Decidability of conjugacy of tree shifts of finite type

Nathalie Aubrun and Marie-Pierre Béal

Université Paris-Est
CNRS, Laboratoire d'informatique Gaspard-Monge

Abstract. A one-sided (resp. two-sided) shift of finite type of dimension one can be described as the set of infinite (resp. bi-infinite) sequences of consecutive edges in a finite-state automaton. While the conjugacy of shifts of finite type is decidable for one-sided shifts of finite type of dimension one, the result is unknown in the two-sided case.
In this paper, we study the shifts of finite type defined by infinite trees. Indeed, infinite trees have a natural structure of one-sided shifts, between the shifts of dimension one and two. We prove a decomposition theorem for these shifts, *i.e.* we show that a conjugacy between two shifts of finite type can be broken down into a finite sequence of elementary transformations called in-splittings and in-amalgamations. We prove that the conjugacy problem is decidable for tree shifts of finite type. This result makes the class of tree shifts closer to the class of one-sided shifts of dimension one than to the class of two-sided ones. Our proof uses the notion of bottom-up tree automata.

1 Introduction

Sofic shifts are bi-infinite sequences labeling paths in a finite automaton. Shifts of finite type are a particular important subclass of sofic shifts. Two-sided (resp. one-sided) shifts of finite type are bi-infinite (resp. right-infinite) sequences of consecutive edges in a finite-state automaton (see [8, 13.8], [5]). They are well understood in the one-sided case since the conjugacy is decidable for such shifts [14]. The proof uses the decomposition theorem (see for instance [5]). This theorem states that every conjugacy between two one-sided shifts of finite type can be decomposed into a finite sequence of splittings and amalgamations, which are elementary operations on automata presenting the two shifts.

In the two-sided case, the decidability of the conjugacy problem between two shifts of finite type is still an open question. In higher dimension, many questions become more difficult. The main reason is that there exists no good representation of multidimensional shifts comparable to finite automata in dimension one. Even if there exists a generalization of finite automata to dimension two, which are called textile systems (see [10], see also the automata for tiling systems in [2]), results are more complex than in dimension one. The decomposition theorem can be extended to two-sided multidimensional shifts of finite type, but an additional operation, called an inversion, is needed (see [4]).

In this paper, we introduce the notion of shifts of finite type defined on infinite trees, that we call tree shifts. Indeed, infinite trees have a natural structure of one-sided symbolic systems equipped with several shift transformations. The ith shift transformation applied to a tree gives the subtree rooted at the child number i of the tree. This defines a new class of shifts between the class of one-sided shifts of dimension one and the class of one-sided shifts of higher dimension. Tree shifts can be described thanks to top-down or bottom-up tree automata which are used in automata theory for many purposes. Tree automata have applications to logic and game theory (see [13], [1], [11], and [12]). The tree automata that we consider here are bottom-up tree automata. They are simpler than Büchi or Muller tree automata since they have all their states final.

We define two elementary operations on tree automata: the in-splitting operation and the in-amalgamation operation. They are very close to those existing on finite automata. In particular two in-amalgamations commute. We prove a decomposition theorem for tree shifts of finite type, *i.e.* we show that a conjugacy between two shifts of finite type can be broken down into a finite sequence of in-splittings and in-amalgamations. We then prove that the conjugacy problem is decidable for this class of shifts. The heart of the proof is the commutation property of in-amalgamations. We prove that two tree shifts of finite type are conjugate if and only if they have the same minimal in-amalgamation. Furthermore, the minimal in-amalgamation of a tree automaton can be computed in a polynomial time in the number of states of the automaton.

The paper is organized as follows. In Section 2 we give basic definitions about tree shifts and tree automata. The decomposition theorem is proved in Section 3. Our main result together with an example are given in Section 4. We end the paper with some concluding remarks.

2 Shifts, automata and infinite trees

2.1 Tree shifts

We give here some basic definitions from symbolic dynamics which apply to infinite trees. We consider infinite trees whose nodes have a fixed number of children and are labeled in a finite alphabet.

Let $\Sigma = \{0, 1, .., d-1\}$ be a finite alphabet of cardinal d. An *infinite tree* t over a finite alphabet A is a complete function from Σ^* to A. Unless otherwise stated, a tree is an infinite tree. A node of a tree is a word of Σ^*. The empty word, that corresponds to the root of the tree, is denoted by ϵ. If x is a node, its children are xi with $i \in \Sigma$. Let t be a tree and let x be a node, we shall sometimes denote $t(x)$ by t_x.

When Σ is fixed, we denote by $\mathcal{T}(A)$ the set of all infinite trees on A, hence the set A^{Σ^*}. On this set we have a natural metric. If t, t' are two trees, we define the distance $d(t, t') = \frac{1}{n+1}$, where n is the length of the shortest word x in Σ^* such that $t(x) \neq t'(x)$ if such a word exists, and $d(t, t) = 0$. This metric induces a topology equivalent to the usual product topology, where the topology in A is the discrete one.

We define the shift transformations σ_i for $i \in \Sigma$ from $\mathcal{T}(A)$ to itself as follows. If t is a tree, $\sigma_i(t)$ is the tree rooted at the i-th child of t, *i.e.* $\sigma_i(t)_x = t_{ix}$ for any $x \in \Sigma^*$. The set $\mathcal{T}(A)$ equipped with the shift transformations σ_i is called the *full shift* of infinite trees over A.

