Logic and branching automata

Nicolas Bedon

LITIS (EA CNRS 4108) – Université de Rouen – France
Nicolas.Bedon@univ-rouen.fr

Abstract. The first result presented in this paper is the closure under complementation of the class of languages of finite N-free posets recognized by branching automata. Relying on this, we propose a logic, named *Presburger-MSO* or *P-MSO* for short, precisely as expressive as branching automata. The P-MSO theory of the class of all finite N-free posets is decidable.

Keywords: N-free posets, series-parallel posets, sp-rational languages, automata, commutative monoids, monadic second-order logic, Presburger logic.

1 Introduction

In computer science, if Kleene automata, or equivalently, rational expressions or finite monoids, are thought of as models of sequential programs, then introducing commutativity allows access to models of programs with permutation of instructions, or to concurrent programming. Among the formal tools for the study of commutativity in programs, let us mention for example Mazurkiewicz's traces, integer vector automata or commutative monoids.

In this paper, we are interested in another approach: the *branching automata*, introduced by Lodaya and Weil [13–16]. Branching automata are a generalisation of Kleene automata for languages of words to languages of finite N-free posets. This class of automata takes into account both sequentiality and the fork-join notion of parallelism, in which an execution flow f that splits into $f_1, \ldots, f_n$ concurrent execution flows, joins $f_1, \ldots, f_n$ before it continues. Divide-and-conquer concurrent programming naturally uses this fork-join principle. Lodaya and Weil generalized several important results of the theory of Kleene automata to branching automata, for example, a notion of rational expression with the same expressivity as branching automata. They also investigated the question of the algebraic counterpart of branching automata: the sp-algebras are sets equipped with two different associative products, one of them being also commutative. Contrary to the theory of Kleene automata, branching automata do not coincide any more with finite sp-algebras, and it is not known if the class of languages recognized by branching automata is closed under complementation.

An interesting particular case is the bounded-width rational languages [15], where the cardinality of the antichains of the posets of languages are bounded by an integer n. They correspond to fork-join models of concurrent programs with n

as the upper bound of the number of execution flows (n is the number of physical processors). Bounded-width rational languages have a natural characterisation in rational expressions, branching automata, and sp-algebras. Taking into account those characterisations, the expressiveness of branching automata corresponds exactly to the finite sp-algebras. Furthermore, Kuske [12] proved that in this case, branching automata coincide also with monadic second-order logic, as it is the case for the rational languages of finite words. As in the general case monadic-second order logic is less expressive than branching automata, the question of an equivalent logic was left open.

This paper contains two new results:

1. first, the closure under complementation of the class of rational languages (Theorem 3);
2. second, we define a logic, named *P-MSO logic*, which basically is monadic second-order logic enriched with Presburger arithmetic, that is exactly as expressive as branching automata (Theorem 6).

The paper is organized as follows. Section 2 recalls basic definitions on posets. Section 3 is devoted to branching automata and rational expressions. Finally P-MSO is presented in Section 4.

All the proofs of the results of this paper are effective. As a consequence, the P-MSO theory of the class of finite N-free posets is decidable.

2 Notation and basic definitions

Let E be a set. We denote by $\mathcal{P}(E)$, $\mathcal{P}^+(E)$ and $\mathcal{M}^{>1}(E)$ respectively the set of subsets of E, the set of non-empty subsets of E and the set of multi-subsets of E with at least two elements. For any integer n, the group of permutations of $\{1, \ldots, n\}$ is denoted by S_n. The cardinality of E is denoted by $|E|$.

A *poset* $(P, <_P)$ is composed of a set P equipped with a partial ordering $<_P$. In this paper we consider only finite posets. For simplicity, by *poset* we always mean *finite* poset. A *chain* of length n in P is a sequence $p_1 <_P \cdots <_P p_n$ of elements of P. An *antichain* E in P is a set of elements of P mutually incomparable for $<_P$. The *width* of P is the size of a maximal antichain of P. An *alphabet* is a finite set whose elements are called *letters*. A poset $(P, <_P, \rho)$ *labelled* by A is composed of a poset $(P, <_P)$ and a map $\rho : P \to A$ which associates a letter A with any element of P. Observe that the posets of width 1 labelled by A correspond precisely to the usual finite words: finite totally ordered sequences of letters. Throughout this paper, we use labelled posets as a generalisation of words. In order to lighten the notation we write P for $(P, <_P, \rho)$ when no confusion is possible. The unique empty poset is denoted by ϵ.

Let $(P, <_P, \rho_P)$ and $(Q, <_Q, \rho_Q)$ be two disjoint posets labelled respectively by the alphabets A and A'. The *parallel product* of P and Q, denoted $P \parallel Q$, is the set $P \cup Q$ equipped with the orderings on P and Q such that the elements of P and Q are incomparable, and labelled by $A \cup A'$ by preservation of the labels from P and Q. It is defined as $(P \cup Q, <, \rho)$ where $\rho(x) = \rho_P(x)$ if $x \in P$,

$\rho(x) = \rho_Q(x)$ if $x \in Q$, and $x < y$ if and only if ($x, y \in P$ and $x <_P y$) or ($x, y \in Q$ and $x <_Q y$).

