Series-parallel posets: algebra, automata and languages [1]

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

P. Weil
LIAFA, Univ. Paris-7 & CNRS
2 place Jussieu
75251 Paris Cedex 05 – France
weil@liafa.jussieu.fr

Abstract. In order to model concurrency, we extend automata theory from the usual word languages (sets of labelled linear orders) to sets of labelled series-parallel posets — or, equivalently, to sets of terms in an algebra with two product operations: sequential and parallel. We first consider languages of posets having *bounded width*, and characterize them using *depth-nilpotent* algebras. Next we introduce *series-rational expressions*, a natural generalization of the usual rational expressions, as well as a notion of *branching* automaton. We show both a Myhill-Nerode theorem and a Kleene theorem. We also look at generalizations.

Introduction

In this paper, we seek to extend automata theory from the usual languages over words (labelled linearly ordered sets) to (labelled) posets which include more information regarding concurrency. This has been done earlier; we explain how our approach differs below.

Let A be a finite nonempty alphabet. A language over A accepted by a finite automaton is said to be *regular*. Languages can also be described by expressions using concatenation, union and iteration; such languages are called *rational*. Yet another way of describing languages is by morphisms into a finite algebra; such languages are *recognizable*. The fundamental theorems of automata theory (Kleene, Myhill, Nerode) show that the regular, rational and recognizable languages coincide. (These terms are used in an interchangeable way. Since we need to distinguish them, we have fixed on this terminology. Please bear with us if you use the terms in a different way.)

Our generalization is from words to posets. We will adopt an intermediate framework, and consider the so-called series-parallel posets (*sp*-posets for short) [11, 7] as the structures accepted by our restricted automata. The advantage this yields is that we can work with an algebra in which two different operations coexist: sequential product (concatenation) and parallel product [3]. The book by Büchi [4] advocates such a fully universal algebra approach (although devoid of any attempted application to the modelling of parallelism or concurrency). Absence of communication is a lacuna in our work.

Our starting point is to introduce an expression $e_1 \| e_2$ for the parallel product. When rational expressions are extended to allow the use of the $\|$ operator,

[1] Part of this work was done while the second author was visiting the Institute of Mathematical Sciences, in Chennai.

the corresponding (poset) languages are called *series-rational*. Using morphisms into a finite algebra yields a definition of *recognizable sp*-languages. But recognizability turns out to be a broader notion than series-rational, allowing for instance the language $a^{\oplus} := \{a, a\|a, a\|a\|a, \ldots\}$.

Series-rational *sp*-languages are easily seen to have *bounded width*, that is, the "parallel-width" of the elements of these languages is uniformly bounded. The algebraic counterpart to bounded width turns out to be interesting in its own right. We have to work with *depth-nilpotent sp*-algebras, where the parallel operation is nilpotent and the nesting within parallel products forms a partial order (see Section 3 for the precise definitions). We first characterize the bounded-width languages. Putting recognizability and bounded width together leads to a Myhill-Nerode theorem.

Theorem 1 *An sp-language L has bounded width iff it is recognized by a morphism φ into a depth-nilpotent sp-algebra, such that $0 \notin \varphi(L)$.*

Theorem 2 *A bounded width sp-language is series-rational iff recognizable.*

Next we introduce a kind of branching automaton with fork and join transitions in addition to the usual transitions on a letter of the alphabet. The *sp*-languages accepted by such automata are called *regular*. Again, our machine model has to rule out unbounded parallelism to disallow languages like $a^{\oplus}$ above. By restricting the use of the fork transitions, we define the *fork-regular sp*-languages. We now have the analogue of the Kleene theorem.

Theorem 3 *An sp-language is series-rational iff it is fork-regular.*

What happens when we drop the different restrictions which we have adopted for dealing with bounded width languages? Thatcher and Wright [13] gave a way of constructing generalized rational expressions for algebras with many operations. We are able to show that:

Theorem 4 *The series-rational sp-languages are properly included in the recognizable and the regular ones, which are in turn properly included in the generalized rational ones.*

Related work. It should be clear to the reader familiar with concurrency theory that we are looking at Petri nets and related objects [12]. Our branching automata (Section 4) are a restricted form of nets, and fork-regularity corresponds to 1-safe behaviour. We do not have a characterization of the class of nets we deal with. From the process algebra community, too, attempts have been made to work with series-parallel posets [1, 2].