A *pattern* is a function $p : L \to A$, where L is a finite subset of Σ^* containing the empty word. The set L is called the *support of the pattern.* A *block* of height n is a pattern with support $\Sigma^{\leq n}$, where n is some nonnegative integer, and $\Sigma^{\leq n}$ denotes the words of length at most n of letters of Σ.

We say that a pattern b of support L is a *block of a tree* t if there is a word $x \in \Sigma^*$ such that $t_{xy} = b_y$ for any $y \in \Sigma^*$. We say that b is a block of t rooted at the node x. If b is not a block of t, one says that t *avoids* p. If b is a block of some tree of subshift X, it is called an *allowed block* of X.

We define a (tree) *subshift* (or *shift*) X of $\mathcal{T}(A)$ as the set $\mathsf{X}_{\mathcal{F}}$ of all trees avoiding each pattern of a set of blocks $\mathcal{F}$. This subshift X is closed and for any shift transformation σ_i, $\sigma_i(X) \subseteq X$. A *subshift of finite type* X of $\mathcal{T}(A)$ is a set $\mathsf{X}_{\mathcal{F}}$ of all trees avoiding each block of a *finite* set of blocks $\mathcal{F}$. The set $\mathcal{F}$ is called *a set of forbidden blocks* of X.

We denote by $\mathcal{L}(X)$ the set of blocks of all trees of a shift X, and by $\mathcal{L}_n(X)$ the set of all blocks of height n of X. If b is a block of height n with $n \geq 1$, we denote by $\sigma_i(b)$ the block of height $n-1$ such that $\sigma_i(b)_x = b_{ix}$ for $x \in \Sigma^{\leq n-1}$. The block b is written $b = (b_\varepsilon, \sigma_0(b), .., \sigma_{d-1}(b))$.

Example 1. In figure 1 is pictured an infinite tree of a shift of finite type $\mathsf{X}_{\mathcal{F}}$ on the binary alphabet $\{0, 1\}$ defined by a finite set $\mathcal{F}$ of forbidden blocks of height 2. The forbidden blocks are those whose labels have a sum equal to 1 modulus 2.

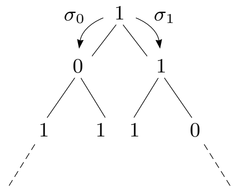

Fig. 1. A tree of the tree shift of finite type $\mathsf{X}_{\mathcal{F}}$ on the alphabet $\{0, 1\}$, where $\mathcal{F}$ is the set of blocks of height 2 whose sum of labels is 1 modulus 2.

Let A, A' be two finite alphabets, X be a subshift of $\mathcal{T}(A)$ and m be a nonnegative integer. A map $\Phi : X \subseteq \mathcal{T}(A) \to \mathcal{T}(A')$ is called a $(m+1)$-*local map* (or a $(m+1)$-*block map*) if there exists a function $\phi : \mathcal{L}_{m+1}(X) \to A'$ such that, for any $x \in \Sigma^*$, $\Phi(t)_x = \phi(t_{|x\Sigma^{\leq m+1}})$, where $t_{|x\Sigma^{\leq m+1}}$ is the pattern q such that $q_y = t_{xy}$ for any $y \in \Sigma^{\leq m+1}$. The smallest integer m satisfying this property is called the *memory* of the block map. A *block map* is a map which is $(m+1)$-local for some nonnegative integer m.

It is known from the Curtis-Lyndon-Hedlund theorem (see [3]) that block maps are exactly the maps $\Phi : X \to Y$ which are continuous and commute with all shifts transformations, *i.e.* such that $\sigma_i(\Phi(t)) = \Phi(\sigma_i(t))$ for any $t \in X$ and any $i \in \Sigma$. The image of X by a block map is also a shift. A one-to-one and onto block map from a shift X onto a shift Y has an inverse which is also a block map. It is called a *conjugacy* from X onto Y. The tree subshifts X and Y are then *conjugate.*

Example 2. Let X the tree shift of finite type defined in Example 1. Let Y be the tree shift of finite type over the alphabet $\{a, b, c\}$, where the allowed blocks of height 2 are (a, a, a), (a, b, c), (a, c, b), (a, c, c), (b, b, a), (b, c, a), (c, a, b) and (c, a, c). The 2-block map $\Phi : X \to Y$, defined by $\phi(0, 0, 0) = a$, $\phi(0, 1, 1) = a$, $\phi(1, 1, 0) = b$, and $\phi(1, 0, 1) = c$, is pictured in Figure 2. The map Φ is a conjugacy. Its inverse is a 1-block map Ψ defined by $\psi(a) = 0$ and $\psi(b) = \psi(c) = 1$.

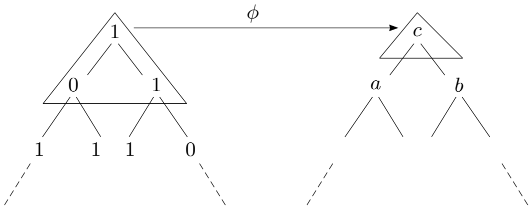

Fig. 2. A 2-block map $\Phi : X \to Y$, where X is the tree shift of Figure 1 and Y a shift of finite type over the alphabet $\{a, b, c\}$. The map Φ is a conjugacy.

