Acta Cybernetica **23** (2017) 269–281.

Synchronous Forest Substitution Grammars*

Andreas Maletti[a]

— dedicated to the memory of Zoltán Ésik (1951–2016) —

Abstract

The expressive power of synchronous forest (tree-sequence) substitution grammars (SFSGs) is studied in relation to multi bottom-up tree transducers (MBOTs). It is proved that SFSGs have exactly the same expressive power as compositions of an inverse MBOT with an MBOT. This result is used to derive complexity results for SFSGs and the fact that compositions of an MBOT with an inverse MBOT can compute tree translations that cannot be computed by any SFSG, although the class of tree translations computable by MBOTs is closed under composition.

Keywords: tree transducer, synchronous grammar, regular tree language, machine translation

1 Introduction

Synchronous forest substitution grammars (SFSGs) [19] or the rational binary tree relations [17] computed by them received renewed interest recently due to their applications in Chinese-to-English machine translation [21, 22]. The fact that [19] and [17] arrived independently and with completely different backgrounds at the same model shows that SFSGs are a natural, practically relevant, and theoretically interesting model for tree translations. Roughly speaking, SFSGs are a synchronous grammar formalism [2] that utilizes only first-order substitution as in a regular tree grammar [7, 8], but allows several components that develop simultaneously for both the input and the output side. This feature allows them to model linguistic discontinuity on both the source and target language. The rational binary tree relations or equivalently the tree translations computed by SFSGs can also be characterized by rational expressions [17] and automata [16].

Multi bottom-up tree transducers (MBOTs) [1, 4] are restricted SFSGs, in which only the output side is allowed to have several components. They were rediscovered

*This article is an extended and revised version of [Maletti: *Synchronous forest substitution grammars.* In Proc. Algebraic Informatics, LNCS 8080, pages 235–246, 2013]

[a]Universität Leipzig, Institute of Computer Science, PO box 100 920, 04009 Leipzig, Germany, E-mail: maletti@informatik.uni-leipzig.de

DOI: 10.14232/actacyb.23.1.2017.15

in [5, 6], but were studied extensively by [3, 11, 1] already in the 70s and 80s. Their properties [13] are desirable in statistical syntax-based machine translation [10]. This led to a closer inspection [4, 15, 9] of their properties in recent years. Overall, their expressive power is rather well-understood by now.

In this contribution, we investigate the expressive power of SFSGs in terms of MBOTs. We show that the expressive power of SFSGs coincides exactly with that of compositions of an inverse MBOT followed by an MBOT. This characterization is natural in terms of bimorphisms and shows that the input and the output tree are independently obtained by a full MBOT from an intermediate tree language, which is always regular [7, 8]. This paves the way to complementary results. In particular, we derive the first complexity results for SFSGs and we demonstrate that the composition in the other order (first an MBOT followed by an inverse MBOT) contains tree translations that cannot be computed by any SFSG. This shows a limitation of MBOTs, which are closed under composition [4]. Overall, we can thus also characterize the expressive power of SFSGs by an arbitrary chain of inverse MBOTs followed by an arbitrary chain of MBOTs.

2 Preliminaries

We use $\mathbb{N}$ for the set of nonnegative integers, and $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$ for the set of positive integers. For all $k \in \mathbb{N}$, the set $\{i \in \mathbb{N}_+ \mid i \leq k\}$ is abbreviated to $[k]$. In particular, $[0] = \emptyset$. For all relations $R \subseteq A \times B$ and subsets $A' \subseteq A$, we let $R(A') = \{b \in B \mid \exists a \in A' \colon (a, b) \in R\}$. Moreover,

$$R^{-1} = \{(b, a) \mid (a, b) \in R\} \qquad \mathrm{dom}(R) = R^{-1}(B) \qquad \mathrm{ran}(R) = \mathrm{dom}(R^{-1}) \ ,$$

which are called the *inverse* of R, the *domain* of R, and the *range* of R, respectively. Given relations $R \subseteq A \times B$ and $S \subseteq B \times C$, the *composition* $R\,;S \subseteq A \times C$ of R followed by S is $R\,;S = \{(a, c) \in A \times C \mid \exists b \in B \colon (a, b) \in R, (b, c) \in S\}$. These notions and notations are lifted to sets and classes of relations as usual. For every $k \in \mathbb{N}$, we also write $A^k = A \times \cdots \times A$ containing the factor A exactly k times.

Given a set Σ, the set $\Sigma^* = \bigcup_{k \in \mathbb{N}} \Sigma^k$ is the set of all words over Σ, which includes the empty word $\varepsilon \in \Sigma^0$. The concatenation of two words $u, w \in \Sigma^*$ is denoted by $u.w$ or just uw. The length $|w|$ of a word $w \in \Sigma^*$ is the unique $k \in \mathbb{N}$ such that $w \in \Sigma^k$. We simply write w_i for the i-th letter of w, so $w_i = \sigma_i$ for all $i \in [k]$ provided that $w = \sigma_1 \cdots \sigma_k$ with letters $\sigma_i \in \Sigma$ for all $i \in [k]$. Given a set A, the set $T_\Sigma(A)$ of all Σ-*trees indexed by* A is the smallest set T such that $A \subseteq T$ and $\sigma\vec{u} \in T$ for all $\sigma \in \Sigma$ and $\vec{u} \in T^*$. Such a sequence $\vec{u}$ of trees is also called *forest*. Consequently, a tree t is either an element of A, or it consists of a root node labeled σ followed by a forest $\vec{u}$ of $|\vec{u}|$ children. To improve the readability, we often write a forest '$t_1 \cdots t_k$' as '$t_1, \ldots, t_k$', where $t_1, \ldots, t_k \in T_\Sigma(A)$. In addition, we identify the tree t with the forest (t). The *positions* $\mathrm{pos}(t) \subseteq \mathbb{N}_+^*$ of

a tree $t \in T_\Sigma(A)$ are inductively defined by

$$\mathrm{pos}(a) = \{\varepsilon\} \qquad \text{and} \qquad \mathrm{pos}(\sigma\vec{u}) = \{\varepsilon\} \cup \bigcup_{i=1}^{|\vec{u}|} \{i.p \mid p \in \mathrm{pos}(\vec{u}_i)\}$$

