RAIRO-Theor. Inf. Appl. **42** (2008) 503–524
DOI: 10.1051/ita:2008014

Available online at:
www.rairo-ita.org

AN AUTOMATA-THEORETIC APPROACH TO THE STUDY OF THE INTERSECTION OF TWO SUBMONOIDS OF A FREE MONOID

Laura Giambruno[1] and Antonio Restivo[1]

Abstract. We investigate the intersection of two finitely generated submonoids of the free monoid on a finite alphabet. To this purpose, we consider automata that recognize such submonoids and we study the product automata recognizing their intersection. By using automata methods we obtain a new proof of a result of Karhumäki on the characterization of the intersection of two submonoids of rank two, in the case of prefix (or suffix) generators. In a more general setting, for an arbitrary number of generators, we prove that if H and K are two finitely generated submonoids generated by prefix sets such that the product automaton associated to $H \cap K$ has a given special property then $\widetilde{rk}(H \cap K) \leq \widetilde{rk}(H)\widetilde{rk}(K)$ where $\widetilde{rk}(L) = \max(0, rk(L) - 1)$ for any submonoid L.

Mathematics Subject Classification. 68Q70, 68Q45, 20M35.

1. Introduction

The main purpose of this paper is to study the intersection of two submonoids of a free monoid by using an automata approach. We are, in particular, interested in submonoids of finite rank.

The study of the intersection of two submonoids of finite rank is not trivial at all. In fact, by a result of Latteux and Leguy (*cf.* [8]), every language is regular if and only if the submonoid generated by it is obtained as homomorphic image of the intersection of two finitely generated monoids: *Let A be an alphabet and R a language of A^*. R is a regular language if and only if there exist two finite languages F_1, F_2 and a morphism g such that $R^* = g({F_1}^* \cap {F_2}^*)$.*

Keywords and phrases. Automata, free monoids, rank, intersection of submonoids.

[1] Dipartimento di Matematica e Applicazioni, Università di Palermo, via Archirafi 34, 90123 Palermo, Italy; lgiambr;restivo@math.unipa.it

Moreover the study of the intersection of two submonoids has been developed in many papers. It was first studied by Tilson in 1972 (*cf.* [12]) who proved that the intersection of two free submonoids of A^*, the free monoid on a finite alphabet A, is free too.

Karhumäki in 1984 (*cf.* [7]) deepened the study of the intersection of two submonoids generated by two elements by giving a characterization of such an intersection. In particular he proved that given two submonoids H and K of A^*, where A is a finite alphabet, if both H and K are of rank two, then $H \cap K$ is a submonoid either of rank at most two or it is generated by a regular language of a special form. In particular if H and K are generated by prefix (or suffix) sets of two words, and $H \cap K$ is not finitely generated, then this intersection has the form $(\alpha\beta^*\gamma)^*$ where $\alpha, \beta, \gamma \in A^*$.

Recently Bruyère *et al.* (*cf.* [2]) have studied the *meet* of two rational codes X and Y, defined as the base of the free monoid $X^* \cap Y^*$. They concentrated on the study of maximal rational codes such that their meet is yet a maximal rational code, showing with many examples the complex behavior of the meet. Finally they proved that any rational code is the meet of two rational maximal codes.

The starting motivation of this paper was the article of Karhumäki (*cf.* [7]) that characterizes the intersection of two submonoids of the free monoid generated by two elements. The proof given by Karhumäki is long and provides no intuition on the real nature of the result. In this paper we prove, in particular, the result of Karhumäki in the case of two submonoids generated by prefix (or suffix) sets using a more intuitive approach based on automata.

When dealing with the intersection of two submonoids of finite rank it is natural to relate it to a more general problem in the theory of free groups known as the 'Hanna Neumann conjecture'. This conjecture deals with the problem of finding an upper bound of the rank of the intersection of two finitely generated subgroups.

In 1956 Hanna Neumann (*cf.* [10]) proved that if H and K are two subgroups of finite rank of a free group then $\widetilde{rk}(H \cap K) \leq 2\widetilde{rk}(H)\widetilde{rk}(K)$, where $\widetilde{rk}(T) = \max(rk(T) - 1, 0)$ with $rk(T)$ denoting the rank of a subgroup T. Then she made the following conjecture, known nowadays as the 'Hanna Neumann conjecture':

$$\widetilde{rk}(H \cap K) \leq \widetilde{rk}(H)\widetilde{rk}(K).$$

In 1991 Walter Neumann (*cf.* [11]) formulated a stronger conjecture known as "Strengthened Hanna Neumann conjecture" (in short SHN) and in 2002 Meakin and Weil (*cf.* [9]) proved that SHN holds for the class of positively generated subgroups of the free group $F(A)$ on A, finite alphabet, that are generated by words on A^*. This last result suggested us to propose the problem of Hanna Neumann for finitely generated submonoids of a free monoid, in the case that their intersection is finitely generated.

Some of the basic tools in dealing with the Hanna Neumann conjecture for free groups makes use of the representation of subgroups of the free group by graphs (or automata). The same tools are still available when dealing with the intersection of two submonoids of the free monoid. For this purpose we refer to the correspondence (*cf.* [1]) between submonoids on the free monoid on a finite alphabet A and automata on A.

Through the study of the product of two automata associated to two finitely generated submonoids H and K, we prove that if H and K are submonoids generated by prefix sets such that the product automaton associated to $H \cap K$ has a given special property then $\widetilde{rk}(H \cap K) \leq \widetilde{rk}(H)\widetilde{rk}(K)$. In the general case we find a family of examples such that $rk(H \cap K) = 2^{\log_2(rk(H))\log_2(rk(K))}$.

Moreover if the two submonoids H and K are generated by prefix sets of two elements, then their intersection $H \cap K$ either is of rank two, or it has the form $(\alpha\beta^*\gamma)^*$ where $\alpha, \beta, \gamma \in A^*$, that is the result of Karhumäki [7].

2. Automata and submonoids

We consider finite state automata on a finite alphabet A. For the notation we refer to [1]. Let A be a finite alphabet. Let us denote by ε the empty word of A^* and by A^+ the set of nonempty words on A. An *automaton* over A, $\mathcal{A} = (Q, I, T, \mathcal{F})$, consists of a finite set Q of *states*, of two subsets I and T of Q called sets of initial and final states, respectively, and of a set $\mathcal{F} \subset Q \times A \times Q$ whose elements are called *edges*. An edge $e = (x, a, y)$ is also denoted by $e : x \xrightarrow{a} y$. The letter a is called the label of the edge. We will say that the edge e goes out from x and comes in y.

Two edges $e : x \xrightarrow{a} y$ and $f : x' \xrightarrow{b} y'$ are consecutive if $y = x'$. A *path* in $\mathcal{A}$ is a finite sequence $p = p_1p_2 \dots p_n$ of consecutive edges $p_i : x_i \xrightarrow{a_i} y_i$. We shall also write $p : x_1 \xrightarrow{w} y_n$ where $w = a_1a_2 \dots a_n$ is the *label* of the path p. The path p is said to start at x_1 and end at y_n. We indicate by $i(p) = x_1$ the starting state and by $f(p) = y_n$ the ending state. The *length* of a path is the number of edges that compose it. For each state $x \in Q$ it is defined the *null path* starting and ending at x, denoted by $1_x : x \longrightarrow x$ having as label ε. Given an automaton $\mathcal{A}$, we will say that a path p of $\mathcal{A}$ visits a vertex x of $\mathcal{A}$ if x is a vertex of p.

When two paths p and q are consecutive (*i.e.* $f(p) = i(q)$) then p and q can be concatenated and we call the resulting path pq. A *subpath* of a path p is a subsequence of consecutive edges. A subpath of a path p is a *prefix* of p if it starts at the same starting state of p. Given two paths p and q starting at the same state x, the *longest prefix path in common* between p and q is a path prefix of p and prefix of q that is the longest with this property. Analogously it can be defined, given two paths p and q ending at the same state x, the *longest suffix path in common* between p and q.

A path p is a *simple path* if all states in the path are distinct. Given two states x and y, if there exists a path from x to y then there exists also a simple path from x to y. A path p is a *cycle* if it is not the null path and if it starts and ends at the same state. We say that a path c is a cycle in x if it starts and ends at x. A cycle c is a *simple cycle* if it has all the intermediate states distinct. Given a cycle c we say that c is *simple in* x if it is a cycle in x such that no intermediate state is equal to x. We remark that a cycle that is simple in x is not in general a simple cycle.