Let X be a tree subshift on A. Let n be a positive integer. The *higher block presentation* of order n of X is the shift $\hat{X}$ on the alphabet $\mathcal{L}_n(X)$ made of trees t such there is a tree $t' \in X$ such that, for any node x, t_x is the block of height n of t' rooted at x. A tree shift is conjugate to any of its higher block presentations.

2.2 Tree automata

In this section we consider bottom-up automata for infinite trees. Such an automata its computation from the infinite branches and moves upward. A *tree automaton* is here a structure $\mathcal{A} = (V, A, \Delta)$ where V is a finite set of states (or vertices), A is a finite set of input symbols, and Δ is a set of transitions of the form $(q_0, .. , q_{d-1}), a \to q$, with $q, q_i \in V$, $a \in A$. A transition $(q_0, .. , q_{d-1}), a \to q$ is called a transition *labeled by* a, *going out of* the d-uple of states $(q_0, .. , q_{d-1})$ and *coming in* the state q. Note that no initial nor final states are specified. This means that all states are both initial and final.

Such an automaton is *deterministic* if the set of transitions defines a partial function δ from $V^d \times A$ to V.

A (bottom-up) *computation* of $\mathcal{A}$ on the infinite tree t is an infinite tree C on V such that, for each node x, there is a transition $(C_{x0}, .., C_{x(d-1)}, t_x) \to C_x \in \Delta$. A tree t is *accepted* by $\mathcal{A}$ if there exists a computation of $\mathcal{A}$ on t. The set of infinite trees accepted by $\mathcal{A}$ is a subshift.

Let m be a nonnegative integer. An *m-deterministic local tree automaton* (or an *m-definite tree automaton*) is a tree automaton $\mathcal{A} = (V, A, \delta)$ such that whenever t and t' are two trees accepted by $\mathcal{A}$ with a same block b of height m rooted at the node x in t and rooted at the node x' in t', for any computation C of t and any computation C' of t', we have $C_x = C'_{x'}$. A tree automaton is *local* (or *definite*) if it is m-local for some nonnegative integer m.

Proposition 1. *Any tree shift of finite type is accepted by a deterministic local tree automaton. Reciprocally any tree shift accepted by a deterministic local tree automaton is of finite type.*

Proof. The proof of this proposition id given in the appendix section.

Let $\mathcal{A} = (V, A, \Delta)$ be a tree automaton such that whenever

$$(q_0, .., q_{d-1}, a) \to q \in \Delta \text{ and } (q_0, .., q_{d-1}, b) \to q \in \Delta, \text{ one has } a = b.$$

Note that the automaton built in the proof of Proposition 1 satisfies this condition. The set of computations in the tree automaton $\mathcal{A}$ defines a subshift called the *vertex shift* (or *Markov shift*) defined by $\mathcal{A}$. Equivalently, a vertex shift is a shift accepted by a 2-local automaton, *i.e.* the tree subshift of finite type $\mathsf{X}_{\mathcal{F}}$ where $\mathcal{F}$ is a set of forbidden blocks of height 2.

A vertex shift is the set of computation of the unlabeled automaton $\mathcal{B} = (V, \Gamma)$ with transitions $(q_0, .., q_{d-1}) \to q$.

Example 3. The tree shift X of Example 1 is a vertex shift accepted by the automaton $\mathcal{A} = (V, \Delta)$ with transitions $(0, 0) \to 0$, $(1, 1) \to 0$, $(1, 0) \to 1$ and $(0, 1) \to 1$. These transitions are given in the following table t where $(p, q) \to t[p, q]$ is a transition.

	0	1
0	0	1
1	1	0

Proposition 2. *Any shift of finite type is conjugate to a vertex shift.*

Proof. Let $X = \mathsf{X}_{\mathcal{F}}$ be a tree shift of finite type defined by a finite set of forbidden blocks of height m for some nonnegative integer m. Let $\mathcal{A} = (V, A, \delta)$ be the deterministic m-local automaton such that $V = \mathcal{L}_m(X)$ and, for $p_i \in V, a \in A$, the block $q = (a, p_0, .., p_{d-1})$ of height $m + 1$ is an allowed block of X, then $\delta(p_0, .., p_{d-1}), a) = \text{trunc}(q)$, where $\text{trunc}(q)$ is the block of height m such that $\text{trunc}(q)_x = q_x$ for $x \in \Sigma^{\leq m}$. The automaton $\mathcal{A}$ accepts X. Let Y be the vertex shift made of the all computations on $\mathcal{A}$. Note that any tree t of X has a unique computation C in $\mathcal{A}$.

We define an $m+1$-block map Φ from X to Y via $\phi : \mathcal{L}_m(X) \to V$ by setting $\phi(p) = p$. The map Φ associate to each tree of X its computation in $\mathcal{A}$. The one-block map Ψ from Y to X given by $\psi : Y \to A$ with $\psi(p) = p_\varepsilon$ is the inverse of ϕ. The shifts X and Y are thus conjugate.

In the sequel, in order to simplify the notations, we restrict us to binary trees ($\Sigma = \{0, 1\}$) but all results extend trivially to the case of trees with d children for any $d \geq 1$.