The algebraically informed reader may see our languages as a special kind of *trace* languages, on which there is a major body of work (see [6] for a recent summary). In traces, the alphabet is extended with an independence relation which allows commutativity of actions to represent concurrency. However, while the independence relation allows coding to represent communication between parallel processes — which is completely missing from our approach — it also prevents two actions from being serially as well as parallelly related. Our approach makes this possible, even for languages such as the solution of the equation $X = a(b\|aX)$.

A very general algebraic approach using operations on hypergraphs has been developed by Courcelle [5]. It is easy to express series-parallel languages in this approach, but automata models operating on graphs have proved hard to find. Our notion of branching automata (Section 4) provides a small step forward on the tightrope, without losing balance with algebra (Section 1).

1 *sp*-posets, *sp*-algebras and *sp*-languages

Let A be an alphabet. An *A-labelled poset* is a poset $(P, \leq)$ equipped with a labelling function $\lambda: P \to A$. This is a generalization of words, which are finite linearly ordered A-labelled sets. We are in fact considering isomorphism classes of labelled posets (*pomsets* [11]). Our posets are always finite and the underlying sets of distinct posets are assumed to be pairwise disjoint.

Let $\mathcal{P} = (P, \leq_P, \lambda_P)$ and $\mathcal{Q} = (Q, \leq_Q, \lambda_Q)$ be A-labelled posets. The *parallel product* of these posets $\mathcal{P}\|\mathcal{Q}$ is $(P \cup Q, \leq_P \cup \leq_Q, \lambda_P \cup \lambda_Q)$. The *sequential product* of the same posets $\mathcal{P} \circ \mathcal{Q}$ is $(P \cup Q, \leq_P \cup \leq_Q \cup (P \times Q), \lambda_P \cup \lambda_Q)$. It is easily verified that both products are associative, that they have the same unit (the empty set), and that the parallel product is commutative.

We say that an A-labelled poset is *series-parallel*, an *sp-poset* for short, if either it is empty, or it can be obtained from singleton A-labelled posets by using a finite number of times the operations of sequential and parallel product.

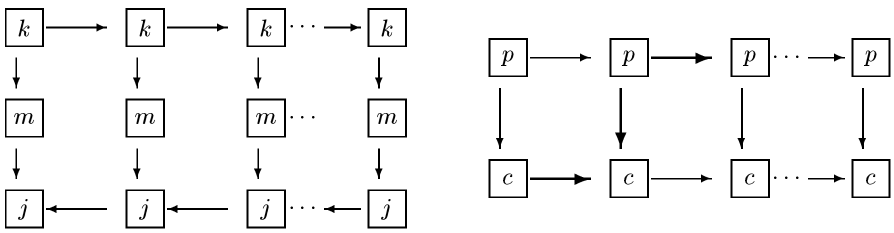

Fig. 1. Posets: series-parallel and not

Example 1.1 Figure 1 shows an example of an *sp*-poset. Indeed, let P_n be the poset in Figure 1 having n occurrences of k. Then $P_1 = \{k\} \circ \{m\} \circ \{j\}$ (usually written kmj) and $P_{n+1} = \{k\} \circ (\{m\}\|P_n) \circ \{j\}$ (written $k(m\|P_n)j$).

The right hand picture in Figure 1 shows a producer-consumer system, a simple example of message-passing. This poset is not series-parallel. The trouble can be traced to the N-shaped subposet shown using thick lines. A poset is N-*free* if it does not contain N as a subposet. That is, more formally, if it does not contain 4 distinct elements a, b, c and d among which the only order relations are $a < c$, $a < d$ and $b < d$. Observe that this property is independent of any labelling of the poset. It is well known that an A-labelled poset $(P, \leq, \lambda)$ is an *sp*-poset if and only if $(P, \leq)$ is N-free [8, 14].

***sp*-algebras.** The notion of *sp*-poset is generalized in the following fashion. An *sp-algebra* is a set S equipped with two binary operations, denoted $\circ$ and $\|$ and called respectively *sequential* and *parallel product*, such that $(S, \circ)$ is a semigroup (that is, the sequential product is associative), and such that $(S, \|)$ is a commutative semigroup. Note that no distributivity is assumed, thus $ab\|ab$ is distinct from $(a\|a)(b\|b)$. The discussion in the previous section shows that the class of *sp*-posets labelled by the alphabet A, forms an *sp*-algebra.

