



**HAL**  
open science

# Algebras, automata and logic for languages of labeled biregular trees

David Janin

► **To cite this version:**

David Janin. Algebras, automata and logic for languages of labeled biregular trees. 40th International Colloquium on Automata, Languages and Programming (ICALP), Jul 2013, Riga, Latvia. hal-00784898v1

**HAL Id: hal-00784898**

**<https://hal.science/hal-00784898v1>**

Submitted on 4 Feb 2013 (v1), last revised 25 Apr 2013 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



LaBRI, CNRS UMR 5800  
Laboratoire Bordelais de Recherche en Informatique

Rapport de recherche RR-1467-13

Algebras, automata and logic for languages of labeled  
birooted trees

February 4, 2013

David Janin,  
LaBRI, IPB, Université de Bordeaux



# Algebras, automata and logic for languages of labeled birooted trees

David Janin

Université de Bordeaux, LaBRI UMR 5800,  
351, cours de la Libération,  
F-33405 Talence, FRANCE  
[janin@labri.fr](mailto:janin@labri.fr)

**Abstract.** In this paper, we study languages of labeled finite birooted trees: Munn’s birooted trees extended with vertex labeling. We define a notion of finite state birooted tree automata that is shown to capture the class of languages that are upward closed w.r.t. the natural order and definable in Monadic Second Order Logic. Then, relying on the inverse monoid structure of labeled birooted trees, we derive a notion of recognizable languages by means of (adequate) premorphisms into finite (adequately) ordered monoids. This notion is shown to capture finite boolean combinations of languages as above. As a particular case, a simple encoding of finite (mono-rooted) labeled trees in an antichain of labeled birooted trees shows that classical regular languages of finite (mono-rooted) trees are also recognized by such premorphisms and finite ordered monoids.

## Introduction

### Motivations and background

Semigroup theory have demonstrated for years its considerable strength for the study and the fine grain analysis of languages of finite words, that is subsets of the free monoid  $A^*$ . Among the simplest known results, this can be illustrated by the fact that a language  $L \subseteq A^*$  is regular if and only if there is a finite monoid  $S$  and a monoid morphism  $\theta : A^* \rightarrow S$  such that  $L = \theta^{-1}(\theta(L))$ . In this case, we say that the language  $L$  is *recognized* by the finite monoid  $S$  (and the morphism  $\theta$ ).

Even better, for every language  $L \subseteq A^*$ , this notion of recognizability induces a notion of syntactic congruence  $\simeq_L$  for the language  $L$  such that the monoid  $S^L = A^* / \simeq_L$  is the smallest monoid that recognizes  $L$ . Many structural properties of the language  $L$  can be decided by analyzing the properties of its syntactic monoid  $S^L$ , e.g. regularity, start freeness, etc (see [11] for some more examples of such properties).

These results motivated the development of entire algebraic theories of languages of various structures by considering richer algebraic frameworks such as, among many others,  $\omega$ -semigroups for languages of infinite words [16, 9],

preclones or forest algebra for languages of trees [3, 2], or  $\omega$ -hyperclones for languages of infinite trees [1]. Aiming at deciding more subtle properties of languages, several extensions of the notion of recognizability by monoids and morphisms were also considered, e.g. recognizability by monoids and relational morphisms [10] or recognizability by ordered monoids and monotonic morphisms [12].

A recent study of languages of overlapping tiles [7, 5, 4], that is subsets of the (inverse) monoid of McAlister [8], has led to consider the notion of quasi-recognizability : recognizability by means of (adequate) premorphisms into (adequately ordered) ordered monoids. As (monotonic) morphisms are particular case of premorphisms, this notion can be seen as a generalization of recognizability by (ordered) monoids and (monotonic) morphisms [12].

However, the notion of quasi-recognizability is yet not settled for one need to restrict both the class of allowed premorphisms and the class finite ordered monoids for that notion to be effective. Without any restriction, the inverse image by a premorphism of a finite subset of a finite ordered monoid may even not be computable. Even more, there are several incomparable candidates for defining such a notion as illustrated, for instance, by a recent study of walking birooted tree automata with invisible pebbles [6].

We aim in this paper at stabilizing the notion of recognizability by adequate premorphisms by applying it to the study of languages of labeled birooted trees. Doing so, it appears that this notion admits a robust logical characterization in terms of languages of birooted trees definable in Monadic Second Order Logic: a typical yardstick to measure and analyse the expressiveness of notions of definability.

## Outline

Birooted labeled trees, called birooted  $F$ -trees, are presented in Section 1. Equipped with an extension of Scheiblich’s product of (unlabeled) birooted trees [13], the resulting structure is an inverse monoid. As such, birooted trees can be ordered by the *natural order* relation that is stable under product: the inverse monoid  $\mathcal{B}(F)$  of labeled birooted  $F$ -trees is also a partially ordered monoid.

Birooted tree automata are defined and studied in Section 2. By construction, languages recognized by these finite automata are upward closed in the natural order. It follows that they fail to capture languages definable by means of Monadic Second Order (MSO) formulae. However, this loss of expressive power is shown to be limited to that upward closure properties. Indeed, we prove (Theorem 2) that every language of birooted trees upward closed and MSO definable is recognized by a finite state birooted tree automata.

As a particular case, when  $F$  is seen a functional signature, embedding the classical  $F$ -terms (see [15]) into birooted  $F$ -trees, we show (Theorem 3) that the birooted tree image of every regular language  $L$  of  $F$ -terms is of the form  $U_L \cap D_L$  with  $U_L$  (resp.  $D_L$ ) some upward (resp. downward) closed MSO definable language.

The algebraic counterpart of birooted tree automata is presented in Section 3 where the notion of adequately ordered monoids and adequate premorphisms are defined. The induced notion of quasi-recognizable languages of birooted  $F$ -trees is shown to be effective (Theorem 4).

Concerning expressive power, it is shown that every birooted tree automaton simply induces an adequate premorphism that recognizes the same language (Theorem 5) and that every quasi-recognizable language is MSO definable (Theorem 6). The picture is made complete by proving (Theorem 7) that quasi-recognizable languages of birooted trees exactly correspond to finite boolean combinations of upward closed MSO definable languages.

Together with Theorem 3, this last result tells that our proposal can also be seen as yet another algebraic characterization of regular languages of trees that complete the one previously obtained by means of preclones [3], forest algebras [2] or ordered monoids and admissible premorphisms [6].

## Related works

It must be mentioned that the notion of birooted  $F$ -tree automaton defined above extends the one already defined [4] for languages of one-dimensional overlapping tiles that is subsets of McAlister monoids [8].

Though strongly related, one can observe that such an extension is by no mean straightforward. Of course going from the linear structure of overlapping tiles to the tree shaped structure of birooted  $F$ -trees already induces a real increase of complexity. However, the main difference comes from directed edges. In overlapping tiles, they all goes in the same direction while, in birooted  $F$ -trees, edges may (almost) arbitrarily go back and forth. Proving Theorem 2 is thus much more involved than proving an analogous result for overlapping tiles.

Comparing our proposal with other known algebraic characterizations of languages of (mono-rooted)  $F$ -trees [3, 2] is not easy.

At first sight, our proposal seems to induce a bigger class of definable languages since we deal with birooted  $F$ -trees. However, a more relevant comparison would be to compare the classification of languages one can obtain, in every approach, by restricting even further the allowed recognizers: be them preclones as in [3], forest algebras [2] or adequately ordered monoid as proposed here. Though quasi-recognizability could lead to simpler extension of the known tools for languages of words - recognizers are just finite monoids - the theory is yet not sufficiently developed for such a comparison to be possible.

Another source of difficulty comes from the fact that adequate premorphisms are *not* morphisms : only (some notion of) *disjoint products* are preserved by premorphisms. To some extent, the notion of quasi-recognizability by premorphisms presented here is, compared to classical recognizability by morphisms, what unambiguous non deterministic automata are compared to deterministic automata. On the negative side, this means that the notion of quasi-recognizability is yet not completely understood. On the positive side, this means that it may lead to radically new outcomes.

## 1 Semigroups and monoids of birooted $F$ -trees

Simply said, a labeled birooted tree is a (non empty) finite connected subgraph of the Cayley graph of the free group  $FG(A)$  with labeled vertices on some finite alphabet  $F$  and with two distinguished vertices respectively called the input root and the output root of the labeled graph. This definition and some of the associated properties are detailed in this section.

