one can show that a recognizable $s\Sigma$-language has bounded Υ-depth if and only if it is accepted by a branching automaton in which every successful run is Υ-acyclic (a weaker requirement than Υ-acyclicity). This can be proved without using the notion of Υ-nilpotent $s\Sigma$-algebra, by adapting the proof of [17, Proposition 4.5].

6.5 Deciding Υ-acyclicity for a branching automaton

Using elements from the emptiness decision algorithm in Section 2.2, we prove the following decidability result.

Proposition 6.11 *It is decidable in polynomial time whether a given branching automaton is Υ-acyclic.*

Proof. By definition, an automaton is Υ-acyclic if it is $\{t\}$-acyclic for each operation $t \in \Sigma$. So we may assume that Υ is a singleton, $\Upsilon = \{t\}$.

Let $\mathcal{A}$ be a branching automaton with state set Q and let P be the set of pairs (p, q) of states such that there is a run of $\mathcal{A}$ from p to q. In the proof of Proposition 2.3, it was shown that P is computable in polynomial time. We now build a directed graph Γ with vertex set P as follows. For each pair of an opening t-transition $p \to (p_1, \ldots, p_n)$ and a closing t-transition $(q_1, \ldots, q_n) \to q$ such that the pairs (p_i, q_i) lie in P, we let an edge of Γ go from (p, q) to each of the (p_i, q_i). It is not difficult to verify that if Γ has a cycle going through vertex (p, q), then there exists a run of $\mathcal{A}$ from p to q on a t-operation, in which the matched pair of the initial and terminal t-transitions occurs again, nested within itself. Thus if Γ has cycles, then $\mathcal{A}$ is not Υ-acyclic. We can now conclude since detecting the existence of cycles in a graph can be done in polynomial time. $\square$

Remark. The proof of Proposition 6.11 actually allows us to compute the set P' of pairs of states (p, q) such that there exist runs from p to q which are not Υ-acyclic. Thus we can decide whether the successful runs of the given automaton are Υ-acyclic (if P' does not contain a pair of an initial and a final state), and we can verify the condition described in the Remark at the end of Section 6.4.

6.6 Bounded-depth subsets in axiomatized $s\Sigma$-algebras

Let us consider a theory $\mathcal{T} = (\Sigma, E)$ in which the axioms are all such that both sides are Υ-operations, or neither is. Then the Υ-depth of an element $x \in S_{\mathcal{T}}(A)$ is well-defined: it is the Υ-depth of any of the pre-images of x in the canonical morphism $\pi \colon S_\Sigma(A) \to S_{\mathcal{T}}(A)$. In particular, a subset L of $S_{\mathcal{T}}(A)$ has bounded Υ-depth if and only if $\pi^{-1}(L)$ has bounded Υ-depth, if and only if $L = \pi(K)$ for some $s\Sigma$-language K with bounded Υ-depth.

In view of Proposition 5.1 and Theorem 6.10, it follows that if L is regular, then L has bounded Υ-depth if and only if it is accepted by an Υ-acyclic branching automaton, if and only if it is described by an Υ-bounded rational expression.

If L is recognizable, it is not difficult to verify that the syntactic $\mathcal{T}$-algebra of L is exactly the syntactic $s\Sigma$-algebra of $\pi^{-1}(L)$. Thus, using Proposition 5.1 again and Theorem 6.6, we see that if L is recognizable, then L has bounded Υ-depth if and only if its syntactic algebra is Υ-nilpotent and $0 \notin \mu(L)$ (where μ is the syntactic morphism of L).

To conclude this section, let us finally consider the case where $\Upsilon = \Sigma$. If the set of recognizable subsets of $S_{\mathcal{T}}(A)$ is closed under sequential and Σ-products and under sequential iteration, then every Σ-bounded rational set is recognizable (see Example 6.8). In that case, regularity and recognizability are equivalent properties for subsets of bounded Σ-depth. This holds if $\mathcal{T}$ is commutative (see Section 5.1), if Σ consists of a single binary associative operation (see Section 5.2), or if Σ consists of a single binary commutative and associative operation. The latter case is that of *sp*-languages (see Section 5.3), and it was dealt with in [17].