It is clear from this definition that *sp*-algebras form a variety of algebras (of signature $(2, 2)$), so there exists a free *sp*-algebra over each set A, denoted $SP(A)$. This free algebra was characterized by Bloom and Ésik [3].

Proposition 1.2 *$SP(A)$ is (isomorphic to) the sp-algebra of non-empty A-labelled sp-posets.*

***sp*-terms.** The elements of the free *sp*-algebra over A, namely $SP(A)$, can be identified with A-labelled *sp*-posets according to Proposition 1.2. We mildly abuse notation and call them *sp-terms* as well.

Let $x \in SP(A)$ be an *sp*-term. We say that x is a *sequential term* if it cannot be written as a parallel product $x = y\|z$. We say that x is a *parallel term* if it cannot be written as a sequential product $x = yz$ $(y, z \in SP(A))$.

The semigroup $(SP(A), \circ)$ is freely generated by the parallel terms, and the commutative semigroup $(SP(A), \|)$ is freely generated by the sequential terms. So, if $x \in SP(A)$ is an *sp*-term, then x admits a factorization of the form $x = x_1\| \ldots \|x_n$, where $n \geq 1$ and each x_i is a sequential term, and this factorization is unique up to the order of the factors. The *sp*-term x also admits a unique factorization of the form $x = x_1 \circ \ldots \circ x_n$, where $n \geq 1$ and each x_i is a parallel term. These factorizations are called, respectively, the *parallel* and the *sequential* factorizations of x.

Example 1.3 Let us follow up on the *sp*-posets of Example 1.1 above. For $n = 3$, the *sp*-poset P_n can be written $P_3 = k\Big(m\|\Big(k\big(m\|(kmj)\big)j\Big)\Big)j$. For each $n \geq 1$, P_{n+1} is a sequential term, whose factorization has length 3: $P_{n+1} = k\circ(m\|P_n)\circ j$. The first and last factor are letters, and the second one is a parallel term whose factorization has length 2.

The width $\mathrm{wd}(x)$ of an *sp*-term $x \in SP(A)$ is inductively defined as follows: $\mathrm{wd}(a) = 1$ for $a \in A$; $\mathrm{wd}(x_1x_2) = \max\{\mathrm{wd}(x_1), \mathrm{wd}(x_2)\}$; $\mathrm{wd}(x_1\|x_2) = \mathrm{wd}(x_1) + \mathrm{wd}(x_2)$. The width of the *sp*-term P_3 in the example above is 3. Intuitively, the width of an *sp*-term x is the number of processors needed to realize x.

***sp*-languages.** An *sp-language* is a subset of some free *sp*-algebra $SP(A)^1$ (with an identity element for both products, denoted 1). The family of *series-rational* *sp*-languages over the alphabet A is defined to be the least class of *sp*-languages containing the finite *sp*-languages and closed under union, the sequential and parallel products and sequential iteration (Kleene star). Although our results pertain only to the series-rational languages, we define the *series-parallel-rational* languages as those also closed under parallel iteration $L^{\oplus}$, defined to be the set $\{t_1\|t_2\|\ldots\|t_n \mid n \geq 1,\ t_i \in L, 1 \leq i \leq n\}$.

We say that an *sp*-language $L \subseteq SP(A)^1$ has *bounded width* if there exists an integer k such that, for each element $x \in L$ we have $\mathrm{wd}(x) \leq k$.

Fact 1.4 *Every series-rational sp-language has bounded width.*

There are bounded-width *sp*-languages which are not series-rational, e.g. non-rational languages contained in the free monoid A^*, since such languages have width 1. From Fact 1.4 we also know that the complement of a series-rational *sp*-language is not series-rational.

2 Recognizable *sp*-languages

As is usual with algebras, we define a *morphism* from an *sp*-algebra S to an *sp*-algebra T to be a mapping which preserves the two products. We say that an *sp*-language $L \subseteq SP(A)$ is *recognized* by a morphism φ from $SP(A)$ into an *sp*-algebra S if $L = \varphi^{-1}\varphi(L)$. In that case, we also say that L is *recognized* by the *sp*-algebra S. An *sp*-language L is *recognizable* if it is recognized by a *finite* *sp*-algebra. If our *sp*-languages may contain 1, then we must use morphisms from $SP(A)^1$ into (finite) *sp*-algebras with identity.

