RAIRO-Theor. Inf. Appl. 43 (2009) 321–338
DOI: 10.1051/ita/2009009

Available online at:
www.rairo-ita.org

TREE AUTOMATA AND AUTOMATA ON LINEAR ORDERINGS

Véronique Bruyère[1], Olivier Carton[2] and Géraud Sénizergues[3]

Abstract. We show that the inclusion problem is decidable for rational languages of words indexed by scattered countable linear orderings. The method leans on a reduction to the decidability of the monadic second order theory of the infinite binary tree [9].

Mathematics Subject Classification. 68Q45, 03D05.

1. Introduction

In his seminal paper [8], Kleene establishes the equivalence between automata and rational expressions. Since then, many kinds of automata accepting infinite words, bi-infinite words, ordinal words and trees have been introduced.

In [1,3], linear structures are considered in a general framework, that is, words indexed by any linear ordering. Automata accepting such words as well as rational expressions are introduced. A Kleene-like theorem is proved for words indexed by linear orderings which are scattered. This class of orderings contains the ordinals and is closed under many natural operations like reversal, sum, ω-sum and taking suborderings.

In [4], the emptiness problem for these automata is shown to be decidable in polynomial time when again scattered orderings are considered.

In this paper, we solve the equivalence problem for these automata. More generally, we show that the inclusion problem is decidable.

Keywords and phrases. Finite automata, words over linear orderings-trees, monadic second order logics.

[1] Université de Mons-Hainaut, France; Veronique.Bruyere@umh.ac.be

[2] LIAFA, Université Paris 7, France; Olivier.Carton@liafa.jussieu.fr

[3] LaBRI, Université de Bordeaux I, France; ges@labri.u-bordeaux.fr

Article published by EDP Sciences

In [10], it has been proved that the class of rational sets of words over countable and scattered ordering is closed under complementation. It follows from [10] that the inclusion problem can be reduced to the emptiness problem. Our paper provides an alternative proof of the decidability of inclusion. Since our proof techniques are different we hope that this approach can be extended to the class of non scattered orderings which is not closed under complementation. Our result has already been presented in [2] in 2003. At that time, the complementation results of [10] were not available and there was no alternative proof of the decidability of inclusion.

Our approach to solve the inclusion problem uses the Rabin tree theorem [9]. The idea is the following. Accepting paths in an automaton are described by a monadic second-order formula using two successor functions. In this way, the inclusion problem for automata on scattered linear orderings is reduced to the inclusion problem for tree automata. This establishes an interesting link between tree automata and automata on linear orderings.

A similar approach is already proposed in [12] to solve a decision problem on words on linear orderings raised by Courcelle [5] and Heilbrunner [7]. That paper is concerned with the decidability of the equivalence of two rational expressions, when each rational expression describes exactly one word over arbitrary countable orderings. So, in terms of automata, the two differences with our result are that the automata only accept *one* word, however the orderings are not restricted to *scattered* ones.

The paper is organized as follows. In Section 2, we recall the basic notions on linear orderings and the definition of automata accepting words indexed by a linear ordering. In Section 3, we show how a binary tree can represent a linear ordering and its set of cuts. Section 4 is devoted to the inclusion problem for automata on scattered linear orderings. In Section 4.2 we give a detailed translation to the inclusion problem for tree automata and in Section 4.3 we sketch a translation to the monadic second order theory of the class of countable orderings. We mention perspectives and open problems in the conclusion.

2. Words and automata on linear orderings

In this section, we recall the definition of words indexed by linear orderings and the definition of automata accepting such words [1,3]. We refer the reader to [11] for a complete introduction to linear orderings.

2.1. Orderings

A *linear ordering* J is a set equipped with an ordering $<$ which is total. The ordering of the integers, of the relative integers and of the rational numbers are linear orderings, respectively denoted by ω, ζ and η. We recall that an *ordinal* is a linear ordering which is also a well-ordering, that is, any non-empty subset has a least element. In the sequel, we freely say that two orderings are equal if they are actually isomorphic.

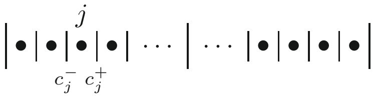

Figure 1. Ordering $J \cup \hat{J}$ for $J = \omega + (-\omega)$.

Given a linear ordering J, two elements i and k of J are said to be *consecutive* if $i < k$ and there exists no j in J such that $i < j < k$. The element i is then called the *predecessor* of k and k is called the *successor* of i.

Let J and K be two linear orderings. We denote by $-J$ the backwards linear ordering obtained by reversing the ordering relation. The linear ordering $J + K$ is the ordering on the disjoint union $J \cup K$ extended with $j < k$ for any $j \in J$ and any $k \in K$. More generally, let K_j be a linear ordering for any $j \in J$. The linear ordering $\sum_{j \in J} K_j$ is the set of pairs (k, j) such that $k \in K_j$. The relation $(k_1, j_1) < (k_2, j_2)$ holds if and only if $j_1 < j_2$ or ($j_1 = j_2$ and $k_1 < k_2$ in K_{j_1}).

Example 2.1. The ordering $\omega + (-\omega)$ is the sum of the ordering of the integers and its backwards ordering. It is different from the ordering $-\omega + \omega$ which is equal to ζ.

A *cut* of a linear ordering J is a pair (K, L) of intervals such that J is the disjoint union $K \cup L$ and for any $k \in K$ and $l \in L$, $k < l$. The set of all cuts of the ordering J is denoted by $\hat{J}$. The set $\hat{J}$ can be linearly ordered as follows. For any cuts $c_1 = (K_1, L_1)$ and $c_2 = (K_2, L_2)$, we define the relation $c_1 < c_2$ if and only if $K_1 \subsetneq K_2$. The cuts $(\emptyset, J)$ and $(J, \emptyset)$ are then the first and the last element of $\hat{J}$. It is sometimes convenient to ignore these two cuts. We denote by $\hat{J}^*$ the set $\hat{J} \setminus \{(\emptyset, J), (J, \emptyset)\}$.

Example 2.2. Let J be the ordinal ω. The set $\hat{J}$ contains the cut $(\{0, \ldots, n - 1\}, \{n, n+1, \ldots\})$ for any integer n and the last cut $(\omega, \emptyset)$. The ordering $\hat{J}$ is thus the ordinal $\omega + 1$. If J is the ordering $\omega + (-\omega)$, then $\hat{J}$ is the ordering $\omega + 1 + (-\omega)$. It contains the particular cut $(\omega, -\omega)$ which has no predecessor and no successor. The ordering $\hat{J}$ for $J = \eta$ is not countable since it contains the usual ordering on the set of real numbers.

The orderings of J and $\hat{J}$ can be extended to an ordering on the disjoint union $J \cup \hat{J}$ as follows. For $j \in J$ and a cut $c = (K, L)$, define the relations $j < c$ and $c < j$ by respectively $j \in K$ and $j \in L$. Note that exactly one of these two relations holds since (K, L) is a partition of J. These relations together with the orderings of J and $\hat{J}$ endows $J \cup \hat{J}$ with a linear ordering.