3 A decomposition theorem for trees

The decomposition theorem for subshifts of infinite words states that any conjugacy between shifts of finite type can be decomposed into a finite sequence of splittings and amalgamations. We will prove an analogous theorem for infinite trees. The crucial lemma will show that the memory of a block map can be reduced using a notion of (input) splittings on tree automata defined below. We first consider the case of tree vertex shifts.

Let X be a binary tree vertex shift defined by a deterministic 2-local automaton $\mathcal{A} = (V, \Delta)$. Set $V = \{p_1, .., p_n\}$. We define an *in-splitting* of $\mathcal{A}$ as an automaton $\tilde{\mathcal{A}} = (\tilde{V}, \tilde{\Delta})$ obtained as follows. First, we in-split the automaton $\mathcal{A}$ by refining the natural partition of Δ; for each vertex $p \in V$, we partition the set Δ_p of transitions coming in p into subsets $\Delta_p^1, .., \Delta_p^{l(p)}$. We set

$$\tilde{V} = \{p_1^1, .., p_1^{l(1)}, p_2^1, .., p_2^{l(2)}, .., p_n^1, .., p_n^{l(n)}\} \tag{1}$$

and $(p^i, q^j) \to r^k \in \tilde{\Delta}$ if $(p, q) \to r \in \Delta_r^k$.

The shift accepted by $\tilde{\mathcal{A}}$ is a vertex shift denoted by $\tilde{X}$ and called the *in-splitting* of X defined by the above partitioning of the transitions.

The same notion of in-splitting is defined for tree shifts of finite type as follows. Let X be a shift of finite type accepted by a deterministic automaton $\mathcal{A} = (V, A, \Delta)$. The in-splitting of $\mathcal{A}$ is $\tilde{\mathcal{A}} = (\tilde{V}, \tilde{\Delta})$. For each vertex $p \in V$, we partition the set Δ_p of transitions coming in p into subsets $\Delta_p^1, .., \Delta_p^{l(p)}$ and set $\tilde{V}$ as in Equation 1. We set $(p^i, q^j), a \to r^k \in \tilde{\Delta}$ if $(p, q), a \to r \in \Delta_r^k$.

A tree *in-amalgamation* of an tree automaton $\mathcal{A}$ is an automaton $\mathcal{B}$ such that $\mathcal{B}$ is an in-splitting of $\mathcal{A}$. An *in-amalgamation* of an tree shift X is a tree shift Y such that Y is an in-splitting of X.

Lemma 1. *Let $\tilde{X}$ be an in-splitting of a tree vertex shift X. Then $\tilde{X}$ and X are conjugate.*

Proof. Let $\tilde{X}$ be an in-splitting of X accepted by $\mathcal{A} = (V, \Delta)$. Define a 1-block map $\Phi : \tilde{X} \to X$ via $\phi(p^i) = p$ for each state $p \in V$, and a 2-block map $\Psi : X \to \tilde{X}$ via $\psi(r, p, q) = r^i$, where $(p, q) \to r \in \Delta_r^i$.

It is not difficult to check that $\Phi(\tilde{X}) \subseteq X$ and $\Psi(X) \subseteq \tilde{X}$. It is clear that $\Phi(\Psi(t))_u = t_u$ for all trees $t \in X$ and each word $u \in \Sigma^*$ since adding and removing superscripts has no effect. Thus we only need to check that $\Psi(\Phi(t))_u =$

t_u for all trees $t \in \tilde{X}$ and each word $u \in \Sigma^*$. Let $t \in \tilde{X}$ and $u \in \Sigma^* = \{0,1\}^*$. We need to show that

$$\Psi(\Phi(t))_u = \psi((\phi(t_u), \phi(t_{u0}), \phi(t_{u1})) = t_u.$$

Indeed, suppose $t_{u0} = p^i$, $t_{uj} = q^j$ and $t_u = r^k$ where $p^i, q^j \in \tilde{V}$. Hence we have $(p,q) \to r \in \Delta_r^k$. We get $\psi((\phi(t_u), \phi(t_{u0}), \phi(t_{u1})) = \psi(r,p,q) = r^k = t_u$.

When the partition of the set of transitions consists of singleton sets, then the in-s splitting $\tilde{X}$ of X is called the *complete in-splitting* of X.

In the remainder of this section, we give a proof of the decomposition theorem of tree vertex shifts. In Lemma 3 we show that a higher block presentation of the tree shift is the composition of a finite sequence of in-splittings and in-amalgamations. Then, by moving to a higher block presentation if necessary, we may assume that the conjugacy Φ between two tree shifts is a 1-block map with an n-block map inverse. If $n = 1$, then this conjugacy is just a relabeling of the symbols of the states and as such a trivial splitting. So we would like a way to reduce the memory of the inverse of Φ. We will reduce the memory in Lemmas 2 by using tree in-splittings and in-amalgamations.