### 1.1 The free groups

Formally, let  $A$  be a finite (edge) alphabet and let  $\bar{A}$  be a disjoint copy of  $A$  with, for every letter  $a \in A$ , its copy  $\bar{a} \in \bar{A}$ . Let  $u \mapsto \bar{u}$  be the mapping from  $(A + \bar{A})^*$  to itself inductively defined by  $\bar{1} = 1$  and  $\overline{u\bar{a}} = \bar{a} \bar{u}$  and  $\overline{a\bar{u}} = a \bar{u}$ , for every  $u \in (A + \bar{A})^*$ , every  $a \in A$ . This mapping is involutive, i.e.  $\overline{\bar{u}} = u$  for every  $u \in (A + \bar{A})^*$ , and it is an anti-morphism, i.e.  $\overline{uv} = \bar{v} \bar{u}$  for every word  $u$  and  $v \in (A + \bar{A})^*$ .

The free group  $FG(A)$  generated by  $A$  is the quotient of  $(A + \bar{A})^*$  by the least congruence  $\simeq$  such that, for every letter  $a \in A$ ,  $a\bar{a} \simeq 1$  and  $\bar{a}a \simeq 1$ . This is indeed a group since, for every  $u \in (A + \bar{A})^*$ , we have  $[u][\bar{u}] = [1]$  hence  $[\bar{u}]$  is the group inverse of  $[u]$ .

It is known that every class  $[u] \in FG(A)$  contains a unique element  $red(u)$  (the reduced form of  $u$ ) that contains no factors of the form  $a\bar{a}$  nor  $\bar{a}a$  for  $a \in A$ . In the sequel, every such class  $[u] \in FG(A)$  is thus represented by its reduced form  $red(u)$ . Doing so, the product  $u \cdot v$  of every two reduced words  $u$  and  $v \in FG(A)$  is directly defined by  $u \cdot v = red(uv)$ .

Elements of  $FG(A)$ , when seen as reduced words, can then be ordered by the prefix order relation  $\leq_p$  defined, for every (reduced word)  $u$  and  $v \in FG(A)$  by  $u \leq_p v$  when there exists (a reduced word)  $w \in FG(A)$  such that  $uw = v$ . The associated predecessor relation  $\prec_p$  is defined, for every  $v$  and  $w \in FG(A)$ , by  $v \prec_p w$  when  $v <_p w$  and  $w = vx$  for some  $x \in A + \bar{A}$ .

### 1.2 Labeled birooted trees

A labeled birooted tree on the edge alphabet  $A$  and the vertex alphabet  $F$  is a pair  $B = \langle t, v \rangle$  where  $t : FG(A) \rightarrow F$  is a partial maps which domain  $dom(t)$  is a prefix closed subset of  $FG(A)$  such that  $v \in dom(t)$ .

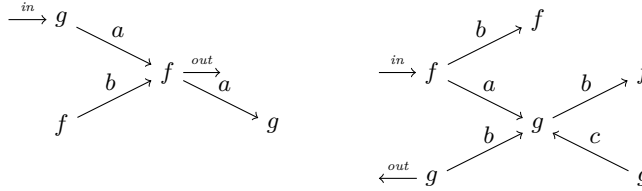
In such a presentation,  $1 \in dom(t)$  is the input root vertex and  $u \in dom(t)$  is the output root vertex. Assuming the edge alphabet  $A$  is implicit, these labeled birooted trees are called *birooted  $F$ -trees* or, when  $F$  is also implicit, simply *birooted trees*.

Examples of birooted  $F$ -trees are depicted in Figure 1 below. With a dangling input edge marking the input root and a dangling output edge marking the output root.

For every birooted tree  $B = \langle t, u \rangle$ , for every  $v \in dom(t)$ , let  $t_v : FG(A) \rightarrow F$  be the partial function defined by  $dom(t_v) = \bar{v} \cdot dom(t)$  and  $t_v(vw) = t(uvw)$

for every  $w \in \text{dom}(t_v)$ . Accordingly, let  $B_v = \langle t_u, \bar{v}u \rangle$  be the  $v$  translation of the birooted tree  $B$ .

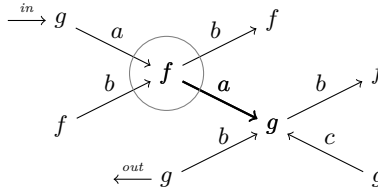
Observe that such a translation slightly differs from the classical notion of subtrees since  $\text{dom}(t_v) = \bar{v} \cdot \text{dom}(t)$  contains as many vertices as  $\text{dom}(t)$ . A notion of sub-birooted tree  $B_v^p$ , with fewer vertices and thus more closely related with the classical notion of subtree, is defined below when proving a decomposition property (Lemma 1).



**Fig. 1.** Two birooted  $F$ -trees  $B_1$  and  $B_2$

The partial product  $\langle r, u \rangle \cdot \langle s, v \rangle$  of two birooted  $F$ -tree  $\langle r, u \rangle$  and  $\langle s, v \rangle$  is defined, when it exists, as the birooted  $F$ -tree  $\langle t, w \rangle$  defined by  $w = u \cdot v$ ,  $\text{dom}(t) = \text{dom}(r) \cup u \cdot \text{dom}(s)$ ,  $t(u') = r(u')$  for every  $u' \in \text{dom}(r)$  and  $t_u(v') = s(v')$  for every  $v' \in \text{dom}(s)$ .

Observe that such a product exists if and only if the tree  $r_u$  and the tree  $s$  agree on  $\text{dom}(r_u) \cap \text{dom}(s)$ , i.e. for every  $v' \in \text{dom}(r_u) \cap \text{dom}(s)$ , we have  $r_u(v') = r(uv') = s(v')$ . It follows that undefined products may arise when  $F$  is not a singleton.



**Fig. 2.** The product  $B_1 \cdot B_2$  of the two birooted  $F$ -trees  $B_1$  and  $B_2$

A (defined) product of two birooted  $F$ -trees is depicted in Figure 2 above. In that picture, the circle marks the synchronization vertex that results from the merging of the output root of  $B_1$  and the input root of  $B_2$ . The  $a$ -labeled edge  $f \xrightarrow{a} g$  emanating from that vertex is the common edge resulting from the fusion of the two (synchronized) birooted  $F$ -trees.



The product is completed by adding a zero element for the undefined case with  $0 \cdot \langle t, v \rangle = \langle t, v \rangle \cdot 0 = 0 \cdot 0 = 0$  for every (defined) birooted tree  $\langle t, v \rangle$ .

One can easily check that the resulting product is associative. The resulting structure is thus a semigroup denoted by  $\mathcal{B}(F)$ : the *semigroup of birooted  $F$ -trees*.

In the case  $F$  is a singleton, every birooted  $F$ -tree can be redefined as a pair  $(P, u)$  with an non empty prefix closed domain  $P \subseteq FG(A)$  and an output root  $u \in P$ . Then, following Scheiblich presentation [13], the semigroup  $\mathcal{B}(F)$  is the free monoid  $FIM(A)$  generated by  $A$ . When  $F$  is not a singleton, we extend the set  $\mathcal{B}(F)$  with a unit denoted by 1. The resulting structure is a monoid  $\mathcal{B}^1(F)$  : the *monoid of birooted  $F$ -trees*.

The monoid of birooted  $F$ -trees is an *inverse monoid*, i.e. for every  $B \in \mathcal{B}^1(F)$  there is a unique  $B^{-1} \in \mathcal{B}^1(F)$  such that  $BB^{-1}B = B$  and  $B^{-1}BB^{-1} = B^{-1}$ . Indeed, we necessarily have  $0^{-1} = 0$ ,  $1^{-1} = 1$  and, for every non trivial birooted  $F$ -tree  $\langle t, u \rangle$  one can check that  $\langle t, u \rangle^{-1} = \langle t_u, \bar{u} \rangle$ .

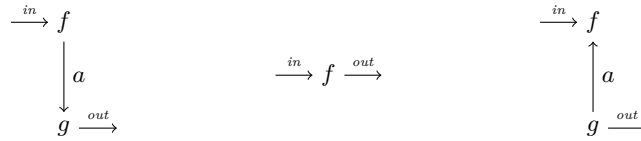
As an inverse monoid, elements of  $\mathcal{B}^1(F)$  can be ordered by the *natural order* defined, for every  $B$  and  $C \in \mathcal{B}^1(F)$  by  $B \leq C$  when  $B = BB^{-1}C$  (equivalently  $B = CB^{-1}B$ ). One can check that 0 is the least element and, for every defined birooted  $F$ -trees  $\langle r, u \rangle$  and  $\langle s, v \rangle$  we have  $\langle r, u \rangle \leq \langle s, v \rangle$  if and only if  $u = v$ ,  $dom(r) \supseteq dom(s)$  and, for every  $w \in dom(s)$ ,  $t(w) = s(w)$ .