Example 2.3. For the ordering $J = \omega + (-\omega)$, the ordering $J \cup \hat{J}$ is pictured in Figure 1 where each element of J is represented by a bullet and each cut by a vertical bar.

For any element $j \in J$, there are two cuts c_j^- and c_j^+ obtained by cutting J on the left and on the right of j (see Fig. 1). These two cuts are defined by

$c_j^- = (K, \{j\} \cup L)$ and $c_j^+ = (K \cup \{j\}, L)$ where K and L are the sets $K = \{k \mid k < j\}$ and $L = \{k \mid j < k\}$. The cuts c_j^- and c_j^+ are consecutive in the ordering $\hat{J}$. Conversely, for any consecutive cuts c and c' in $\hat{J}$, there is a unique element j in J such that $c = c_j^-$ and $c' = c_j^+$. In the ordering $J \cup \hat{J}$, one has $c_j^- < j < c_j^+$. Note that if j and j' are consecutive elements in J, then the equality $c_j^+ = c_{j'}^-$ holds.

2.2. Scattered linear orderings

A linear ordering J is said to be *dense* if for any $i < k$ in J, there is $j \in J$ such that $i < j < k$. A linear ordering is *scattered* if it contains no dense subordering of cardinality ≥ 2. The following characterization of countable and scattered linear orderings is due to Hausdorff. Notation $\mathbf{0}$ and $\mathbf{1}$ is used for the empty ordering and the one-element ordering. Notation $\mathcal{N}$ and $\mathcal{O}$ is used for the class of finite orderings and the class of countable ordinals.

Theorem 2.4 (Hausdorff [6], also Thm. 5.24 p. 86 of [11])**.** *A countable linear ordering J is scattered if and only if J belongs to $\bigcup_{\alpha\in\mathcal{O}} V_\alpha$ where the classes V_α are inductively defined by*

(1) $V_0 = \{\mathbf{0}, \mathbf{1}\}$;
(2) $V_\alpha = \{\sum_{j\in J} K_j \mid J \in \mathcal{N} \cup \{\omega, -\omega, \zeta\}$ *and* $K_j \in \bigcup_{\beta<\alpha} V_\beta\}$.

Remark 2.5. (1) It follows from the previous theorem that if J is a countable and scattered linear ordering, then $\hat{J}$ is also a countable and scattered linear ordering.
(2) Conversely, suppose that $\hat{J}$ is countable.
(2.1) Since $x \mapsto (\{y \in J \mid y \leq x\}, \{y \in J \mid y > x\})$ is a strictly increasing map from $(J, \leq)$ into $(\hat{J}, \leq)$, J is countable too.
(2.2) If J has some dense subset J_0 with at least two elements, then it also contains a countable dense subset J_1. The ordered set J_1 is isomorphic with $\mathbb{Q}$ (by the classical "back and forth" argument of Cantor), so that $\hat{J}_1 \approx \mathbb{R}$. But $(K_1, L_1) \mapsto (\{x \in J \mid \exists k \in K_1, x \leq k\}, \{x \in J \mid \forall k \in K_1, x > k\})$ is an injective map from $\hat{J}_1$ into $\hat{J}$, showing that $\hat{J}$ has a cardinality greater or equal to Card($\mathbb{R}$), contradicting the hypothesis of point (2). Thus the hypothesis of point (2.2) is impossible, which shows that J is scattered.

2.3. Words on linear orderings

Let A be a finite alphabet. For a linear ordering J, a *word* $(a_j)_{j\in J}$ of *length* J over A is a function from J to A which labels any element j of J by a letter a_j of A. The word whose length is the ordering $\mathbf{0}$ is called the *empty word* and it is denoted by ε. The set of all words over the alphabet A is denoted by $A^\diamond$.

Let $x = (a_j)_{j\in J}$ and $y = (b_k)_{k\in K}$ be two words of length J and K. The *product* xy (or the *concatenation*) of x and y is the word $z = (c_i)_{i\in J+K}$ of length $J + K$ such that $c_i = a_i$ if $i \in J$ and $c_i = b_i$ if $i \in K$. More generally, let J be a linear ordering and for each $j \in J$, let x_j be a word of length K_j. The *product* $\prod_{j\in J} x_j$ is the word z of length $K = \sum_{j\in J} K_j$ defined as follows. Suppose that each word

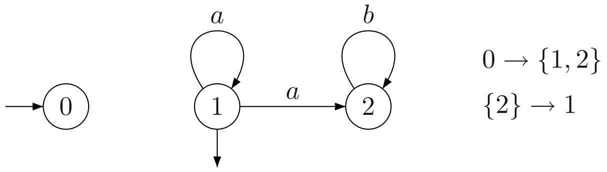

FIGURE 2. Automaton of Example 2.8.

x_j is equal to $(a_{k,j})_{k \in K_j}$ and recall that K is the set of all pairs (k,j) such that $k \in K_j$. The product z is then equal to $(a_{k,j})_{(k,j)\in K}$.

Example 2.6. The word $a^\omega b^{-\omega}$ of length $\omega + (-\omega)$ is the product of the word a^ω of length ω and $b^{-\omega}$ of length $-\omega$. Another example is the word $\prod_{i\in -\omega} a^i ab^\omega = \dots a^3b^\omega a^2b^\omega ab^\omega$ of length $\sum_{i\in-\omega} \omega$. It is obtained as the product $\prod_{i\in-\omega} x_i$ of words $x_i = a^i ab^\omega$ of length ω.

Two words $x = (a_j)_{j\in J}$ and $y = (b_k)_{k\in K}$ of length J and K are *isomorphic* if there is an ordering isomorphism f from J into K such that $a_j = b_{f(j)}$ for any j in J. This obviously defines an equivalence relation on words. In this paper, we identify isomorphic words and what we call a word is actually a class of isomorphic words.

2.4. Automata

We now recall the notion of automaton on words on linear orderings as introduced in [1,3]. They are usual (Kleene) automata with additional limit transitions of the form $P \to p$ and $p \to P$ where p is a state and P a subset of states.

Definition 2.7. Let A be a finite alphabet. An *automaton* $\mathcal{A}$ over A is a 5-tuple (Q, A, E, I, F) where Q is a finite set of states, $E \subseteq (Q \times A \times Q) \cup (\mathcal{P}(Q) \times Q) \cup (Q \times \mathcal{P}(Q))$ is the set of transitions, $I \subseteq Q$ is the set of initial states and $F \subseteq Q$ is the set of final states.

Since the alphabet and the set of states are finite, the set of transitions is also finite. Transitions are either *successor* transitions of the form $p \xrightarrow{a} q$, or *left limit* transitions of the form $P \to q$, or *right limit* transitions of the form $q \to P$, where P is a subset of Q.

Example 2.8. The automaton pictured in Figure 2 has three successor transitions which are pictured like a labeled graph. It has also a left limit transition $\{2\} \to 1$ and a right limit transition $0 \to \{1, 2\}$. State 0 is initial and state 1 is final.