The *sequential product* of P and Q, denoted by $P \cdot Q$ or PQ for simplicity, is the poset $(P \cup Q, <, \rho)$ labelled by $A \cup A'$, such that $\rho(x) = \rho_P(x)$ if $x \in P$, $\rho(x) = \rho_Q(x)$ if $x \in Q$, and $x < y$ if and only if one of the following conditions is true:

- $x \in P$, $y \in P$ and $x <_P y$;
- $x \in Q$, $y \in Q$ and $x <_Q y$;
- $x \in P$ and $y \in Q$

Observe that the parallel product is an associative and commutative operation on posets, whereas the sequential product does not commute (but is associative). The parallel and sequential products can be generalized to finite sequences of posets. Let $(P_i)_{i \le n}$ be a sequence of posets. We denote by $\prod_{i \le n} P_i = P_0 \cdot \dots \cdot P_n$ and $\|_{i \le n} P_i = P_0 \parallel \dots \parallel P_n$.

The class of *series-parallel* posets, denoted SP, is defined as the smallest set containing the posets with zero and one element and closed under finite parallel and sequential product. It is well known that this class corresponds precisely to the class of N-free posets [22, 23], in which the exact ordering relation between any four elements x_1, x_2, x_3, x_4 cannot be $x_1 < x_2$, $x_3 < x_2$ and $x_3 < x_4$. The class of series-parallel posets over an alphabet A is denoted $SP(A)$ (or $SP^+(A)$ when the empty poset is not considered).

A *block* B of a poset $(P, <)$ is a nonempty subset of P such that, if $b, b' \in B$ such that $b < b'$, then for all elements of $p \in P$, if $b \le p \le b'$ then $p \in B$. We say that B is *connected* if, for any different and incomparable $b, b' \in B$ there exists $b'' \in B$ such that $b, b' \le b''$ or $b'' \le b, b'$. A subset G of P is *good* if, for all $p \in P$, if p is comparable to an element of G and incomparable to another, then $p \in G$.

3 Rational languages and automata

A *language* over an alphabet A is a subset of $SP(A)$. The sequential and parallel product of labelled posets can naturally be extended to languages. If $L_1, L_2 \subseteq SP(A)$, then $L_1 \cdot L_2 = \{P_1 \cdot P_2 \mid P_1 \in L_1, P_2 \in L_2\}$ and $L_1 \parallel L_2 = \{P_1 \parallel P_2 \mid P_1 \in L_1, P_2 \in L_2\}$.

3.1 Rational languages

Let A and B be two alphabets and let $P \in SP(A)$, $L \subseteq SP(B)$ and $\xi \in A$. We define the language $L \circ_\xi P$ of posets labelled by $A \cup B$ by substituting non-uniformly in P each element labelled by ξ by a labelled poset of L. This substitution $L \circ_\xi$ is the homomorphism from $(SP(A), \|, \cdot)$ into the powerset algebra $(\mathcal{P}(SP(A \cup B)), \|, \cdot)$ with $a \mapsto \{a\}$ for all $a \in A$, $a \neq \xi$, and $\xi \mapsto L$. It can be easily extended from labelled posets to languages of posets. Using this, we define the substitution and the iterated substitution on languages. By the way the usual Kleene rational operations [11] are recalled. Let L and L' be languages of $SP(A)$:

$$L \circ_\xi L' = \bigcup_{P\in L'} L \circ_\xi P$$
$$L^{*\xi} = \bigcup_{i\in\mathbb{N}} L^{i\xi} \text{ with } L^{0\xi} = \{\xi\} \text{ and } L^{(i+1)\xi} = (\bigcup_{j\leq i} L^{j\xi}) \circ_\xi L$$
$$L^* = \{\prod_{i<n} P_i : n \in \mathbb{N}, P_i \in L\}$$

A language $L \subseteq SP(A)$ is *rational* if it is empty, or obtained from the letters of the alphabet A using usual rational operators : finite union $\cup$, finite concatenation $\cdot$, and finite iteration *, and using also the finite parallel product $\|$, substitution $\circ_\xi$ and iterated substitution $^{*\xi}$, provided that in $L^{*\xi}$ any element labelled by ξ in a labelled poset $P \in L$ is incomparable with another element of P. This latter condition excludes from the rational languages those of the form $(a\xi b)^{*\xi} = \{a^n\xi b^n : n \in \mathbb{N}\}$, for example, which are known to be not Kleene rational. Observe also that the usual Kleene rational languages are a particular case of the rational languages defined above, in which the operators $\|$, $\circ_\xi$ and $^{*\xi}$ are not used.

Example 1. Let $A = \{a, b, c\}$ and $L = c \circ_\xi (a \parallel (b\xi))^{*\xi}$. Then L is the smallest language containing c and such that if $p \in L$, then $a \parallel (bx) \in L$.

$$L = \{c, a \parallel (bc), a \parallel (b(a \parallel (bc))), \dots\}$$

Let L be a language where the letter ξ is not used. In order to lighten the notation we use the following abreviation:

$$L^{\circledast} = \{\epsilon\} \circ_\xi (L \parallel \xi)^{*\xi} = \{\|_{i<n} P_i : n \in \mathbb{N}, P_i \in L\}$$

L^* is the sequential iteration of L whereas $L^{\circledast}$ is its parallel iteration.