for every $a \in A$, $\sigma \in \Sigma$, and $\vec{u} \in T_\Sigma(A)^*$. For each forest $\vec{u} \in T_\Sigma(A)^*$, we let $\mathrm{pos}(\vec{u}) = \bigcup_{i=1}^{|\vec{u}|} \{\#^{i-1}.p \mid p \in \mathrm{pos}(\vec{u}_i)\}$. Positions are totally ordered via the (standard) lexicographic ordering $\preceq$ on $\mathbb{N}_+^*$, which can be extended to $(\mathbb{N}_+ \cup \{\#\})^*$ with the convention that the additional letter $\#$ is larger than all numbers; i.e., $n \prec \#$ for every $n \in \mathbb{N}_+$. Let $t, t' \in T_\Sigma(A)$ and $p \in \mathrm{pos}(t)$. The label of t at position p is $t(p)$, the subtree rooted at position p is $t|_p$, and the tree obtained by replacing the subtree at position p by t' is denoted by $t[t']_p$. Formally, they are defined by $a(\varepsilon) = a|_\varepsilon = a$ and $a[t']_\varepsilon = t'$ for every $a \in A$ and

$$t(p) = \begin{cases} \sigma & \text{if } p = \varepsilon \\ \vec{u}_i(p') & \text{if } p = i.p' \text{ with } i \in \mathbb{N}_+ \end{cases} \qquad t|_p = \begin{cases} t & \text{if } p = \varepsilon \\ \vec{u}_i|_{p'} & \text{if } p = i.p' \text{ with } i \in \mathbb{N}_+ \end{cases}$$

$$t[t']_p = \begin{cases} t' & \text{if } p = \varepsilon \\ \sigma(\vec{u}_1, \ldots, \vec{u}_{i-1}, \vec{u}_i[t']_{p'}, \vec{u}_{i+1}, \ldots, \vec{u}_{|\vec{u}|}) & \text{if } p = i.p' \text{ with } i \in \mathbb{N}_+ \end{cases}$$

for all $t = \sigma\vec{u}$ with $\sigma \in \Sigma$ and $\vec{u} \in T_\Sigma(A)^*$. We immediately also extend this notion to forests $\vec{u} \in T_\Sigma(A)^*$ for all $\#^i p \in \mathrm{pos}(\vec{u})$ with $i \in \mathbb{N}$ and $p \in \mathrm{pos}(\vec{u}_{i+1})$ by $\vec{u}(\#^i p) = \vec{u}_{i+1}(p)$ and $\vec{u}|_{\#^i p} = \vec{u}_{i+1}|_p$ and

$$\vec{u}[t']_{\#^i p} = (\vec{u}_1, \ldots, \vec{u}_{i-1}, \vec{u}_i[t']_p, \vec{u}_{i+1}, \ldots, \vec{u}_{|\vec{u}|}) \ .$$

In the following, let $\vec{u} \in T_\Sigma(A)^*$ be a forest. By our identification of trees $T_\Sigma(A)$ with forests $T_\Sigma(A)^1$ of length 1, this choice includes trees. A position $p \in \mathrm{pos}(\vec{u})$ is a *leaf* in $\vec{u}$ if $p.1 \notin \mathrm{pos}(\vec{u})$. For every selection $S \subseteq A \cup \Sigma$ of labels, we let $\mathrm{pos}_S(\vec{u}) = \{p \in \mathrm{pos}(\vec{u}) \mid \vec{u}(p) \in S\}$ and $\mathrm{pos}_s(\vec{u}) = \mathrm{pos}_{\{s\}}(\vec{u})$ for every $s \in A \cup \Sigma$. The forest $\vec{u} \in T_\Sigma(A)$ is *linear in* $S \subseteq A$ if $|\mathrm{pos}_s(\vec{u})| \leq 1$ for every $s \in S$. The *variables* of $\vec{u}$ are $\mathrm{var}(\vec{u}) = \{a \in A \mid \mathrm{pos}_a(\vec{u}) \neq \emptyset\}$. Given a selection $S \subseteq A$ and a mapping $\theta\colon S \to T_\Sigma(A)^*$ such that $|\theta(s)| = |\mathrm{pos}_s(\vec{u})|$ for all $s \in S$, also called *suitable substitution for S in $\vec{u}$*, the forest $\vec{u}\theta$ is obtained from $\vec{u}$ by replacing for every $s \in S$ the leaves $\mathrm{pos}_s(\vec{u})$ in lexicographic order by the trees $\theta(s)$. Formally, for every $s \in S$, let $\mathrm{pos}_s(\vec{u}) = \{p_{s1}, \ldots, p_{sk_s}\}$ with $p_{s1} \prec \cdots \prec p_{sk_s}$. Then

$$\vec{u}\theta = \vec{u}[\theta(s_1)_1]_{p_{s_1 1}} \cdots [\theta(s_1)_{|\theta(s_1)|}]_{p_{s_1 k_{s_1}}} \cdots [\theta(s_\ell)_1]_{p_{s_\ell 1}} \cdots [\theta(s_\ell)_{|\theta(s_\ell)|}]_{p_{s_\ell k_{s_\ell}}} \ ,$$

where $S = \{s_1, \ldots, s_\ell\}$.

Given two sets Σ and Δ with $\square \notin \Delta$, a mapping $d\colon \Sigma \to (\Delta \cup \{\square\})$ is a *delabeling*. Thus, a delabeling is similar to a relabeling [7, 8], but it can also map symbols to a special symbol $\square$, which will yield that those symbols are deleted, when they occur with exactly one child and project on the delabeling of the child. The delabeling

induces a mapping $\tau_d \colon T_\Sigma(A) \to T_{\Sigma\cup\Delta}(A)$ such that $\tau_d(a) = a$ for all $a \in A$ and

$$\tau_d(\sigma\vec{u}) = \begin{cases} \tau_d(\vec{u}_1) & \text{if } d(\sigma) = \square \text{ and } |\vec{u}| = 1 \\ \sigma(\tau_d(\vec{u}_1), \dots, \tau_d(\vec{u}_{|\vec{u}|})) & \text{if } d(\sigma) = \square \text{ and } |\vec{u}| \neq 1 \\ d(\sigma)(\tau_d(\vec{u}_1), \dots, \tau_d(\vec{u}_{|\vec{u}|})) & \text{otherwise} \end{cases}$$

for all $\sigma \in \Sigma$ and $\vec{u} \in T_\Sigma(A)^*$.