We now come to the definition of a path in an automaton on linear orderings. Let x be a word of length J. Roughly speaking, a path associated with x is a labeling $(q_c)_{c\in\hat{J}}$ of each cut of J by a state of the automaton such that local properties are satisfied. If two cuts are consecutive there must be a successor transition labeled by the letter of x in between the two cuts. If a cut has no

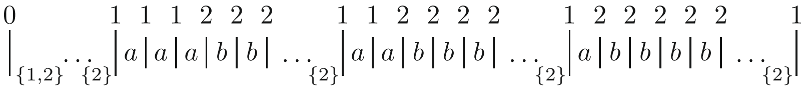

FIGURE 3. A path labeled by $x = \prod_{i \in -\omega} a^i a b^\omega$.

predecessor, a left limit transition must be used and if it has no successor, a right limit transition must be used.

To be more precise, the following notion of limit is needed. Let $\gamma = (q_c)_{c \in \hat{J}}$ be a path and let c be a fixed element of $\hat{J}$. The *left* and *right limit set* of γ at c are the two subsets $\lim_{c^-} \gamma$ and $\lim_{c^+} \gamma$ of Q defined as follows:

$$\lim_{c^-} \gamma = \{q \in Q \mid \forall c' < c \, \exists d \quad c' < d < c \text{ and } q = q_d\},$$
$$\lim_{c^+} \gamma = \{q \in Q \mid \forall c < c' \, \exists d \quad c < d < c' \text{ and } q = q_d\}.$$

Note that if the cut c is not the first cut of $\hat{J}$ and if it has no predecessor, the limit set $\lim_{c^-} \gamma$ is non-empty since the set Q is finite. A similar remark holds for right limit sets.

Definition 2.9. Let $\mathcal{A}$ be an automaton and let $x = (a_j)_{j \in J}$ be a word of length J. A *path* γ labeled by x is a sequence of states $\gamma = (q_c)_{c \in \hat{J}}$ of length $\hat{J}$ such that

- For any consecutive cuts c_j^- and c_j^+, $q_{c_j^-} \xrightarrow{a_j} q_{c_j^+}$ is a successor transition.
- For any cut c which is not the first cut and which has no predecessor, $\lim_{c^-} \gamma \to q_c$ is a left limit transition.
- For any cut c which is not the last cut and which has no successor, $q_c \to \lim_{c^+} \gamma$ is a right limit transition.

Since the ordering $\hat{J}$ has a first and a last element, a path always has a first and a last state which are indexed by the first and the last cut. A path is *accepting* if and only if its first state is initial and its last state is final. A word is *accepted* by the automaton if and only if it is the label of an accepting path. A set of words is *recognizable* if it is the set $\mathrm{L}(\mathcal{A})$ of words accepted by some automaton $\mathcal{A}$.

Example 2.10. Consider the automaton $\mathcal{A}$ of Figure 2 and let x be the word $\prod_{i \in -\omega} a^i a b^\omega$. An accepting path γ labeled by x is pictured in Figure 3. It can be verified that this automaton $\mathcal{A}$ accepts the set $(a^* a b^\omega)^{-\omega}$.

Remark 2.11. The automata that we have defined work for any linear ordering. It should be noted that in Section 4, the set $\mathrm{L}(\mathcal{A})$ of words accepted by an automaton is restricted to words whose length is a countable and scattered ordering.

3. Trees and orderings

In this section, we show that a countable linear ordering J can be seen as the set of leaves of a binary tree ordered with the infix ordering. Moreover, if J is scattered, we show that the set $\hat{J}^*$ of non-trivial cuts of J can be seen as the set of internal nodes, again ordered with the infix ordering.

3.1. Trees

A *tree* t is a prefix closed subset of $\{0,1\}^*$. An element of a tree is called a *node*. If $v0$ and $v1$ are nodes, they are called the *left* and the *right successor* of the node v. A *leaf* is a node with no successor and an *internal node* is a node with at least one successor. A node v is *non-branching* if it has only one successor. It is *complete* if it has either zero or two successors. A tree is *complete* if all its nodes are complete. The subtree rooted in a node v is made of the nodes of the form vw for w in $\{0,1\}^*$. If v is an internal complete node, the subtree rooted in the left successor of v is called the *left subtree* of v and the subtree rooted in the right successor of v is called the *right subtree* of v. A tree is *locally finite* if any of its subtrees contains a leaf.

We recall the *prefix ordering* $\sqsubseteq$ on the nodes of a tree t. We write $u \sqsubseteq v$ to denote that $v = uw$ for some word w. We also recall the *infix ordering* $<$. Two nodes satisfy $u < v$ if either $v0 \sqsubseteq u$, or $u1 \sqsubseteq v$, or $w0 \sqsubseteq u$ and $w1 \sqsubseteq v$ for some node w of t. Note that this is a linear ordering on the nodes which is countable. For a tree t, we call *frontier* of t the set of leaves of t ordered with the infix ordering.

Proposition 3.1. *Let J be a countable ordering. There exists a complete and locally finite tree t such that J is the frontier of t.*

Proof. Let us consider the tree $t = (00 + 11)^*(\varepsilon + 0 + 1 + 10)$ introduced in [12]. The set of leaves of t is $(00 + 11)^*10$ and its frontier is a dense countable ordering. It follows that the frontier of t is the ordering η of the rational numbers [11]. Since any countable ordering can be embedded in η, any ordering is thus the frontier of a tree. However this tree is not necessarily locally finite and complete.

We show how a locally finite and complete tree with the same frontier can be constructed from a given tree. This tree is constructed in two steps. The first step gives a locally finite tree with the same frontier. The second step gives a locally finite and complete tree from a locally finite tree.

The first step is quite easy. The nodes which have no leaf in their subtree are removed. This gives of course a locally finite tree. Furthermore, the frontier of the new tree is the same as the frontier of the starting tree since no leaf has been removed.

The second step starts from a locally finite tree. If a tree is not complete, it has some non-branching nodes. The idea is to remove all its non-branching nodes. Let $v_1, \ldots, v_n \ldots$ be a sequence of nodes such that each v_{i+1} is the only successor of v_i. This sequence is finite since the tree is locally finite. Let v_n be the last node. Then

v_n is either a leaf or a node with two successors. We remove the nodes $v_1, \ldots, v_{n-1}$ from the tree. If the node v_1 was the root of the tree, the node v_n becomes the new root. If the node v_1 was the left (respectively right) successor of a node v_0 with two successors, the node v_n becomes the left (respectively right) successor of the node v_0. If this procedure is performed on all sequences of non-branching nodes, one gets a tree which is locally finite and complete. Furthermore, the frontier is left unchanged. □

3.2. Trees and cuts

In this section, we consider the cuts of the frontier of trees. We show that if a tree is locally finite and complete, the cuts of its frontier can be viewed in the tree. More precisely, we show that these cuts are in a one-to-one correspondence with the internal nodes and some particular branches of the tree. Then we consider trees with a scattered frontier. In this case, the cuts correspond to the internal nodes.