A language L is $\|$*-rational* if it is rational without using the operators $\cdot$, $\circ_\xi$, * and $^{*\xi}$ (but $^{\circledast}$ is allowed).

Remark 1. Any rational language L which does not make use of sequentiality (i.e. $PP' \notin L$ for all $P, P' \in SP^+(A)$) is $\|$-rational.

A subset L of $A^{\circledast}$ is *linear* if it has the form

$$L = a_1 \parallel \cdots \parallel a_k \parallel \left(\cup_{i\in I}(a_{i,1} \parallel \cdots \parallel a_{i,k_i})\right)^{\circledast}$$

where the a_i and $a_{i,j}$ are elements of A and I is a finite set. It is *semi-linear* if it is a finite union of linear sets. We refer to [5] for a proof of the following result:

Theorem 1. *Let A be an alphabet and $L \subseteq A^{\circledast}$. Then L is $\|$-rational if and only if it is semi-linear.*

3.2 Branching automata

Branching automata are a generalisation of usual Kleene automata. They were introduced by Lodaya and Weil [13–15].

A *branching automaton* (or just *automaton* for short) over an alphabet A is a tuple $\mathcal{A} = (Q, A, E, I, F)$ where Q is a finite set of states, $I \subseteq Q$ is the set of *initial states*, $F \subseteq Q$ the set of *final states*, and E is the set of *transitions* of $\mathcal{A}$. The set of transitions of E is partitioned into $E = (E_{seq}, E_{fork}, E_{join})$, according to the different kinds of transitions:

- $E_{seq} \subseteq (Q \times A \times Q)$ contains the *sequential* transitions, which are usual transitions of Kleene automata;
- $E_{fork} \subseteq Q \times \mathcal{M}^{>1}(Q)$ and $E_{join} \subseteq \mathcal{M}^{>1}(Q) \times Q$ are respectively the sets of *fork* and *join* transitions.

Sequential transitions $(p, a, q) \in Q \times A \times Q$ are sometimes denoted by $p \stackrel{a}{\to} q$.

We now turn to the definition of paths in automata. The definition we use in this paper is different, but equivalent to, the one of Lodaya and Weil [13–16]. Paths in automata are posets labelled by transitions. A *path* γ from a state p to a state q is either the empty poset (in this case $p = q$), or a non-empty poset labelled by transitions, with a unique minimum and a unique maximum element. The minimum element of γ is mapped either to a sequential transition of the form (p, a, r) for some $a \in A$ and $r \in Q$ or to a fork transition of the form (p, R) for some $R \in \mathcal{M}^{>1}(Q)$. Symmetrically, the maximum element of γ is mapped either to a sequential transition of the form (r', a, q) for some $a \in A$ and $r' \in Q$ or to a join transition of the form (R', q) for some $R' \in \mathcal{M}^{>1}(Q)$. The states p and q are respectively called *source* (or *origin*) and *destination* of γ. Two paths γ and γ' are *consecutive* if the destination of γ is also the source of γ'. Formally, the paths γ labelled by $P \in SP(A)$ in $\mathcal{A}$ are defined by induction on the structure of P:

- the empty poset ϵ is a path from p to p, labelled by $\epsilon \in SP(A)$, for all $p \in Q$;
- for any transition $t = (p, a, q)$, then t is a path from p to q, labelled by a;
- for any finite set of paths $\{\gamma_0, \ldots, \gamma_k\}$ (with $k > 1$) respectively labelled by $P_0, \ldots, P_k$, from $p_0, \ldots, p_k$ to $q_0, \ldots, q_k$, if $t = (p, \{p_0, \ldots, p_k\})$ is a fork transition and $t' = (\{q_0, \ldots, q_k\}, q)$ a join transition, then $\gamma = t(\|_{j \leq k} \gamma_j)t'$ is a path from p to q and labelled by $\|_{j \leq k} P_j$;
- for any non-empty finite sequence $\gamma_0, \ldots, \gamma_k$ of consecutive paths respectively labelled by $P_0, \ldots, P_k$, then $\prod_{j<k+1} \gamma_j$ is a path labelled by $\prod_{j<k+1} P_j$ from the source of γ_0 to the destination of γ_k;

Observe that paths are labelled posets of three different forms: ϵ, t or tPt' for some transitions t, t' and some labelled poset P. In an automaton $\mathcal{A}$, a path γ from p to q labelled by $P \in SP(A)$ is denoted by $\gamma : p \overset{P}{\underset{\mathcal{A}}{\Longrightarrow}} q$. A state s is a *sink* if s is the destination of any path originating in s.

A labelled poset is *accepted* by an automaton if it is the label of a path, called *successful*, leading from an initial state to a final state. The language $L(\mathcal{A})$ is the set of labelled posets accepted by the automaton $\mathcal{A}$. A language L is *regular* if there exists an automaton $\mathcal{A}$ such that $L = L(\mathcal{A})$.