A Appendix: an equivalent definition of regular sp-languages

Branching automata were first considered in [15, 17], exclusively for the purpose of accepting *sp*-languages. There, a somewhat different definition of a branching automaton was given, which we will call here *sp-branching automata*. In this appendix, we verify that branching automata and *sp*-branching automata accept the same class of *sp*-languages.

In an *sp*-branching automaton (see [15, 17]), the opening and closing $\|$-transitions (called fork and join transitions) may have arbitrary arity (at least 2): a fork transition of arity n ($n \geq 2$) is an element of $Q \times Q^n$, written $p \longrightarrow (p_1, \ldots, p_n)$. Similarly, a join transition of arity n ($n \geq 2$) is an element of $Q^n \times Q$, written $(q_1, \ldots, q_n) \longrightarrow q$.

The definition of the runs of an *sp*-branching automaton differ from the definitions in the previous sections, only on parallel elements of $SP(A)$. If $x \in SP(A) \| SP(A)$, $\mathcal{A}$ has a run on x from p to q if $\mathcal{A}$ has a fork transition $p \longrightarrow (p_1, \ldots, p_n)$ and a join transition $(q_1, \ldots, q_n) \longrightarrow q$ of the same arity, if x has a factorization of the form $x = x_1 \| \cdots \| x_n$, and if there is a permutation σ of $\{1, \ldots, n\}$ such that $\mathcal{A}$ has a run on x_i from p_i to $q_{\sigma(i)}$ for each $1 \leq i \leq n$.

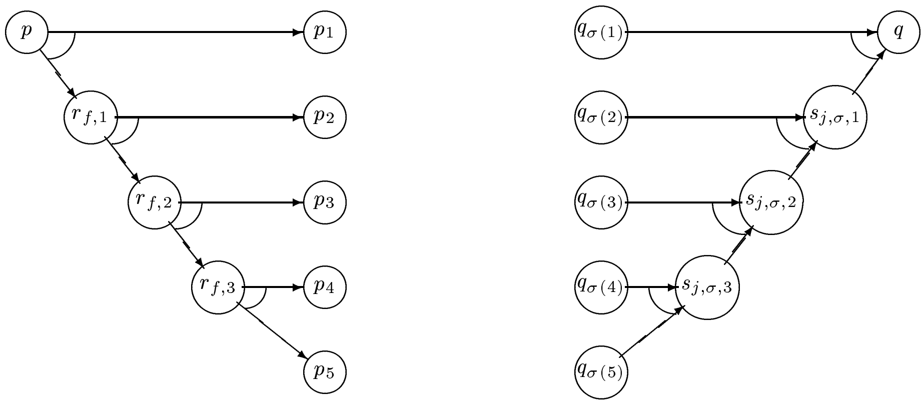
Then we have the following proposition, which is an immediate consequence of Lemmas A.2 and A.3 below.

Proposition A.1 *An sp-language is accepted by an sp-branching automaton if and only if it is regular.*

Lemma A.2 *Every sp-language which is accepted by an sp-branching automaton, is regular.*

Proof. Let $\mathcal{A}$ be an *sp*-branching automaton. Let $\mathcal{B}$ be the branching automaton built from $\mathcal{A}$ as follows. Firstly, all the states of $\mathcal{A}$ are states of $\mathcal{B}$, and $\mathcal{A}$ and $\mathcal{B}$ have the same initial and final states; all the sequential transitions of $\mathcal{A}$ and all its arity 2 fork and join transitions are transitions of $\mathcal{B}$; and the other fork and join transitions of $\mathcal{A}$ (i.e. with arity at least 3) are not transitions of $\mathcal{B}$.

Next, for each fork transition of arity $k \geq 3$ in $\mathcal{A}$, say $f = p \longrightarrow (p_1, \ldots, p_k)$, we add to $\mathcal{B}$ $k-2$ new states and $k-1$ new binary fork transitions as in the 5-ary example below. We proceed similarly with join transitions of arity k.