A *branch* of a tree t is an infinite sequence $v_0v_1v_2\ldots$ of nodes such that v_0 is the root of t and each node v_{i+1} is a successor of the node v_i. A branch $v_0v_1v_2\ldots$ can be identified with the infinite word $b = b_0b_1b_2\ldots$ where $b_i = 0$ if v_{i+1} is the left successor of v_i and $b_i = 1$ if v_{i+1} is the right successor of v_i. A branch is called *straight* if b belongs to $(0+1)^*(0^\omega + 1^\omega)$ and it is called *zigzagging* otherwise.

We first show that the set made of all the nodes and all the branches of t can be endowed with a linear ordering which extends the infix ordering on nodes. This ordering is also denoted by the symbol $<$. Let $b = v_0v_1v_2\ldots$ and $b' = v'_0v'_1v'_2\ldots$ be two branches of t. Let k be the least integer such that $v_k = v'_k$ and $v_{k+1} \neq v'_{k+1}$. By definition, $b < b'$ holds if v_{k+1} and v'_{k+1} are respectively the left and right successor of $v_k = v'_k$. Let $b = v_0v_1v_2\ldots$ and v be a node of t. Let k be the integer such that v is in the subtree rooted in v_k but not in the subtree rooted in v_{k+1}. In particular, if v is a node of b, then $v_k = v$. By definition, the relation $v < b$ holds if v_{k+1} is the right successor of v_k and $b < v$ holds otherwise. In the former case, $v < v_i$ holds for any i greater than k and in the latter case, $v_i < v$ holds for any i greater than k. It is pure routine to check that this defines a linear ordering.

Let t be a locally finite and complete tree and let J be the frontier of t. Let J' be the set of all the internal nodes and the zigzagging branches of t. The set J' is of course ordered with the ordering that we have just defined. We claim that J' is isomorphic to the ordering $\hat{J}^*$ of non-trivial cuts of J.

Proposition 3.2. *Let t be a locally finite and complete tree. The set of internal nodes and zigzagging branches of t is isomorphic to the ordering $\hat{J}^*$ of non-trivial cuts of the frontier J of t.*

Proof. In order to prove the proposition, we define an isomorphism f from J' to $\hat{J}^*$. Let v be an element of J', that is, either an internal node or a zigzagging branch of t. We define $f(v)$ as the cut (K, L) where the sets K and L of leaves are given by $K = \{l \text{ leaf} \mid l < v\}$ and $L = \{l \text{ leaf} \mid v < l\}$. Since the ordering $<$ is linear, either $l < v$ or $v < l$ holds and (K, L) is a cut of the frontier of t. Let us show that (K, L) is a non-trivial cut. If v is an internal node, its left and right subtrees contain a leaf because the tree t is locally finite. Therefore K and L are non-empty. If v is a zigzagging branch $v_0 v_1 v_2 \ldots$, there is an integer k such that v_{k+1} is the right successor of v_k. The left subtree of v_k contains a leaf and K is non-empty. By symmetry, L is also non-empty.

We now prove that the function f is one to one. We first prove that f is injective. Let v and v' be two internal nodes of the tree t. By symmetry, we assume that $v < v'$. Let us show that there is a leaf l of t such that $v < l < v'$. Therefore the cuts $f(v)$ and $f(v')$ are different. We first suppose that $v'0 \sqsubseteq v$. The right subtree of v contains a leaf l since t is locally finite. This leaf l satisfies $v < l < v'$. The two other cases $v1 \sqsubseteq v'$ and $w0 \sqsubseteq v$, $w1 \sqsubseteq v'$ for some node w of t are similar. Let b be a branch $v_0 v_1 v_2 \ldots$ of t and let v be a node. Let k be the integer such that v is in the subtree rooted in v_k but not in the subtree rooted in v_{k+1}. By symmetry, we may assume that v_{k+1} is the right successor of v_k. Then, v is either equal to v_k or it is in the left subtree of v_k. In both cases, the left subtree of v_{k+1} contains a leaf l. This leaf satisfies $v < l$ and $l < b$, showing that the cuts $f(v)$ and $f(b)$ are different. Finally, let $b = v_0 v_1 v_2 \ldots$ and $b' = v'_0 v'_1 v'_2 \ldots$ be two different zigzagging branches of t. Let k be the integer such that $v_k = v'_k$ but $v_{k+1} \neq v'_{k+1}$. By symmetry, we may assume that v_{k+1} and v'_{k+1} are respectively the left and right successors of v_k. The right subtree of v_{k+1} contains a leaf l. Thus $b < l < b'$ and $f(b) \neq f(b')$.

We now prove that f is onto, that is, any non-trivial cut $c = (K, L)$ of J is equal to either $f(v)$ for some internal node v or to $f(b)$ for some zigzagging branch b. We proceed as follows. We define by induction a sequence $v_0 v_1 v_2 \ldots$ of internal nodes. Either, the definition stops after a finite number of steps and the cut c is equal to $f(v_n)$ or the definition gives an infinite sequence $v_0 v_1 v_2 \ldots$ of internal nodes. This infinite sequence turns to be a zigzagging branch b such that $c = f(b)$. Let v_0 be the root of the tree. Since c is a non-trivial cut, K and L are non-empty and the frontier of t contains at least two leaves. Therefore, v_0 cannot be a leaf and v_0 is an internal node. We suppose that we have already defined the nodes $v_0, \ldots, v_n$ of internal nodes such that each v_{i+1} is a successor of v_i. We also suppose by induction that the subtree rooted in v_n intersects both K and L. If $f(v_n)$ is equal to c, we have done. Otherwise, the left subtree or the right subtree of v_n intersects both K and L. Note that exactly one of the two subtrees satisfies this property. We define v_{n+1} as the successor of v_n whose subtree intersects both K and L. Note that v_{n+1} is an internal node since the subtree rooted in v_{n+1} intersects both K and L.

It remains to prove that when the sequence $v_0 v_1 \ldots v_n \ldots$ is infinite, one gets a zigzagging branch b of t such that $c = f(b)$. Suppose by contradiction that b is straight. By symmetry, we may assume that b is a branch of the form $(0+1)^* 1^\omega$. Let k be the least integer such that v_{i+1} is the right successor of v_i for every $i \geq k$. We claim that any leaf in the subtree rooted in v_k belongs to K. Let l be such a leaf. Let $m \geq k$ be the integer such that l is in the subtree rooted in v_m but not in the subtree rooted in v_{m+1}. Since v_{m+1} is the right successor of v_m, l is in the left subtree of v_m. By the choice of v_{m+1}, this subtree cannot intersect both K

and L. Therefore, it only intersects K and l belongs to K. Now if $k = 0$, v_k is the root of t and L is empty. This contradicts the fact that c is a non-trivial cut. If $k \geq 1$, the subtree rooted in v_k does not intersect K and L. This contradicts the choice of v_k. This proves that b is zigzagging.

We finally prove that $c = f(b)$. Let l be a leaf of t and let k be the integer such that l is in the subtree rooted in v_k but not in the subtree rooted in v_{k+1}. If l belongs to K, v_{k+1} must be the right successor of v_k. Therefore, $l < b$. Symmetrically, if l belongs to L, then $b < l$.