Theorem 2 (Lodaya and Weil [13]). *Let A be an alphabet, and $L \subseteq SP(A)$. Then L is regular if and only if it is rational.*

Example 2. Figure 1 represents the automaton $\mathcal{A} = (\{1,2,3,4,5,6\}, \{a,b\}, E, \{1\}, \{1,6\})$, with $E_{seq} = \{(2,a,4),(3,b,5)\}$, $E_{fork} = \{(1,\{1,1\}),(1,\{2,3\})\}$ and $E_{join} = \{(\{6,6\},6),(\{4,5\},6)\}$, and an accepting path labelled by $a \parallel b \parallel a \parallel b$. Actually, $L(\mathcal{A}) = (a \parallel b)^{\circledast}$.

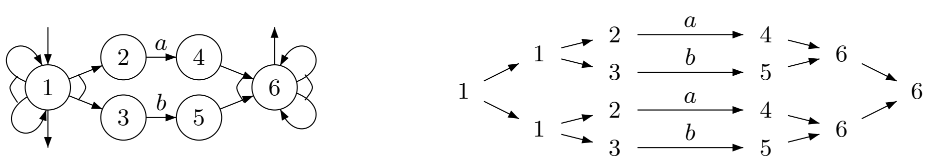

Fig. 1. An automaton $\mathcal{A}$ with $L(\mathcal{A}) = (a \parallel b)^{\circledast}$ and an accepting path labelled by $a \parallel b \parallel a \parallel b$.

It is known from Lodaya and Weil [15] that the regular languages of $SP(A)$ are closed under finite union and finite intersection, but the closure under complementation was still unexplored.

The first result of this paper is stated by the following Theorem which implies that the class of regular languages of N-free posets is closed under boolean operations.

Theorem 3. *Let A be an alphabet. The class of regular languages of $SP(A)$ is effectively closed under complement.*

The proof relies on an algebraic approach of regular languages, which was first introduced by Lodaya and Weil [13–15]. Algebras considered here are of the form $(S, \cdot, \parallel)$ (or just S for short) such that $(S, \cdot)$ and $(S, \parallel)$ are respectively a semigroup and a commutative semigroup, which may be infinites. The first step consists in the construction of a morphism $\varphi : SP(A) \to S$, where S is build from an automaton $\mathcal{A}$ and $L(\mathcal{A}) = \varphi^{-1}(X)$ for some $X \subseteq S$. Then we show that $\varphi^{-1}(S - X)$ is regular by a reduction of the problem to the finitely generated commutative semigroup case, and we conclude by the use of the following result

Theorem 4 (Eilenberg and Schützenberger [5]). *If X and Y are rational subsets of a commutative monoid M, then $Y - X$ is also a rational subset of M.*

As emphasized in [18], if M is finitely generated then Theorem 4 is effective.

4 P-MSO

In this section we define a logical formalism called P-MSO, which is a mix between Presburger [17] and monadic second-order logic, and that has exactly the same expressivity as branching automata. As all the constructions involved in the proof are effective, then the P-MSO theory of the class of finite N-free posets is decidable.

Let us recall useful elements of monadic second-order logic, and settle some notation. For more details about MSO logic we refer e.g. to Thomas' survey paper [4, 20]. The monadic second-order (MSO) logic is classical in set theory, and was first set up by Büchi-Elgot-Trakhtenbrot for words [2, 6, 21]. In our case, the domain of interpretation is the class of finite N-free posets.

Monadic second-order logic is an extension of first-order logic that allows to quantify over elements as well as subsets of the domain of the structure. A MSO-formula is given by the following grammar

$$\begin{aligned}\psi ::= & R_a(x) \mid x \in X \mid x < y \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \neg\psi \\ & \mid \exists x\psi \mid \exists X\psi \mid \forall x\psi \mid \forall X\psi\end{aligned}$$

where $a \in A$, x, y and X are respectively first- and second-order variables, $R_a(x)$ is interpreted as "x is labelled by a" (also denoted $a(x)$ for readability), and all other symbols have their usual meaning. The language L_ψ of ψ is the class of posets $(P, <, \rho)$ labelled over A that satisfy ψ. Logical equivalence of formulæ corresponds to the equality of their languages. In order to enhance readability of formulæ we use several notations and abbreviations for properties expressible in MSO. The following are usual and self-understanding: $\phi \to \psi$, $X \subseteq Y$, $x = y$. We also write $\exists^X x\psi$ for $\exists x\ x \in X \wedge \psi$, and extend this notion of *relative quantification* to universal quantification and second-order variables. MSO logic is strictly less expressive than automata. There is no MSO-formula that defines the language $(a \parallel b)^\circledast$. On the contrary, MSO-definability implies rationality.