Thus all the fork and join transitions of $\mathcal{B}$ have arity 2. We now need to verify that $\mathcal{A}$ and $\mathcal{B}$ accept the same *sp*-language. The details of the proof are tedious, but here is a sketch of a proof, by induction on $|x|$, that for each pair of states p, q in $\mathcal{A}$, if there is a run on $x \in SP(A)$ from p to q in $\mathcal{A}$, then there is one in $\mathcal{B}$. The situation is trivial unless the first and last transitions of the run on x in $\mathcal{A}$ are a fork transition f and a join transition j of arity $k \geq 3$, x factors as $x = x_1 \| \cdots \| x_k$ and there is a permutation $\sigma \in S_k$ such that, for each i, there is a run in $\mathcal{A}$ from p_i to $q_{\sigma(i)}$ labelled x_i (where the p_i and the q_j are as in the previous paragraphs). Since $|x_i| < |x|$ for each i, it is immediately verified by induction, that $\mathcal{B}$ has a run on $x_{k-1-i} \| \cdots \| x_k$ from $r_{f,k-2-i}$ to $s_{j,\sigma,k-2-i}$, and eventually, that $\mathcal{B}$ has a run on x from p to q. Thus the language accepted by $\mathcal{A}$ is contained in that accepted by $\mathcal{B}$.

The verification of the converse implication is similar. □

Lemma A.3 *Every regular sp-language is accepted by an sp-branching automaton.*

Proof. Let $\mathcal{A}$ be a branching automaton. We construct an *sp*-branching automaton $\mathcal{B}$ by taking 3 copies of the state set of $\mathcal{A}$, written $\mathcal{A}_0$, $\mathcal{A}_1$ and $\mathcal{A}_2$ (the indices are taken modulo 3), and called the components of $\mathcal{B}$. The initial and final states of $\mathcal{B}$ are those of $\mathcal{A}$, taken in $\mathcal{A}_0$. The sequential transitions of $\mathcal{B}$ all take place within the components, and all the components have the same sequential transitions, namely those of $\mathcal{A}$.

For each fork transition of $\mathcal{A}$, $p \longrightarrow (p_1, p_2)$, $\mathcal{B}$ has fork transitions $p \longrightarrow (p_1, p_2)$ with p in $\mathcal{A}_i$, p_1 in $\mathcal{A}_{i+1}$ and p_2 in $\mathcal{A}_{i+2}$ for $i = 0, 1, 2$. Similarly, for each join transition $(q_1, q_2) \longrightarrow q$ of $\mathcal{A}$, $\mathcal{B}$ has join transitions $(q_1, q_2) \longrightarrow q$ with q in $\mathcal{A}_i$, q_1 in $\mathcal{A}_{i+1}$ and q_2 in $\mathcal{A}_{i+2}$ for $i = 0, 1, 2$.

Let $x \in SP(A)$. It is immediately verified, by induction on $|x|$, that if $\mathcal{A}$ has a run on x from p to q, then $\mathcal{B}$ has a run on x from p to q with p and q taken in $\mathcal{A}_0$ (resp. $\mathcal{A}_1$, $\mathcal{A}_2$). As a result, the *sp*-language accepted by $\mathcal{A}$ is contained in that accepted by $\mathcal{B}$.

We now prove, again by induction on $|x|$, that if $\mathcal{B}$ has a run on x from p to q, then p and q are in the same component, and $\mathcal{A}$ has a run on x from p to q. By the usual arguments, this is easily verified if $x \in A$ or if x can be written as a sequential product.

If x cannot be written as a sequential product, then $\mathcal{B}$ has a fork transition $p \longrightarrow (p_1, p_2)$, a join transition $(q_1, q_2) \longrightarrow q$, and either a run on y from p_1 to q_1 and a run on z from p_2 to q_2, or a run on y from p_1 to q_2 and a run on z from p_2 to q_1. In the first case, by induction, p_1 and q_1 are in the same component, p_2 and q_2 are in the same component, and $\mathcal{A}$ has runs on y from p_1 to q_1 and from p_2 to q_2. Therefore, if p is in $\mathcal{A}_i$, then p_1 and q_1 are in $\mathcal{A}_{i+1}$, p_2 and q_2 are in $\mathcal{A}_{i+2}$, and q must be in $\mathcal{A}_i$ by definition of $\mathcal{B}$. Also, $p \longrightarrow (p_1, p_2)$ and $(q_1, q_2) \longrightarrow q$ are transitions of $\mathcal{A}$, so $\mathcal{A}$ has a run on x from p to q.