The *language recognized by* $\mathcal{A}$, $L(\mathcal{A})$, is the set of words that are labels of paths from an initial state to a final one. A state x in $\mathcal{A}$ is *accessible* if there is a path starting at an initial state and ending at x. A state x in $\mathcal{A}$ is *coaccessible* if there is a path starting at x and ending at a final state. An automaton $\mathcal{A}$ is a *trim* automaton if all the states of the automaton are accessible and coaccessible. An automaton $\mathcal{A} = (Q, I, T, \mathcal{F})$ is *unambiguous* if, for each $x, y \in Q$, for each $w \in A^*$, there exists at most one path starting at x and ending at y with label w.

An automaton $\mathcal{A}$ over A is a *deterministic automaton* if $\mathrm{card}(I) = 1$ and if, for each state x and for each a in A, there is at most one edge starting in x with label a. If $\mathcal{A} = (Q, i, T, \mathcal{F})$ is a deterministic automaton it can be defined the function $\delta : Q \times A \longrightarrow Q$ such that $\delta(x, a) = y$ if $x \xrightarrow{a} y \in \mathcal{F}$ and $\delta(x, a) = \emptyset$ otherwise. This function δ is extended to words in A^* by setting, for all $x \in Q$, $\delta(x, \varepsilon) = x$ and, for $w \in A^*$ and $a \in A$, $\delta(x, wa) = \delta(\delta(x, w), a)$.

This function is called the *transition function.* With this notation we have that $L(\mathcal{A}) = \{w \in A^* \mid \delta(i, w) \in T\}$.

From now on we will consider automata with non empty and non trivial languages.

We say that $\mathcal{A}$ is a *monoidal automaton* if it is a trim automaton with a unique final state equal to a unique initial one. Such a special state is denoted by 1. Let us note that in a monoidal automaton, for each state x, there exist a simple path from 1 to x and a simple path from x to 1. It is easy to prove the following (*cf.* [1]):

Proposition 2.1. *Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a monoidal automaton. The automaton $\mathcal{A}$ recognizes the submonoid generated by the set of labels of the cycles that are simple in* 1.

In general given a submonoid H of A^* there exists a unique minimal set of generators (*cf.* [1]). We define the rank of H as the cardinality of the minimal set of generators. It is denoted by $rk(H)$. We say that a submonoid H of A^* is cyclic if $rk(H) = 1$. We also define the reduced rank of a submonoid H as $\widetilde{rk}(H) = \max(rk(H) - 1, 0)$.

A submonoid $H \subseteq A^*$ is free if there exists a set of generators X such that every element in H can be factorized in a unique way in words of X. If H is free the minimal set of generators of H is called the *base* of H.

Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a monoidal automaton. Let us denote by $C_{\mathcal{A}}$ the set of cycles that are simple in 1 and by $Y_{\mathcal{A}}$ the set of their labels. In general if $\mathcal{A}$ is monoidal then $Y_{\mathcal{A}}$ is not the minimal set of generators (see an example in Fig. 1). If we suppose that $\mathcal{A}$ is unambiguous then $Y_{\mathcal{A}}$ is the minimal set of generators and moreover the submonoid generated by $\mathcal{A}$ is free.

Proposition 2.2. *Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be an unambiguous monoidal automaton. The automaton $\mathcal{A}$ recognizes a free submonoid with base the set of labels of the cycles that are simple in* 1.

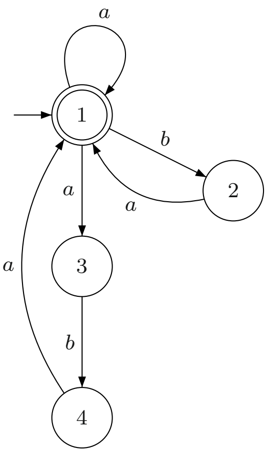

FIGURE 1. $\mathcal{A}$ monoidal automaton with $Y_{\mathcal{A}} = \{a, ba, aba\}$ and $L(\mathcal{A}) = \{a, ba\}^*$.

Proof. Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be an unambiguous monoidal automaton. By Proposition 2.1 the set of labels of cycles that are simple in 1, $Y_{\mathcal{A}}$, is a set of generators for $L(\mathcal{A})$. Moreover, by the unambiguity of $\mathcal{A}$, every element in $L(\mathcal{A})$ can be factorized in a unique way in words of $Y_{\mathcal{A}}$, otherwise there would be a word in $L(\mathcal{A})$ label of two different cycles in 1. □

So to a monoidal automaton $\mathcal{A}$ it is associated the submonoid $H = L(\mathcal{A})$ of A^*. Conversely to each submonoid X^* of A^* generated by a finite set X it is associated $\mathcal{F}_X$ the *flower automaton of* X (*cf.* [1,3]). It is built in the following way. First we build $\mathcal{S}_X$ the *solar automaton* recognizing X in this way: we build, for each word $x \in X$, an automaton with $|x| + 1$ states and merge all the initial states. Note that this automaton is a tree with root the initial state 1. Then we merge all the final states with the initial state 1. Doing this we obtain the flower automaton of X. Such an automaton is a monoidal automaton recognizing X^* such that all the cycles visit the unique initial-final state 1, all the cycles that are simple in 1 intersect themselves only in 1 and have as labels the words of X. See an example in Figure 2.

Let us define now a class of monoidal automata recognizing finitely generated submonoids: the class of semi-flower automata.

Definition 2.3. Let $\mathcal{A}$ be an automaton. $\mathcal{A}$ is a semi-flower automaton if it is a monoidal automaton such that all the cycles visit the unique initial-final state.

Hence in a semi-flower automaton the cycles that are simple in 1 intersect themselves not necessarily only in 1. In particular given a finite set $X \subseteq A^*$ the flower automaton associated to X is a semi-flower automaton in which all the cycles that are simple in 1 intersect themselves only in 1. It is interesting to observe that in a semi-flower automaton every cycle that is simple in 1 is in particular a simple cycle. It is easy to prove the following proposition.

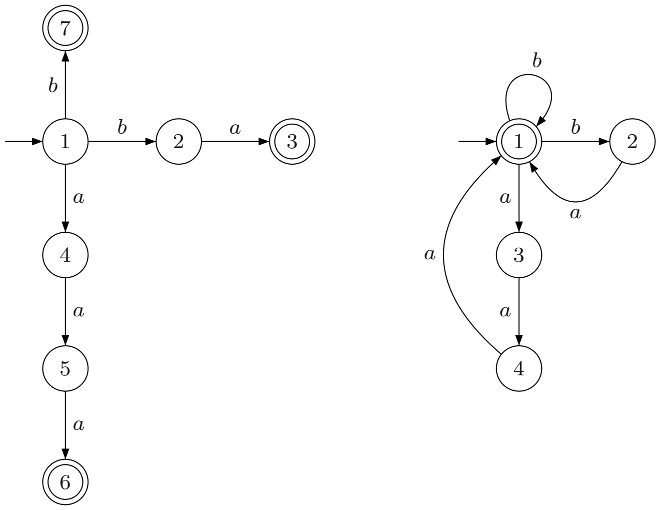

Figure 2. $S(X)$, $F(X)$ for $X = \{b, ba, aaa\}$.

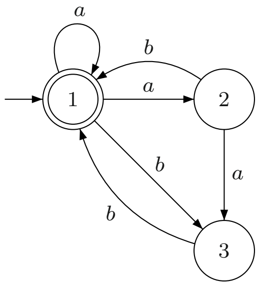

Figure 3. $\mathcal{A}$ semi-flower automaton, $L(\mathcal{A}) = \{a, ab, aab, bb\}^*$.

Proposition 2.4. *Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a semi-flower automaton, then $\mathcal{A}$ recognizes a finitely generated submonoid.*

An example of a semi-flower automaton is shown in Figure 3. The converse of Proposition 2.4 is not true in general as it is shown in the example of Figure 4.

However, with the supplementary hypothesis of unambiguity we get also the converse as stated in the following proposition.

Proposition 2.5. *Let $\mathcal{A}$ be an unambiguous monoidal automaton such that $L(\mathcal{A}) = H$. The submonoid H is finitely generated if, and only if $\mathcal{A}$ is a semi-flower automaton.*

Proof. Let $\mathcal{A}$ be an unambiguous monoidal automaton that recognizes H finitely generated submonoid. By Proposition 2.2, H is free with base $Y_{\mathcal{A}}$, the set of labels of the cycles that are simple in 1. By hypothesis, $|Y_{\mathcal{A}}| = rk(H) < \infty$.

