Synchronous Forest Substitution Grammars

Andreas Maletti*

Universität Stuttgart, Institute for Natural Language Processing
Pfaffenwaldring 5b, 70569 Stuttgart, Germany
andreas.maletti@ims.uni-stuttgart.de

Abstract. The expressive power of synchronous forest (tree-sequence) substitution grammars (SFSG) is studied in relation to multi bottom-up tree transducers (MBOT). It is proved that SFSG have exactly the same expressive power as compositions of an inverse MBOT with an MBOT. This result is used to derive complexity results for SFSG 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 MBOT is closed under composition.

1 Introduction

Synchronous forest substitution grammars (SFSG) [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 SFSG are a natural, practically relevant, and theoretically interesting model for tree translations. Roughly speaking, SFSG 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 tree translations computed by SFSG) can also be characterized by rational expressions [17] and automata [16].

Multi bottom-up tree transducers (MBOT) [1,4] are restricted SFSG, in which only the output side is allowed to have several components. They were rediscovered 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 SFSG in terms of MBOT. We show that the expressive power of SFSG 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

* The author gratefully acknowledges the financial support by the German Research Foundation (DFG) grant MA / 4959 / 1-1.

T. Muntean, D. Poulakis, and R. Rolland (Eds.): CAI 2013, LNCS 8080, pp. 235–246, 2013.

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 SFSG 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 MBOT, which are closed under composition [4]. Overall, we can thus also characterize the expressive power of SFSG by an arbitrary chain of inverse MBOT followed by an arbitrary chain of MBOT.

2 Preliminaries

The set of nonnegative integers is $\mathbb{N}$. We write $[k]$ for the set $\{i \in \mathbb{N} \mid 1 \leq i \leq k\}$, and we treat functions (or maps) as special relations. 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 $R_1 \subseteq A \times B$ and $R_2 \subseteq B \times C$, the *composition* $R_1 ; R_2 \subseteq A \times C$ of R_1 and R_2 is $R_1 ; R_2 = \{(a,c) \in A \times C \mid \exists b \in B \colon (a,b) \in R_1, (b,c) \in R_2\}$. These notions and notations are lifted to sets of relations as usual. Given a set Σ, the set of all words over Σ is Σ^*, of which ε is the empty word. The concatenation of two words $u, w \in \Sigma^*$ is denoted by uw. The length of a word $w = \sigma_1 \cdots \sigma_k$ with $\sigma_i \in \Sigma$ for all $i \in [k]$ is $|w| = k$. We simply write w_i for the i^{th} letter of w (i.e., $w_i = \sigma_i$) for all $i \in [k]$. For every $k \in \mathbb{N}$, we let $\Sigma^k = \{w \in \Sigma^* \mid k = |w|\}$.

A ranked alphabet (Σ, rk) consists of an alphabet Σ and a map $\mathrm{rk} \colon \Sigma \to \mathbb{N}$. The symbol $\sigma \in \Sigma$ has rank $\mathrm{rk}(\sigma)$, and we let $\Sigma_k = \{\sigma \in \Sigma \mid \mathrm{rk}(\sigma) = k\}$ for all $k \in \mathbb{N}$. We usually denote the ranked alphabet (Σ, rk) by just Σ and write $\sigma^{(k)}$ to indicate that $\mathrm{rk}(\sigma) = k$. The set $T_\Sigma(N)$ of all *Σ-trees indexed by* the set N is the smallest set T such that $N \subseteq T$ and $\sigma(\boldsymbol{t}) \in T$ for all $\sigma \in \Sigma$ and $\boldsymbol{t} \in T^{\mathrm{rk}(\sigma)}$. Such a sequence $\boldsymbol{t}$ of trees is also called *forest*. Consequently, a tree t is either an element of N or it consists of a root node labeled σ followed by a forest $\boldsymbol{t}$ of $\mathrm{rk}(\sigma)$ children. To improve the readability, we often write a forest $t_1 \cdots t_k$ as $t_1, \ldots, t_k$. The *positions* $\mathrm{pos}(t), \mathrm{pos}(\boldsymbol{u}) \subseteq \mathbb{N}^*$ of a tree $t \in T_\Sigma(N)$ and a forest $\boldsymbol{u} \in T_\Sigma(N)^*$ are inductively defined by (i) $\mathrm{pos}(n) = \{\varepsilon\}$, (ii) $\mathrm{pos}(\sigma(\boldsymbol{t})) = \{\varepsilon\} \cup \mathrm{pos}(\boldsymbol{t})$, and (iii) $\mathrm{pos}(\boldsymbol{u}) = \bigcup_{i=1}^{|\boldsymbol{u}|} \{ip \mid p \in \mathrm{pos}(u_i)\}$ for every $n \in N$, $\sigma \in \Sigma_k$, and $\boldsymbol{t} \in T_\Sigma(N)^k$. This yields an undesirable difference between $\mathrm{pos}(t)$ and $\mathrm{pos}(\boldsymbol{u})$ with $\boldsymbol{u} = (t)$. Note that positions are totally ordered via the (standard) lexicographic ordering on $\mathbb{N}^*$. Let $t, t' \in T_\Sigma(N)$ 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 $n(\varepsilon) = n|_\varepsilon = n$ and $n[t']_\varepsilon = t'$ for every $n \in N$ and