Lemma 2. *Let X_k for $k = 1$ and 2, be two tree vertex shift defined by the two automata $\mathcal{A}_1 = (V_1, \Delta)$ and $\mathcal{A}_2 = (V_2, \Lambda)$ respectively. Suppose $\Phi : X_1 \to X_2$ is a 1-block conjugacy with an n-block inverse. If $n \geq 1$, then there are tree vertex shifts $\tilde{X}_k$ such that the following diagram commutes:*

$$\begin{tikzcd} X_1 \arrow[r, "\Phi"] \arrow[d, "\Psi_1"'] & X_2 \arrow[d, "\Psi_2"] \\ \tilde{X}_1 \arrow[r, "\tilde{\Phi}"] & \tilde{X}_2 \end{tikzcd}$$

where Ψ_1 and Ψ_2 are in-splittings of X_1 and X_2 respectively, and where $\tilde{\Phi}$ is a 1-block conjugacy with an $(n-1)$-block inverse.

Proof. For $p \in V_1$, we partition Δ_p into $\bigcup_{s,t} \in V_2 \times V_2 = \{(q,r) \to p \mid \phi(q) = s$ and $\phi(r) = t\}$. Then we set

$$\tilde{V_1} = \{p^{s,t} : p \in V_1, s,t \in V_2 \text{ with } \phi(q) = s \text{ and } \phi(r) = t \text{ for some } (q,r) \to p \in \Delta_p\},$$

We set $(p^{s,t}, q^{u,v}) \to r^{p,q} \in \tilde{\Delta}$ if $(p,q) \to r \in \tilde{\Delta}$.

As shown in Lemma 1, $\Psi_1 : X_1 \to \tilde{X}_1$ is a conjugacy via $\Psi_1(t)_u = \psi_1(t_u, t_{u0}, t_{u1}) = t_u^{\phi(t_{u0}),\phi(t_{u1})}$, where $u \in \{0,1\}^*$.

Now let $\tilde{X}_2$ be the complete in-splitting of X_2. So

$$\tilde{V_2} = \{r^{p,q} : p,q,r \in V_2, \text{ with } (p,q) \to r \in \Lambda_r\},$$

We set $(r^{p,q}, s^{u,v}) \to t^{r,s} \in \tilde{\Lambda}$ if $(r,s) \to t \in \Lambda$. As shown in Lemma 1, $\Psi_2 : X_2 \to \tilde{X}_2$ is a conjugacy via $\Psi_2(t)_u = \psi_2(t_u, t_{u0}, t_{u1}) = t_u^{t_{u0},t_{u1}}$, where $u \in \{0,1\}^*$.

Now we define the 1-block map $\tilde{\Phi} : \tilde{X}_1 \to \tilde{X}_2$ via

$$\tilde{\Phi}(t)_u = \phi(t_u)^{\phi(t_{u0}),\phi(t_{u1})},$$

where $u \in \{0,1\}^*$. Clearly, the diagram commutes and thus Φ is one-to-one and onto. It remains to check that $\tilde{\Phi}^{-1} = \Psi_1 \circ \Phi^{-1} \circ \Psi_2$ is an $(n-1)$-block map. That is, we must show that for any tree $t \in \tilde{X}_2$, the coordinates in a block of height $n-1$ of t rooted at the node ε determines $\tilde{\Phi}^{-1}(t)_\varepsilon$. But this follows from the observation that the block of height $n-1$ of t rooted at ε determines all $\Psi_2^{-1}(t)_{v0}$ and all $\Psi_2^{-1}(t)_{v1}$ for $v \in \{0,1\}^{n-1}$, and therefore the block of height n at the root of t.

The proof of the above lemma is similar to the proof of the analogous result for subshifts of $\Sigma^{\mathbb{Z}}$ (see [8, Lemma 7.3.1]) or subshifts of $\Sigma^{\mathbb{Z}^2}$ (see [4]).

In general a conjugacy between tree vertex shifts is an n-block map but the following lemma shows that moving to a higher block presentation we may assume it is a 1-block map.

Lemma 3. *Let X_k, for $k = 1$ and 2, be tree vertex shifts. Let n be a positive integer. Suppose $\Phi : X_1 \to X_2$ is an n-block conjugacy and let $\hat{X}_1$ be the higher block presentation of X_1 of order n. There exists a map $\eta : X_1 \to \hat{X}_1$ which is a sequence of tree in-splittings, such that $\Phi \circ \eta^{-1}$ is a 1-block conjugacy.*

Proof. Clearly, $\Phi \circ \eta^{-1}$ is a 1-block conjugacy. We need to show that η is a sequence of tree in-splitting.

If X is the vertex shift defined by $\mathcal{A} = (V, \Delta)$ and n is a nonnegative integer, a higher block presentation of X of order n is the vertex shift $\hat{X}$ defined by $\hat{\mathcal{A}} = (\hat{V}, \hat{\Delta})$, where $\hat{V}$ is the set of allowed blocks of X of height n. There is a transition $(p,q) \to r$ in $\hat{\Delta}$, where $p, q, r \in \mathcal{L}_n(X)$ if and only if $r_{0u} = p_u$ and $r_{1u} = q_u$ for any $u \in \{0,1\}^{n-1}$.

A complete in-splitting of the tree shift X yields a higher block presentation of X of order 2 by $\Psi(t)_u = t_u^{t_{u0},t_{u1}}$.

By iterating this construction on $\Psi(X)$, we can find a sequence of in-splittings η such that $\bar{\eta} = \eta \circ \Psi_1^{-1}$ is a 1-block conjugacy, or simply a relabeling. Then $\eta = \bar{\eta} \circ \Psi_1$ and we have the result.