Example 2.1 Let $a \in A$. Then the *sp*-language $\{a\}$ is recognized by the 2-element *sp*-algebra $S = \{x, 0\}$ given by $x \circ x = x\|x = 0$. (Consider the morphism from $SP(A)$ into S which maps a to x and the other letters of A to 0.)

Let $a^{\oplus}$ be the *sp*-language consisting of all the parallel powers of a. It is recognized by the 2-element *sp*-algebra $\{x, 0\}$ given by $x\|x = x$ and $xx = 0$. One needs to consider the morphism from $SP(A)$ into S mapping a to x and the other letters to 0.

This example can easily be generalized. Note that the free semigroup A^+ and the free commutative semigroup $A^{\oplus}$ are naturally embedded in $SP(A)$ (the latter as the semigroup generated by A under the parallel product). Then an *sp*-language L contained in A^+ (resp. $A^{\oplus}$) is recognizable if and only if L is a recognizable subset of A^+ (resp. $A^{\oplus}$).

The *sp*-language $\{a^n b^n \mid n \geq 1\}$ is not recognizable, since it is not recognizable in A^+. Similarly, the *sp*-language $(a\|b)^{\oplus}$, which consists of all the parallel powers of $a\|b$, is not recognizable. Indeed it is not recognizable in $A^{\oplus}$ (its inverse image in A^+, the set of all words having as many a as b, is not recognizable).

Let $L \subseteq SP(A)$ be an *sp*-language. If $\xi \notin A$, we define a *ξ-term* to be a term in $SP(A \cup \{\xi\})$ which has exactly one occurrence of ξ. Such terms represent "contexts". If c is a ξ-term and t is a term, then $c[\xi, t]$ denotes the result of the substitution of t for the occurrence of ξ in c.

The *syntactic congruence* of L is defined as follows. Let $u, v \in SP(A)$. Intuitively, we say that u is $\sim_L$-equivalent to v if, for each *sp*-term $t \in L$ such that u (resp. v) is a subterm of t, replacing u by v (resp. v by u) in t yields another element of L. More precisely, we let $u \sim_L v$ if, for each ξ-term c (where $\xi \notin A$), we have $c[\xi, u] \in L \Longleftrightarrow c[\xi, v] \in L$.

The quotient *sp*-algebra $\mathrm{Synt}(L) = SP(A)/\sim_L$ is called the *syntactic sp-algebra* of L and the projection morphism $\mu_L : SP(A) \to \mathrm{Synt}(L)$ is called the *syntactic morphism* of L. This morphism recognizes L.

As usual, an absorbing element for both products is called a *zero*. Such an element is unique if it exists, and is written 0.

Lemma 2.2 *Let L be an sp-language and let s be an sp-term which is not a subterm of any element of L. Then* $\mathrm{Synt}(L)$ *has a zero and* $\mu_L(s) = 0$.

3 Bounded-width *sp*-languages

There are recognizable *sp*-languages which are not series-rational. For instance, the series-parallel-rational *sp*-language $a^{\oplus}$ is recognizable by Example 2.1, but not series-rational by Fact 1.4. To rule out languages like $a^{\oplus}$, we give an algebraic characterization of the *sp*-languages having bounded width.

Consider an *sp*-algebra S. If $s, t \in S$, we let $s \prec t$ if and only if $s = u\big(p \| (qtr)\big)v$ for some $u, p, q, r, v \in S^1$ such that $p \| (qtr) \neq qtr$. The relation $<_{\mathrm{n}}$ ($\leq_{\mathrm{n}}$) is defined to be the (reflexive) transitive closure of $\prec$, and it is called the *parallel nesting relation*. $\leq_{\mathrm{n}}$ is a quasi-order on S, but need not be a partial order.