In order to capture the expressiveness of automata with logic we need to add Presburger expressivity to MSO. Presburger logic is the first-order logic over the structure $(\mathbb{N}, +)$ where $+ = \{(a, b, c) : a + b = c\}$. A language $L \subseteq \mathbb{N}^n$ is a *Presburger set* of $\mathbb{N}^n$ if $L = \{(x_1, \ldots, x_n) : \varphi(x_1, \ldots, x_n) \text{ is true }\}$ for some Presburger formula $\varphi(x_1, \ldots, x_n)$. If $\varphi(x_1, \ldots, x_n)$ is given then L is called the *Presburger set* of $\varphi(x_1, \ldots, x_n)$ (or of φ for short). Presburger logic provides tools to manipulate semi-linear sets of $A^\circledast$ with formulæ. Indeed, let $A = \{a_1, \ldots, a_n\}$ be an alphabet ($n > 0$). As a word u of $A^\circledast$ can be thought of as a n-tuple $(|u|_{a_1}, \ldots, |u|_{a_n})$ of non-negative integers, where $|u|_a$ denotes the number of occurences of letter a in u, then $A^\circledast$ is isomorphic to $\mathbb{N}^n$.

Example 3. Let $A = \{a, b, c\}$ and $L = \{u \in A^\circledast : |u|_a \leq |u|_b \leq |u|_c\}$. Then L is isomorphic to $\{(n_a, n_b, n_c) \in \mathbb{N}^3 : n_a \leq n_b \leq n_c\}$, and thus the Presburger set of

$$\varphi(n_a, n_b, n_c) \equiv (\exists x\ n_b = n_a + x) \wedge (\exists y\ n_c = n_b + y)$$

Semi-linear sets and Presburger sets are connected by the following Theorem:

Theorem 5 (Ginsburg and Spanier [8], Theorem 1.3). *Let A be an alphabet and $L \subseteq A^\circledast$. Then L is semi-linear if and only if it is a Presburger set. Furthermore, the construction of one description from the other is effective.*

The *P-MSO logic* is a melt of Presburger and MSO logics. From the syntactic point of view, P-MSO logic contains MSO logic, and in addition formulæ of the form

$$Q(Z, (\psi_1(R_1), x_1), \ldots, (\psi_n(R_n), x_n), \varphi(x_1, \ldots, x_n))$$

where Z is the name of a (free) second-order variable, $\psi_i(R_i)$ (for each $i \in 1 \ldots n$) a P-MSO formula having no free first-order variables, and only quantifications relative to R_i, and $\varphi(x_1, \ldots, x_n)$ a Presburger formula with n free variables $x_1, \ldots, x_n$. Considering the formula $\psi(Z) = Q(Z, (\psi_1(R_1), x_1), \ldots, (\psi_n(R_n), x_n), \varphi(x_1, \ldots, x_n))$ the only variable that counts as *free* in $\psi(Z)$ is Z. Note that as n can be any positive integer then P-MSO does not really fit into the framework of usual formal propositional logic (where the arity of connectors are usually fixed).

As in monadic second-order logic, the class of syntactically correct P-MSO formulæ is closed under boolean operations, and existential and universal quantification over first and second-order variables of a P-MSO formula that are interpreted over elements or sets of elements of the domain of the structure. Semantics of P-MSO formulæ is defined below by extension of semantics of Presburger and MSO logics. The notions of a language and definability naturally extend from MSO to P-MSO.

Before continuing with formal definitions, let us give some intuition on the meaning of $\psi(Z) = Q(Z, (\psi_1(R_1), x_1), \ldots, (\psi_n(R_n), x_n), \varphi(x_1, \ldots, x_n))$. Let X be an interpretation of a second-order variable Z in P, such that X is a good block of P. That means, X is the poset associated with a sub-term of a term on A (a full binary tree whose leaves are elements of A, and nodes are a sequential or a parallel product) describing P, and is the parallel composition of $m \geq 1$ connected blocks: $X = X_1 \parallel \cdots \parallel X_m$. Take n different colors $c_1, \ldots, c_n$. To each X_i we associate a color c_j with the condition that X_i satisfies $\psi_j(X_i)$. Observe that this coloring may not be unique, and may not exist. Denote by x_j the number of uses of c_j in the coloring of X. Then $P, X \models \psi(Z)$ if there exists such a coloring with $x_1, \ldots, x_n$ satisfying the Presburger condition $\varphi(x_1, \ldots, x_n)$.

More formally, let $P \in SP(A)$, $\psi(Z) = Q(Z, (\psi_1(R_1), x_1), \ldots, (\psi_n(R_n), x_n), \varphi(x_1, \ldots, x_n))$ be a P-MSO formula, $X \subseteq P$ be an interpretation of Z in P such that X is a good block of P. Then $P, X \models \psi(Z)$ if there exist non negative integers $v_1, \ldots, v_n$ and a partition $(Z_{1,1}, \ldots, Z_{1,v_1}, \ldots, Z_{n,1}, \ldots, Z_{n,v_n})$ of X into connected blocks $Z_{i,j}$ such that

- $(v_1, \ldots, v_n)$ belongs to the Presburger set of $\varphi(x_1, \ldots, x_n)$,
- $z \in Z_{i,j}$, $z' \in Z_{i',j'}$ implies that z and z' are incomparable, for all possible (i, j) and (i', j') with $(i, j) \neq (i', j')$,
- $P, Z_{i,j} \models \psi_i(Z_{i,j})$ for all $i \in 1 \ldots n$ and $j \in 1 \ldots v_i$.