Lemma 4. *Let X be tree subshift of finite type, there is a vertex shift Y and a conjugacy from X to Y which is a sequence of in-splittings.*

Proof. Let $X = \mathsf{X}_{\mathcal{F}}$ be a tree shift of finite type defined by a finite set of forbidden blocks of height m for some nonnegative integer m.

Let $\hat{X}$ be the higher block presentation of X of order m. The shift $\hat{X}$ is the vertex shift defined by $\hat{\mathcal{A}} = (\hat{V}, \hat{\Delta})$, where $\hat{V}$ is the set of allowed blocks of X of height m. There is a transition $(p,q) \to r$ in $\hat{\Delta}$, where $p, q, r \in \mathcal{L}_m(X)$ if and only if $r_{0u} = p_u$ and $r_{1u} = q_u$ for any $u \in \{0,1\}^{m-1}$.

We know from Lemma 3 that $\hat{X}$ is obtained from X with a sequence of in-splittings.

We are now ready to state the main result of this section.

Theorem 1. *Let X_1 and X_2 be two tree shifts of finite type. Every conjugacy between X_1 and X_2 is the composition of a finite sequence of tree in-splittings and tree in-amalgamations.*

Proof. By Lemma 4, we can view any tree shift of finite type as a vertex shift. The theorem follows then from Lemma 1 and Lemma 3.

4 Commutation of in-amalgamations

Proposition 3. *Suppose X_1 is a tree vertex shift and X_2, X_3 are tree vertex shifts obtained from X_1 by in-amalgamations. Then there is a tree vertex shift X_4 that can be obtained from both X_3 and X_4 by in-amalgamations.*

$$\begin{tikzcd}
& X_1 \arrow[dl, "\Phi"'] \arrow[dr, "\Psi"] & \\
X_2 \arrow[dr, "\Omega"'] & & X_3 \arrow[dl, "\Theta"] \\
& X_4 &
\end{tikzcd}$$

Fig. 3. The commutation of in-amalgamations. If X_2, X_3 are vertex shifts which are in-amalgamations of X_1, the there is a vertex shift X_4 which is a common amalgamation of X_2, X_3.

In Figure 3, the maps Φ and Ψ are in-amalgamations. As a consequence of Proposition 3 the maps Ω and Θ are also in-amalgamations.

Proof. Let us first assume that there is an in-amalgamation $\Phi : X_1 \to X_2$ and a in-amalgamation $\Psi : X_1 \to X_3$. Let us assume that states $p^1, .., p^{l(p)}$ of V_1 are amalgamated to a state p of V_2.

By definition of an in-amalgamation, this implies that if $(q, r) \to p^i \in \Delta_1$, then $(q, r) \to p^j \notin \Delta_1$ for any states $q, r \in V_1$ and any $1 \leq i \neq j \leq l(p)$. This implies also that $(p^i, q) \to r \in \Delta_1$ if an only if $(p^j, q) \to r \in \Delta_1$, and $(q, p^i) \to r \in \Delta_1$ if and only if $(q, p^j) \to r \in \Delta_1$ for any states $q, r \in V_1$ and any $1 \leq i, j \leq l(p)$.

Suppose also that states $q^1, .., q^{l(q)}$ of V_1 are amalgamated to a state q of V_3.

Let us first assume that the states $p^1, .., p^{l(p)}$ and $q^1, .., q^{l(q)}$ are all distinct. We define X_4 as the in-amalgamation of X_2 obtained by amalgamating the states $p, q^1, .., q^{l(q)}$ to a state q. It is also the in-amalgamation of X_3 obtained by amalgamating the states $q, p^1, .., p^{l(p)}$ to a state q.

Let us now assume that $p^1 = q^1, .., p^l = q^l$ for some integer $1 \leq l \leq \min(l(p), l(q))$. This implies that, for any $1 \leq i \leq l(p)$, $1 \leq j \leq l(q)$, one has $(p^i, q) \to r \in \Delta_n$ if and only if $(p^j, q) \to r \in \Delta_n$, and $(q, p^i) \to r \in \Delta_n$ if and only if $(q, p^j) \to r \in \Delta_n$ for $n = 1$ and $n = 2$.

We define X_4 as the in-amalgamation of X_2 obtained by amalgamating the states $p, q^{l+1}, .., q^{l(q)}$ to the state p. It is also the in-amalgamation of X_3 obtained by amalgamating the states $q, p^{l+1}, .., p^{l(p)}$ to a state p. Hence, if $\varPhi$ and $\varPsi$ are in-amalgamations, then $\varOmega$ and $\varTheta$ also.

The previous theorem allows us to define the notion of *minimal in-amalgamation* of an edge shift X. It is defined as the vertex shift defined by an automaton $\mathcal{A} = (V, \varDelta)$ with the smallest number of vertices which is obtained by in-amalgamations of X.

Corollary 1. *Any vertex shift has a unique minimal in-amalgamation.*

Proof. Let us assume that X has two minimal amalgamations X_2 and X_3. By Proposition 3, X_2 and X_3 have a common in-amalgamation Y. By minimality, $Y = X_2 = X_3$.