Example 3.1 Let $x, y \in SP(A)$ be *sp*-terms. Then $s <_{\mathrm{n}} t$ if and only if t is a subterm of s and at least one context of t in s (a ξ-term) is not a sequential term having ξ as a factor in its sequential decomposition. In particular, $\leq_{\mathrm{n}}$ is a partial order on $SP(A)$, and $s <_{\mathrm{n}} t$ implies $\mathrm{wd}(s) > \mathrm{wd}(t)$.

We will be concerned with *sp*-algebras S in which, for each $s \in S$, there exists an integer n such that each $<_{\mathrm{n}}$-chain with least element s has length at most n. Such algebras are called *depth-graded*. Example 3.1 shows that this is the case if $S = SP(A)$, the free *sp*-algebra.

In addition, if S is a depth-graded *sp*-algebra, there is a notion of *parallel nesting depth*, or simply *depth*, for the elements of S:

$$\mathrm{dp}(s) = \begin{cases} 1 & \text{if } s \text{ is } \leq_{\mathrm{n}}\text{-maximal,} \\ 1 + \max\{\mathrm{dp}(t) \mid s <_{\mathrm{n}} t\} & \text{otherwise.} \end{cases}$$

In particular, the depth of s is the maximum cardinality of a $<_{\mathrm{n}}$-chain with least element s. Note that the maximal $<_{\mathrm{n}}$-chains are $\prec$-chains.

If L is a subset of a depth-graded *sp*-algebra S, we let $\mathrm{dp}(L) = \sup_{x \in L} \mathrm{dp}(x)$. If $\mathrm{dp}(L) < \infty$, we say that L has *bounded depth*.

It is easily verified (see Example 3.1) that if $x \in SP(A)$ is an *sp*-term, then $\mathrm{dp}(x) \leq \mathrm{wd}(x)$. However, we do not necessarily have equality between the width and the depth of an *sp*-term: if $u = a(a\|a)$ and if x is the parallel product of n copies of u, then one verifies that $\mathrm{wd}(x) = 2n$ and $\mathrm{dp}(x) = n+1$. Yet very wide *sp*-terms must be very deep, as is shown by:

Proposition 3.2 *An sp-language has bounded width iff it has bounded depth.*

Depth-nilpotent *sp*-algebras

We say that an *sp*-algebra S has *bounded depth* if there exists an integer d such that each $\prec$-chain is of length at most d. If S is a bounded-depth *sp*-algebra, then S is trivially depth-graded, $\mathrm{dp}(S) < \infty$ and any element of maximal depth is $\leq_{\mathrm{n}}$-minimal. The converse also holds:

Lemma 3.3 *An sp-algebra S has bounded depth if and only if it is depth-graded and it has a $\leq_{\mathrm{n}}$-minimal element.*

Recall that a semigroup is said to be *nilpotent* if it has a zero and there exists an integer $c \geq 1$ such that any product of c or more elements is equal to 0. In that case, it has a zero which is its only idempotent [10]. Now we say that an *sp*-algebra S is *depth-nilpotent* if it has bounded depth and 0 is the only idempotent element of S for the parallel product. We can show that a bounded depth *sp*-algebra S is depth-nilpotent if and only if the semigroup $(S, \|)$ is nilpotent. This leads to the following property, and our first main theorem.

Lemma 3.4 *Let $\varphi: T \to S$ be a morphism between sp-algebras, and assume S is depth-nilpotent. If $t, t' \in T$ and $t \prec t'$, then either $\varphi(t) = 0$ or $\varphi(t) \prec \varphi(t')$.*

Theorem 1 *Let L be a sp-language. The following properties are equivalent.*

(1) *L has bounded width;*
(2) $\mathrm{Synt}(L)$ *is depth-nilpotent and $0 \notin \mu_L(L)$;*
(3) *L is recognized by a morphism $\varphi: SP(A) \to S$ into a depth-nilpotent sp-algebra S, such that $0 \notin \varphi(L)$.*

Since recognizable *sp*-languages have finite syntactic *sp*-algebras, Theorem 1 immediately implies the following result.