Finally, let us recall the regular tree languages [7, 8]. A *(finite-state) tree automaton* (TA) is a tuple $G = (Q, \Sigma, I, R)$ such that Q is a finite set of *states*, Σ is an alphabet of symbols, $I \subseteq Q$ is a set of *initial states*, and $R \subseteq Q \times \Sigma \times Q^*$ is a finite set of *rules*. A rule $(q, \sigma, \vec{r}) \in R$ is typically written $q \to \sigma\vec{r}$. Given sentential forms $\xi, \zeta \in T_\Sigma(Q)$ we write $\xi \Rightarrow_G \zeta$ if there exists a rule $q \to \sigma\vec{r} \in R$ and an occurrence $p \in \mathrm{pos}_q(\xi)$ of q in ξ such that $\zeta = \xi[\sigma\vec{r}]_p$. The tree automaton G generates the tree language $L(G) = \{t \in T_\Sigma \mid \exists q \in I \colon q \Rightarrow_G^* t\}$, where $\Rightarrow_G^*$ is the reflexive and transitive closure of $\Rightarrow_G$. A tree language $L \subseteq T_\Sigma$ is *regular* if there exists a TA G such that $L = L(G)$. The class of regular tree languages is denoted by 'Reg'. Moreover, 'FTA' denotes the class of partial identities computed by the regular tree languages; i.e., $\mathrm{FTA} = \{\mathrm{id}_L \mid L \in \mathrm{Reg}\}$, where $\mathrm{id}_L = \{(t, t) \mid t \in L\}$.

3 Synchronous forest substitution grammars

In this section, we introduce our main model, the *(finite-state) synchronous forest-substitution grammar* (SFSG), which is the natural finite-state generalization of the (local non-contiguous) synchronous tree-sequence substitution grammars of [19]. Although we often speak about grammars in the following, we will continue to use 'states' instead of 'nonterminals'. SFSGs naturally coincide in expressive power with the binary rational relations studied by [17, 16], which we will show later. We immediately present it in a form inspired by tree bimorphisms [1] and tree grammars with multi-variables [17].

Definition 1. *A* (finite-state) synchronous forest-substitution grammar *(SFSG) is a tuple $G = (Q, \Sigma, \Delta, I, R)$, where*

- *Q is a finite set of states,*
- *Σ and Δ are alphabets of input and output symbols,*
- *$I \subseteq Q$ is a set of initial states, and*
- *$R \subseteq T_\Sigma(Q)^* \times Q \times T_\Sigma(Q)^*$ is a finite set of* rules.

It is a multi bottom-up tree transducer *(MBOT) if $R \subseteq T_\Sigma(Q) \times Q \times T_\Sigma(Q)^*$ and a* multiple regular tree grammar *(MRTG) if $R \subseteq T_\Sigma(Q) \times Q \times \{\varepsilon\}$.*

In simple terms, an SFSG consists of a finite set of rules that specify a state, for which the rule applies together with a sequence of input tree fragments and a sequence of output tree fragments. In an application of such a rule all fragments replace occurrences of the guarding state at the same time in the input and output tree. This also yields that all occurrences of the same state in those fragments are implicitly linked and prepared to be replaced in parallel in a future rule application.

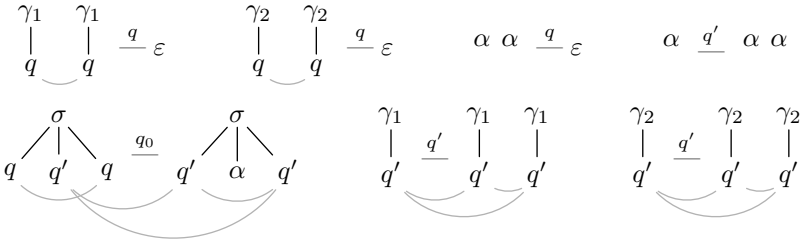

Figure 1: Example rules of the SFSG of Example 1.

An MBOT is a restricted SFSG, in which only a single input tree fragment is allowed in each rule. Compared to its traditional definition [4] the linearity of the single input tree fragment in the states Q is not required here, but as we will see nonlinear rules will not be useful in our version of MBOTs. To make the rules more readable, we also write $\ell_1 \cdots \ell_k \xrightarrow{q} r_1 \cdots r_{k'}$ or $\vec{\ell} \xrightarrow{q} \vec{r}$ for a rule $(\ell_1, \ldots, \ell_k, q, r_1, \ldots, r_{k'}) \in R$.

Example 1. Let $G = (Q, \Sigma, \Sigma, \{q_0\}, R)$ be the SFSG such that

- $Q = \{q_0, q, q'\}$ and $\Sigma = \{\alpha, \gamma_1, \gamma_2, \sigma\}$, and
- for every $\gamma \in \{\gamma_1, \gamma_2\}$ the following rules are in R:

$$\rho_0 = \sigma(q, q', q) \xrightarrow{q_0} \sigma(q', \alpha, q') \quad \rho_\gamma = \gamma(q)\,\gamma(q) \xrightarrow{q} \varepsilon \quad \rho_\alpha = \alpha\,\alpha \xrightarrow{q} \varepsilon$$
$$\rho'_\gamma = \gamma(q') \xrightarrow{q'} \gamma(q')\,\gamma(q') \quad \rho'_\alpha = \alpha \xrightarrow{q'} \alpha\,\alpha \ .$$

The rules are illustrated in Figure 1, where we indicate the implicit links by splines. Clearly, this SFSG G is neither an MBOT nor an MRTG, although the rules for q are valid MRTG rules and the rules for q' are valid MBOT rules.