Theorem 2. *Let X_1 and X_2 be two tree subshifts of finite type. It is decidable whether X_1 and X_2 are conjugate.*

Proof. By Proposition 2, one may assume that X_1 and X_2 are vertex shifts. By Theorem 1, there is a sequence of tree in-splittings and tree in-amalgamations from X_1 to X_2.

Let us consider first that this sequence is decomposed into a sequence of tree in-splittings from X_1 to X followed (up to a relabeling of X), by a sequence of tree in-amalgamations from X to X_2. This case is illustrated in Figure 4. By Proposition 3, there are vertex shift at the confluence of any two dashed edges of Figure 4. As a consequence, X_1 and X_2 have a common amalgamation and thus the same minimal amalgamation. Conversely, if X_1 and X_2 have the same minimal amalgamation, there is a sequence of tree in-splittings and tree in-amalgamations from X_1 to X_2.

We now consider the case where there is a sequence of tree in-splittings and tree in-amalgamations from X_1 to X_2. This sequence is decomposed into a sequence of the form described in the previous case and the same result holds by transitivity.

Example 4. Let X_1 and X_2 be two tree vertex shifts over the alphabet $V = \{a, b, c\}$. The shift X_1 is accepted by $\mathcal{A}_1 = (V, \varDelta_1)$ and the shift X_2 is accepted by $\mathcal{A}_2 = (V, \varDelta_2)$ where $\varDelta_1$ and $\varDelta_2$ are given in the two following tables.

$$\varDelta_1 = \begin{array}{c|c|c|c|} & a & b & c \\ \cline{2-4} a & a & c & c \\ \cline{2-4} b & b & a & a \\ \cline{2-4} c & b & a & a \\ \cline{2-4} \end{array} \qquad \varDelta_2 = \begin{array}{c|c|c|c|} & a & b & c \\ \cline{2-4} a & c & a & a \\ \cline{2-4} b & a & b & b \\ \cline{2-4} c & a & b & b \\ \cline{2-4} \end{array} \qquad \varDelta_3 = \begin{array}{c|c|c|} & a & b \\ \cline{2-3} a & a & b \\ \cline{2-3} b & b & a \\ \cline{2-3} \end{array} \qquad \varDelta_4 = \begin{array}{c|c|c|} & a & b \\ \cline{2-3} a & b & a \\ \cline{2-3} b & a & b \\ \cline{2-3} \end{array}$$

Since the second and third row of $\varDelta_1$ and the second and third column of $\varDelta_1$ are equal, the vertices b and c can be amalgamated. There is an in-amalgamation from $\mathcal{A}_1$ to $\mathcal{A}_3 = (V_3, \varDelta_3)$ where $V_3 = \{a, b\}$ and $\varDelta_3$ is given by the following tables.

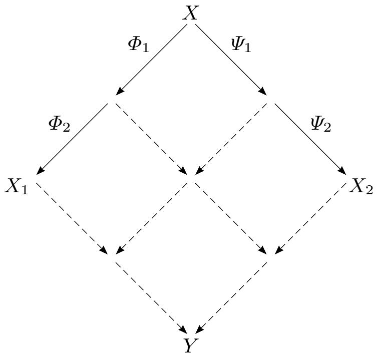

Fig. 4. A sequence of tree in-splittings from X_1 to X is followed (up to a relabeling of X), by a sequence of tree in-amalgamations from X to X_2. Any edge represents an in-amalgamation. The shifts X_1 and X_2 have the same minimal amalgamation Y.

No more in-amalgamation is possible from $\mathcal{A}_3$ and thus $\mathcal{A}_3$ is minimal. Similarly, the second and third row of Δ_2 and the second and third column of Δ_2 are equal, the vertices b and c can be amalgamated. There is an in-amalgamation from $\mathcal{A}_2$ to $\mathcal{A}_4 = (V_4, \Delta_4)$ where $V_4 = \{a, b\}$ and Δ_4 is given by the following tables.

Finally, relabeling the states of $\mathcal{A}_4$ by exchanging a and b gives $\mathcal{A}_3$. Hence, X_1 and X_2 have the same minimal amalgamation and are conjugate.

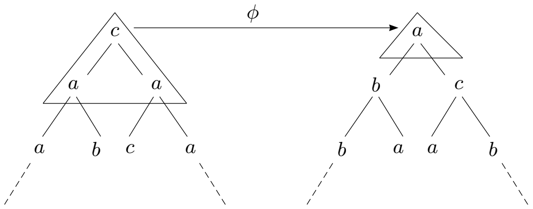

Fig. 5. A 2-block map $\Phi : X \to Y$, where X is the tree shift of Figure 1 and Y a shift of finite type over the alphabet $\{a, b, c\}$.

The 2-block map $\Phi : X_2 \to X_1$ of Figure 5 is a conjugacy. It is defined by $\phi(a, a, b) = b$, $\phi(a, b, a) = c$, $\phi(a, a, c) = b$, $\phi(a, c, a) = c$, $\phi(b, b, b) = a$, $\phi(b, b, c) = a$, $\phi(b, c, b) = a$, $\phi(b, c, c) = a$, $\phi(c, a, a) = a$.

5 Conclusion

We have shown that it decidable whether two tree shifts of finite type are conjugate.