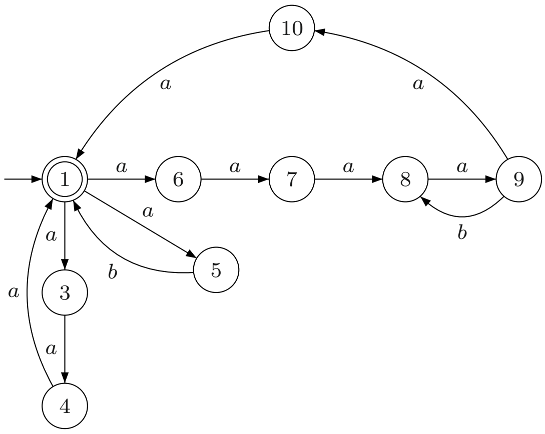

Figure 4. $\mathcal{A}$ not semi-flower automaton, $L(\mathcal{A}) = \{ab, aaa\}^*$.

Let us suppose, by contradiction, that there exists a cycle c not visiting 1. In particular c is a cycle in a state $x \neq 1$. Let v be the label of c. Let p_1 be a simple path from 1 to x, u_1 its label. Let p_2 be a simple path from x to 1, u_2 its label. We have that $\forall i \geq 0$, $u_1 v^i u_2$ is the label of $p_1 c^i p_2$ cycle that is simple in 1. Let $Y' = \{u_1 v^i u_2, \ \ \forall \ i \geq 0\}$. Finally we get $Y' \subseteq Y_{\mathcal{A}}$, contradiction since $\infty = |Y'| \leq |Y_{\mathcal{A}}| < \infty$. The other implication is proved in Proposition 2.4. □

We will recall now some definitions on graphs with the purpose of studying semi-flower automata with some graph's properties.

A *directed graph* $\Gamma = (V, E)$ (*cf.* [4]) is a pair (V, E) where V is a finite set and $E \subseteq V \times V$ is a binary relation on V. A *directed multigraph* is like a graph but it can have more than one edge between two vertices and self-loops (edges from a vertex to itself). More formally a directed multigraph is a 4-uple (V, E, i, f) where V and E are two disjoint sets and $i, f : V \longrightarrow E$ are maps.

Given a directed multigraph $\Gamma = (V, E)$ we say that a vertex $x \in V$ is a *branch point* (in short *bp*) if the degree of x (*i.e.* the number of edges incident to x) is greater than two. We say that a vertex $x \in V$ is a *branch point going out* (in short *bpo*) if x is a branch point and if the number of edges going out from x is at least two and we say that $x \in V$ is a *branch point going in* (in short *bpi*) if x is a branch point and if the number of edges coming in x is at least two.

Let $\mathcal{A}$ be an automaton. We can think of $\mathcal{A}$ as a labelled multigraph whose set of vertices is the set of states Q and the set of labelled edges is $\mathcal{F}$.

Let now $\mathcal{A}$ be a monoidal automaton. Let $BPI(\mathcal{A})$ be the set of vertices of $\mathcal{A}$ that are bpi's and let $BPO(\mathcal{A})$ be the set of vertices of $\mathcal{A}$ that are bpo's. We have the following proposition that links the existence of bpi's with the existence of bpo's.

Proposition 2.6. *Let $\mathcal{A}$ be a monoidal automaton. $BPI(\mathcal{A}) = \emptyset$ if and only if $BPO(\mathcal{A}) = \emptyset$.*

Proof. Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a monoidal automaton with $BPO(\mathcal{A}) \neq \emptyset$. Let so $x \in Q$ be a bpo. Let e_1 and e_2 be two edges starting at x and let us consider p a simple path from $f(e_1)$ to 1 and q a simple path from $f(e_2)$ to 1. Let us consider r the longest suffix path in common between e_1p and e_2q. Let us distinguish two cases: when $x = 1$ and when $x \neq 1$.

Let $x = 1$. If p, q are null paths then 1 is a bpi. If p is not null and q is the null path then e_2 is not a suffix of e_1p since p is simple and so $i(r)$ is a bpi. If p is null and q is not null we find analogously a bpi. Finally if p and q are not null paths then e_1p and e_2q are not suffixes each other since p and q are simple paths ending at 1. As before $i(r)$ is a bpi and so in all cases it results $BPI(\mathcal{A}) \neq \emptyset$.

Let $x \neq 1$. Let us prove that e_1p and e_2q are not suffixes each other. If, by contradiction, e_1p is a proper suffix of e_2q then $e_2q = q'e_1p$, with q' not null and $f(q') = x$. Let s be the simple path from 1 to x. The path s is not a suffix path of q' since q is simple and $x \neq 1$. Analogously q' is not a suffix path of s since s is simple. So if r is the longest suffix path in common between s and q' then $i(r)$ is a bpi and so $BPI(\mathcal{A}) \neq \emptyset$.
The converse is proved in an analogue way. □

In the following theorem we have a characterization of unambiguous monoidal automata with no bpi's. In particular, such automata recognize cyclic submonoids.

Theorem 2.7. *Let $\mathcal{A}$ be a monoidal automaton. If $BPI(\mathcal{A}) = \emptyset$ then $L(\mathcal{A})$ is cyclic. Moreover, if $\mathcal{A}$ is unambiguous then, if $L(\mathcal{A})$ is cyclic then $BPI(\mathcal{A}) = \emptyset$.*

Proof. Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a monoidal automaton such that $BPI(\mathcal{A}) = \emptyset$. If $|Q| = 1$ then trivially $L(\mathcal{A})$ is cyclic. Let $|Q| > 1$ and let $x \in Q$, $x \neq 1$. Let p be a simple path from 1 to x and q a simple path from x to 1 then the cycle pq is, in particular, simple in 1. It is the unique cycle that is simple in 1. In fact if there is another cycle that is simple in 1, let it be c, then considering r the longest suffix path in common between pq and c we get that the initial state of r is a bpi, that is a contradiction. So if u is the label of pq then $L(\mathcal{A}) = \{u\}^*$.

Conversely let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be an unambiguous monoidal automaton such that $L(\mathcal{A})$ is cyclic. Then $\mathcal{A}$ is semi-flower. If $BPI(\mathcal{A}) \neq \emptyset$ then there exists $x \in BPO(\mathcal{A})$. It follows that there exist two different cycles that are simple in 1 and so, by the unambiguity of $\mathcal{A}$, two free generators for $L(\mathcal{A})$, that is a contradiction. □

The semi-flower automata with a unique bpi have also interesting properties as we will see. The automaton in Figure 3 is a semi-flower automaton such that $BPI(\mathcal{A}) = \{1, 3\}$. The automaton in Figure 5 is a semi-flower automaton such that $BPI(\mathcal{A}) = \{1\}$.

It is interesting to note that if $\mathcal{A}$ is a monoidal automaton with initial-final state 1 such that $BPI(\mathcal{A}) = \{1\}$ then $\mathcal{A}$ is a semi-flower automaton, as explained in the following proposition.

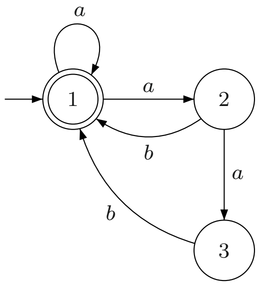

Figure 5. $\mathcal{A}$ semi-flower automaton, $|BPI(\mathcal{A})| = 1$.

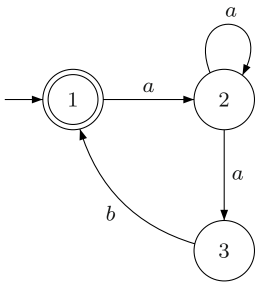

Figure 6. $\mathcal{A}$ monoidal automaton with a unique bpi that is not a semi-flower automaton.

Proposition 2.8. *Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a monoidal automaton such that $BPI(\mathcal{A}) = \{1\}$. It follows that $\mathcal{A}$ is a semi-flower automaton.*

Proof. Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a monoidal automaton having $BPI(\mathcal{A}) = \{1\}$. Let us suppose that there exists a cycle c in x not visiting 1. Let p be the simple path from 1 to x and let us consider r the longest suffix path in common between p and c. The path p is not a suffix of c since c does not visit 1. Moreover c is not a suffix of p since p is simple. So the initial vertex of r is a bpi different from 1 and we get the thesis. □

In general if $\mathcal{A}$ is a monoidal automaton such that $|BPI(\mathcal{A})| = 1$ then $\mathcal{A}$ is not necessarily a semi-flower automaton as we can see in the example in Figure 6.

Let us see now, given a semi-flower automaton with exactly one bpi, how to link the rank of the submonoid generated by it with the characteristics of the automaton. Let us give before some definitions and propositions.