Therefore the function f is injective and onto. By definition, it is order preserving. This completes the proof of the proposition. □

In Proposition 3.2, only zigzagging branches are considered. Indeed, if b is a straight branch, it defines the same cut as the one of some internal node. For instance, if b is the branch $u01^\omega$, then it defines the cut $f(u)$ of the internal node u.

As already mentioned in Remark 2.5, a countable ordering J is scattered if and only if its set $\hat{J}$ of cuts is countable. By the previous proposition, the frontier of a locally finite and complete tree is scattered if the set of zigzagging branches of the tree is countable. Note that the set of straight branches is countable. Therefore, the set of zigzagging branches is countable if and only if the set of all branches is countable. The next proposition follows.

Proposition 3.3. *The frontier of a locally finite and complete tree is scattered if and only if the set of its branches is countable.*

A tree is said to be *straight* if all its branches are straight. The set of branches of a straight tree is countable and its frontier is thus scattered. Conversely we have the next proposition.

Proposition 3.4. *Any countable and scattered linear ordering is the frontier of a straight, locally finite and complete tree.*

Proof. The proof is based on Hausdorff's characterization of countable and scattered linear orderings (see Thm. 2.4). We prove by induction on the ordinal α that each ordering in the class V_α is the frontier of a straight, locally finite and complete tree. The empty ordering $\mathbf{0}$ and the one-element ordering $\mathbf{1}$ are respectively the frontier of the empty tree and the frontier of the tree with just one leaf. Suppose that the ordering K is equal to the sum $\sum_{j \in J} K_j$ where J is either a finite ordering, ω, $-\omega$ or ζ. By the induction hypothesis, each ordering K_j is the frontier of a straight, locally finite and complete tree t_j where each tree t_j is a prefix-closed subset of $\{0, 1\}^*$.

Suppose first that J is the finite ordering $\{0, \ldots, n\}$. We define the tree t by

$$t = \{\varepsilon, 1, 1^2, \ldots, 1^{n-1}\} \cup \bigcup_{j=0}^{n-1} 1^j 0 t_j \cup 1^n t_n.$$

It is straightforward to verify that t is prefix-closed and that it is a straight, locally finite and complete tree. Furthermore, its frontier is the ordering K. The other cases are treated similarly as follows. If J is the ordering ω, we define the tree t by $t = 1^* \cup \bigcup_{j\in\omega} 1^j 0 t_j$. The case $J = -\omega$ is symmetrical. If J is the ordering ζ, we define the tree t by $t = \{\varepsilon\} \cup 0t_0 \cup 1t_1$ where t_0 and t_1 are two straight, locally finite and complete trees such that $\mathrm{fr}(t_0) = \sum_{j\in-\omega} K_j$ and $\mathrm{fr}(t_1) = \sum_{j\in\omega} K_j$. □

The next result is a consequence of Proposition 3.2 since a straight tree has no zigzagging branch.

Corollary 3.5. *Let t be a straight, locally finite and complete tree. The set of the internal nodes of t is isomorphic to the ordering $\hat{J}^*$ of non-trivial cuts of the frontier J of t.*

3.3. Labeled trees

Let A be an alphabet. An *A-labeled tree* or *labeled tree* is a map $t : \mathrm{dom}(t) \to A$ where $\mathrm{dom}(t)$ is a tree (in the sense of Sect. 3.1) called the *domain*. The *word frontier* $\mathrm{fr}(t)$ of a labeled tree t is the word whose length is the frontier of t and whose labeling is given by the labeling of t. More precisely, if J is the frontier of the tree t, the word frontier of t is the word $x = (x_j)_{j\in J}$ where each letter x_j is equal to $t(j)$.

4. Logic and decidability

This section is devoted to the proof of the main result of this paper stated in the next theorem.

Theorem 4.1. *Given two automata $\mathcal{A}$ and $\mathcal{B}$ on countable and scattered linear orderings, it is decidable whether* $\mathrm{L}(\mathcal{A}) \subseteq \mathrm{L}(\mathcal{B})$.

It should be noted that in this statement, the sets $\mathrm{L}(\mathcal{A})$ and $\mathrm{L}(\mathcal{B})$ of words accepted by the automata $\mathcal{A}$ and $\mathcal{B}$ respectively, are restricted to words whose length is a countable and scattered linear ordering.

The following corollary is a trivial consequence of this theorem.

Corollary 4.2. *Given two automata $\mathcal{A}$ and $\mathcal{B}$ on countable and scattered linear orderings, it is decidable whether* $\mathrm{L}(\mathcal{A}) = \mathrm{L}(\mathcal{B})$.

The proof of Theorem 4.1 makes use of Rabin's tree theorem. Given an automaton $\mathcal{A}$, we are going to construct a logical formula $\varphi_{\mathcal{A}}$ such that every A-labeled tree t satisfies $\varphi_{\mathcal{A}}$ if and only if the word frontier of t is accepted by $\mathcal{A}$. To decide whether $\mathrm{L}(\mathcal{A})$ is included in $\mathrm{L}(\mathcal{B})$ will be reduced to decide whether formula $\varphi_{\mathcal{A}} \to \varphi_{\mathcal{B}}$ is valid. This proof only works for trees t that are straight, locally finite and complete. It follows that $\mathrm{L}(\mathcal{A})$ is restricted to words on countable and scattered linear orderings.

In Section 4.1, we recall the logical language that we use. In Section 4.2 we show how to construct formula $\varphi_{\mathcal{A}}$ from the automaton $\mathcal{A}$ and end with the proof of Theorem 4.1. In Section 4.3 we sketch another proof of Theorem 4.1 leaning on a different (though analogous) reduction.

4.1. Logical language

In [9], Rabin suggests to describe a set of A-labeled trees by a logical formula. The trees that are considered are *full* trees with domain $\{0,1\}^*$. In order to handle any locally finite and complete tree t labeled by $A = \{a_1, \dots, a_n\}$, we introduce a new letter a_0 and we label by a_0 any node outside the domain of t. The resulting full tree is denoted by $\bar{t}$.

Let A be a fixed alphabet. We recall from [9] the *monadic second-order logical language* $\mathcal{L}_A$ which allows to describe full A-labeled trees. This language has first-order variables $x, y, \dots$ ranging over nodes of the trees and second-order variables $X, Y, \dots$ ranging over subsets of nodes. It has also the symbol $=$ for equality, symbols S_0 and S_1 for the two successors functions on $\{0,1\}^*$, and unary symbols R_a, $a \in A$, for nodes labeled by a. As usual, atomic formulas are of the form $x = y$, $S_0(x) = y$, $S_1(x) = y$, $R_a(x)$ and $X(x)$. Arbitrary formulas are constructed by induction from atomic formulas in combination with the logical constructors $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$ and the quantifiers $\exists$ and $\forall$ acting on first-order and second-order variables. A *sentence* is a formula without free variables. Given a sentence φ and a full A-labeled tree t, the relation $t \models \varphi$ saying that t *satisfies* φ is defined in the standard way. The set of trees satisfying a formula φ is denoted by $T(\varphi)$.

A sentence φ is called *valid* if $T(\neg\varphi) = \emptyset$, that is, if every tree t satisfies φ.