It remains to define the semantics of SFSGs. We use a bottom-up variant of the classical fixed-points semantics of an SFSG G. It closely corresponds to a semantics based on the evaluation of derivation trees (and that of bimorphisms), which we also define as well. We inductively define the pairs of input and output tree sequences generated by each state, which we call pre-translations. Each pre-translation for a state $q \in Q$ is obtained from a rule $\rho = \vec{\ell} \xrightarrow{q} \vec{r}$ of R by replacing all occurrences of a state $q' \in \mathrm{var}(\vec{\ell}.\vec{r})$ by the corresponding components of a pre-translation for q'.

Definition 2. *Let $G = (Q, \Sigma, \Delta, I, R)$ be an SFSG. A* pre-translation *for $q \in Q$ is a pair $\langle \vec{u}, \vec{v} \rangle$ consisting of an input tree sequence $\vec{u} \in T^*_\Sigma$ and an output tree sequence $\vec{v} \in T^*_\Delta$. For every state $q \in Q$, the* pre-translations $G_q \subseteq T^*_\Sigma \times T^*_\Delta$ generated by q *are defined to be the smallest set T_q such that $\langle \vec{\ell}\theta, \vec{r}\theta' \rangle \in T_q$ for all rules $\rho = \vec{\ell} \xrightarrow{q} \vec{r} \in R$ and suitable substitutions $\theta\colon \mathrm{var}(\vec{\ell}) \to T^*_\Sigma$ and $\theta'\colon \mathrm{var}(\vec{r}) \to T^*_\Delta$ for $\mathrm{var}(\vec{\ell})$ in $\vec{\ell}$ and for $\mathrm{var}(\vec{r})$ in $\vec{r}$, respectively, with pre-translations $\langle \theta(q'), \theta'(q') \rangle$ of $T_{q'}$ for every $q' \in \mathrm{var}(\vec{\ell}.\vec{r})$. The derivation tree corresponding to the newly constructed pre-translation $\langle \vec{\ell}\theta, \vec{r}\theta' \rangle$ is $\rho(t_{q_1}, \ldots, t_{q_k})$, where $\mathrm{var}(\vec{\ell}.\vec{r}) = \{q_1, \ldots, q_k\}$*

with $q_1 <_N \cdots <_Q q_k$ for some fixed total order $\leq_Q$ on Q and $t_{q'}$ is the derivation tree corresponding to the pre-translation $\langle \theta(q'), \theta'(q') \rangle$ for every $q' \in \operatorname{var}(\vec{\ell}.\vec{r})$. The derivation tree language $D_q(G) \subseteq T_R$ contains all derivation trees for the pre-translations $\langle \vec{u}, \vec{v} \rangle \in G_q$.

Example 2. Let us recall the SFSG G of Example 1. The rules $\rho_\alpha = \alpha\, \alpha \xrightarrow{q} \varepsilon$ and $\rho'_\alpha = \alpha \xrightarrow{q'} \alpha\, \alpha$ immediately yield the corresponding pre-translations $\langle \alpha\, \alpha, \varepsilon \rangle \in G_q$ and $\langle \alpha, \alpha\, \alpha \rangle \in G_{q'}$ with derivation trees ρ_α and ρ'_α, respectively. The former pre-translation can be combined with the rule ρ_γ for $\gamma \in \{\gamma_1, \gamma_2\}$ to obtain the pre-translation $\langle \gamma(\alpha)\, \gamma(\alpha), \varepsilon \rangle \in G_q$ with derivation tree $\rho_\gamma(\rho_\alpha)$, and more generally, the pre-translations

$$\langle \gamma_{i_1}(\cdots(\gamma_{i_k}(\alpha))\cdots)\, \gamma_{i_1}(\cdots(\gamma_{i_k}(\alpha))\cdots), \varepsilon \rangle \in G_q$$

for all $k \in \mathbb{N}$ and $i_1, \ldots, i_k \in \{1, 2\}$. The derivation tree corresponding to the displayed pre-translation is $\rho_{\gamma_{i_1}}(\cdots(\rho_{\gamma_{i_k}}(\rho_\alpha))\cdots)$. Similarly, if we use the rules ρ'_γ with $\gamma \in \{\gamma_1, \gamma_2\}$ on the already mentioned pre-translation $\langle \alpha, \alpha\, \alpha \rangle \in G_{q'}$ and the such obtained pre-translations, then we derive the pre-translation

$$\langle \gamma_{i_1}(\cdots(\gamma_{i_k}(\alpha))\cdots), \gamma_{i_1}(\cdots(\gamma_{i_k}(\alpha))\cdots)\, \gamma_{i_1}(\cdots(\gamma_{i_k}(\alpha))\cdots) \rangle \in G_{q'}$$

using the derivation tree $\rho'_{\gamma_{i_1}}(\cdots(\rho'_{\gamma_{i_k}}(\rho'_\alpha))\cdots)$ for all $k \in \mathbb{N}$ and $i_1, \ldots, i_k \in \{1, 2\}$. Plugging those pre-translations into the rule ρ_0, we obtain pre-translations of the form $\langle \sigma(t, t', t), \sigma(t', \alpha, t') \rangle \in G_{q_0}$. We illustrate the last step of the combination process in Figure 2.

The tree translation computed by an SFSG is now simply the set of all those pre-translations computed by the initial states that have sequences of length 1 for the input and output side. The restriction to sequences of length 1 is necessary to obtain a relation on trees. Finally, we also formally define the tree language generated by an SFSG although this notion is most suitable for MRTGs.