Corollary 3.5 *Let L be a sp-language. The following properties are equivalent.*

(1) *L is recognizable and it has bounded width;*
(2) $\mathrm{Synt}(L)$ *is finite depth-nilpotent and $0 \notin \mu_L(L)$;*
(3) *L is recognized by a morphism $\varphi: SP(A) \to S$ into a finite depth-nilpotent sp-algebra S, such that $0 \notin \varphi(L)$.*

We complete this section by connecting bounded-width, recognizability and series-rationality. As an aside, note that finite depth-nilpotent algebras can recognize (using the zero) complements of recognizable bounded-width languages as well (e.g. $SP(A)$). It can be shown that finite depth-nilpotent *sp*-algebras form a pseudovariety, and using the next theorem, that series-rational languages form a positive variety (in the sense of [10]).

Theorem 2 *An sp-language is series-rational if and only if it is recognizable and bounded-width.*

Proof sketch. By adapting the classical proof for languages in the free semigroup, we prove that recognizable languages are closed under the two products, boolean operations and sequential iteration. The forward direction then follows from Fact 1.4. To prove the converse, we use the algebraic characterization of bounded-width recognizable *sp*-languages established in Corollary 3.5 and a McNaughton-Yamada-like construction. □

4 Branching automata and regular *sp*-languages

In order to provide an automaton characterization of series-rational *sp*-languages, we introduce a new kind of automata.

A *branching automaton* over A is a tuple $\mathcal{A} = (Q, T_{seq}, T_{fork}, T_{join}, I, F)$ where Q is the (finite) set of states, I and F are subsets of Q, respectively the set of initial and of final states, $T_{seq} \subseteq Q \times A \times Q$ is the set of sequential transitions, $T_{fork} \subseteq Q \times \mathcal{P}_{ns}(Q)$ and $T_{join} \subseteq \mathcal{P}_{ns}(Q) \times Q$ are respectively the sets of fork and join transitions. Here $\mathcal{P}_{ns}(Q)$ ("nonempty and nonsingleton") stands for subsets of Q of cardinality at least 2.

An element $(q, a, q') \in T_{seq}$ is said to be an a-labelled transition from q to q'. As usual, we write $q \xrightarrow{a} q'$. An element $(q, \{q_1, \ldots, q_n\})$ of T_{fork} is said to be a *fork transition of arity* n. We denote it by $q \to \{q_1, \ldots, q_n\}$. Similarly, an element $(\{q_1, \ldots, q_n\}, q)$ of T_{join}, written $\{q_1, \ldots, q_n\} \to q$, is said to be a *join transition of arity* n.

The existence of a *run* of $\mathcal{A}$ on the *sp*-poset t from state p to state q is inductively defined as follows: there is a run on 1 from p to q only if $p = q$, and there is a run on letter a from p to q if $p \xrightarrow{a} q$. If a poset t has sequential factorization $t_1 \ldots t_n$ ($n \geq 2$), there is a run on t from p to q if there exist states $p = p_0, p_1, \ldots p_n = q$ and there exist runs on t_m from p_{m-1} to p_m, for each $1 \leq m \leq n$.

Suppose now that a poset t has parallel factorization $t_1 \| \ldots \| t_n$ ($n \geq 2$). There is a run on t from p to q if there is a fork at p, sub-runs for the factors and then a matching join ending at q. However, the automaton can do the forking in levels. We formalize this as follows:

Let $\pi = \{r_1, \ldots, r_s\}$ be a subset of $\{1, \ldots, n\}$. By t_π we mean the parallel term $t_{r_1} \| \ldots \| t_{r_s}$. Now we say there is a run on t from p to q if there exists a partition $\{\pi_1, \ldots, \pi_m\}$ of $\{1, \ldots, n\}$ and there are states p_ℓ, q_ℓ, for $1 \leq \ell \leq m$ such that there is a fork $k = p \to \{p_1, \ldots, p_m\}$, there are runs on t_{π_ℓ} from p_ℓ to q_ℓ, $1 \leq \ell \leq m$, and there is a matching join $j = \{q_1, \ldots, q_m\} \to q$. Figure 2

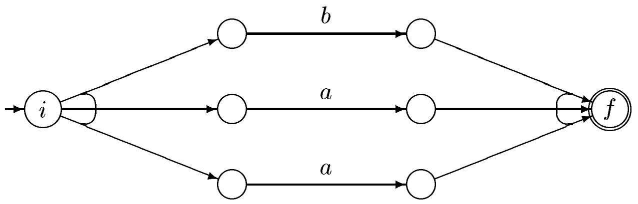

Fig. 2. Automaton accepting $a \| a \| b$

shows an automaton with a run on $a\|a\|b$ from state i to state f. The partition of $\{1, 2, 3\}$ here is simple: it just has three singletons $\{\{1\}, \{2\}, \{3\}\}$.