Theorem 4.3 (Rabin [9])**.** *Let φ be a sentence of the monadic second-order language $\mathcal{L}_A$. It is decidable whether φ is valid.*

4.2. Reduction to logics over trees

We are now equipped to construct a logical formula $\varphi_{\mathcal{A}}$ describing the set of words accepted by an automaton $\mathcal{A}$ when these words are seen as word frontier of trees.

Proposition 4.4. *Let $A = \{a_1, \dots, a_n\}$ be an alphabet and a_0 be a new letter. For any automaton $\mathcal{A}$ over A and on countable and scattered linear orderings, one can construct a sentence $\varphi_{\mathcal{A}}$ over the language $\mathcal{L}_{A \cup \{a_0\}}$ such that*

$$T(\varphi) = \{\bar{t} \quad | \quad \mathrm{fr}(t) \in \mathrm{L}(\mathcal{A}) \setminus \{\varepsilon\}\}.$$

Before proceeding to the proof of the proposition, we illustrate the main idea by an example.

Example 4.5. We consider again the word $x = \prod_{i \in -\omega} a^i ab^\omega$ accepted by the automaton $\mathcal{A}$ of Figure 2. The accepting path of Figure 3 can be represented as the labeled tree pictured in Figure 4. Indeed as the length J of x is a countable and scattered ordering, it can be seen as the frontier of a straight, locally finite tree thanks to Proposition 3.4. By Corollary 3.5, the non-trivial cuts of J can be identified with the internal nodes of this tree.

Proof. Before giving the definition of $\varphi_{\mathcal{A}}$, it is convenient to introduce special predicates, in particular for expressing that a full $A \cup \{a_0\}$-labeled tree has the

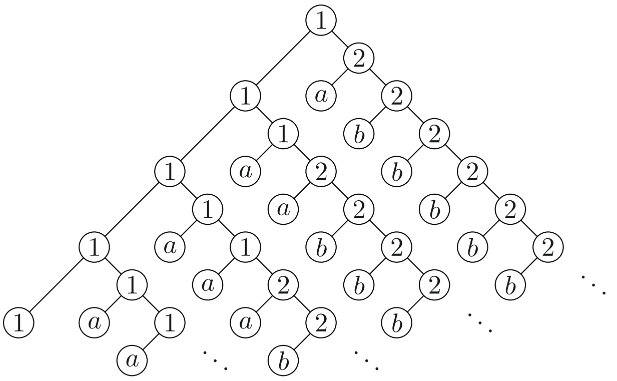

FIGURE 4. Tree.

form $\bar{t}$ where t is a straight, locally finite and complete A-labeled tree. The trivial definitions of some of these predicates have been omitted.

(1) Ordering predicates
- The prefix ordering $x \sqsubseteq y$ on nodes x and y is defined as follows

$$\forall X \quad [X(x) \wedge \forall z(X(z) \rightarrow X(S_0(z)) \wedge X(S_1(z)))] \rightarrow X(y).$$

- For the infix ordering $x < y$, we have the following formula

$$S_0(y) \sqsubseteq x \ \vee \ S_1(x) \sqsubseteq y \ \vee \ \exists z \ (S_0(z) \sqsubseteq x \wedge S_1(z) \sqsubseteq y).$$

- Given two nodes x and y not labeled by a_0, it is then easy to define a predicate $\mathrm{Succ}(x, y)$ stating that y is the successor of x with respect to the infix ordering.

(2) Node predicates
- $\mathrm{Root}(x)$ means that the node x is the root of the tree.
- $\mathrm{Father}(x, y)$ means that the node x is the father of the node y.
- $\mathrm{Brother}(x, y)$ means that nodes x and y are brothers.
- $\mathrm{Leaf}(x)$ means that x is a leaf in the tree obtained by removing a_0-labeled nodes. Thus $\mathrm{Leaf}(x)$ stands for

$$\neg R_{a_0}(x) \wedge R_{a_0}(S_0(x)) \wedge R_{a_0}(S_1(x)).$$

- $\mathrm{Intern}(x)$ means that x is an internal node in the tree obtained by removing a_0-labeled nodes. Thus $\mathrm{Intern}(x)$ stands for

$$\neg R_{a_0}(x) \wedge \neg \, \mathrm{Leaf}(x).$$

(3) Branch predicates

- The predicate Branch(X) states that X is the set of nodes of a branch. It states that the root of the tree is in X, that any two elements of X are $\sqsubseteq$-comparable and that for any node x in X, one of its successors is also in X. It is defined as follows

$$\begin{aligned} & \exists x \quad \mathrm{Root}(x) \wedge X(x) \\ \wedge \quad & \forall x, y \ \ (X(x) \wedge X(y) \to (x \sqsubseteq y \vee y \sqsubseteq x)) \\ \wedge \quad & \forall x \ \ X(x) \to (X(S_0(x)) \vee X(S_1(x))). \end{aligned}$$

- The predicate StraightBranch(X) states that a branch is straight. The first part of the formula describes a branch of the form $u0^\omega$ and the second part a branch of the form $u1^\omega$. It is defined as follows

$$\begin{aligned} & \exists x \ \ X(x) \wedge \forall y \ (X(y) \wedge x \sqsubseteq y \to X(S_0(y))) \\ \vee \quad & \exists x \ \ X(x) \wedge \forall y \ (X(y) \wedge x \sqsubseteq y \to X(S_1(y))). \end{aligned}$$

(4) Tree predicate

Using all the previous predicates, it can be expressed by a predicate that a full tree $t' : \{0,1\}^* \to A \cup \{a_0\}$ is equal to $\bar{t}$ where t is a non-empty, straight, locally finite, complete and A-labeled tree. This predicate expresses the following facts.

(a) At least one node of t' is not labeled by a_0 (t is non-empty) and for any node u, there is $v \in \{0,1\}^*$ such that $t'(uv) = a_0$ (t is locally finite).

(b) The two successors and the brother of a node labeled by a_0 are also labeled by a_0 (t is complete).

(c) Any branch with no node labeled by a_0 is straight (t is straight).