Definition 3. *Let $G = (Q, \Sigma, \Delta, I, R)$ be an SFSG. It computes the tree translation $\tau_G \subseteq T_\Sigma \times T_\Delta$ defined by $\tau_G = (\bigcup_{q \in I} G_q) \cap (T_\Sigma \times T_\Delta)$. The tree language $L(G) \subseteq T_\Sigma$ generated by G is $L(G) = (\bigcup_{q \in I} G_q) \cap (T_\Sigma \times \{\varepsilon\})$. Two SFSGs are (translation) equivalent if their computed tree translations coincide and language equivalent if their generated tree languages coincide. The classes* `SFSG` *and* `MBOT` *contain all tree translations computable by SFSGs and MBOTs, respectively, and the class* `MRTG` *denotes the class of all tree languages generated by MRTGs.*

In the rest of this section, we present a normal form for MBOTs and an alternative characterization of SFSGs in terms of classical bimorphisms [1] using a tree language of `MRTG` as center language. The former result demonstrates that our MBOTs are as expressive as the notion discussed in [4]. We conclude with some simple properties of `SFSG`, but we start with the normal form for MBOTs.

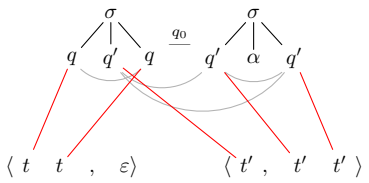

Figure 2: Illustration of the combination of a rule with pre-translations.

Lemma 1. *For every MBOT* $G = (Q, \Sigma, \Delta, I, R)$ *there is a translation equivalent MBOT* $G' = (Q, \Sigma, \Delta, I, R')$ *such that* t *is linear in* Q *and* $\mathrm{var}(\vec{r}) \subseteq \mathrm{var}(t)$ *for every* $t \xrightarrow{q} \vec{r} \in R'$.

Proof. We set $R' = \{t \xrightarrow{q} \vec{r} \in R \mid t \text{ linear in } Q, \mathrm{var}(\vec{r}) \subseteq \mathrm{var}(t)\}$, which makes sure that the MBOT G' obeys the required restrictions. The translation equivalence of G and G' remains a proof obligation. We first observe that $|\vec{u}| = 1$ for every state $q \in Q$ and pre-translation $\langle \vec{u}, \vec{v} \rangle \in G_q$ due to the rule shape of G. Now, let $\rho = t \xrightarrow{q} \vec{r} \in R$ be a rule that admits a state $q' \in \mathrm{var}(\vec{r}) \setminus \mathrm{var}(t)$. To build a pre-translation utilizing ρ (whose derivation tree has root label ρ), we need a pre-translation $\langle \varepsilon, \vec{v} \rangle \in G_{q'}$ because $q' \in \mathrm{var}(t.\vec{r})$, but $q' \notin \mathrm{var}(t)$. Such pre-translations do not exist, hence the rule ρ is useless (i.e., there are no derivation trees that contain ρ), which proves that deleting it does not affect the semantics. Similarly, let $\rho = t \xrightarrow{q} \vec{r} \in R$ be a rule such that t is not linear in Q; i.e., there exists a state $q' \in Q$ such that $|\mathrm{pos}_{q'}(t)| \geq 2$. To utilize such a rule, we need a pre-translation $\langle \vec{u}, \vec{v} \rangle \in G_{q'}$ with $|\vec{u}| = |\mathrm{pos}_{q'}(t)| \geq 2$, which again do not exist. Consequently, both types of rules can be deleted without effect, which proves that G and G' are translation equivalent. □

Consequently, our class `MBOT` coincides with the notion of [4], so we can freely use the known properties of `MBOT`. Already in [12, 4] MBOTs were transformed into a normal form before composition. In this normal form, at most one (input or output) symbol is allowed in each rule. For our purposes, a slightly less restricted variant, in which at most one input symbol may occur in each rule is sufficient since we compose the input parts of two MBOTs. Let us recall the relevant normalization result [4].

Lemma 2 (see [4, Lemma 14])**.** *For every MBOT* $G = (Q, \Sigma, \Delta, I, R)$ *there exists a translation equivalent MBOT* $G' = (Q', \Sigma, \Delta, I', R')$ *in normal form, which means that* $|\mathrm{pos}_\Sigma(t)| \leq 1$ *for every rule* $t \xrightarrow{q} \vec{r} \in R'$.

Proof. By Lemma 1 we can construct a translation equivalent MBOT G'' in the sense of [4]. With the help of [4, Lemma 14], we can then construct a translation equivalent MBOT G' in normal form. □

For MBOTs in normal form, we can now define the determinism property, which we use to avoid the k-morphisms of [1]. We note that deterministic MBOTs are

slightly more expressive than k-morphisms.

Definition 4. *An MBOT $(Q, \Sigma, \Delta, I, R)$ in normal form is* deterministic *if $|I| = 1$, $t \notin Q$ for every $t \xrightarrow{q} \vec{r} \in R$, and for every $q \in Q$ and $\sigma \in \Sigma$ there exists at most one rule $t \xrightarrow{q} \vec{r} \in R$ with $t(\varepsilon) = \sigma$. It is a* deterministic linear top-down tree transducer with regular look-ahead *(deterministic LTOP^R) if additionally $|\vec{r}| \leq 1$ for all $t \xrightarrow{q} \vec{r} \in R$.*

We conclude with the presentation of some simple properties of SFSG including one characterization of it in terms of bimorphisms. We will develop another bimorphism characterization in the next section.

Lemma 3. *We observe that (i) $\mathtt{SFSG} = \mathtt{SFSG}^{-1}$, (ii) both the domain $\mathrm{dom}(\tau)$ and the range $\mathrm{ran}(\tau)$ of a tree translation $\tau \in \mathtt{SFSG}$ are not necessarily regular, and (iii) $\mathtt{MBOT} \subsetneq \mathtt{SFSG}$.*

Proof. The first property is immediate because the syntactic definition of SFSGs is completely symmetric. The tree translation τ_G computed by the SFSG G of Example 1 is such that both its domain and its range are not regular, which proves the second property. Finally, the inclusion in the third item is obvious, and its strictness follows because $\mathrm{dom}(\tau)$ is regular for every $\tau \in \mathtt{MBOT}$ by Lemma 1 and [4, Theorem 25], so $\tau_G \notin \mathtt{MBOT}$. □

Theorem 1. *For every SFSG G there exists an MRTG G_0 and two deterministic LTOP^{R}s G_1 and G_2 such that $\tau_G = \{(\tau_{G_1}(t), \tau_{G_2}(t)) \mid t \in L(G_0)\}$.*