Observe that, as far as trees only are concerned, the natural order is the reverse of the (often called) prefix order on trees. In particular, the bigger is the size of a birooted tree, the smaller is the birooted tree in the natural order.

### 1.3 Strong decomposition of birooted trees

One can easily check that the monoid of birooted  $F$ -trees is finitely generated. We prove here a stronger statement that will be extensively used in the remainder of the text.

A birooted tree is said *elementary* when it is either 0 or 1, or of the form  $B_f = \langle \{1 \mapsto f\}, 1 \rangle$  for some  $f \in F$  or of the form  $B_{fag} = \langle \{1 \mapsto f, x \mapsto g\}, 1 \rangle$  for some vertex label  $f$  and  $g \in F$  and some letter  $x \in A + \bar{A}$ .



**Fig. 3.** The elementary birooted  $F$ -trees  $B_{fag}$ ,  $B_f$  and  $B_{f\bar{a}g}$

The *left projection*  $B^L$  (resp. the *right projection*  $B^R$ ) of a birooted tree  $B = \langle t, u \rangle$  is defined by  $B^L = B^{-1}B$  (resp.  $B^R = BB^{-1}$ ) or, equivalently,  $B^L = \langle t_u, 1 \rangle$  (resp.  $B^R = \langle t, 1 \rangle$ ).

The product  $B_1 \cdot B_2$  of two birooted trees  $B_1$  and  $B_2$  is a *disjoint product* when  $B_1 \cdot B_2 \neq 0$  and 1 is the unique birooted  $F$ -tree such that  $B_1^L \leq 1$  and  $B_2^R \leq 1$ , i.e.  $B_1^L \vee B_2^R = 1$ .

This restricted product is called a disjoint product because, when  $B_1 = \langle t_1, u_1 \rangle$  and  $B_2 = \langle t_2, u_2 \rangle$ , the product  $B_1 \cdot B_2$  is a disjoint product if and only if  $t(u_1) = t_2(1)$  and  $\text{dom}(t_1) \cap u_1 \cdot \text{dom}(t_2) = \{u_1\}$ , i.e. the set of edges in  $B_1 \cdot B_2$  is the disjoint union of the set of edges of  $B_1$  and the set of (translated) edges of  $B_2$ .

**Lemma 1 (Strong decomposition).** *For every  $B \in \mathcal{B}(F)$ , the birooted  $F$ -tree  $B$  can be decomposed into a finite combination by disjoint product and right resets of elementary birooted trees.*

*Proof.* Let  $B = \langle t, u \rangle$  be a birooted  $F$ -tree. We aim at proving it can be decomposed as stated above. We first define some specific sub-birooted trees of  $B$  that will be used for such a decomposition.

For every vertex  $v$  and  $w \in \text{dom}(t)$  such that  $v \prec_p w$ , let  $B_{v,w}^p$  be the two vertices birooted  $F$ -tree defined by  $B_{v,w}^p = B_{fxg}$  where  $f = t(v)$ ,  $g = t(w)$  and  $vx = w$ .

Let  $U = \{v \in \text{dom}(t) : 1 \leq_p v \leq_p u\}$  be the set of vertices that appears on the shortest path from the input root 1 to the output root  $u$ . For every  $v \in \text{dom}(t)$ , let  $D^p(v) = \{w \in \text{dom}(t) : v \leq_p w, (v = w \vee w \notin U)\}$  be the set of vertices above (on the prefix order)  $v$  and, except from  $v$  itself, that are not in  $U$ . Then, for every  $v \in \text{dom}(t)$ , let  $B_v^p = \langle t_v | D^p(v), 1 \rangle$  be the idempotent birooted tree obtained from  $B$  by restricting the subtree  $t_v$  rooted at the vertex  $v$  to the domain  $D^p(v)$ .

Then, given  $u_0 = 1 \prec_p u_1 \prec_p u_2 \prec_p \dots \prec_p u_{n-1} \prec_p u_n = u$  the increasing sequence (under the prefix order) of all the prefixes of the output root  $u$ , we have

$$B = B_{u_0}^p B_{u_0, u_1}^p B_{u_1}^p \cdots B_{u_{n-1}}^p B_{u_{n-1}, u_n}^p B_{u_n}^p$$

with only disjoint products.

In order to conclude the proof, It remains to prove, by induction on the size of birooted trees, that every idempotent sub-birooted tree of the form  $B_v^p$  for some  $v \in \text{dom}(t)$  can also be decomposed into a finite combination by disjoint product and right projection of elementary birooted trees.

Let then  $v \in \text{dom}(t)$ . In the case  $v$  is a leaf (w.r.t. the prefix order) then  $B_v^p = B_{t(v)}$  and we are done. Otherwise, we have  $B_v^p = \langle r, 1 \rangle$  for some  $F$ -tree  $r$  and we observe that

$$B_v^p = \prod \{(B_{v,w}^p \cdot B_w^p)^R : w \in \text{dom}(r), v \prec_p w\}$$

with only disjoint products. This concludes the proof.  $\square$

The above decomposition of  $B$  as a combination of elementary birooted trees by disjoint products and right projections is called a *strong decomposition* of the birooted  $F$ -tree  $B$ .

## 2 Birooted $F$ -tree automata

In this section, we define the notion of birooted  $F$ -tree automata that is shown to capture the class of languages of birooted  $F$ -trees that are upward closed w.r.t. the natural order and definable in Monadic Second Order Logic (MSO).

### 2.1 Definition

A *birooted  $F$ -tree (finite) automaton* is a quintuple  $\mathcal{A} = \langle Q, \delta, \Delta, W \rangle$  defined by a (finite) set of states  $Q$ , a (non deterministic) state table  $\delta : F \rightarrow \mathcal{P}(Q)$ , a (non deterministic) transition table  $\Delta : A \rightarrow \mathcal{P}(Q \times Q)$  and an acceptance condition  $W \subseteq Q \times Q$ .

A *run* of the automaton  $\mathcal{A}$  on a non trivial birooted  $F$ -tree  $B = \langle t, u \rangle$  is a mapping  $\rho : \text{dom}(t) \rightarrow Q$  such that for every  $v \in \text{dom}(t)$ :

- ▷ State coherence:  $\rho(v) \in \delta(t(v))$ ,
- ▷ Transition coherence: for every  $a \in A$ , if  $va \in \text{dom}(t)$  then  $(\rho(v), \rho(va)) \in \Delta(a)$  and if  $v\bar{a} \in \text{dom}(t)$  then  $(\rho(v\bar{a}), \rho(v)) \in \Delta(a)$ .

The run  $\rho$  is an *accepting run* when  $(\rho(1), \rho(u)) \in W$ . The set  $L(\mathcal{A}) \subseteq \mathcal{B}(F)$  of birooted  $F$ -tree  $B$  such that there is an accepting run of  $\mathcal{A}$  on  $B$  is the language recognized by the automaton  $\mathcal{A}$ .

### 2.2 Expressive power

Every non trivial birooted  $F$ -tree  $B = \langle t, u \rangle$  can be seen as a (tree-shaped) FO-structure  $\mathcal{M}_B$  with domain  $\text{dom}(\mathcal{M}_B) = \text{dom}(t)$ , constant  $\text{in}_B = 1$  and constant  $\text{out}_B = u$ , unary relation  $S_f = t^{-1}(f)$  for every  $f \in F$  and binary relation  $R_a = \{(v, w) \in \text{dom}(t) \times \text{dom}(t) : va = w\}$  for every  $a \in A$ .

We say that a language  $L \subseteq \mathcal{B}(F)$  is definable in monadic second order logic (MSO) when there exists a closed MSO formula  $\varphi$  on the FO-signature  $\{\text{in}, \text{out}\} \cup \{S_f\}_{f \in F} \cup \{R_a\}_{a \in A}$  such that  $L = \{B \in \mathcal{B}(A) : \mathcal{M}_B \models \varphi\}$ .

The following theorem gives a rather strong characterization of the languages recognized by finite state birooted  $F$ -tree automata.

**Theorem 2.** *Let  $L \subseteq \mathcal{B}(F)$  be a language of birooted  $F$ -trees. The language is recognized by a finite birooted  $F$ -tree automaton if and only if  $L$  is upward closed (in the natural order) and MSO definable.*

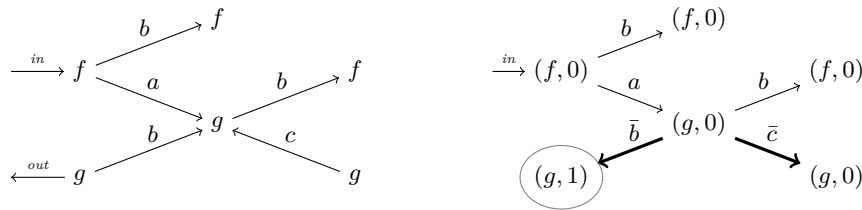
*Proof.* Let  $L \subseteq \mathcal{B}(F)$  be a language of birooted  $F$ -trees. We first prove the easiest direction, from birooted tree automata to MSO. Then, we prove the slightly more difficult direction from MSO to birooted tree automata.