Example 4. Let L be the language of Example 3, and $\varphi(n_a, n_b, n_c)$ be the Presburger formula of Example 3. For all $\alpha \in A$, set $\psi_\alpha(X) \equiv \mathtt{Card}_1(X) \wedge \forall^X x\ \alpha(x)$, where $\mathtt{Card}_1(X)$ is a MSO formula (thus a P-MSO formula) which is true if and only if the interpretation of X has cardinality 1. Then L is the language of the following P-MSO sentence:

$$\forall P\ (\forall p\ p \in P) \rightarrow Q(P, (\psi_a(X), n_a), (\psi_b(X), n_b), (\psi_c(X), n_c), \varphi(n_a, n_b, n_c))$$

Theorem 6. *Let A be an alphabet, and $L \subseteq SP(A)$. Then L is rational if and only if is P-MSO definable.*

The proof uses usual arguments adapted to the case of N-free posets.

The inclusion from left to right relies on the ideas of Büchi on words: the encoding of accepting paths of a branching automaton $\mathcal{A}$ into a P-MSO formula. Each letter of the poset is mapped to a sequential transition of $\mathcal{A}$, and each part of the poset of the form $P = P_1 \parallel \cdots \parallel P_n$ $(n > 1)$, as great as possible relatively to inclusion and such that each P_i is a connected block of P, is mapped to a pair (p, q) of states; informally speaking, p and q are the states that are supposed to respectively begin and finish the part of the path labelled by P. The formula guarantees that pairs of states and sequential transitions are chosen consistently with the transitions of $\mathcal{A}$, and that, if $P = P_1 \parallel \cdots \parallel P_n$ as above and $p_i \overset{P_i}{\underset{\mathcal{A}}{\Longrightarrow}} q_i$ for all $i \in 1 \ldots n$, then there exists a combination of fork transitions that connects p to $p_1, \ldots, p_n$, a sequence of join transitions that connects $q_1, \ldots, q_n$ to q, such that a path $p \overset{P}{\underset{\mathcal{A}}{\Longrightarrow}} q$ in $\mathcal{A}$ is formed.

The inclusion from right to left relies on well-known techniques from words adapted to posets. In this part of the proof posets are not just labelled by elements of the alphabet A, but by elements of $A \times \mathcal{P}(V_1) \times \mathcal{P}(V_2)$, where V_1 and V_2 are sets that contain respectively the names of the free first and second-order variables of the formula (we do not consider here the variables that are interpreted over nonnegative integers). When formulæ are sentences, then the posets are labelled by $A \times \emptyset \times \emptyset$, which is similar to A. Observe that an interpretation of the variables $\{x_1, \ldots, x_n\} = V_1$, $\{X_1, \ldots, X_m\} = V_2$ in P induces a unique poset labelled by elements of $A \times \mathcal{P}(V_1) \times \mathcal{P}(V_2)$, and reciprocally. This allows us to use indifferently one representation or the other in order to lighten the notation. This labelling of posets by elements of $A \times \mathcal{P}(V_1) \times \mathcal{P}(V_2)$ has a unique restriction: the name of a free first-order variable x must appear at most once in the labels of elements of the poset. An automaton $\mathcal{A}_r$ that accepts a poset if and only if this condition is verified on its label can easily be constructed. We may assume, up to an intersection with $\mathcal{A}_r$ (the regular languages are closed under intersection), that all the constructions of automata below have posets in $L(\mathcal{A}_r)$ as inputs.