In the second case, by the same reasoning, q_1 is in $\mathcal{A}_{i+2}$ and q_2 is in $\mathcal{A}_{i+1}$. But $\mathcal{B}$ has no join transition $(r_1, r_2) \longrightarrow r$ with r_1 in $\mathcal{A}_{i+2}$ and r_2 in $\mathcal{A}_{i+1}$, so this situation does not occur.

Thus the branching automaton $\mathcal{A}$ and the *sp*-branching automaton $\mathcal{B}$ accept the same *sp*-language, and this concludes the proof. □

References

[1] J.-M. Autebert, J. Berstel, L. Boasson. Context-free languages and pushdown automata, in *Handbook of formal languages*, vol. 1 (G. Rozenberg, A. Salomaa eds) Springer (1997).

[2] S. Bloom and Z. Ésik. Free shuffle algebras in language varieties, *Theoretical Computer Science* **163** (1996) 55–98.

[3] J.R. Büchi. *Finite automata, their algebras and grammars: Towards a theory of formal expressions* (D. Siefkes, ed.), Springer (1989).

[4] B. Courcelle. On recognizable sets and tree automata, in *Resolution of equations in algebraic structures* (H. Aït-Kaci and M. Nivat, eds.), Academic Press (1989) 93–126.

[5] B. Courcelle. The monadic second-order logic of graphs V: On closing the gap between definability and recognizability, *Theoretical Computer Science* **80** (1991) 153–202.

[6] B. Courcelle. Basic notions of universal algebra for language theory and graph grammars, *Theoretical Computer Science* **163** (1996) 1–54.

[7] R. Downey and M.R. Fellows. *Parameterized complexity*, Springer (1999).

[8] S. Eilenberg and J. Wright. Automata in general algebras, *Information and Control* **11** (1967) 52–70.

[9] Z. Ésik. Free algebras for generalized automata and language theory, in *Algebraic systems, formal languages and Computations* (M. Ito ed.), RIMS Publications **1116** (2000) 52–58.

[10] F. Gécseg and M. Steinby. Tree languages, in *Handbook of formal language theory*, vol. 3 (G. Rozenberg, A. Salomaa eds) Springer (1996).

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

[12] D. Kuske. Infinite series-parallel pomsets: logic and languages, in *ICALP 2000* (U. Montanari et al., eds.), Lect. Notes in Computer Science **1853**, Springer (2000) 648–662.

[13] D. Kuske. A model theoretic proof of Büchi-type theorems and first order logic for N-free pomsets, in *STACS 2001* (A. Ferreira, H. Reichel eds.), Lect. Notes in Computer Science **2010**, Springer (2001).

[14] D. Lapoire. Recognizability equals monadic second-order definability for sets of graphs of bounded treewidth, in *STACS 98* (M. Morvan et al, eds.), Lect. Notes in Computer Science **1373**, Springer (1998) 618–628.

[15] K. Lodaya and P. Weil. Series-parallel posets: algebra, automata and languages, in *STACS 98* (M. Morvan et al, eds.), Lect. Notes in Computer Science **1373**, Springer (1998) 555–565.

[16] K. Lodaya and P. Weil. A Kleene iteration for parallelism, in *FST & TCS 98* (V. Arvind, R. Ramanujam eds.), Lect. Notes in Computer Science **1530**, Springer (1998) 355–366.

[17] K. Lodaya and P. Weil. Series-parallel languages and the bounded-width property, *Theoretical Computer Science* **237** (2000) 347–380.

[18] J. Mezei and J. Wright. Algebraic automata and context-free sets, *Information and Control* **11** (1967) 3–29.

[19] J.W. Thatcher and J.B. Wright. Generalized finite automata with an application to a decision problem of second order logic, *Math. Syst. Theory* **2** (1968) 57–82.

[20] J. Valdes, R.E. Tarjan and E.L. Lawler. The recognition of series-parallel digraphs, *SIAM J. Comput.* **11** (1981) 298–313.