Further work will include the case of sofic trees. We conjecture that the results that we have obtained for tree shifts of finite type can be extended to sofic tree shifts using techniques similar to the one used for shifts of sequences (see [9], [6], [7]). The decomposition theorem that we have proved for tree shifts of finite type will also allow us to define a notion of strong shift equivalence between tree shifts and to deduce that two tree shifts of finite type are equivalent if and only if their transition matrices are related by a sequence of simple algebraic matrix conditions.

References

1. H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata, 2007. release October, 12th 2007.
2. E. Coven, A. Johnson, N. Jonoska, and K. Madden. The symbolic dynamics of multidimensional tiling systems. *Ergodic Theory and Dynamical Systems*, 23(02):447–460, 2003.
3. G. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Theory of Computing Systems*, 3(4):320–375, 1969.
4. A. Johnson and K. Madden. The decomposition theorem for two-dimensional shifts of finite type. *Proceedings-American Mathematical Society*, 127:1533–1544, 1999.
5. B. P. Kitchens. *Symbolic dynamics.* Universitext. Springer-Verlag, Berlin, 1998. One-sided, two-sided and countable state Markov shifts.
6. W. Krieger. On sofic systems. I. *Israel J. Math.*, 48(4):305–330, 1984.
7. W. Krieger. On sofic systems. II. *Israel J. Math.*, 60(2):167–176, 1987.
8. D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, 1995.
9. M. Nasu. Topological conjugacy for sofic systems and extensions of automorphisms of finite subsystems of topological markov shifts. In *Proceedings of Maryland special year in Dynamics 1986–87*, volume 1342 of *Lecture Notes in Mathematics*, pages 564–607. Springer verlag, 1988.
10. M. Nasu. *Textile Systems for Endomorphisms and Automorphisms of the Shift.* American Mathematical Society, 1995.
11. M. Nivat and A. Podelski, editors. *Tree automata and languages*, volume 10 of *Studies in Computer Science and Artificial Intelligence.* North-Holland Publishing Co., Amsterdam, 1992. Papers from the workshop held in Le Touquet, June 1990.
12. D. Perrin and J. Pin. *Infinite words.* Elsevier Boston, 2004.
13. W. Thomas. Automata on infinite objects. In *Handbook of theoretical computer science, Vol. B*, pages 133–191. Elsevier, Amsterdam, 1990.
14. R. F. Williams. Classification of subshifts of finite type. In *Recent advances in topological dynamics (Proc. Conf. Topological Dynamics, Yale Univ., New Haven, Conn., 1972; in honor of Gustav Arnold Hedlund)*, pages 281–285. Lecture Notes in Math., Vol. 318. Springer, Berlin, 1973.

Proof of Proposition 1

Proposition 4. *Any tree shift of finite type is accepted by a deterministic local tree automaton. Reciprocally any tree shift accepted by a deterministic local tree automaton is of finite type.*

Proof.

- Let $X = \mathsf{X}_{\mathcal{F}}$ be a tree shift of finite type defined by a finite set of forbidden blocks. Without loss of generality, we can assume that $\mathcal{F}$ is the set of all not allowed blocks of height m for some nonnegative integer m.
  We define a deterministic tree automaton $\mathcal{A} = (V, A, \delta)$ such that $V = \mathcal{L}_m(X)$. For $p_i \in V, a \in A$, if the block $q = (a, p_0, .., p_{d-1})$ of height $m+1$ is an allowed block of X, then $\delta((p_0, .., p_{d-1}), a) = \mathrm{trunc}(q)$, where $\mathrm{trunc}(q)$ is the block of height m such that $\mathrm{trunc}(q)_x = q_x$ for $x \in \Sigma^{\leq m}$. The partial function δ is undefined otherwise.
  The automaton $\mathcal{A}$ is deterministic and m-local. Let Y be the tree subshift accepted by $\mathcal{A}$. It is clear that $Y \subseteq \mathsf{X}_{\mathcal{F}}$. Conversely, let $t \in \mathsf{X}_{\mathcal{F}}$. We define a computation C of t as follows. For any $x \in \Sigma^*$, set C_x be the block of t of size m rooted at the node x. Hence $\mathsf{X}_{\mathcal{F}} \subseteq Y$ which proves the first part of the claim.
- Let X be a tree shift and $\mathcal{A} = (V, A, \delta)$ a m-definite tree automaton accepting X. We define $\mathcal{F}$ as the set of blocks of height $\leq m+1$ such that any infinite tree containing such a block is not accepted by $\mathcal{A}$. Then one immediately has $X \subseteq \mathsf{X}_{\mathcal{F}}$. Suppose $t \in \mathsf{X}_{\mathcal{F}}$, we construct a computation C of $\mathcal{A}$ on t on the following way. For any $x \in \Sigma^*$ we denote by p_x the block $t_{x\Sigma^{\leq m+1}}$. Then there exist a tree t' with a computation C' of $\mathcal{A}$ and a word $y \in \Sigma^*$ such that $t'_{y\Sigma^{\leq m+1}} = p_x$. We set $C_x = C'_y$ and since $\mathcal{A}$ is a m-definite automaton, C is a computation of $\mathcal{A}$ on t. Finally $X = \mathsf{X}_{\mathcal{F}}$ where $\mathcal{F}$ is a finite set of blocks which proves the claim.