*From birooted tree automata to MSO.* Assume that  $L$  is recognizable by a finite state birooted tree automaton  $\mathcal{A}$ . Without loss of generality, since  $\mathcal{A}$  is finite, we assume that the set  $Q$  of states of  $\mathcal{A}$  is such that  $Q \subseteq \mathcal{P}([1, n])$  for some  $n \geq \log_2 |Q|$ .

Then, checking that a birooted tree  $\langle t, u \rangle$  belongs to  $L(\mathcal{A})$  just amounts to checking that there exists an accepting run. This can easily be described by an existential formula of monadic second order logic of the form  $\exists X_1 X_2 \cdots X_n \varphi(in, out)$  with  $n$  set variables  $X_1, X_2, \dots, X_n$  and a first order formula  $\varphi(in, out)$ .

Indeed, every mapping  $\rho : dom(v) \rightarrow Q$  is encoded by saying, for every vertex  $v \in dom(t)$ , that  $\rho(v) = \{k \in [1, n] : v \in X_k\}$ . Then, checking that the mapping  $\rho$  encoded in such a way is indeed an accepting run amounts to checking that it satisfies state and transition coherence conditions and acceptance condition. This is easily encoded in the FO-formula  $\varphi(x, y)$ .

*From MSO to birooted tree automata.* Conversely, assume that  $L$  is upward closed for the natural order and that  $L$  is definable in *MSO*. Observe that every  $B = \langle t, u \rangle$  can just be seen as a (deterministic) tree rooted in the input root vertex 1 with edges labeled on the alphabet  $A + \bar{A}$  (with edge “direction” being induced by the prefix order on  $FG(A)$ ), vertices labeled on the alphabet  $F \times \{0, 1\}$  (with 1 used to distinguish the output root  $u$  from the other vertices). An example of such an encoding of birooted trees into trees is depicted in Figure 4.



**Fig. 4.** From a birooted  $F$ -tree to a mono-rooted tree with  $(A + \bar{A})$ -labeled edges

Since  $L$  is definable in *MSO*, applying (an adapted version of) the theorem of Doner, Thatcher and Wright (see for instance [15]), there exists a finite state tree automaton  $\mathcal{A}$  that recognizes  $L$ . We conclude our proof by defining from the (finite) tree automaton  $\mathcal{A}$  a (finite) birooted tree automaton  $\mathcal{A}'$  such that  $L(\mathcal{A}) = L(\mathcal{A}')$ .

The major difficulty defining  $\mathcal{A}'$  is that the (one root) tree automaton  $\mathcal{A}$  reads a tree from the (input) root to the leaves hence following the prefix relation order  $\leq_p$ . Moreover, in birooted trees, such a prefix order is *not* encoded in the direction of edges. It follows that, translating the tree automaton  $\mathcal{A}$  into an equivalent birooted tree automaton  $\mathcal{A}'$ , we need to encode (and propagate) that direction information into states.

But this can be achieved by observing that for every vertex  $v$  and  $w$  such that  $v \prec_p w$ , the edge from  $v$  to  $w$  is uniquely defined by the letter  $x \in (A + \bar{A})$  such that  $vx = w$ . It follows that every such a vertex  $w$  (distinct from the input root 1) will be marked in automaton  $\mathcal{A}'$  by a state that will encode that letter  $x$ ; distinguishing thus the unique predecessor vertex  $v$  of  $w$  from all successor vertices  $w'$  such that  $w \prec_p w'$ . This argument is detailed below.

*The (mono-rooted) tree automaton  $\mathcal{A}$  recognizing  $L$ .* Let  $\mathcal{A}$  be a non deterministic a top down *tree automaton* on the *edge alphabet*  $A + \bar{A}$  that recognizes  $L$ .

Adapting classical definitions to birooted trees seen as trees, we can define  $\mathcal{A}$  by  $\mathcal{A} = \langle Q, I_0, \delta, T \rangle$  with a finite set of states  $Q$ , an set of initial states  $I \in Q$  (though a priori reducible to a single initial state, this possibility is used below for normalizing  $\mathcal{A}$ ), a (non deterministic) transition function  $\delta : Q \times F \times \{0, 1\} \rightarrow \mathcal{P}(\mathcal{P}((A + \bar{A}) \times Q))$  and a specification of accepting states  $T : F \rightarrow \mathcal{P}(Q)$  that recognized the birooted tree language  $L$  with birooted trees interpreted as trees as described above.

With these notations, an accepting run of automaton  $\mathcal{A}$  on a birooted  $F$ -tree  $B = \langle t, u \rangle$  (seen as a tree rooted in 1) is a mapping  $\rho : \text{dom}(t) \rightarrow Q$  such that  $\rho(1) \in I_0$  and, for every vertex  $v \in \text{dom}(t)$  :

- ▷ inner vertex condition: if  $v$  is non maximal (in the prefix order) then the set of pairs  $\{(x, \rho(vx)) \in A \times Q : x \in (A + \bar{A}), v \prec_p vx, vx \in \text{dom}(t)\}$  either belongs to  $\delta(\rho(v), t(v), 0)$  when  $u \neq v$  or belongs to  $\delta(\rho(v), t(v), 1)$  when  $u = v$ ,
- ▷ leaf vertex condition: if  $v$  is maximal (in the prefix order) then  $\rho(v) \in T(t(v))$ .

The language of birooted trees  $L(\mathcal{A}) \subseteq \mathcal{B}(\mathcal{A})$  recognized by  $\mathcal{A}$  is defined as the set of birooted trees for which there exists an accepting run of  $\mathcal{A}$  on  $B$ . By assumption, we have  $L = L(\mathcal{A})$ .

We observe that, for every  $p \in Q$ , every  $f \in F$  and every  $m \in \{0, 1\}$ , the following assumptions can be done on  $\delta(p, f, m)$  without altering the language  $L(\mathcal{A})$ .

Since all the trees are deterministic in the edge alphabet  $A + \bar{A}$  we can assume that for every  $X \in \delta(p, f, m)$  and  $x \in A + \bar{A}$ , there exists at most one state  $q \in Q$  such that  $(x, q) \in X$ , i.e.  $X$  models a partial function from  $A + \bar{A}$  to  $Q$ .

Since we are only interested in accepting runs, we can also assume that we have  $\emptyset \in \delta(p, f, x)$  if and only if  $p \in T(f)$ . This means that the inner vertex condition and the leaf vertex condition can be merged into a single condition : the inner vertex condition extended to all vertices.

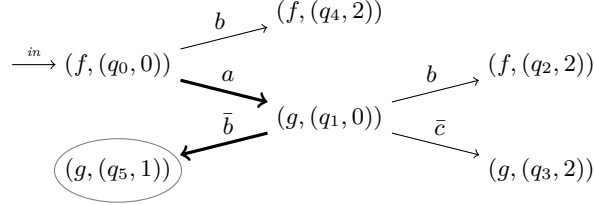
Since  $L$  is upward closed w.r.t. the natural order, we may also assume that  $\delta(p, f, m) \subseteq \mathcal{P}((A + \bar{A}) \times Q)$  is closed under taking non empty subsets, i.e. for every set non empty  $X \in \mathcal{P}((A + \bar{A}) \times Q)$ , if there is  $Y \in \delta(p, f, m)$  such that  $X \subseteq Y$  then  $X \in \delta(p, f, m)$ .

A little harder, we can also assume that  $Q = Q_0 \uplus Q_1 \uplus Q_2$  such that  $I_0 \subseteq Q_0 \uplus Q_1$  and for every for every  $p \in Q$ , every  $f \in F$ :

- ▷ if  $p \in Q_0$  then  $\delta(p, f, 1) = \emptyset$  (henceforth such a transition is unrealizable) and, for every  $X \in \delta(p, f, 0)$  there exists one and only one pair  $(x, q) \in X$  such that  $q \in Q_0 \cup Q_1$ , and, for every other pair  $(y, q') \in X$  we have  $q' \in Q_2$ ,
- ▷ if  $p \in Q_1$  then  $\delta(p, f, 0) = \emptyset$  (idem) and  $\delta(p, f, 1) \subseteq \mathcal{P}((A + \bar{A}) \times Q_2)$ ,
- ▷ if  $p \in Q_2$  then  $\delta(p, f, 1) = \emptyset$  (idem) and  $\delta(p, f, 0) \subseteq \mathcal{P}((A + \bar{A}) \times Q_2)$ .