Getting back to the previous definition, we say that the states and the transitions forming the sub-runs on the t_{π_ℓ} are *nested within* the fork k in the run on t. In the example in Figure 2, all states except i and f are within the fork in any run. There are no nested forks.

Finally, a branching automaton $\mathcal{A}$ *accepts* the *sp*-poset t if there exists a run of $\mathcal{A}$ on t from an initial state to a final state. An *sp*-language L is *regular* if it is the set of *sp*-posets accepted by a finite branching automaton.

As in the case of recognizable languages (Section 2), the notion of regularity is too general. It is easy to find a branching automaton accepting the non-bounded-width language $a^{\oplus}$. Further, Figure 3 shows that even the non-

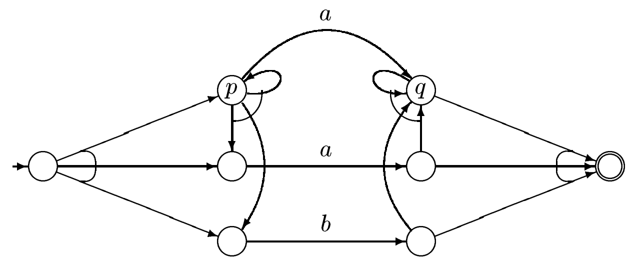

Fig. 3. An automaton which is not fork-acyclic

recognizable language $a\|(a\|b)^{\oplus}$ (parallel products of a and b having one more a than b) is accepted by an automaton. Note the looping fork at state p and the looping join at state q in Figure 3. It is the presence of a loop from inside one branch of a fork to another (or to outside the fork) that allows unbounded parallel-width terms to be accepted.

A run ρ of $\mathcal{A}$ on t is *fork-acyclic* if in the induced sub-run of ρ on any parallel sub-term of t with outermost fork k, k does not occur nested within itself. An *sp*-language is *fork-regular* if it is the set of *sp*-posets of the fork-acyclic runs from an initial state to a final state of a finite branching automaton. If all runs of an automaton are fork-acyclic, we call the automaton *fork-acyclic*. The automaton in Figure 2 is fork-acyclic. We can now prove our next main theorem.

Theorem 3 *An sp-language is series-rational iff it is accepted by a fork-acyclic automaton, iff it is fork-regular.*

Proof sketch. By extending the usual inductive construction for rational languages, we can prove that series-rational languages are accepted by finite fork-acyclic automata. To prove the converse, we inductively build up expressions for languages using a set of forks. The base case, when no forks are used, is provided by the usual McNaughton-Yamada construction for regular languages. $\square$

5 Generalizations

All three notions which we have used and related so far in the context of bounded-width languages, namely recognizability, series-rationality and fork-regularity, admit natural generalizations.

For recognizability, dropping the depth-nilpotency condition, we get the recognizable *sp*-languages, recognized by a finite *sp*-algebra. As we saw, $a^{\oplus}$ is a recognizable language which is not bounded-width. Dropping the fork-acyclicity condition, our automata accept the regular *sp*-languages, which include the language $a^{\oplus}$ as well as non-recognizable languages like $a\|(a\|b)^{\oplus}$ (Figure 3).

We borrow our generalization of series-rationality from [4] and [13]. Let B be an alphabet containing A. The generalized *sp*-rational expressions over B are defined by closing the letters of B using this rule:
If e_1 and e_2 are generalized rational expressions, and $\xi \in B$, then $e_1 \cup e_2$, $e_1 \circ e_2$, $e_1 \| e_2$, $e_1 \circ_\xi e_2$ and ${e_1}^{*\xi}$ are generalized rational expressions.
With each generalized rational expression, we can associate an *sp*-language in $SP(B)$. We indicate how this is done for the expressions using the ξ variables.