Proof. Let $G = (Q, \Sigma, \Delta, I, R)$ be the SFSG. We start with the construction of the MRTG $G_0 = (Q \cup \{\star\}, \Sigma \cup \Delta \cup \{\gamma\}, \emptyset, \{\star\}, R_0)$ such that $\star \notin Q$, $\gamma \notin \Sigma \cup \Delta$, and

$$R_0 = \{\gamma(q_0, q_0) \xrightarrow{\star} \varepsilon \mid q_0 \in I\} \cup \{\vec{\ell}.\vec{r} \xrightarrow{q} \varepsilon \mid \vec{\ell} \xrightarrow{q} \vec{r} \in R\} \ .$$

Let $\Gamma = \Sigma \cup \Delta \cup \{\gamma\}$. The two deterministic LTOP$^{\mathrm{R}}$s G_1 and G_2 simply project on the first and second subtree, respectively. We omit their straightforward, albeit technical specification and the obvious correctness proof. □

Using Theorem 1 the relation of SFSG to the binary rational relations of [17] should be apparent. The main difference that remains is that we cannot specify the order, in which components are substituted. However, this does not restrict the expressive power. For the converse inclusion between SFSG and certain bimorphism, we restrict ourselves to linear tree homomorphisms [7, 8], which are slightly cumbersome to define in our notation. Note that the LTOP$^{\mathrm{R}}$s constructed in the previous proof are actually linear tree homomorphisms. We let LHOM denote the class of all linear tree homomorphisms, and assume that each tree homomorphism h is extended to act on state leaves as the identity; i.e., $h(q) = q$ for all $q \in Q$.

Theorem 2. *For all MRTGs $G_0 = (Q, \Gamma, \emptyset, I, R_0)$ and all tree homomorphisms $h_1 \colon T_\Gamma \to T_\Sigma$ and $h_2 \colon T_\Gamma \to T_\Delta$ there exists an SFSG $G = (Q, \Sigma, \Delta, I, R)$ such that $\tau_G = \{(\tau_{G_1}(t), \tau_{G_2}(t)) \mid t \in L(G_0)\}$.*

Proof. We let $R = \{h_1(\ell_1)\cdots h_1(\ell_k) \xrightarrow{q} h_2(\ell_1)\cdots h_2(\ell_k) \mid \ell_1\cdots\ell_k \xrightarrow{q} \varepsilon \in R_0\}$. We again omit the straightforward correctness proof. □

Consequently, SFSG can be characterized by bimorphisms [1] with linear tree homomorphisms and a center language from MRTG.

4 Composition and decomposition

For our second characterization of SFSG, we first characterize it in terms of MBOT. Since we already showed that MBOT $\subsetneq$ SFSG in Lemma 3, we need a composition of MBOTs to characterize the expressive power of SFSGs. The relevant decomposition is presented in Theorem 3, and the corresponding composition is presented in Theorem 5.

Theorem 3 (see [17, Proposition 4.5])**.** *For every SFSG G, there exist two deterministic MBOTs G_1 and G_2 such that $\tau_G = \tau_{G_1}^{-1} ; \tau_{G_2}$.*

Proof. Let $G = (Q, \Sigma, \Delta, I, R)$ be the SFSG. As usual, we assume a total order $\leq$ on Q, and whenever we explicitly list states like $\{q_1, \ldots, q_k\}$, we assume that $q_1 < \cdots < q_k$. We construct the two MBOTs $G_1 = (Q, R, \Sigma, I, R_1)$ and $G_2 = (Q, R, \Delta, I, R_2)$ such that

- $R_1 = \{\rho(q_1, \ldots, q_k) \xrightarrow{q} \vec{\ell} \mid \rho = \vec{\ell} \xrightarrow{q} \vec{r} \in R,\ \mathrm{var}(\vec{\ell}.\vec{r}) = \{q_1, \ldots, q_k\}\}$, and
- $R_2 = \{\rho(q_1, \ldots, q_k) \xrightarrow{q} \vec{r} \mid \rho = \vec{\ell} \xrightarrow{q} \vec{r} \in R,\ \mathrm{var}(\vec{\ell}.\vec{r}) = \{q_1, \ldots, q_k\}\}$.

Obviously, both G_1 and G_2 are deterministic MBOTs. A straightforward induction can be used to prove that G_1 and G_2 translate derivation trees of $D_q(G)$ with $q \in Q$ into the corresponding input and output tree, respectively. Since each derivation tree $t \in D_q(G)$ uniquely determines the corresponding input and output tree, we immediately obtain that $\tau_G = \tau_{G_1}^{-1} ; \tau_{G_2}$. A more detailed proof can be found in [17]. □

In the proof of Theorem 3 the rule ρ uniquely determines the state q. Nevertheless, the constructed MBOTs have (potentially) several states as we need to check the finite-state behavior of the SFSG. It follows straightforwardly from the proof of Theorem 3 that each SFSG can be characterized by a regular derivation tree language and two deterministic MBOTs mapping the derivation trees to the input and output trees. This view essentially coincides with the bimorphism approach [1], and SFSGs are equally expressive as the bimorphisms of [1], in which both the input and output morphisms are allowed to be k-morphisms. We reuse this characterization later on, so we make it explicit here.

Theorem 4. SFSG $=$ dMBOT^{-1} ; FTA ; dMBOT*, where* dMBOT *is the class of all tree translations computed by deterministic MBOTs.*

Now we are ready to state our first composition result. We first prove it using several known results on decompositions and compositions together with a few new results.

Theorem 5. $\mathtt{MBOT}^{-1} ; \mathtt{MBOT} \subseteq \mathtt{SFSG}$.