Indeed, possibly taking  $Q \times \{0, 1, 2\}$  with  $Q_0 = Q \times \{0\}$ ,  $Q_1 = Q \times \{1\}$  and  $Q_2 = Q \times Q$  as new set of states with  $I_0 \times \{0, 1\}$  as new set of initial states, this just amounts to extending the transition function in such a (quite straightforward) way that, for every accepting run  $\rho : \text{dom}(t) \rightarrow Q$  on a birooted tree  $B = \langle t, u \rangle$ , for every  $v \in \text{dom}(t)$ , if  $v <_p u$  then  $\rho(v) \in Q_0$ , if  $v = u$  then  $\rho(v) \in Q_1$  and  $\rho(v) \in Q_2$  in all other cases.

In other words, the states of  $Q_0$  can only be used on the vertices encountered along the (shortest) path from the input root (included when distinct from the output root) to the output root (excluded), states of  $Q_1$  can only be used on the output root, and states of  $Q_2$  can only be used on all other vertices. This situation is depicted in the Figure 5 below.



**Fig. 5.** A run of  $\mathcal{A}$  on a tree  $(A + \bar{A})$ -labeled edges

It follows that the definition of automaton  $\mathcal{A}$  can be simplified into  $\mathcal{A} = \langle Q, Q_1, I_0, \delta \rangle$  with set of states  $Q = Q_0 \uplus Q_1 \uplus Q_2$  as above, initial states  $I_0 \subseteq \mathcal{P}(Q_0 \uplus Q_1)$ , transition table  $\delta : Q \times F \rightarrow \mathcal{P}(\mathcal{P}((A + \bar{A}) \times Q))$  restricted as above, with the *additional condition* that, in accepting runs, the output root must be labeled by a state of  $Q_1$ .

We are now ready to simulate the tree automaton  $\mathcal{A}$ , with the distinguished set of state  $Q_1$  for the output root by a birooted tree automaton  $\mathcal{A}'$ .

*The birooted tree automaton  $\mathcal{A}'$  simulating  $\mathcal{A}$ .* Let  $\mathcal{A}' = \langle Q', \delta', \Delta', W' \rangle$  be the birooted tree automaton defined as follows:

- ▷ the set of states  $Q' \subseteq (1 + A + \bar{A}) \times Q \times F \times \mathcal{P}((A + \bar{A}) \times Q)$  is defined as the set of triples of the form  $(x, p, f, X)$  such that any of the following condition is satisfied:
  - input root vertex states:  $x = 1$ ,  $p \in I_0$  and  $X \in \delta(p, f)$ ,
  - inner vertex states:  $x \neq 1$  and  $X \neq \emptyset$  with  $X \in \delta(p, f)$ .

- leaf vertex state: no condition on  $x$  (since the root can be a leaf),  $p \in T$  and  $X = \emptyset$  with  $X \in \delta(p, f)$ .
- ▷ the state table  $\delta' : F \rightarrow Q'$  is defined, for every  $f \in F$  by  $\delta'(f) \subseteq Q'$  is the set of states of  $\mathcal{A}'$  that are of the form  $(x, p, f, X)$ ,
- ▷ the transition table  $\Delta' : A \rightarrow \mathcal{P}(Q' \times Q')$  is defined, for every  $a \in A$  by  $\Delta'(a) \subseteq Q' \times Q'$  is the set of pairs states of the form  $((x, p, f, X), (y, q, g, Y))$  such that:
  - either  $y = a$  and  $(a, q) \in X$  (and  $X \in \delta(p, f)$  by definition of states),
  - or  $x = \bar{a}$  and  $(\bar{a}, p) \in Y$  (and  $Y \in \delta(q, f)$  by definition of states),
- ▷ the acceptance condition  $W'$  is defined as the set of states of  $\mathcal{A}'$  of the form  $((1, p, X), (y, q, Z))$  in  $Q' \times Q'$  such that  $p \in I_0$  and  $q \in Q_1$ .

We conclude the proof by showing that  $L(\mathcal{A}') = L(\mathcal{A})$ . In order do so, let  $B = \langle t, u \rangle$  be a birooted  $F$ -tree.

*From a tree run to a birooted tree run.* Let  $\rho : \text{dom}(t) \rightarrow Q$  be an accepting run of automaton  $\mathcal{A}$  on the birooted tree  $B$  (hence with  $\rho(u) \in Q_1$ ). We define  $\rho' : \text{dom}(t) \rightarrow Q'$  by taking, for every  $v \in \text{dom}(t)$ , the state  $\rho'(v) = (x, \rho(v), t(v), X) \in Q'$  with:

- ▷ if  $v = 1$  then  $x = 1$  and if  $v \neq 1$  then  $x \in A + \bar{A}$  with  $v = wx$  for some  $w <_p v$ ,
- ▷  $X = \{(x, p) \in (A + \bar{A}) \times Q : vx \in \text{dom}(t), v <_p vx, p = \rho(vx)\}$ .

By definition of  $\mathcal{A}'$ , the mapping  $\rho'$  built from  $\rho$  as above is indeed an accepting run of  $\mathcal{A}'$  on  $B$ .

*From birooted tree run to tree run.* Conversely, let  $\rho' : \text{dom}(t) \rightarrow Q'$  be an accepting run of automaton  $\mathcal{A}'$  on  $B$ . We define  $\rho : \text{dom}(t) \rightarrow Q$  by taking, for every  $v \in \text{dom}(t)$ ,  $\rho(v) = p$  when  $\rho'(v)$  is of the form  $(x, p, f, X)$ .

The following fact is easily proved by induction on the distance of vertices from the root. For every  $v \in \text{dom}(t)$ :

- ▷ if the state  $\rho'(v)$  is of the form  $(1, p, f, X)$  then  $v = 1$ ,  $p = \rho(v)$ ,  $t(v) = f$ , and  $X \in \delta(p, f)$ ,
- ▷ if the state  $\rho'(v)$  is of the form  $(x, p, f, X)$  with  $x \in A + \bar{A}$ , then, given  $w = vx$  (in  $FG(A)$ ) we have  $w <_p v = w\bar{x}$ ,  $p = \rho(v)$ ,  $f = t(v)$  and, given  $\alpha(t, x) = \{y \in A + \bar{A} : y \neq x, vy \in \text{dom}(t)\}$ , for every  $y \in B$ ,  $v \leq_p vy$  and  $X = \{(y, \rho(vy)) \in (A + \bar{A}) \times Q : y \in \alpha(t, x)\}$  with  $X \in \delta(\rho(v), f)$ .

These properties explicit how indeed the automaton  $\mathcal{A}'$  in the accepting run  $\rho'$  marks the shortest path from every vertex to the (input) root. These properties also prove that the mapping  $\rho$  defined from  $\rho'$  is indeed an accepting run of  $\mathcal{A}$  on the birooted tree  $B$ .  $\square$

From now on, a language of birooted  $F$ -trees that is definable by a finite birooted  $F$ -tree automaton is called a *regular language of birooted  $F$ -trees*.

### 2.3 Tree languages vs birooted tree languages

We aim now at relating languages of birooted  $F$ -trees and languages of  $F$ -trees. Assume till the end of that section that the set  $F$  is now a finite functional signatures that is a finite set of symbols equipped with some arity mapping  $\rho : F \rightarrow \mathcal{P}(A)$  that maps every function symbol  $f$  the set of its arguments' names  $\rho(f) \subseteq A$ .

A  $F$ -tree (also called  $F$ -term) is a function  $t : A^* \rightarrow F$  with prefix closed finite domain  $\text{dom}(t)$  such that for every  $u \in \text{dom}(t)$ , every  $a \in A$ , if  $ua \in \text{dom}(t)$  then  $a \in \rho(t(u))$ . Such a finite tree  $t$  is said to be complete when, moreover, for every  $u \in \text{dom}(t)$ , for every  $a \in A$ , if  $a \in \rho(t(u))$  then  $ua \in \text{dom}(t)$ .

Every  $F$ -tree  $t$  is encoded into a birooted  $F$ -tree  $\langle t, 1 \rangle$  called the birooted image of tree  $t$ . By extension, for every set  $X$  of  $F$ -tree, the language  $L_X = \{\langle t, 1 \rangle \in \mathcal{B}(F) : t \in X\}$  of birooted tree images of trees of  $X$  is called the birooted tree image of the language  $X$ .