Let us consider now a graph $\Gamma = (V, E)$. We recall that an undirected graph Γ is a *tree* if it is connected and acyclic. Given $x, z \in V$ we say that x is reachable from z if there exists a path in Γ from z to x. There exist different algorithms for visiting a graph. One of such algorithms is the *breadth-first search* (BFS in shorts). Given a graph $\Gamma = (V, E)$ and a distinguished *source* vertex s, BFS systematically explores the edges of Γ to "discover" every vertex that is reachable from s. Moreover it discovers every vertex reachable from s only once. It also produces a tree T called

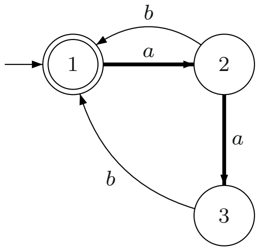

FIGURE 7. $\mathcal{A}$ semi-flower automaton. T BFS tree in 1.

breadth-first tree with root s that contains all such reachable vertices. For the procedure of breadth-first search and breadth-first tree we recall [4].

In Figure 7 we see an example of a semi-flower automaton with pointed out the edges of the breadth-first tree in 1.

Remark 2.9. The BFS procedure applied to automata chooses only one edge among multiples edges between two same states. Moreover it produces a tree.

Let us give now a theorem that links the rank of the submonoid recognized by a semi-flower automaton with a unique bpi to the characteristics of the automaton.

Theorem 2.10. *Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a semi-flower automaton. If $|BPI(\mathcal{A})| = 1$ then $rk(L(\mathcal{A})) \leq |\mathcal{F}| - |Q| + 1$.*

Proof. Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a semi-flower automaton with a unique bpi. Let $BPI(\mathcal{A}) = \{x\}$. By Proposition 2.1 , a set of generators for $L(\mathcal{A})$ is $Y_{\mathcal{A}}$, the finite set of labels of the cycles that are simple in 1. So, we have to count the number of cycles that are simple in 1. Let us prove that there is a bijection between $C_{\mathcal{A}}$, the set of cycles that are simple in 1, and $\mathcal{F}(x)$, the set of edges ending at x. Then we will have $rk(L(\mathcal{A})) \leq |Y_{\mathcal{A}}| \leq |C_{\mathcal{A}}| = |\mathcal{F}(x)|$. For this purpose let us prove the following three points:

1. *If $c \in C_{\mathcal{A}}$ then c visits x.*
If $x = 1$ it is done. Let so $x \neq 1$ and let q be a simple path in $\mathcal{A}$ from x to 1. If we consider the longest suffix path in common between q and c we get that if c does not visit x then there exists a bpi different from x (the proof is analogous of that one of Prop. 2.8).

2. *There is a unique simple path q in $\mathcal{A}$ from x to 1.*
If $x = 1$ then the unique simple path from 1 to 1 is the null path. Let so $x \neq 1$. If, by contradiction, there exists q' simple path from x to 1, $q' \neq q$, then q and q' cannot be suffix each other since they are simple paths with the same initial vertex. So if we consider t the longest suffix path in common between q and q' we have that $i(r)$ is a bpi different from x.

3. *For each edge e ending at x there exists a unique simple path $p_e : 1 \longrightarrow i(e)$.*
If $i(e) = 1$ then the unique simple path from 1 to 1 is the null path. If by contradiction there exists another simple path q_e from 1 to $i(e)$ then we get as

before another bpi considering the longest suffix path in common between p_e and q_e.
Let us define now a map from $C_{\mathcal{A}}$ to $\mathcal{F}(x)$. Let

$$\varphi : C_{\mathcal{A}} \longrightarrow \mathcal{F}(x)$$

be such that

$$for\ each\ c \in C_{\mathcal{A}},\ \ \varphi(c) = e_c$$

where $e_c \in \mathcal{F}$ is the unique edge in c ending at x.

- φ is well defined since c is a simple cycle so it visits x only once;
- φ is injective. Let $c_1, c_2 \in C_{\mathcal{A}}$ such that $\varphi(c_1) = \varphi(c_2) = e_c$. Then $c_1 = c_2 = p_{e_c} e_c q$;
- φ is surjective. For each $e \in \mathcal{F}(x)$, let p be the simple path from 1 to $i(e)$ then $peq \in C_{\mathcal{A}}$ and $\varphi(peq) = e$.

Since φ is a bijection then $|C_{\mathcal{A}}| = |\mathcal{F}(x)|$.
Let us consider $T = (Q(T), \mathcal{F}(T))$ the BFS tree associated to the BFS procedure applied to $\mathcal{A}$ with root the state x. Since every state of $\mathcal{A}$ is reachable from 1 then the BFS tree is a spanning tree. In the following we prove that the edges of $\mathcal{A}$ that are not contained in T are all the edges ending at x, that is $\mathcal{F} \setminus \mathcal{F}(T) = \mathcal{F}(x)$.

- Let us prove that $\mathcal{F} \setminus \mathcal{F}(T) \subseteq \mathcal{F}(x)$. Let $e \in \mathcal{F} \setminus \mathcal{F}(T)$. Then either the edge e ends at x or $f(e)$, the final vertex of e, is visited by the BFS procedure by another edge of $\mathcal{A}$ ending at $f(e)$, let us call it g. Since $g \neq e$ then $f(e)$ is a bpi and $e \in \mathcal{F}(x)$.
- $\mathcal{F}(x) \subseteq \mathcal{F} \setminus \mathcal{F}(T)$ since the BFS procedure is applied in x.

So we have that

$$|\mathcal{F} \setminus \mathcal{F}(T)| = |\mathcal{F}(x)| = |C_{\mathcal{A}}|$$

and by the properties of trees we have

$$|\mathcal{F}(T)| = |Q(T)| - 1 = |Q| - 1$$

and so

$$|C_{\mathcal{A}}| = |\mathcal{F} \setminus \mathcal{F}(T)| = |\mathcal{F}| - |Q| + 1.$$

Finally by Proposition 2.4

$$rk(H) \leq |C_{\mathcal{A}}| = |\mathcal{F}| - |Q| + 1. \qquad \square$$

Let us note that the statement of Theorem 2.10 is not true without the hypothesis of $|BPI(\mathcal{A})| = 1$ as we can see in Figure 8.

It follows from the proof of Theorem 2.10 and Proposition 2.2 that in the unambiguous case the following holds:

Proposition 2.11. *If $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ is an unambiguous semi-flower automaton such that $|BPI(\mathcal{A})| = 1$ then $rk(L(\mathcal{A})) = |\mathcal{F}| - |Q| + 1$.*

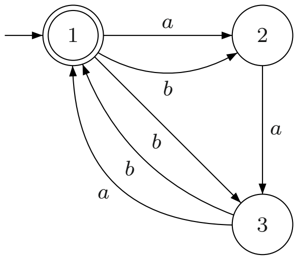

FIGURE 8. $\mathcal{A}$ semi-flower automaton, $\text{BPI}(\mathcal{A}) = \{1, 2, 3\}$, $L(\mathcal{A}) = \{aab, aaa, bab, baa, bb, ba\}^*$ and $rk(L(\mathcal{A})) = 6 > |\mathcal{F}| - |Q| + 1 = 4$.

We remark that a similar result holds for free groups (*cf.* [9]): if $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ is an inverse automaton associated to a subgroup H then $rk(H) = |\mathcal{F}| - |Q| + 1$. From now on let us consider A an alphabet of cardinality n.

We have defined a bpo of an automaton as a vertex in which there are at least two different edges going out from it. Let us consider now, for each $i \geq 0$, the bpo's of an automaton in which there are i different edges going out from them. Let $\mathcal{A} = (Q, I, T, \mathcal{F})$ be an automaton over A . For each vertex x of $\mathcal{A}$, let m_x be the number of edges going out from x. It is easy to see that $|\mathcal{F}| - |Q| = \sum_{x \in Q}(m_x - 1)$. For each $i \geq 0$, let us consider the set

$$BPO_i(\mathcal{A}) = \{x \in Q \mid m_x = i\}.$$

When no confusion arises we will write $BPO_i = BPO_i(\mathcal{A})$.

One has $Q = \cup_{i \geq 0} BPO_i$ and, for each $i \neq j$, $BPO_i \cap BPO_j = \emptyset$.