The predicate GoodTree is defined as follows.

$$\begin{aligned} & \exists x \ \ \neg R_{a_0}(x) \\ \wedge \quad & \forall x \, \exists y \ \ x \sqsubseteq y \wedge R_{a_0}(y) \\ \wedge \quad & \forall x \ R_{a_0}(x) \to (R_{a_0}(S_0(x)) \wedge R_{a_0}(S_1(x))) \\ \wedge \quad & \forall x \, \forall y \ (R_{a_0}(x) \wedge \mathrm{Brother}(x,y)) \to R_{a_0}(y) \\ \wedge \quad & \forall X \ (\mathrm{Branch}(X) \wedge \forall x \ (X(x) \to \neg R_{a_0}(x))) \to \mathrm{StraightBranch}(X). \end{aligned}$$

Let $\mathcal{A} = (Q, A, E, I, F)$ be an automaton on linear orderings. Without loss of generality, we may assume that $Q = \{1, \ldots, m\}$. We have to find a sentence $\varphi_{\mathcal{A}}$ which expresses that a non-empty word $\alpha = (a_j)_{j \in J}$ of length J is accepted by $\mathcal{A}$, where J is a countable and scattered linear ordering. This means that there is an accepting path $\gamma = (q_c)_{c \in \hat{J}}$ labeled by α. By Proposition 3.4, let t be a non-empty, straight, locally finite, complete and A-labeled tree t such that $\mathrm{fr}(t) = \alpha$. By Corollary 3.5, we know that the set $\hat{J}^*$ of non-trivial cuts of J can be identified with the set of internal nodes of t. Thus, the sentence $\varphi_{\mathcal{A}}$ will state the existence of a partition $Z_1, \ldots, Z_m$ of the internal nodes of t such that nodes of Z_i are labeled with state i, $1 \leq i \leq m$, with respect to path γ. In other words, sentence $\varphi_{\mathcal{A}}$ will state that the labeling of the cuts by states represents a path respecting the transitions of $\mathcal{A}$, starting in an initial state and ending in a final state. The main difficulties are the description of the left and right limits of γ at some cut and the treatment of the first and last cuts of γ which have no corresponding internal nodes in t.

Let P be a subset of Q and let x be an internal node (*i.e.*, a cut $c \in \hat{J}^*$). Let us define a predicate $\text{LeftLimit}_P(x)$ which states that the left limit at node x is equal to P. It is given by the following formula

$$\bigwedge_{i \in P} (i \in \lim_{x^-} \gamma) \quad \wedge \quad \bigwedge_{i \in Q \setminus P} \neg(i \in \lim_{x^-} \gamma)$$

where formula $i \in \lim_{x^-} \gamma$ stands for

$$\forall y \;\; (\text{Intern}(y) \wedge y < x) \rightarrow (\exists z \;\; \text{Intern}(z) \wedge (y < z) \wedge (z < x) \wedge Z_i(z)).$$

A predicate $\text{RightLimit}_P(x)$ is defined similarly.

We now treat the particular case where c is the last cut $(J, \emptyset)$ of $\hat{J}$. Even if no node x corresponds to c, it can be expressed that the left limit at c is equal to P thanks to predicate LastLeftLimit_P. Its definition is very close to the definition of $\text{LeftLimit}_P(x)$ but there is no mention to x.

$$\bigwedge_{i \in P} (i \in \lim_{c^-} \gamma) \quad \wedge \quad \bigwedge_{Q \setminus P} \neg(i \in \lim_{c^-} \gamma)$$

where formula $i \in \lim_{c^-} \gamma$ stands for

$$\forall y \;\; \text{Intern}(y) \rightarrow (\exists z \;\; \text{Intern}(z) \wedge (y < z) \wedge Z_i(z)).$$

A predicate FirstRightLimit_P is defined similarly for the particular case of the right limit at the first cut $(\emptyset, J)$ of $\hat{J}$.

The expected formula $\varphi_{\mathcal{A}}$ is constructed as follows. It is equal to the formula

$$\text{GoodTree} \quad \wedge \quad \exists Z_1, \ldots, Z_m \;\; \varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_8$$

where formulas $\varphi_1, \ldots, \varphi_8$ are given and explained below. Let t' be a full $A \cup \{a_0\}$-labeled tree satisfying $\varphi_{\mathcal{A}}$. Formula GoodTree verifies that the tree t' is equal to $\bar{t}$ where the tree t is non-empty, straight, locally finite, complete and A-labeled. Formula φ_1 verifies that the sets $Z_1, \ldots, Z_m$ form a partition of the internal nodes of t. It is defined as follows

$$\forall x \left[\text{Intern}(x) \leftrightarrow \bigvee_{i=1}^{m} Z_i(x) \quad \wedge \quad \bigwedge_{1 \leq i < j \leq m} \neg(Z_i(x) \wedge Z_j(x)) \right].$$

The partition $Z_1, \ldots, Z_m$ of the internal nodes is then viewed as a labeling of each internal node by a state. Each node in Z_i is labeled by the state i of Q. Since internal nodes are in one to one correspondence with the cuts, this gives a labeling of each non-trivial cut by a state. Formulas $\varphi_2, \ldots, \varphi_8$ verify that this labeling defines an accepting path in $\mathcal{A}$. Formula φ_2 verifies that it is compatible with the successor transitions of the automaton $\mathcal{A}$. It is defined as follows

$$\forall\, x, y, z \ \ \big(\mathrm{Intern}(x) \wedge \mathrm{Succ}(x,y) \wedge \mathrm{Leaf}(y) \wedge \mathrm{Succ}(y,z) \wedge \mathrm{Intern}(z)\big) \to \Big(\bigvee_{(i,a,j)\in E} Z_i(x) \wedge R_a(y) \wedge Z_j(z)\Big).$$

Formulas φ_3 and φ_4 verify respectively that the labeling is compatible with the left and right limit transitions of $\mathcal{A}$. Formula φ_3 is defined as follows

$$\forall x \ \ \big(\mathrm{Intern}(x) \wedge \neg(\exists y \ \ \mathrm{Succ}(y,x))\big) \to \Big(\bigvee_{(P,j)\in E} Z_j(x) \wedge \mathrm{LeftLimit}_P(x)\Big)$$

and formula φ_4 is defined analogously as follows

$$\forall x \ \ \big(\mathrm{Intern}(x) \wedge \neg(\exists y \ \ \mathrm{Succ}(x,y))\big) \to \Big(\bigvee_{(i,P)\in E} Z_i(x) \wedge \mathrm{RightLimit}_P(x)\Big).$$

Formulas φ_5 and φ_6 deal with the last transition of the path. This transition is either a successor transition or a limit transition and it must lead to a final state. Formula φ_5 handles the former case and it is defined as follows

$$\forall\, x, y \ \ \big(\mathrm{Intern}(x) \wedge \mathrm{Succ}(x,y) \wedge \mathrm{Leaf}(y) \wedge \forall z \ \ (y < z \to R_{a_0}(z))\big) \to \Big(\bigvee_{(i,a,j)\in E, j\in F} Z_i(x) \wedge R_a(y)\Big).$$

Formula φ_6 handles the latter case and it is defined as follows

$$\forall y \ \ \mathrm{Leaf}(y) \to \big(\exists z \ \ y < z \wedge \neg R_{a_0}(z)\big) \to \Big(\bigvee_{(P,j)\in E, j\in F} \mathrm{LastLeftLimit}_P\Big).$$

Formulas φ_7 and φ_8 deal with the first transition of the path. Formula φ_7 is defined as follows

$$\forall\, y, z \ \ \big(\mathrm{Leaf}(y) \wedge \mathrm{Succ}(y,z) \wedge \mathrm{Intern}(z) \wedge \forall z \ \ (z < y \to R_{a_0}(z))\big) \to \Big(\bigvee_{(i,a,j)\in E, i\in I} R_a(y) \wedge Z_j(z)\Big).$$

Formula φ_8 is finally defined as follows

$$\forall y \ \ \mathrm{Leaf}(y) \to \big(\exists z \ \ z < y \wedge \neg R_{a_0}(z)\big) \to \Big(\bigvee_{(i,P)\in E, i\in I} \mathrm{FirstRightLimit}_P\Big).$$

Note that in $\varphi_{\mathcal{A}}$, the predicate GoodTree imposes that t is not the empty tree. Therefore t' satisfies $\varphi_{\mathcal{A}}$ if and only if $t' = \bar{t}$ with $\mathrm{fr}(t)$ being a non-empty word accepted by $\mathcal{A}$. This completes the proof of the proposition. □

We are now ready to prove Theorem 4.1.