**Theorem 3.** *For every regular language  $X$  of complete finite  $F$ -trees, we have  $L_X = U_X \cap D_X$  for some regular language  $U_X$  of birooted  $F$ -trees and the complement  $D_X$  of some regular language  $\mathcal{B}(F) - D_X$  of birooted  $F$ -trees.*

*Proof.* Let  $X$  be a regular language of finite  $F$ -tree. We observe first that for every complete  $F$ -tree  $t_1$  and  $t_2$ , their birooted images  $\langle t_1, 1 \rangle$  and  $\langle t_2, 1 \rangle$  are incomparable in the natural order. It follows that the element of  $L_X$  form an anti-chain in the natural order. It follows that, given  $U_x = \{y \in \mathcal{B}(F) : \exists x \in L_X, x \leq y\}$  the upward closure of  $L_X$  and  $D_x = \{y \in \mathcal{B}(F) : \exists x \in L_X, y \leq x\}$  the downward closure of  $L_X$ , we have  $L_X = U_x \cap D_x$ . We conclude the proof by observing that if  $X$  is regular then it is definable in MSO. This implies that the languages  $L_X$ ,  $D_X$  and  $U_X$  are also definable in MSO. We conclude by applying Theorem 2, that ensures that both  $U_X$  and  $\mathcal{B}(X) - D_X$  are regular languages of birooted trees.  $\square$

## 3 Quasi-recognizable languages of birooted $F$ -trees

In [5] (also generalized in [4]), a notion of recognizability by premorphisms, called quasi-recognizability, is proposed to define languages of positive (or arbitrary) overlapping tiles. This notion is extended here to languages of birooted  $F$ -trees.

### 3.1 Definition

Let  $S$  be a monoid partially ordered by a relation  $\leq_S$  (or just  $\leq$  when there is no ambiguity). We always assume that the order relation  $\leq$  is stable under product, i.e. if  $x \leq y$  then  $xz \leq yz$  and  $zx \leq zy$  for every  $x, y$  and  $z \in S$ . The set  $U(S)$  of *subunits* of the partially ordered monoid  $S$  is defined by  $U(S) = \{y \in S : y \leq 1\}$ .

A partially ordered monoid  $S$  is an *adequately ordered monoid* when all subunits of  $S$  are idempotents, and for every  $x \in S$ , both the minimum of right



local units  $x^L = \min\{y \in U(S) : xy = x\}$  and the minimum of left local units  $x^R = \min\{y \in U(S) : yx = x\}$  exist and belong to  $U(S)$ .

For every  $x \in S$ , the subunits  $x^L$  and  $x^R$  are also called the *left projection* and the *right projection* of  $x$ . Since subunits are assumed to be idempotents, one can check that they commute and thus, ordered by the monoid order, form a meet semilattice with the product as the meet operator. It follows that when  $x$  is itself a subunit, we have  $x = x^L = x^R$ . In other words, in an adequately ordered monoid, both left and right projection mappings are indeed projections from  $S$  onto  $U(S)$ .

*Examples.* Every monoid  $S$  extended with the trivial order  $x \leq y$  when  $x = y$  is a adequately ordered monoid with  $x^L = x^R = 1$  for every  $x \in S$ . These adequately ordered monoids are called trivial.

Every inverse monoid  $S$  ordered by the natural order is also an adequately ordered monoid with  $x^L = x^{-1}x$  and  $x^R = xx^{-1}$  for every  $x \in S$ . As a particular case, the monoid  $\mathcal{B}^1(F)$  ordered by the natural order is also an adequately ordered monoid. The subunits of  $\mathcal{B}^1(F)$  are, when distinct from 0 or 1, the birooted  $F$ -trees of the form  $\langle t, 1 \rangle$  and, indeed, for every birooted  $F$ -tree  $B = \langle t, u \rangle$  we have  $B^R = \langle t, 1 \rangle$  and  $B^L = \langle t_u, 1 \rangle$ .

For every set  $Q$ , the relation monoid  $\mathcal{P}(Q \times Q)$  ordered by inclusion is also an adequately ordered monoid with, for every  $X \subseteq Q \times Q$ ,  $X^L = \{(q, q) \in Q \times Q : (p, q) \in X\}$  and  $X^R = \{(p, p) \in Q \times Q : (p, q) \in X\}$ .

A mapping  $\theta : S \rightarrow T$  between two adequately ordered monoids is a *premorph* when  $\theta(1) = 1$  and, for every  $x$  and  $y \in S$ , we have  $\theta(xy) \leq_T \theta(x)\theta(y)$  and if  $x \leq_S y$  then  $\theta(x) \leq_T \theta(y)$ . A premorphism  $\theta : S \rightarrow T$  is an *adequate premorphism* when for every  $x$  and  $y \in S$  we have  $\theta(x^L) = (\theta(x))^L$ ,  $\theta(y^R) = (\theta(y))^R$  and, if  $xy \neq 0$  with  $x^L \vee y^R = 1$ , i.e. the product  $xy$  is a disjoint product, then  $\theta(xy) = \theta(x)\theta(y)$ .

A language  $L \subseteq \mathcal{B}(F)$  of birooted tree is a *quasi-recognizable language* when there exists a finite adequately ordered monoid  $S$  and an adequate premorphism  $\theta : \mathcal{B}(F) \rightarrow S$  such that  $L = \theta^{-1}(\theta(L))$ .

**Theorem 4.** *Let  $\theta : FIM(A) \rightarrow S$  be an adequate premorphism with finite  $S$ . For every  $B \in \mathcal{B}(F)$  the image  $\theta(B)$  of the birooted  $F$ -tree  $B$  by the adequate premorphism  $\theta$  is uniquely determined by the structure of  $B$ , the structure of  $S$  and the image by  $\theta$  of elementary birooted  $F$ -trees.*

*Proof.* This essentially follows from the adequacy assumption and the strong decomposition property (Lemma 1).  $\square$

### 3.2 From birooted tree automata to quasi-recognizable languages

Now we want to show that every finite state birooted automaton induces an adequate premorphism that recognizes the same language.

**Theorem 5.** *Let  $L \subseteq \mathcal{B}(F)$  be a language of birooted  $F$ -trees. If  $L$  is recognizable by a finite state birooted tree automaton then it is recognizable by an adequate premorphism into a finite adequately ordered monoid.*

*Proof.* Let  $L \subseteq \mathcal{B}(F)$  and let  $\mathcal{A} = \langle Q, \delta, \Delta, T \rangle$  be a finite birooted tree automaton such that  $L = L(\mathcal{A})$ .

We define the mapping  $\varphi_{\mathcal{A}} : \mathcal{B}(F) \rightarrow \mathcal{P}(Q \times Q)$  by saying that  $\varphi_{\mathcal{A}}(B)$  is, for every birooted  $F$ -tree  $B = \langle t, u \rangle \in \mathcal{B}(F)$ , the set of all pairs of state  $(p, q) \in Q \times Q$  such that there exists a run  $\rho : \text{dom}(t) \rightarrow Q$  such that  $p = \rho(1)$  and  $q = \rho(u)$ . The mapping  $\varphi_{\mathcal{A}}$  is extended to 0 by taking  $\varphi_{\mathcal{A}}(0) = \emptyset$  and, to 1 by taking  $\varphi_{\mathcal{A}}(1) = I_Q = \{(q, q) \in Q \times Q : q \in Q\}$ .

The fact  $\mathcal{P}(Q \times Q)$  is an adequately ordered monoid have already been detailed in the examples above. By definition we have  $L = \varphi^{-1}(\mathcal{X})$  with  $\mathcal{X} = \{X \subseteq Q \times Q : X \cap T \neq \emptyset\}$ . It remains to show that  $\varphi_{\mathcal{A}}$  is indeed an adequate premorphism.

The fact  $\varphi_{\mathcal{A}}$  is monotonic is immediate. Indeed, for every birooted  $F$ -tree  $\langle s, u \rangle$  and  $\langle t, v \rangle$ , if  $\langle s, u \rangle \leq \langle t, v \rangle$  this means that  $u = v$  thus, for every run  $\rho : \text{dom}(s) \rightarrow Q$  of  $\mathcal{A}$  on  $\langle s, u \rangle$ , the mapping  $\rho$  restricted to  $\text{dom}(t)$  is clearly a run of  $\mathcal{A}$  on  $\langle s | \text{dom}(t), u \rangle = \langle t, u \rangle$ .

Left and right projections preservation immediately follows from their characterizations in both  $\mathcal{B}(F)$  and  $\mathcal{P}(Q \times Q)$  and the definition of  $\varphi_{\mathcal{A}}$ .