Let $m_{\mathcal{A}} = \max_{x \in Q}\{m_x\}$. It follows that

$$|\mathcal{F}| - |Q| = \sum_{i=0,\ldots,m_{\mathcal{A}}} \left(\sum_{x \in BPO_i} (m_x - 1) \right).$$

For each $i = 0, \ldots m_{\mathcal{A}}$, if $x \in BPO_i$ then $m_x - 1 = i - 1$ and we get

$$|\mathcal{F}| - |Q| = \sum_{i=0,\ldots,m_{\mathcal{A}}} |BPO_i|(i - 1).$$

If $\mathcal{A}$ is a monoidal automaton we get in particular the following:

Proposition 2.12. *Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a monoidal automaton. One has $|\mathcal{F}| - |Q| = \sum_{i=2,\ldots,m_{\mathcal{A}}} |BPO_i|(i - 1)$.*

Proof. Since $L(\mathcal{A}) \neq \emptyset$, $L(\mathcal{A}) \neq \{\varepsilon\}$ and $\mathcal{A}$ is trim with a unique final state equal to the initial one then $1 \leq m_x \leq m_{\mathcal{A}}$. □

If, moreover, $\mathcal{A}$ is a deterministic monoidal automaton on the alphabet A of cardinality n it trivially follows that:

Corollary 2.13. *Let A be an alphabet of cardinality n. Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a deterministic monoidal automaton. One has $|\mathcal{F}| - |Q| = \sum_{i=2,\ldots,n} |BPO_i|(i-1)$.*

Let us give now a theorem that links the rank of the submonoid generated by a given a semi-flower automaton with a unique bpi with the cardinality of the sets of BPO_i.

Theorem 2.14. *Let A be an alphabet of cardinality n. Let $\mathcal{A}$ be a semi-flower automaton on A with a unique bpi then*

$$rk(L(\mathcal{A})) \leq \sum_{i=2,\ldots,m_{\mathcal{A}}} (i-1)|BPO_i| + 1.$$

Moreover if $\mathcal{A}$ is unambiguous then it follows the equality and if $\mathcal{A}$ is deterministic then

$$rk(L(\mathcal{A})) = \sum_{i=2,\ldots,n} (i-1)|BPO_i| + 1.$$

Let us consider now deterministic semi-flower automata. The submonoids recognized by these automata are generated by finite prefix sets.

Proposition 2.15. *Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a deterministic semi-flower automaton. Then $\mathcal{A}$ recognizes a free submonoid generated by a finite prefix set.*

Proof. Let $\mathcal{A} = (Q, 1, 1, \mathcal{F})$ be a deterministic semi-flower automaton. By Propositions 2.2 and 2.4, $L(\mathcal{A})$ is a free finitely generated submonoid with base the set of labels of the simple cycles in 1. Let us prove that this is a prefix set. By contradiction, let u, v be labels of c_u and c_v, simple cycles in 1, such that $v = uw$ with w not empty word. The automaton $\mathcal{A}$ is deterministic so $c_v = c_u c_w$ with c_w the cycle with label w. This is a contradiction because c_v is simple in 1. □

In general given a submonoid X^* of A^* generated by a finite prefix set X we can easily construct an automaton recognizing X^*: the *literal automaton of* X^* (*cf.* [1]). It is the automaton

$$\mathcal{A}_X = (Q, \varepsilon, \varepsilon, \delta)$$

where $Q = X(A^+)^{-1}$ is the set of the proper prefixes of X and where

$$\delta(u, a) = \begin{cases} ua & \text{if } \ ua \in X(A^+)^{-1} \\ \varepsilon & \text{if } \ ua \in X \\ \emptyset & \text{otherwise.} \end{cases}$$

It is immediate that $L(\mathcal{A}_X) = X^*$.

Proposition 2.16. *Let X be a finite prefix set then $\mathcal{A}_X$ is a deterministic semi-flower automaton with at most the state ε as bpi.*

Proof. Let $\mathcal{A}_X = (Q, \varepsilon, \varepsilon, \delta)$. By construction $\mathcal{A}_X$ is a deterministic monoidal automaton. Since $L(\mathcal{A}_X) = X^*$ and X is finite then, by Proposition 2.5, $\mathcal{A}_X$ is

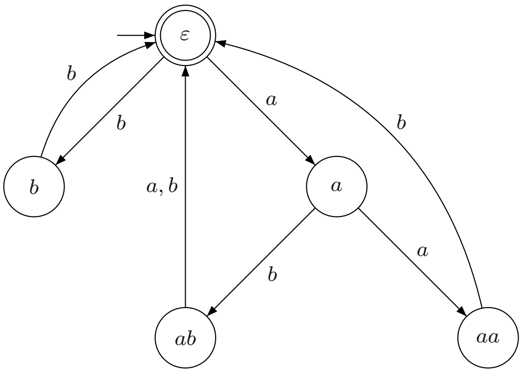

Figure 9. $\mathcal{A}_X$ with $X = \{bb, aba, abb, aab\}$.

a semi-flower automaton. If it has not bpi's then it is done. Let us suppose that there exists at least one bpi.

Let us suppose, by contradiction, that there exists $x \in Q$ bpi for $\mathcal{A}_X$ and $u \neq \varepsilon$. So in $\mathcal{A}_X$ there exist two different edges ending at x, $e_1 : y_1 \xrightarrow{a} x$ and $e_2 : y_2 \xrightarrow{b} x$. Since $x \neq \varepsilon$ it is $x = y_1 a = y_2 b$. Since a, b are letters in A it follows that $a = b$ then $y_1 = y_2$ and so $e_1 = e_2$, contradiction since $e_1 \neq e_2$. □

So we have that if $\mathcal{A}$ is a deterministic semi-flower automaton with at most one bpi then $L(\mathcal{A})$ is a submonoid generated by a finite prefix set and conversely given a finite prefix set X there exists a deterministic semi-flower automaton with at most the initial vertex as bpi recognizing X^*. So the class of deterministic semi-flower automata with at most the initial vertex as bpi recognizes the class of submonoids generated by finite prefix sets.

In the example in Figure 9 $\mathcal{A}_X$ is a deterministic semi-flower automaton with a unique bpi (the vertex ε), with three bpo's (the vertices ε, a and ab) and one has that $rk(X^*) = 4 = |BPO_2| \; + 1$.

Let us note that in all figures every edge with label a, b has to be understood as two edges with labels a and b, respectively.

3. Intersection of two submonoids

In this section we investigate the intersection of two submonoids of A^* finitely generated by prefix sets by studying the product of two deterministic semi-flower automata recognizing the two given submonoids.

For the definitions and the properties of the product automaton we recall [5]. It is well known that the product of two automata recognizing submonoids recognizes the intersection of the two submonoids.

Some properties of two given automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are saved in the product $\mathcal{A}_1 \times \mathcal{A}_2$ as shown in the following proposition.

Proposition 3.1. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two automata.*

(1) *If $\mathcal{A}_1$ and $\mathcal{A}_2$ are automata with a unique final state equal to a unique initial state then $\mathcal{A}_1 \times \mathcal{A}_2$ is an automaton with a unique final state equal to a unique initial state.*
(2) *If $\mathcal{A}_1$ and $\mathcal{A}_2$ are unambiguous automata then $\mathcal{A}_1 \times \mathcal{A}_2$ is a unambiguous automaton.*
(3) *If $\mathcal{A}_1$ and $\mathcal{A}_2$ are deterministic automata then $\mathcal{A}_1 \times \mathcal{A}_2$ is a deterministic automaton.*

On the other hand the product of two trim automata is not necessarily a trim automaton. Moreover the product of two semi-flower automata is not necessarily a semi-flower automaton (see example in Fig. 10). This is in agreement with the fact that the intersection of two finitely generated submonoids is not necessarily finitely generated. Further, if the product of two semi-flower automata with a unique bpi is a semi-flower automaton, then it is not necessarily a semi-flower automaton with a unique bpi (see example in Fig. 11).

Let now $\mathcal{A}_1$ and $\mathcal{A}_2$ be two deterministic automata on A

$$\mathcal{A}_1 = (Q_1, q_1, F_1, \delta_1),\ \mathcal{A}_2 = (Q_2, q_2, F_2, \delta_2).$$

The product automaton is defined as

$$\mathcal{A}_1 \times \mathcal{A}_2 = (Q_1 \times Q_2, (q_1, q_2), F_1 \times F_2, \delta)$$

with for each $(x, y) \in Q_1 \times Q_2$, for each $a \in A$

$$\delta((x, y), a) = (\delta_1(x, a), \delta_2(y, a)).$$

Let us now give a lemma that links the bpo's in the product of two deterministic automata with the bpo's of the respective automata:

Lemma 3.2. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two deterministic automata on A, alphabet of cardinality n. One has $BPO_t(\mathcal{A}_1 \times \mathcal{A}_2) \subseteq \cup_{t \leq r, s \leq n}(BPO_r(\mathcal{A}_1) \times BPO_s(\mathcal{A}_2))$.*

Proof. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two deterministic automata on $A = \{\alpha_1, \ldots, \alpha_n\}$, $\mathcal{A}_1 = (Q_1, q_1, F_1, \delta_1)$ and $\mathcal{A}_2 = (Q_2, q_2, F_2, \delta_2)$.