We build, by induction on the structure of $\varphi(x_1, \ldots, x_n, X_1, \ldots, X_m)$, an automaton $\mathcal{A}_\varphi$ such that $P, x_1, \ldots, x_n, X_1, \ldots, X_m \models \varphi(x_1, \ldots, x_n, X_1, \ldots, X_m)$ if and only if $P, x_1, \ldots, x_n, X_1, \ldots, X_m \in L(\mathcal{A}_\varphi)$. The case $n = m = 0$ gives the inclusion from right to left of Theorem 6. For formulæ of the form $x < y$ it suffices to build an automaton that checks if the poset has two elements p_1 and p_2 respectively labelled by (a_1, X_1, X_2) and (a_2, Y_1, Y_2) such that $p_1 < p_2$, $x \in X_1$ and $y \in y_1$. An automaton that checks if the poset contains an element labelled by (a, X_1, X_2) with $x \in X_1$ can easily be constructed for formulæ of the form $a(x)$. The case of formulæ of the form $x \in X$ is similar. Constructions of automata for the boolean connectors $\vee$, $\wedge$ and $\neg$ are a consequence of Theorem 3 and the closure under finite union and intersection of regular languages. For formulæ of the form $\exists x \phi$ or $\exists X \phi$, constructions are a consequence of the closure under projection of regular languages. We finally turn to the last case where the formula ψ has the form $Q(Z,(\psi_1(R_1),x_1),\ldots,(\psi_n(R_n),x_n),\varphi(x_1,\ldots,x_n))$. Recall here that $x_1,\ldots,x_n$ are variables that are interpreted over nonnegative integers, and that each ψ_i, $i \in 1\ldots n$, has one free variable R_i, which is second-order, all quantifications relative to R_i and no free first-order variables. By induction hypothesis, there is an automaton $\mathcal{A}_{\psi_i}$ such that $P,R \models \psi_i(R)$ if and only if $P,R \in L(\mathcal{A}_{\psi_i})$. According to the semantics of $Q(Z,(\psi_1(R_1),x_1),\ldots,(\psi_n(R_n),x_n),\varphi(x_1,\ldots,x_n))$, the only interpretations of R in P verify (1) $R = P$ and (2) P is a connected block. The conjunction of (1) and (2) is a MSO-definable property of R, and thus it can be checked by an automaton $\mathcal{B}$. As a consequence of the closure under intersection of regular languages there exists an automaton $\mathcal{A}'_{\psi_i}$ such that $L_i = L(\mathcal{A}'_{\psi_i}) = L(\mathcal{A}_{\psi_i}) \cap L(\mathcal{B})$. Now, let $B = \{b_1,\ldots,b_n\}$ be a new alphabet disjoint from A. As a consequence of Theorems 5, 1 and 2 there is an automaton $\mathcal{C}$ over the alphabet B such that $L(\mathcal{C})$ is the Presburger set of $\varphi(x_1,\ldots,x_n)$ over B. Then $L_\psi = L_1 \circ_{b_1} (\ldots(L_n \circ_{b_n} L(\mathcal{C})))$ thus L_ψ is regular according to Theorem 2.

Example 5. Let L be the language over the alphabet $A = \{a,b\}$ composed of the sequential products of posets of the form $P = P_1 \parallel \cdots \parallel P_n$ such that each P_i is a nonempty totally ordered poset (i.e., a word), and that the number of P_i that starts with an a is $\frac{2}{3}n$. Set $L_1 = aA^*$ and $L_2 = bA^*$. Then L is the language of the rational expression $((L_1 \parallel L_1 \parallel L_2)^{\circledast})^*$. We define L by a P-MSO sentence as follows. Given two elements of the poset denoted by first order variables x and y, one can easily write a MSO formula $\texttt{Succ}(x,y)$ (resp. $\texttt{Pred}(x,y)$) that is true if and only if x is a successor (resp. predecessor) of y. Set

$$\begin{aligned}
\texttt{Lin}(X) \equiv& \forall^X x \forall^X y \forall^X z\ ((\texttt{Succ}(y,x) \wedge \texttt{Succ}(z,x)) \rightarrow y = z) \\
&\qquad\qquad\qquad \wedge ((\texttt{Pred}(y,x) \wedge \texttt{Pred}(z,x)) \rightarrow y = z) \\
\psi_1(X) \equiv& \texttt{Lin}(X) \wedge \exists^X x\ a(x) \wedge \forall^X y\ x = y \vee x < y \\
\psi_2(X) \equiv& \texttt{Lin}(X) \wedge \exists^X x\ b(x) \wedge \forall^X y\ x = y \vee x < y \\
\varphi(n_a,n_b) \equiv& n_a = 2n_b
\end{aligned}$$

Then L is the language of the following P-MSO sentence

$$\begin{aligned}
\psi \equiv \forall P(\forall p\ p \in P) \rightarrow \exists X_1 \exists X_2\ P = X_1 \oplus X_2 \\
\wedge \forall U((\texttt{MaxBlock}(U,X_1) \vee \texttt{MaxBlock}(U,X_2)) \rightarrow \\
Q(U,(\psi_1(R_1),n_a),(\psi_2(R_2),n_b),\varphi(n_a,n_b))
\end{aligned}$$

with $X = U \oplus V \equiv \texttt{Partition}(U,V,X) \wedge (\forall u \forall v\ u \in U \wedge v \in V \rightarrow \neg u \parallel v)$. In the formula above, $\texttt{Partition}(U,V,X)$ and $u \parallel v$ respectively express with MSO formulæ that (U,V) partitions X, and that u and v are different and not comparable. The MSO formula $\texttt{MaxBlock}(U,X)$ express that U is a block of X, maximal relatively to inclusion.

5 Conclusion

As all the constructions involved in the proof of Theorem 6 are effective, and emptiness is decidable for languages of branching automata, P-MSO is decidable:

Theorem 7. *Let A be an alphabet. The P-MSO theory of $SP(A)$ is decidable.*

In [15], Lodaya and Weil asked for logical characterizations of several classes of rational languages. As it is equivalent to branching automata, P-MSO is the natural logic to investigate such questions, that are still open.

Among the works connected to ours, let us mention Esik and Németh [7], which itself has been influenced by the work of Hoogeboom and ten Pas [9, 10] on text languages. They study languages of biposets from an algebraic, automata and regular expressions based point of view, and the connections with MSO. A biposet is a set equipped with two partial orderings; thus, N-free posets are a generalisation of N-free biposets, where commutation is allowed in the parallel composition.