Proof. Let G_1 and G_2 be the given input MBOTs. Without loss of generality, let G_1 and G_2 be in normal form (see Lemma 2). With the help of the construction of [4, Lemma 6] applied to both G_1 and G_2 we obtain delabelings d_1 and d_2, regular tree languages $L_1, L_2 \in \mathrm{Reg}$, and deterministic MBOTs G_1' and G_2' such that

$$\tau_{G_1} = d_1^{-1} ; \mathrm{id}_{L_1} ; \tau_{G_1'} \qquad \text{and} \qquad \tau_{G_1} = d_2^{-1} ; \mathrm{id}_{L_2} ; \tau_{G_2'} \ .$$

This situation is depicted in Figure 3. We observe that

$$\tau_{G_1}^{-1} ; \tau_{G_2} = (d_1^{-1} ; \mathrm{id}_{L_1} ; \tau_{G_1'})^{-1} ; (d_2^{-1} ; \mathrm{id}_{L_2} ; \tau_{G_2'}) = \tau_{G_1'}^{-1} ; \mathrm{id}_{L_1} ; d_1 ; d_2^{-1} ; \mathrm{id}_{L_2} ; \tau_{G_2'} \ .$$

Next, we show that the composition $d_1 ; d_2^{-1}$ can equivalently be expressed as the composition $e_2^{-1} ; e_1$ for some delabelings e_1 and e_2 following the construction of [3, Sect. II-1-4-2-1]. To this end, let $d_1 \colon \Sigma \to \Delta \cup \{\Box\}$, and we set $\Sigma' = \{\underline{\sigma} \mid \sigma \in \Sigma,\, d_1(\sigma) = \Box\}$, which is an alphabet containing copies of the elements of Σ that are erased by d_1. Similarly, let $d_2 \colon \Gamma \to \Delta \cup \{\Box\}$, and we set $\Gamma' = \{\overline{\gamma} \mid \gamma \in \Gamma,\, d_2(\gamma) = \Box\}$ to an alphabet that contains copies of those elements of Γ that are erased by d_2. Moreover, let

$$\Delta'' = \{\langle \sigma, \gamma \rangle \mid \sigma \in \Sigma,\, \gamma \in \Gamma,\, d_1(\sigma) = d_2(\gamma) \neq \Box\}$$

and $\Delta' = \Sigma' \cup \Gamma' \cup \Delta''$. Then we construct the two delabelings $e_1 \colon \Delta' \to \Sigma \cup \{\Box\}$ and $e_2 \colon \Delta' \to \Gamma \cup \{\Box\}$ as follows:

$$\begin{array}{lll} e_2(\underline{\sigma}) = \sigma & e_2(\overline{\gamma}) = \Box & e_2(\langle \sigma, \gamma \rangle) = \sigma \\ e_1(\underline{\sigma}) = \Box & e_1(\overline{\gamma}) = \gamma & e_1(\langle \sigma, \gamma \rangle) = \gamma \end{array}$$

for all $\underline{\sigma} \in \Sigma'$, $\overline{\gamma} \in \Gamma'$, and $\langle \sigma, \gamma \rangle \in \Delta''$. We leave the formal proof of $d_1 ; d_2^{-1} = e_2^{-1} ; e_1$ to the interested reader, but mention that it can be achieved by a simple induction. Thus, we arrive at

$$\tau_{G_1}^{-1} ; \tau_{G_2} = \tau_{G_1'}^{-1} ; \mathrm{id}_{L_1} ; d_1 ; d_2^{-1} ; \mathrm{id}_{L_2} ; \tau_{G_2'} = (\tau_{G_1'}^{-1} ; \mathrm{id}_{L_1} ; e_2^{-1}) ; (e_1 ; \mathrm{id}_{L_2} ; \tau_{G_2'})$$

using the just explained exchange of the delabelings. Since inverse delabelings preserve regular tree languages, we let $L_1' = e_2^{-1}(L_1)$ and $L_2' = e_1^{-1}(L_2)$, which are clearly both regular, so also their intersection $L_1' \cap L_2'$ is regular [7, 8]. Consequently,

$$\tau_{G_1}^{-1} ; \tau_{G_2} = (\tau_{G_1'}^{-1} ; e_2^{-1}) ; \mathrm{id}_{L_1' \cap L_2'} ; (e_1 ; \tau_{G_2'}) \ ,$$

which we can further simplify to $\tau_{G_1''}^{-1} ; \mathrm{id}_{L_1' \cap L_2'} ; \tau_{G_2''}$ by composing the delabelings e_1 and e_2 with the deterministic MBOTs G_1' and G_2' to obtain the deterministic MBOTs G_1'' and G_2'', respectively, using [4, Theorem 23]. With this final step, we obtain a bimorphism representation of $\tau_{G_1}^{-1} ; \tau_{G_2}$ and according to Theorem 4 we have $\tau_{G_1}^{-1} ; \tau_{G_2} \in \mathtt{SFSG}$. □

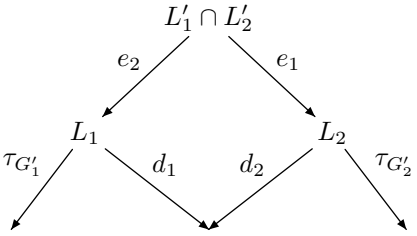

Figure 3: Illustration of the approach used in the proof of Theorem 5.

Problem	String level	Tree level
Parsing	$\mathcal{O}\big(\lvert G\rvert \cdot (\lvert w_1\rvert \cdot \lvert w_2\rvert)^{2r+2}\big)$	$\mathcal{O}\big(\lvert G\rvert \cdot \lvert t_1\rvert \cdot \lvert t_2\rvert\big)$
Translation	$\mathcal{O}\big(\lvert G\rvert \cdot \lvert w_1\rvert^{r+2}\big)$	$\mathcal{O}\big(\lvert G\rvert \cdot \lvert t_1\rvert\big)$

Table 1: Complexity results for an SFSG G and input strings (w_1, w_2) as well as trees (t_1, t_2), where $r = \max\{|\vec{\ell}.\vec{r}| \mid \vec{\ell} \xrightarrow{q} \vec{r} \in R\}$ is the length of the longest sequence of input and output tree fragments in a rule of G.

Corollary 1 (of Theorems 3 and 5)**.** $\mathtt{SFSG} = \mathtt{MBOT}^{-1} ; \mathtt{MBOT}$.