It remains to show that  $\varphi_{\mathcal{A}}$  is submultiplicative and preserves disjoint product. Let  $\langle s, u \rangle$  and  $\langle t, v \rangle$  be two birooted trees. In the case  $\langle s, u \rangle \cdot \langle t, v \rangle = 0$  we are done, since  $\varphi_{\mathcal{A}}(0) = \emptyset$ . Otherwise, let  $\rho$  be a run of  $\mathcal{A}$  on the product  $\langle s, u \rangle \cdot \langle t, v \rangle$ . By definition of the product, the mapping  $\rho_1 : \text{dom}(s) \rightarrow Q$  defined by  $\rho_1(w) = \rho(w)$  for every  $w \in \text{dom}(s)$  is clearly a run of  $\mathcal{A}$  on  $\langle s, u \rangle$ . Similarly, the run  $\rho_2 : \text{dom}(t) \rightarrow Q$  defined by  $\rho_2(w) = \rho(uw)$  for every  $w \in \text{dom}(t)$  is also a run of  $\mathcal{A}$  on  $\langle t, v \rangle$ . Now, since  $\rho_1(u) = \rho_2(1)$  and that construction applies for every run  $\rho$ , this shows that  $\varphi_{\mathcal{A}}(\langle s, u \rangle \cdot \langle t, v \rangle) \subseteq \varphi_{\mathcal{A}}(\langle s, u \rangle) \cdot \varphi_{\mathcal{A}}(\langle t, v \rangle)$ .

Last, assume that the product  $\langle s, u \rangle \cdot \langle t, v \rangle$  is disjoint. This means that  $s(u) = t(1)$  and  $\text{dom}(s) \cap u \cdot \text{dom}(t) = \{u\}$ .

Let  $(p, q) \in \varphi_{\mathcal{A}}(\langle s, u \rangle) \cdot \varphi_{\mathcal{A}}(\langle t, v \rangle)$ . By definition of the product of relations, this means that there exists  $q' \in Q$ , such that  $(p, q') \in \varphi_{\mathcal{A}}(\langle s, u \rangle)$  and  $(q', q) \in \varphi_{\mathcal{A}}(\langle t, v \rangle)$ . But then, by definition of  $\varphi_{\mathcal{A}}$  this means that there exists a run  $\rho_1 : \text{dom}(s) \rightarrow Q$  of  $\mathcal{A}$  on  $\langle s, u \rangle$  and a run  $\rho_2 : \text{dom}(t) \rightarrow Q$  of  $\mathcal{A}$  on  $\langle t, v \rangle$  such that  $\rho_1(1) = p$ ,  $\rho_1(u) = q'$ ,  $\rho_2(1) = q'$  and  $\rho_2(v) = q$ .

Let then  $\rho : \text{dom}(s) \cup u \cdot \text{dom}(t) \rightarrow Q$  defined by  $\rho(w) = \rho_1(w)$  for every  $w \in \text{dom}(s)$ , and  $\rho(uw) = \rho_2(w)$  for every  $w \in \text{dom}(t)$ . Since the product of the two birooted  $F$ -trees is a disjoint product, we have  $\text{dom}(s) \cap u \cdot \text{dom}(t) = \{u\}$  with  $\rho_1(u) = q' = \rho_2(1)$  hence  $\rho$  is well defined. As it is clearly a run of  $\mathcal{A}$  on the (non zero) product  $\langle s, u \rangle \cdot \langle t, v \rangle$  with  $\rho(1) = p$  and  $\rho(uv) = q$ , this means we indeed have  $(p, q) \in \varphi_{\mathcal{A}}(\langle s, u \rangle \cdot \langle t, v \rangle)$ .

As this holds for arbitrary pair of states  $(p, q) \in \varphi_{\mathcal{A}}(\langle s, u \rangle) \cdot \varphi_{\mathcal{A}}(\langle t, v \rangle)$  this proves that  $\varphi_{\mathcal{A}}(\langle s, u \rangle) \cdot \varphi_{\mathcal{A}}(\langle t, v \rangle) \subseteq \varphi_{\mathcal{A}}(\langle s, u \rangle \cdot \langle t, v \rangle)$  and thus concludes the proof.  $\square$

### 3.3 From quasi-recognizability to MSO

The following theorem tells how quasi-recognizability and MSO definability are related.

**Theorem 6.** *Let  $\theta : FIM(A) \rightarrow S$  be an adequate premorphism with finite  $S$ . For every  $X \subseteq S$ , the language  $\theta^{-1}(X)$  is definable in Monadic Second Order Logic.*

*Proof.* Let  $\theta : FIM(A) \rightarrow S$  as above and let  $X \subseteq S$ . Uniformly computing the value of  $\theta$  on every birooted tree by means of an MSO formula is done by adapting Shelah's decomposition techniques [14]. More precisely, we show that the strong decomposition provided by Lemma 1 is indeed definable in MSO. Then, the computation of the value of  $\theta$  on every birooted rooted  $B$  can be done from the value of  $\theta$  on the elementary birooted trees and the sub-birooted  $F$ -trees that occur in such a decomposition.

More precisely, we first show that the predecessor relation  $\prec_p$  (and thus, by transitive closure, the prefix relation  $\leq_p$  as well) is definable in MSO. This amounts to saying that there exists an MSO formula  $\varphi_p(x, y)$  such that, for every birooted tree  $\langle t, u \rangle$ , for every vertex  $v$  and  $w \in \text{dom}(t)$ , we have  $\langle t, u \rangle \models \varphi_p(v, w)$  if and only if  $v \prec_p w$ .

Indeed, defining  $\varphi_p(x, y)$  amounts to saying that there exists a partition of  $\text{dom}(t)$  in three sets of vertices  $X_0, X_1$  and  $X_2$  such that the (input) root 1 belongs to  $X_0$ , all its neighbors (or immediate successors) belong to  $X_1$ , and for every vertex  $z \in Z$  distinct from the input root, given  $i \in \{0, 1, 2\}$  such that  $z \in X_i$ , given  $j = i - 1 \pmod 3$  and  $k = i + 1 \pmod 3$ , the vertex  $z$  has a single neighbor in  $X_j$  (the unique predecessor of  $z$  in the predecessor relation  $\prec_p$ ) and all other neighbors of  $z$  belong to  $X_k$  (the successors of  $z$  in the predecessor relation  $\prec_p$ ).

As a consequence, since the reflexive and transitive closure of a definable binary relation is also definable in MSO, there exists a formula  $\varphi_p^*(x, y)$  such that  $\langle t, u \rangle \models \varphi_p^*(v, w)$  if and only if  $v \leq_p w$ .

This also means that for every birooted tree  $B = \langle t, u \rangle$ , the set  $U = \{z \in \text{dom}(t) : 1 \leq_p z \leq_p u\}$  is also MSO definable in every birooted tree  $\langle t, u \rangle$  and, as well, for every vertex  $v \in \text{dom}(t)$ , the sub-birooted tree  $B_v^p$ . Here, by saying the birooted  $F$ -tree  $B_v^p$  is definable in MSO we mean that its domain  $D^p(v)$  (defined in the proof of Lemma 1) is definable and thus its structure: the vertex labels and the edge relations, is just obtained by restricting those of  $B$  to the domain  $D^p(v)$ .

The next step is then the following. Given a finite collection of set variables  $\{Y_s\}_{s \in U(S)}$ , one variable  $Y_s$  per element  $s \in U(S)$ , writing  $\bar{Y}$  for the tuple of such variables, we claim that there exists a formula  $\varphi(\bar{Y})$  such that for every birooted  $F$ -tree  $\langle t, u \rangle$  for every  $v \in \text{dom}(t)$ , for every subunit  $s \in U(S)$ , we have  $\langle t, u \rangle \models \exists \bar{Y}(v \in Y_s \wedge \varphi(\bar{Y}))$  if and only if  $\theta(B_v^p) = s$ .

Indeed, this amounts to saying that  $\{Y_s\}_{s \in U(S)}$  form a partition (with possible empty sets) of  $\text{dom}(t)$  such that, for every vertex  $v \in \text{dom}(t)$ , if  $v$  is a

leaf w.r.t. the prefix order  $\leq_p$  then  $s = \theta(B_v^p) = \theta(B_{t(v)})$  and we check that  $v$  belongs to  $Y_s$  or, if  $v$  is not a leaf, then we must have  $v \in X_s$  with, by adequacy assumption on  $\theta$ ,  $s$  uniquely determined by

$$s = \prod \{(\theta(B_{v,w}^p) \cdot s_w)^R : v \prec_p w\}$$

with  $s_w \in S$  is the unique element of  $S$  such that  $w \in Y_{s_w}$ . Indeed, by the proof of Lemma 1, we know that  $B_v^p = \prod \{(B_{v,w}^p \cdot B_w^p)^R : w \in \text{dom}(r), v \prec_p w\}$  with disjoint products only and the adequacy assumption applies. As the product is of a bounded size, we can check that  $v \in Y_s$ .