Proof. Let $\mathcal{A}$ and $\mathcal{B}$ be two automata on linear orderings. Consider formulas $\varphi_{\mathcal{A}}$ and $\varphi_{\mathcal{B}}$ constructed in Proposition 4.4. They describe all the non-empty words accepted respectively by $\mathcal{A}$ and $\mathcal{B}$. The case of the empty word must be treated separately.

Let us first show that the inclusion $\mathrm{L}(\mathcal{A})\backslash\{\varepsilon\} \subseteq \mathrm{L}(\mathcal{B})\backslash\{\varepsilon\}$ is decidable. This is equivalent to show the decidability of the satisfiability $t \models \varphi_{\mathcal{A}} \rightarrow \varphi_{\mathcal{B}}$ for all full $A \cup \{a_0\}$-labeled trees. This is of course a consequence of Theorem 4.3. It is decidable whether the sentence $\varphi_{\mathcal{A}} \rightarrow \varphi_{\mathcal{B}}$ is valid.

Secondly let us show that the inclusion $\mathrm{L}(\mathcal{A}) \cap \{\varepsilon\} \subseteq \mathrm{L}(\mathcal{B}) \cap \{\varepsilon\}$ is decidable. This is immediate since an automaton accepts the empty word if and only if it has a state which is both initial and final. □

4.3. Reduction to logics over orderings

Let us sketch here a variant of the proof detailed in subsection 4.2: it consists in expressing the property $\mathrm{L}(\mathcal{A}) \subseteq \mathrm{L}(\mathcal{B})$ by a logical formula over *countable orderings* (instead of trees). Since the monadic second-order theory of the class of countable orderings is decidable this also gives a decidability proof for the inclusion problem.

Let us call *monadic second-order logical language* $\mathcal{L}_o$ the set of well-formed monadic second order formulas over a binary predicate $\preceq$ (expressing an ordering relation). We use now the following

Theorem 4.6 (Thm. 2.1, p. 11 of [9])**.** *Let φ be a sentence of the monadic second-order language $\mathcal{L}_o$. It is decidable whether φ is valid in every countable linearly ordered set.*

Let us associate with every countable scattered ordering J the ordered set $O(J)$ defined by $O(J) := J \cup \hat{J}$ and

$$\forall x, y \in J, x \leq_{O(J)} y \Leftrightarrow x \leq_J y; \quad \forall x, y \in \hat{J}, x \leq_{O(J)} y \Leftrightarrow x \leq_{\hat{J}} y;$$

$$\forall x \in J, (K, L) \in \hat{J}, \quad x \in K \Rightarrow x \leq_{O(J)} (K, L), \quad x \in L \Rightarrow (K, L) \leq_{O(J)} x.$$

By Remark 2.5, J is countable and scattered if and only if $O(J)$ is countable. From the automata $\mathcal{A}$ and $\mathcal{B}$, one can write a MSO formula φ which holds in $O(J)$ if and only if, for every labelling of J over A, if there exists a labelling of $\hat{J}$ over $Q_{\mathcal{A}}$ which is an accepting path of $\mathcal{A}$, then there exists a labelling of $\hat{J}$ over $Q_{\mathcal{B}}$ which is an accepting path of $\mathcal{B}$. One can also write a MSO formula ψ such that ψ holds in a countable ordering J' if and only if there exists some countable scattered ordering J such that $J' \approx O(J)$. Finally, the formula $\psi \rightarrow \varphi$ is valid for all countable orderings if and only if $\mathrm{L}(\mathcal{A}) \subseteq \mathrm{L}(\mathcal{B})$. Thus Theorem 4.1 is reduced to Theorem 4.6.

5. Conclusion

Let us sketch a few problems that are raised by our work.

Does Theorem 4.1 still hold for arbitrary countable orderings, that is, is the inclusion problem decidable for general rational languages of words indexed by arbitrary countable linear orderings?

Let us call *normal* every A-labeled tree fulfilling conditions a-b-c of the predicate GoodTree. Let us say that a set T of normal trees is *saturated* if and only if for any normal trees t and t', $t \in T$ and the equality $\mathrm{fr}(t) = \mathrm{fr}(t')$ imply $t' \in T$. We have proved that for any rational set L of words indexed by scattered linear orderings, there exists a saturated rational set T of normal trees such that

$$L = \{\mathrm{fr}(t) \mid t \in T\}.$$

Conversely it is natural to ask whether, given a saturated rational set T of normal trees, the language $\{\mathrm{fr}(t) \mid t \in T\}$ is rational.

The following related question seems also interesting: is it decidable whether a rational set of normal trees is saturated or not.

Acknowledgements. We thank one of the anonymous referees for suggesting the contents of Section 4.3.

References

[1] V. Bruyère and O. Carton, Automata on linear orderings, edited by J. Sgall, A. Pultr and P. Kolman, *MFCS'2001. Lect. Notes Comput. Sci.* **2136** (2001) 236–247.

[2] V. Bruyère, O. Carton and G. Sénizergues, Tree automata and automata on linear orderings, in *Proceedings WORDS'03. Lect. Notes Comput. Sci.* **27** (2003) 222–231. TUCS General Publication.

[3] V. Bruyère and O. Carton, Automata on linear orderings. *J. Comput. System Sci.* **73** (2007) 1–24.

[4] O. Carton, Accessibility in automata on scattered linear orderings, edited by K. Diks and W. Rytter, *MFCS'2002. Lect. Notes Comput. Sci.* **2420** (2002) 155–164.

[5] B. Courcelle, Frontiers of infinite trees. *RAIRO Theoretical Informatics* **12** (1978) 319–337.

[6] F. Hausdorff, Grundzüge einer Theorie der geordneten Mengen. *Math. Ann.* **65** (1908) 435–505.

[7] S. Heilbrunner, An algorithm for the solution of fixed-point equations for infinite words. *RAIRO Theoretical Informatics* **14** (1980) 131–141.

[8] S.C. Kleene, Representation of events in nerve nets and finite automata, edited by C.E. Shannon, *Automata studies*, 3–41. Princeton University Press, Princeton (1956).

[9] M.O. Rabin, Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.* **141** (1969) 1–35.

[10] C. Rispal and O. Carton, Complementation of rational sets on countable scattered linear orderings. *J. Found. Comput. Sci.* **16** (2005) 767–786.

[11] J.G. Rosenstein, *Linear Orderings*. Academic Press, New York (1982).

[12] W. Thomas, On frontiers of regular sets. *RAIRO Theoretical Informatics* **20** (1986) 371–381.

Communicated by C. Choffrut.
Received June 15, 2007. Accepted November 5, 2008.