MSO and Presburger logic were also mixed in other works, but for languages of trees instead of N-free posets. Motivated by reasoning about XML documents, Dal Zilio and Lugiez [3], and independently Seidl, Schwentick and Muscholl [19], defined a notion of tree automata which combines regularity and Presburger arithmetic. In particular in [19], MSO is enriched with Presburger conditions on the children of nodes in order to select XML documents, and proved equivalent to unranked tree automata. Observe that unranked trees are a particular case of N-free posets. The logic named *Unordered Presburger MSO logic* in [19] is contained in our P-MSO logic.

The quality of this paper has been enhanced by the comments of the anonymous referees. One of them noticed that Theorem 3 might also be retrieved using the notion of Commutative Hedge automata (see e.g. [1]), as N-free posets can be assimilated to terms over the operations of parallel and sequential products.

References

1. Ahmed Bouajjani and Tayssir Touili. On computing reachability sets of process rewrite systems. In Jürgen Giesl, editor, *RTA*, volume 3467 of *Lecture Notes in Computer Science*, pages 484–499. Springer, 2005.
2. J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeit. Math. Logik. Grund. Math.*, 6:66–92, 1960.
3. Silvano Dal-Zilio and Denis Lugiez. XML Schema, Tree Logic and Sheaves Automata. In Robert Nieuwenhuis, editor, *RTA*, volume 2706 of *Lecture Notes in Computer Science*, pages 246–263. Springer, 2003.
4. Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory.* Springer monographs in mathematics. Springer, 2nd edition, 1999.
5. Samuel Eilenberg and Marcel-Paul Schützenberger. Rational sets in commutative monoids. *Journal of Algebra*, 13(2):173–191, 1969.
6. Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–51, January 1961.

7. Zoltán Ésik and Z.L. Németh. Automata on series-parallel biposets. In W. Kuich, G. Rozenberg, and A. Salomaa, editors, *DLT'2001*, volume 2295 of *Lect. Notes in Comput. Sci.*, pages 217–227. Springer-Verlag, 2002.
8. Seymour Ginsburg and Edwin H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.
9. H. J. Hoogeboom and P. ten Pas. Text languages in an algebraic framework. *Fund. Inform.*, 25:353–380, 1996.
10. H. J. Hoogeboom and P. ten Pas. Monadic second-order definable languages. *Theory Comput. Syst.*, 30:335–354, 1997.
11. Stephen C. Kleene. Representation of events in nerve nets and finite automata. In Shannon and McCarthy, editors, *Automata studies*, pages 3–42, Princeton, New Jersey, 1956. Princeton University Press.
12. Dietrich Kuske. Infinite series-parallel posets: logic and languages. In *ICALP 2000*, volume 1853 of *Lect. Notes in Comput. Sci.*, pages 648–662. Springer-Verlag, 2000.
13. Kamal Lodaya and Pascal Weil. A Kleene iteration for parallelism. In *Foundations of Software Technology and Theoretical Computer Science*, pages 355–366, 1998.
14. Kamal Lodaya and Pascal Weil. Series-parallel posets: algebra, automata and languages. In M. Morvan, Ch. Meinel, and D. Krob, editors, *STACS'98*, volume 1373 of *Lect. Notes in Comput. Sci.*, pages 555–565. Springer-Verlag, 1998.
15. Kamal Lodaya and Pascal Weil. Series-parallel languages and the bounded-width property. *Theoret. Comput. Sci.*, 237(1–2):347–380, 2000.
16. Kamal Lodaya and Pascal Weil. Rationality in algebras with a series operation. *Inform. Comput.*, 171:269–293, 2001.
17. Mojzesz Presburger. Über die vollstandigkeit eines gewissen systems der arithmetic ganzer zahlen, in welchem die addition als einzige operation hervortritt. In *Proc. Sprawozdaniez I Kongresu Matematykow Krajow Slowianskich, Warsaw*, pages 92–101, 1930. English translation: On the completeness of certain system of arithmetic of whole numbers in which addition occurs as the only operation. *Hist. Philos. Logic*, 12:92–101, 1991.
18. Jacques Sakarovitch. *Éléments de théorie des automates*. Vuibert, 2003. English (and revised) version: *Elements of automata theory*, Cambridge University Press, 2009.
19. Helmut Seidl, Thomas Schwentick, and Anca Muscholl. Counting in trees. In Jörg Flum, Erich Grädel, and Thomas Wilke, editors, *Logic and Automata*, volume 2 of *Texts in Logic and Games*, pages 575–612. Amsterdam University Press, 2008.
20. Wolfgang Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume III, pages 389–455. Springer-Verlag, 1997.
21. Boris Avraamovich Trakhtenbrot. Finite automata and monadic second order logic. *Siberian Math.*, 3:101–131, 1962. (Russian). English translation in AMS Transl. 59 (1966), 23-55.
22. Jacobo Valdes. Parsing flowcharts and series-parallel graphs. Technical Report STAN-CS-78-682, Computer science departement of the Stanford University, Standford, Ca., 1978.
23. Jacobo Valdes, Robert E. Tarjan, and Eugene L. Lawler. The recognition of series parallel digraphs. *SIAM J. Comput.*, 11:298–313, 1982.