Then, for every birooted tree  $B = \langle t, u \rangle$ , given the ordered prefixes of  $u$  described by  $u_0 = 1 \prec_p u_1 \prec_p u_2 \prec_p \dots \prec_p u_{n-1} \prec_p u_n = u$  the value of  $\theta(B)$  can be computed as the element  $s \in S$  defined by

$$s = \theta(B_{u_0}^p) \theta(B_{u_0, u_1}^p) \theta(B_{u_1}^p) \dots \theta(B_{u_{n-1}}^p) \theta(B_{u_{n-1}, u_n}^p) \theta(B_{u_n}^p)$$

Indeed, by Lemma 1 we have  $B = B_{u_0}^p B_{u_0, u_1}^p B_{u_1}^p \dots B_{u_{n-1}}^p B_{u_{n-1}, u_n}^p B_{u_n}^p$  with disjoint product only hence, by adequacy of  $\theta$ , the claim. As all these values are computable, either as image by  $\theta$  of elementary birooted trees, or, by induction, by observing that for every prefix  $u'$  of  $u$  we have  $\theta(B_{u'}^p) = s'$  if and only if  $u' \in Y_{s'}$ . Then, checking that  $v \in Y_s$  by “computing” in MSO the value  $s$  can be done, say, by a left to right “traversal” of the path from 1 to  $u$ , simulating the underlying finite state word automaton induced by  $S$  on the (images of) elementary birooted trees. This concludes the proof.  $\square$

### 3.4 Quasi-recognizable languages vs MSO definable languages

For the picture to be complete, it remains to characterize the class of quasi-recognizable languages w.r.t. the class of languages definable in Monadic Second Order Logic.

**Theorem 7.** *Let  $L \subseteq \mathcal{B}(F)$  be a language of birooted  $F$ -trees. The following properties are equivalent:*

- (1) *the language  $L$  is quasi-recognizable,*
- (2) *the language  $L$  is a finite boolean combination of upward closed MSO definable languages,*
- (3) *the language  $L$  is a finite boolean combination of languages recognized by finite state birooted tree automata.*

*Proof.* Let  $L \subseteq \mathcal{B}(F)$  be a language of birooted  $F$ -trees.

(1) *implies* (2). We assume that  $L$  is recognized by some adequate premorphism  $\theta : \mathcal{B}(F) \rightarrow S$ . By definition, we have  $L = \theta^{-1}(\theta(L))$  hence

$$\theta^{-1}(\theta(L)) = \bigcup_{x \in \theta(L)} \theta^{-1}(D_x) \cap \theta^{-1}(U_x)$$

with  $U_x = \{y \in S : x \leq y\}$  and  $D_x = \{y \in S : y \leq x\}$  for every  $x \in S$ . Indeed, for every  $x \in S$ , we have  $\theta^{-1}(x) = \theta^{-1}(U_x) \cap \theta^{-1}(D_x)$ . The inclusion  $\theta^{-1}(x) \subseteq \theta^{-1}(U_x) \cap \theta^{-1}(D_x)$  is immediate. Conversely, let  $B \in \theta^{-1}(U_x) \cap \theta^{-1}(D_x)$ . Since  $B \in \theta^{-1}(U_x)$  we have  $x \leq \theta(B)$  and since  $B \in \theta^{-1}(D_x)$  we have  $\theta(B) \leq x$  hence  $\theta(B) = x$  and thus  $B \in \theta^{-1}(x)$ .

We prove (2) by observing that both  $\theta^{-1}(U_x)$  and  $\overline{\theta^{-1}(D_x)} = \theta^{-1}(S - D_x)$  are upward closed (and recognized by  $\theta$ ) hence, by Theorem 6, they are MSO definable.

(2) *implies* (3). This immediately follows from Theorem 2 that ensures that every upward closed languages MSO definable is recognized by a finite state birooted tree automaton.

(3) *implies* (1). Assume that  $L$  is a finite boolean combination of languages recognized by birooted tree automata. We want to show that  $L$  is quasi-recognizable.

By Theorem 5, every such a regular language is quasi-recognizable. Since the class of quasi-recognizable languages is obviously closed under complement it suffices to prove that it is closed under intersection.

But this is easily done using classical algebraic tools on monoids [12]. Indeed, given two adequate premorphisms  $\theta_1 : \mathcal{B}(F) \rightarrow S_1$  and  $\theta_2 : \mathcal{B}(F) \rightarrow S_2$ , the mapping  $\theta : \mathcal{B}(F) \rightarrow S_1 \times S_2$  defined by  $\varphi(B) = (\varphi_1(B), \varphi_2(B))$  is an adequate premorphism in the product monoid  $S_1 \times S_2$  ordered by the product order. Then, for every  $X \subseteq S_1$  and  $Y \subseteq S_2$  we have  $\varphi_1^{-1}(X) \cap \varphi_2^{-1}(Y) = \varphi^{-1}(X \times Y)$ . This concludes the proof.  $\square$

**Corollary 8.** *The birooted image of every regular languages of  $F$ -tree is recognizable by an adequate premorphism in a finite adequately ordered monoid.*

*Proof.* This follows from Theorem 3 and Theorem 7.  $\square$

## 4 Conclusion

Studying languages of birooted  $F$ -trees, structures that generalize  $F$ -terms, we have thus defined a notion of automata, a related notion of quasi-recognizability and we have characterized quite in depth their expressive power in relationship with language definability in Monadic Second Order Logic (Theorem 7).

As a particular case, this provided a new algebraic characterization of the regular languages of finite  $F$ -trees that, at first sight, seems incomparable with the preclones approach [3] or the forest algebra approach [2] (Corollary 8).

Still, recognizability by adequate premorphism needs to be better understood. For instance, an intriguing alternative notion of premorphisms, called admissible premorphisms, arises when studying walking automata on birooted trees. Though unrelated with the notion of adequate premorphisms, the recognizability by admissible premorphisms is just as expressive as recognizability by adequate premorphisms [6].

This suggests that some more general notion of MSO definable premorphisms could be studied in place of both notions of adequate or admissible premorphisms.

## References

1. A. Blumensath. Recognisability for algebras of infinite trees. *Theor. Comput. Sci.*, 412(29):3463–3486, 2011.
2. M. Bojańczyk and I. Walukiewicz. Forest algebras. In *Logic and Automata*, pages 107–132, 2008.
3. Z. Ésik and P. Weil. On logically defined recognizable tree languages. In *Found. of Soft. Techno. and Theoretical Computer Science (FSTTCS)*, pages 195–207, 2003.
4. D. Janin. Overlapping tile automata. Technical Report RR-1465-12, LaBRI, Université de Bordeaux, 2012.
5. D. Janin. Quasi-recognizable vs MSO definable languages of one-dimensional overlapping tiles. In *Mathematical Foundations of computer Science (MFCS)*, volume 7464 of *LNCS*, pages 516–528, 2012.
6. D. Janin. Walking automata in the free inverse monoid. Technical Report RR-1464-12 (revised february 2013), LaBRI, Université de Bordeaux, 2012.
7. D. Janin. On languages of one-dimensional overlapping tiles. In *International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 7741 of *LNCS*, pages 244–256, 2013.
8. M. V. Lawson. McAlister semigroups. *Journal of Algebra*, 202(1):276 – 294, 1998.
9. D. Perrin and J.-E. Pin. Semigroups and automata on infinite words. In J. Fountain, editor, *Semigroups, Formal Languages and Groups*, NATO Advanced Study Institute, pages 49–72. Kluwer academic, 1995.
10. J.-E. Pin. Relational morphisms, transductions and operations on languages. In *Formal Properties of Finite Automata and Applications*, volume 386 of *LNCS*, pages 34–55. Springer, 1989.
11. J.-E. Pin. Finite semigroups and recognizable languages: an introduction. In J. Fountain, editor, *Semigroups, Formal Languages and Groups*, NATO Advanced Study Institute, pages 1–32. Kluwer academic, 1995.
12. J.-E. Pin. Chap. 10. Syntactic semigroups. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal language theory, Vol. I*, pages 679–746. Springer Verlag, 1997.
13. H. E. Scheiblich. Free inverse semigroups. *Semigroup Forum*, 4:351–359, 1972.
14. S. Shelah. The monadic theory of order. *Annals of Mathematics*, 102:379–419, 1975.
15. W. Thomas. Chap. 7. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Language Theory, Vol. III*, pages 389–455. Springer Verlag, 1997.
16. T. Wilke. An algebraic theory for regular languages of finite and infinite words. *Int. J. Alg. Comput.*, 3:447–489, 1993.