We conclude with some additional properties of SFSG and their consequences for MBOT using our main result of Corollary 1. In particular, it is known [9] that the output string language of an MBOT is a language generated by an LCFRS (linear context-free rewriting system) [20, 18]. Using Corollary 1, we can conclude that both the input and the output string language of an SFSG are generated by an LCFRS as well. Similarly, together with Theorem 1 we can also conclude that the input and output tree languages are in MRTG. Moreover, we can import several complexity results from MBOT [14] to SFSG as indicated in Table 1.

Lemma 4 (see [16, Example 5])**.** SFSG *is not closed under composition.*

Corollary 2. $\mathtt{MBOT} ; \mathtt{MBOT}^{-1} \not\subseteq \mathtt{SFSG}$.

Proof. Assume on the contrary that $\mathtt{MBOT} ; \mathtt{MBOT}^{-1} \subseteq \mathtt{SFSG}$. Then

$$\begin{aligned}\mathtt{SFSG} ; \mathtt{SFSG} &\subseteq (\mathtt{MBOT}^{-1} ; \mathtt{MBOT}) ; (\mathtt{MBOT}^{-1} ; \mathtt{MBOT}) \subseteq \mathtt{MBOT}^{-1} ; \mathtt{SFSG} ; \mathtt{MBOT} \\ &\subseteq \mathtt{MBOT}^{-1} ; (\mathtt{MBOT}^{-1} ; \mathtt{MBOT}) ; \mathtt{MBOT} \quad \subseteq \mathtt{MBOT}^{-1} ; \mathtt{MBOT} = \mathtt{SFSG}\end{aligned}$$

using Corollary 1, our assumption, Corollary 1, the closure under composition for MBOT [4, Theorem 23], and Corollary 1 once more. However, the result contradicts Lemma 4, thus our assumption is false, proving the result. □

References

[1] Arnold, André and Dauchet, Max. Morphismes et bimorphismes d'arbres. *Theoret. Comput. Sci.*, 20(1):33–93, 1982.

[2] Chiang, David. An introduction to synchronous grammars. In *Proc. 44th ACL*. ACL, 2006. Part of a tutorial given with K. Knight.

[3] Dauchet, Max. *Transductions de forêts — Bimorphismes de magmoïdes*. Première thèse, Université de Lille, 1977.

[4] Engelfriet, Joost, Lilin, Eric, and Maletti, Andreas. Composition and decomposition of extended multi bottom-up tree transducers. *Acta Inform.*, 46(8):561–590, 2009.

[5] Fülöp, Zoltán, Kühnemann, Armin, and Vogler, Heiko. A bottom-up characterization of deterministic top-down tree transducers with regular look-ahead. *Inf. Process. Lett.*, 91(2):57–67, 2004.

[6] Fülöp, Zoltán, Kühnemann, Armin, and Vogler, Heiko. Linear deterministic multi bottom-up tree transducers. *Theoret. Comput. Sci.*, 347(1–2):276–287, 2005.

[7] Gécseg, Ferenc and Steinby, Magnus. *Tree Automata*. Akadémiai Kiadó, 1984. 2nd edition available at `https://arxiv.org/abs/1509.06233`.

[8] Gécseg, Ferenc and Steinby, Magnus. Tree languages. In Rozenberg, Grzegorz and Salomaa, Arto, editors, *Handbook of Formal Languages*, volume 3, chapter 1, pages 1–68. Springer, 1997.

[9] Gildea, Daniel. On the string translations produced by multi bottom-up tree transducers. *Comput. Linguist.*, 38(3):673–693, 2012.

[10] Knight, Kevin and Graehl, Jonathan. An overview of probabilistic tree transducers for natural language processing. In *Proc. 6th CICLing*, volume 3406 of LNCS, pages 1–24. Springer, 2005.

[11] Lilin, Eric. Propriétés de clôture d'une extension de transducteurs d'arbres déterministes. In *Proc. 6th CAAP*, volume 112 of LNCS, pages 280–289. Springer, 1981.

[12] Maletti, Andreas. Compositions of extended top-down tree transducers. *Inform. and Comput.*, 206(9–10):1187–1196, 2008.

[13] Maletti, Andreas. Why synchronous tree substitution grammars? In *Proc. 2010 HLT-NAACL*, pages 876–884. ACL, 2010.

[14] Maletti, Andreas. An alternative to synchronous tree substitution grammars. *J. Nat. Lang. Engrg.*, 17(2):221–242, 2011.

[15] Maletti, Andreas. How to train your multi bottom-up tree transducer. In *Proc. 49th ACL*, pages 825–834. ACL, 2011.

[16] Radmacher, Frank G. An automata theoretic approach to rational tree relations. In *Proc. 34th SOFSEM*, volume 4910 of LNCS, pages 424–435. Springer, 2008.

[17] Raoult, Jean-Claude. Rational tree relations. *Bull. Belg. Math. Soc. Simon Stevin*, 4(1):149–176, 1997.

[18] Seki, Hiroyuki, Matsumura, Takashi, Fujii, Mamoru, and Kasami, Tadao. On multiple context-free grammars. *Theoret. Comput. Sci.*, 88(2):191–229, 1991.

[19] Sun, Jun, Zhang, Min, and Tan, Chew Lim. A non-contiguous tree sequence alignment-based model for statistical machine translation. In *Proc. 47th ACL*, pages 914–922. ACL, 2009.

[20] Vijay-Shanker, K., Weir, David J., and Joshi, Aravind K. Characterizing structural descriptions produced by various grammatical formalisms. In *Proc. 25th ACL*, pages 104–111. ACL, 1987.

[21] Zhang, Min, Jiang, Hongfei, Aw, Aiti, Li, Haizhou, Tan, Chew Lim, and Li, Sheng. A tree sequence alignment-based tree-to-tree translation model. In *Proc. 46th ACL*, pages 559–567. ACL, 2008.

[22] Zhang, Min, Jiang, Hongfei, Li, Haizhou, Aw, Aiti, and Li, Sheng. Grammar comparison study for translational equivalence modeling and statistical machine translation. In *Proc. 22nd CoLing*, pages 1097–1104. ACL, 2008.