If e_1 and e_2 are generalized rational expressions, then $L(e_1 \circ_\xi e_2)$ is obtained from the elements of $L(e_2)$ by substituting an element of $L(e_1)$ for each occurrence of ξ (different occurrences may be replaced by different elements). $L({e_1}^{*\xi}) = L(e_1)^{*\xi}$, where for any *sp*-language U, we let $U^{*\xi} = \bigcup_{i \geq 0} U_i^\xi$, with $U_0^\xi = \{\xi\}$ and $U_{i+1}^\xi = (U_0^\xi \cup \cdots \cup U_i^\xi) \circ_\xi U$.

The regular as well as the generalized rational *sp*-languages are closed under both sequential and parallel iteration. ($L^* = 1 \circ_\xi (\xi L)^{*\xi}$ and $L^{\oplus} = L \circ_\xi (\xi \| L)^{*\xi}$.) The regular languages properly include the series-parallel-rational languages, e.g. $1 \circ_\xi (a \| (b\xi))^{*\xi}$ is a separating language. The generalized rational languages include the context-free languages in A^* (e.g. $1 \circ_\xi (a\xi b)^{*\xi} = \{a^n b^n \mid n \geq 0\}$). Using a result of Thatcher and Wright [13], and pushing the McNaughton-Yamada construction of Theorem 3 one step further, we show:

Theorem 4 *The series-rational sp-languages are properly included in the recognizable and the regular ones, which are in turn properly included in the generalized rational ones.*

$$\begin{array}{ccccc}
\text{series-rational} & \subset & \text{depth-nilpotent recognizable} & \subset & \text{recognizable} \\
(a\|b)^* & & SP(A) & & a^{\oplus} \\
\cap & & & & \cap \\
\text{series-parallel-rational} & \subset & \text{regular} & \subset & \text{generalized rational} \\
(a\|b)^{\oplus} & & 1\circ_{\xi}(a\|b\xi)^{*\xi} & & 1\circ_{\xi}(a\xi b)^{*\xi}
\end{array}$$

Theorem 2 and Theorem 3 showed that the three formalisms of algebraic recognizability, rational expressions and automaton acceptance can be made to coincide in the context of bounded-width *sp*-languages. When generalized, these formalisms open up a hierarchy. We are left with the question of characterizing the smaller subclasses within the generalized rational *sp*-languages. We conjecture that the recognizable *sp*-languages are included in the regular ones.

Our work can be generalized to algebras with another operation $+$ representing choice. Boudol and Castellani [2] have shown that the free algebra (modulo associativity of $\circ$ and associativity and commutativity of $\|$ and $+$) is characterized by a class of labelled event structures satisfying an "X-property" (going beyond N-freeness). From an example of Milner [9, Example 3], it appears that a syntax of right-linear ξ-expressions has to be used.

References

1. L. Aceto. Full abstraction for series-parallel pomsets, in *Proc. TAPSOFT 91*, *LNCS* **493**, Springer (1991) 1–40.
2. G. Boudol and I. Castellani. Concurrency and atomicity, *TCS* **59** (1988) 25–84.
3. S. Bloom and Z. Ésik. Free shuffle algebras in language varieties, *TCS* **163** (1996) 55–98.
4. J.R. Büchi. *Finite automata, their algebras and grammars: Towards a theory of formal expressions* (D. Siefkes, ed.), Springer (1989).
5. B. Courcelle. Graph rewriting: an algebraic and logical approach, in *Handbook of Theoretical Computer Science* **B** (J. van Leeuwen, ed.), Elsevier (1990) 193–242.
6. V. Diekert and G. Rozenberg. *The book of traces*, World Scientific (1995).
7. J.L. Gischer. The equational theory of pomsets, *TCS* **61** (1988) 199–224.
8. J. Grabowski. On partial languages, *Fund. Inform.* **IV** (1981) 427–498.
9. R. Milner. A complete inference system for a class of regular behaviours, *JCSS* **28** (1984) 439–466.
10. J.-E. Pin. Syntactic semigroups, in *Handbook of Formal Language Theory* **1** (G. Rozenberg and A. Salomaa, eds.), Springer (1997) 679–746.
11. V. Pratt. Modelling concurrency with partial orders, *IJPP* **15**(1) (1986) 33–71.
12. W. Reisig. *Petri nets, an introduction*, Springer (1985).
13. 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.
14. J. Valdes, R.E. Tarjan and E.L. Lawler. The recognition of series-parallel digraphs, *SIAM J. Comput.* **11**(2) (1981) 298–313.