If $(x, y) \in BPO_t(\mathcal{A}_1 \times \mathcal{A}_2)$ then there exist t different edges going out from (x, y). So, for each $i = 1, \ldots, t$, there exists $(x_i, y_i) \in Q_1 \times Q_2$, $\alpha_i \in A$ such that

$$\delta((x, y), \alpha_i) = (x_i, y_i)$$

and, since $\mathcal{A}_1 \times \mathcal{A}_2$ is deterministic, for $i \neq j$ $\alpha_i \neq \alpha_j$.

By definition, for each $i = 1, \ldots, t$

$$\delta((x, y), \alpha_i) = (\delta(x, \alpha_i), \delta(y, \alpha_i)) = (x_i, y_i).$$

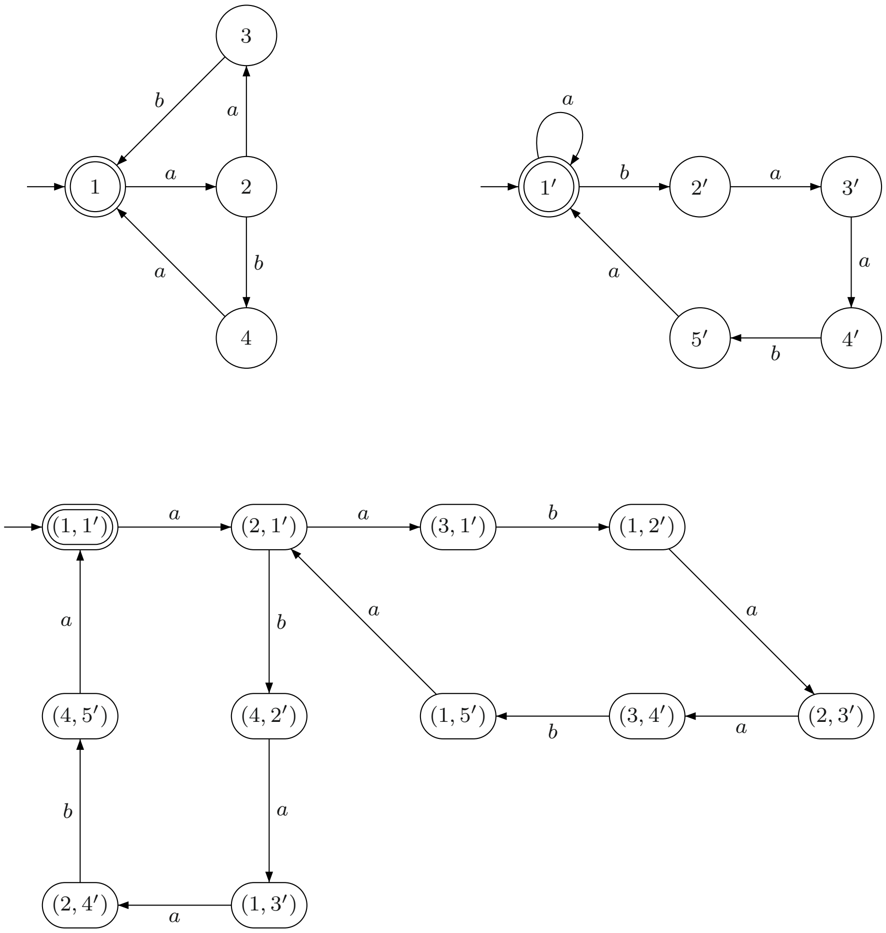

FIGURE 10. $\mathcal{A}_H$, $\mathcal{A}_K$ and $\mathcal{A}_H \times \mathcal{A}_K$ with $H = \{aab, aba\}^*$ and $K = \{a, baaba\}^*$.

Then for each $i = 1, \ldots, t$,

$$\delta(x, \alpha_i) = x_i, \quad \delta(y, \alpha_i) = y_i.$$

Since $\alpha_i \neq \alpha_j$, for each i, j with $i \neq j$, then there are at least t different edges going out from x and so $x \in BPO_r(\mathcal{A}_1)$ with $r \geq t$. Analogously $y \in BPO_s(\mathcal{A}_2)$ with $s \geq t$ and so the thesis follows. □

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be deterministic automata on A alphabet of cardinality n. Let $a_i := |BPO_i(\mathcal{A}_1)|$ and $b_i := |BPO_i(\mathcal{A}_2)|$ for each $i = 1, \ldots, n$. In terms of cardinality of the sets of bpo's Lemma 3.2 becomes:

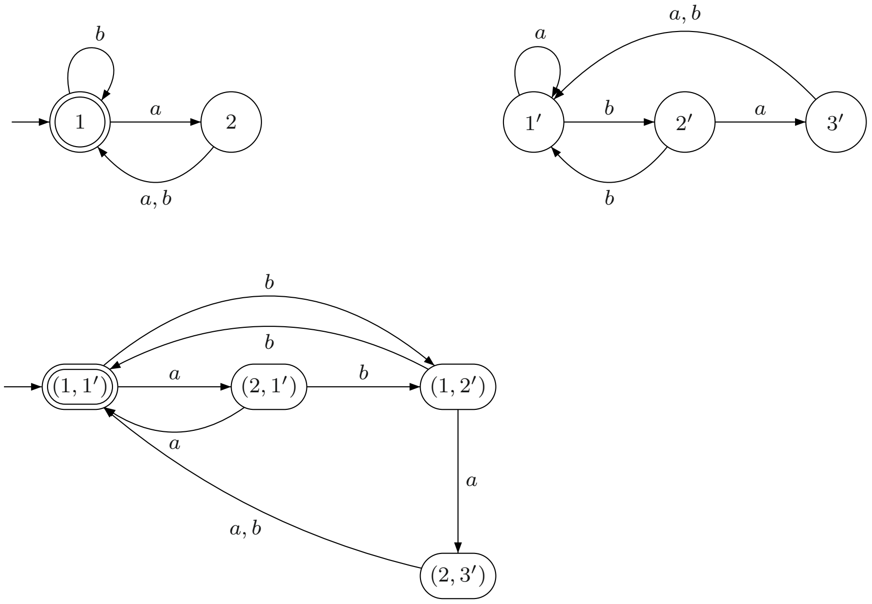

FIGURE 11. $\mathcal{A}_H$, $\mathcal{A}_K$ and $\mathcal{A}_H \times \mathcal{A}_K$ with $H = \{b, aa, ab\}^*$ and $K = \{a, bb, baa, bab\}^*$.

Corollary 3.3. $|BPO_t(\mathcal{A}_1 \times \mathcal{A}_2)| \leq \sum_{t \leq r, s \leq n} a_r b_s$.

Let $\mathcal{A}$ be an automaton. We denote by $\mathcal{A}^T$ the automaton obtained from $\mathcal{A}$ taking only the accessible and coaccessible states.

Let us give now a lemma that will allow us to prove the inequality of Hanna Neumann for submonoids recognized by deterministic semi-flower automata with a unique bpi. Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ be two multisets of the set of the natural numbers $\mathbb{N}$.
Let

$$P_{AB} = \sum_{t=2,\ldots,n} (t-1) \left(\sum_{t \leq r \leq n} a_r \sum_{t \leq s \leq n} b_s \right)$$

and

$$Q_{AB} = \left(\sum_{i=2,\ldots,n} (i-1) a_i \right) \left(\sum_{j=2,\ldots,n} (j-1) b_j \right).$$

Lemma 3.4. *Let* $A = \{a_1, \ldots, a_n\}$ *and* $B = \{b_1, \ldots, b_n\}$ *be multisets of* $\mathbb{N}$. *One has* $P_{AB} \leq Q_{AB}$. *Moreover if there exist* $k, l > 2$ *such that* $a_k \neq 0$ *and* $b_l \neq 0$ *then* $P_{AB} < Q_{AB}$.

Proof. Developing P_{AB} and Q_{AB} we get

$$P_{AB} = \sum_{2\leq k,l\leq n} \alpha_{kl} a_k b_l \quad \text{and} \quad Q_{AB} = \sum_{2\leq k,l\leq n} \beta_{kl} a_k b_l.$$

Our thesis is that $\alpha_{kl} \leq \beta_{kl}$, for each k, l. From the definitions of P_{AB} and Q_{AB} it follows that

$$\alpha_{kl} = 1 + 2 + \ldots + (\overline{k} - 1) \quad \text{and} \quad \beta_{kl} = (k-1)(l-1)$$

with $\overline{k} = \min(k, l)$.

One has the inequality $1 + 2 + \ldots + (\overline{k} - 1) < (\overline{k} - 1)(\overline{k} - 1)$ for $\overline{k} > 2$ and the equality $1 + 2 + \ldots + (\overline{k} - 1) = (\overline{k} - 1)(\overline{k} - 1)$ for $\overline{k} = 2$.

So $\alpha_{kl} \leq \beta_{kl}$, for each $k, l \geq 2$. Moreover $\alpha_{kl} < \beta_{kl}$ for $\overline{k} > 2$. So we get that $P_{AB} \leq Q_{AB}$.

Let us suppose that there exists $k > 2$ such that $a_k \neq 0$ and there exists $l > 2$ such that $b_l \neq 0$. Since $\alpha_{kl} < \beta_{kl}$ and $a_k b_l \neq 0$ we get $P_{AB} < Q_{AB}$. □

Let now H and K be submonoids finitely generated by prefix sets. Let $\mathcal{A}_H$ and $\mathcal{A}_K$ be deterministic semi-flower automata with a unique bpi recognizing H and K, respectively. Since $\mathcal{A}_H$ and $\mathcal{A}_K$ are deterministic monoidal automata then $(\mathcal{A}_H \times \mathcal{A}_K)^T$ is still a deterministic monoidal automaton.

Remark 3.5. If $(\mathcal{A}_H \times \mathcal{A}_K)^T$ is a semi-flower automaton that has not bpi's then $\widetilde{rk}(H \cap K) \leq \widetilde{rk}(H)\widetilde{rk}(K)$. In fact, if $H \cap K = \{\varepsilon\}$ then $\widetilde{rk}(H \cap K) = 0$ otherwise by Theorem 2.7, $H \cap K$ is cyclic and so $\widetilde{rk}(H \cap K) = 0$.

If $(\mathcal{A}_H \times \mathcal{A}_K)^T$ is a semi-flower automaton with a unique bpi we get the Hanna Neumann inequality as it is stated in the following theorem:

Theorem 3.6. *If $(\mathcal{A}_H \times \mathcal{A}_K)^T$ is a semi-flower automaton with a unique bpi then $\widetilde{rk}(H \cap K) \leq \widetilde{rk}(H)\widetilde{rk}(K)$. Moreover the strict inequality holds if there exist $i, j > 2$ such that $BPO_i(\mathcal{A}_H) \neq \emptyset$ and $BPO_j(\mathcal{A}_K) \neq \emptyset$.*

Proof. Let $(\mathcal{A}_H \times \mathcal{A}_K)^T$ be a deterministic semi-flower automaton with a unique bpi. Since $BPO_t((\mathcal{A}_H \times \mathcal{A}_K)^T) \subseteq BPO_t(\mathcal{A}_H \times \mathcal{A}_K)$, for each $t = 1, \ldots, n$, applying Corollary 3.3 and Theorem 2.14 we get:

$$\widetilde{rk}(H \cap K) \leq \sum_{t=2\ldots n} (t-1) \left(\sum_{t\leq r\leq n} a_r \sum_{t\leq s\leq n} b_s \right).$$

On the other hand by Theorem 2.14 it is

$$\widetilde{rk}(H)\widetilde{rk}(K) = \left(\sum_{i=2,\ldots n} (i-1) a_i \right) \left(\sum_{j=2,\ldots n} (j-1) b_j \right).$$

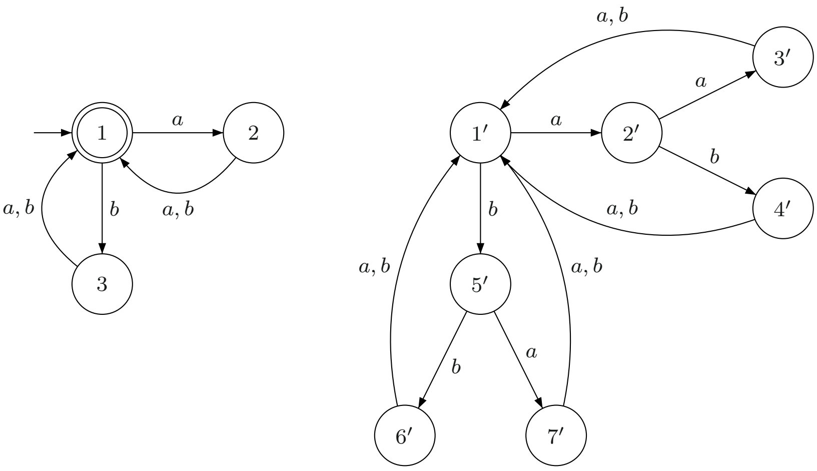

FIGURE 12. $\mathcal{A}_{A^2}$, $\mathcal{A}_{A^3}$ with A^2 the set of words in $A = \{a, b\}$ of length 2 and A^3 the set of words in $A = \{a, b\}$ of length 3.

Applying Lemma 3.4 to the multisets $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$ we get that $\widetilde{rk}(H \cap K) \leq \widetilde{rk}(H)\widetilde{rk}(K)$.

Let us suppose that there exist two indexes $i, j > 2$ such that $BPO_i(\mathcal{A}_H) \neq \emptyset$ and $BPO_j(\mathcal{A}_K) \neq \emptyset$. Then $|BPO_i(\mathcal{A}_H)| = a_i \neq 0$, $i > 2$ and $|BPO_j(\mathcal{A}_H)| = b_j \neq 0$, $j > 2$. And by Lemma 3.4 we get that $\widetilde{rk}(H \cap K) < \widetilde{rk}(H)\widetilde{rk}(K)$. □

If H and K are submonoids finitely generated by prefix sets such that $(\mathcal{A}_H \times \mathcal{A}_K)^T$ is a deterministic semi-flower automaton with more than one bpi then it is not more true that $\widetilde{rk}(H \cap K) \leq \widetilde{rk}(H)\widetilde{rk}(K)$.

There is a family of examples such that $rk(H \cap K) = 2^{\log_2(rk(H)) \log_2(rk(K))}$:

Example 3.7. Let p and q be two positive coprime integers. Let A be a binary alphabet and let $H = (A^p)^*$ and $K = (A^q)^*$, the submonoids generated by the sets of words of length p and q respectively.
One has $rk(H) = 2^p$ and so $p = \log_2(rk(H))$. It is $H \cap K = (A^{pq})^*$ and $rk(H \cap K) = 2^{pq} = 2^{\log_2(rk(H)) \log_2(rk(K))}$.

See examples in Figures 12 and 13.

4. Prefix case with two generators

If H and K are submonoids generated by prefix sets of two elements of A^* we get the result of Karhumäki (*cf.* [7]).

Let H be a submonoid of A^* generated by a finite prefix set of two elements X. Let $\mathcal{A}_X$ be the literal automaton of X^*, let us call it $\mathcal{A}_H$ for simplicity of notation. In the beginning we prove that $\mathcal{A}_H$ has a unique bpo with two edges starting at it and all the other vertices have just one edge starting at them.

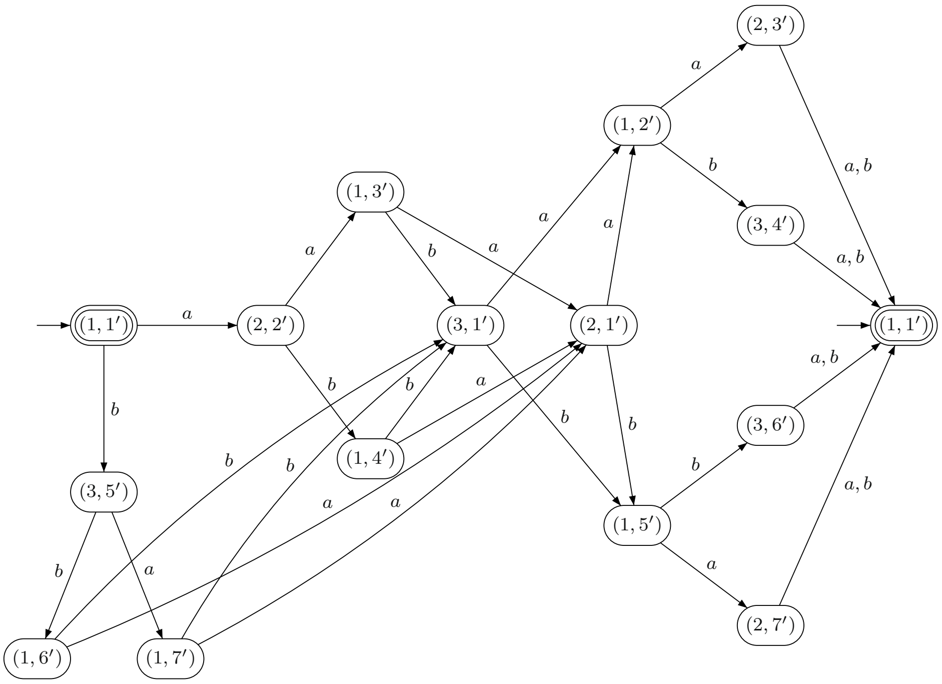

FIGURE 13. $\mathcal{A}_{A^2} \times \mathcal{A}_{A^3}$.

Proposition 4.1. *Let H be a submonoid generated by a prefix set of two elements. Let $\mathcal{A}_H = (Q, 1, 1, \delta)$ be the literal automaton of H.*

For each $x \in Q$, it is $1 \leq m_x \leq 2$. Moreover there is at most one x such that $m_x = 2$.

Proof. Let H be a submonoid finitely generated by a prefix set of two elements $X = \{u, v\}$. Let $\mathcal{A}_H = (Q, 1, 1, \delta)$ be the literal automaton of H. Since $\mathcal{A}_H$ is monoidal then $1 \leq m_x$. Moreover each state has at more two edges going out from it otherwise, since $\mathcal{A}_H$ is deterministic semi-flower, there would be three free generators for H. Finally there is at most one state such that $m_x = 2$ otherwise there would be four free generators for H. □

For the product automaton of two literal automata, associated to submonoids generated by prefix sets of two elements, it holds the following:

Lemma 4.2. *Let H and K be submonoids generated by prefix sets of two elements such that $H \cap K \neq \{\varepsilon\}$. Let $(\mathcal{A}_H \times \mathcal{A}_K)^T = (Q, 1, 1, \delta)$.*

For each $w \in Q$, it is $1 \leq m_w \leq 2$. Moreover there is at most one vertex $z \in Q$ such that $m_z = 2$.

Proof. Since $(\mathcal{A}_H \times \mathcal{A}_K)^T$ is monoidal and $H \cap K \neq \{\varepsilon\}$ then, for each $x \in Q$, it is $m_x \geq 1$. Let $\mathcal{A} = (\mathcal{A}_H \times \mathcal{A}_K)^T$. Let $a_i := |BPO_i(\mathcal{A}_H)|$ and $b_i := |BPO_i(\mathcal{A}_K)|$ for each $i = 1, \ldots, n$.

By Corollary 3.3 we have that for each $t \geq 0$

$$|BPO_t(\mathcal{A}^T)| \leq \sum_{t \leq r,s \leq n} a_r b_s.$$

By Proposition 4.1 we get $a_2 = 1$, $b_2 = 1$ and for each $s > 2$ $a_s = 0$ and $b_s = 0$.

For each $t > 2$ we obtain $|BPO_t(\mathcal{A}^T)| \leq 0$ and so, for each $x \in Q$, $m_x \leq 2$. For $t = 2$ we get $|BPO_2(\mathcal{A}^T)| \leq a_2 b_2 = 1$ and so there is only one $x \in Q$ such that $m_x = 2$. □

We can now prove the result of Karhumäki (*cf.* [7]):

Theorem 4.3. *Let H and K be submonoids generated by prefix sets of two elements. Then either $H \cap K$ is generated by at most two elements or by a regular language of the form $(\alpha(\beta)^*\gamma)^*$, for some $\alpha, \beta, \gamma \in A^*$.*

Proof. Let H and K be submonoids generated by prefix sets of two elements. Let us denote by $\mathcal{A}$ the product automaton $(\mathcal{A}_H \times \mathcal{A}_K)^T = (Q, 1, 1, \mathcal{F})$.

By Lemma 4.2 there is at most one state in $\mathcal{A}$ with two edges going out from it and all the others states have only one edge going out from them.

If $\mathcal{A}$ has not bpo's then, by Proposition 2.6 and since $\mathcal{A}$ is a monoidal automaton, $\mathcal{A}$ has not bpi's. So either $H \cap K = \{\varepsilon\}$ or, by Proposition 2.7, $H \cap K$ is a cyclic submonoid. In all cases $rk(H \cap K)$ is at most one.

Let us suppose now that $\mathcal{A}$ has one bpo and let us denote it by x. Let p be the unique simple path from 1 to x. There is a simple path starting at x and ending at 1 since x is coaccessible, let us call it q.

By Lemma 4.2, since $\mathcal{A}$ is trim, it follows that there is only one simple path, different from q, starting at x and ending at a state y of pq.

If y is a state of q different from x then $\mathcal{A}$ is a semi-flower automaton and y is the unique bpi. Applying Theorem 3.6 we get that the intersection is generated by at most two elements.

If y is a state of p different from 1 then $\mathcal{A}$ is not a semi-flower automaton since there is a cycle in y not visiting 1. So $H \cap K$ is infinitely generated. Let us call p_1 the prefix subpath of p ending at y and p_2 the suffix subpath of p starting at y. We have that:

$$p_1 : 1 \xrightarrow{\alpha_1} y, \quad p_2 : y \xrightarrow{\alpha_2} x, \quad q : x \xrightarrow{\gamma} 1, \quad r : x \xrightarrow{\beta} y.$$

Let $\alpha = \alpha_1\alpha_2$. We have that $p(rp_2)^n q$, for each $n \geq 0$, are the only cycles that are simple in 1. So, by Proposition 2.2, one has that $H \cap K$ is generated by $\alpha(\beta\alpha_2)^n\gamma$, for each $n \geq 0$, that is the thesis. □

It follows trivially the same result for H and K submonoids finitely generated by suffix sets of two elements.

5. Conclusions

The results reported in the present paper show that the automata-theoretic tools we have introduced are useful for studying the intersection of two submonoids of a free monoid. We think that such methods could be extended to more general cases than the ones considered in this paper, in order to discover upper bounds on the rank of the intersection, when it is finite, depending on the ranks of the submonoids.

Two directions of research appear of particular interest. A first direction is to relate properties of the product of non deterministic automata to the study of the intersection of two submonoids generated by finite non prefix sets of words. Another possible direction is to consider, even in the prefix case, the semi-flower automata with a fixed number of bpi's greater than one.

As a more general problem, one could study the intersection of a finite number of submonoids of rank two trying to discover in the case of finite rank of the intersection a result analogous of that one of Karhumäki (*cf.* [7]).

References

[1] J. Berstel and D. Perrin. *Theory of Codes*. Academic Press (1985).

[2] V. Bruyére, D. Derencourt, M. Latteux. The meet operation in the lattice of codes. *Theoretical Computer Science* **191** (1998) 117–129.

[3] J. Clement, J. Duval, G.Guaiana, D. Perrin, G. Rindone. Paarsing with a finite dictionary. *Theoretical Computer Science* **340** (2005) 432–442.

[4] H.Cormen, E. Leiserson, L. Rivest. *Introduction to Algorithms.* The MIT Press (1990).

[5] J.E. Hopcroft, J.D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Weisley Publishing Company (1979).

[6] A.G. Howson. On the intersection of finitely generated free groups. *J. London Math. Soc.* **29** (1954) 428–434.

[7] J. Karhumäki. A note on intersection of free submonoids of a free monoid. *Semigroup Forum* **29** (1984) 183–205.

[8] M. Latteux and J. Leguy. On the composition of morphism and inverse morphisms. *Lecture Notes in Computer Science* **154** (1983) 420–432.

[9] J. Meakin and P. Weil. Sugroups of free groups: a contribution to the Hanna Neumann conjecture. *Geometriae Dedicata* **94** (2002) 33–43.

[10] H. Neumann. On intersections of finitely generated subgroups of free groups. *Publ. Math. Debrecen* **4** (1956) 186–189.

[11] W.D. Neumann. On intersections of finitely generated subgroups of free groups. *Lect. Notes Math.* **1456** (1990) 161–170.

[12] B. Tilson. The intersection of free submonoids of a free monoid is free. *Semigroup Forum* **4** (1972) 345–350.