$$t(p) = \begin{cases} \sigma & \text{if } p = \varepsilon \\ \boldsymbol{t}(p) & \text{if } p \neq \varepsilon \end{cases} \qquad t|_p = \begin{cases} t & \text{if } p = \varepsilon \\ \boldsymbol{t}|_p & \text{if } p \neq \varepsilon \end{cases} \qquad t[t']_p = \begin{cases} t' & \text{if } p = \varepsilon \\ \boldsymbol{t}[t']_p & \text{if } p \neq \varepsilon \end{cases}$$

$$\boldsymbol{u}(ip') = u_i(p') \qquad \boldsymbol{u}|_{ip'} = u_i|_{p'} \qquad \boldsymbol{u}[t']_{ip'} = u_i[t']_{p'}$$

for all $t = \sigma(\boldsymbol{t})$ with $\sigma \in \Sigma_k$ and $\boldsymbol{t} \in T_\Sigma(N)^k$, $\boldsymbol{u} \in T_\Sigma(N)^*$, $1 \leq i \leq |\boldsymbol{u}|$, and $p' \in \mathrm{pos}(u_i)$. As demonstrated, these notions are also defined for forests $\boldsymbol{u}$. A position $p \in \mathrm{pos}(t)$ is a *leaf* (in t) if $p1 \notin \mathrm{pos}(t)$. For every $S \subseteq N \cup \Sigma$, we let $\mathrm{pos}_S(t) = \{p \in \mathrm{pos}(t) \mid t(p) \in S\}$ and $\mathrm{pos}_s(t) = \mathrm{pos}_{\{s\}}(t)$ for every $s \in N \cup \Sigma$. The tree $t \in T_\Sigma(N)$ is *linear* in $S \subseteq N$ if $|\mathrm{pos}_s(t)| \leq 1$ for every $s \in S$. The *variables* of t are $\mathrm{var}(t) = \{n \in N \mid \mathrm{pos}_n(t) \neq \emptyset\}$, and $\mathrm{var}(\boldsymbol{u}) = \bigcup_{i=1}^{|u|} \mathrm{var}(\boldsymbol{u}_i)$ for all $\boldsymbol{u} \in T_\Sigma(N)^*$. Given $S \subseteq N$, $\boldsymbol{u} \in T_\Sigma(N)^*$, and $\theta\colon S \to T_\Sigma(N)^*$ such that $|\theta(s)| = |\mathrm{pos}_s(\boldsymbol{u})|$ for every $s \in S$, the forest $\boldsymbol{u}\theta$ is obtained from $\boldsymbol{u}$ by replacing for every $s \in S$ the occurrences $\mathrm{pos}_s(\boldsymbol{u}) = \{p_1, \ldots, p_k\}$ with $p_1 < \cdots < p_k$ of (the leaf) s in $\boldsymbol{u}$ by the trees $\theta(s)_1, \ldots, \theta(s)_k$, respectively.

Given ranked alphabets Σ and Δ, a mapping $d\colon \bigcup_{k\in\mathbb{N}} \Sigma_k \to (\Delta_k \cup \{\Box\})$ is a *delabeling* if $d(\sigma) \in \Delta_k$ for all $\sigma \in \Sigma_k$ with $k \neq 1$. Thus, a delabeling is similar to a relabeling [7,8], but it can also erase unary symbols. It induces a mapping $d\colon T_\Sigma \to T_\Delta$ such that $d(\sigma(\boldsymbol{t})) = d(t_1)$ if $d(\sigma) = \Box$ and $d(\sigma)(d(t_1), \ldots, d(t_k))$ otherwise for all $\sigma \in \Sigma_k$ and $\boldsymbol{t} \in T_\Sigma^k$. Finally, let us recall the regular tree languages [7,8]. A *regular tree grammar* (RTG) is a tuple $G = (N, \Sigma, I, R)$ such that N is a finite set of *nonterminals*, Σ is a ranked alphabet of symbols, $I \subseteq N$ is a set of *initial nonterminals*, and $R \subseteq N \times T_\Sigma(N)$ is a finite set of *rules*. A rule $(n, r) \in R$ is typically written $n \to r$, and for every $n \in N$, we let $R_n = \{n \to r \mid n \to r \in R\}$. Given $\xi, \zeta \in T_\Sigma(N)$ we write $\xi \Rightarrow_G \zeta$ if there exists a a rule $n \to r \in R$ and a position $p \in \mathrm{pos}_n(\xi)$ such that $\zeta = \xi[r]_p$. The regular tree grammar G generates the tree language $L(G) = \{t \in T_\Sigma \mid \exists n \in I\colon n \Rightarrow_G^* t\}$, 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 regular tree grammar 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

The *(stateful) synchronous forest substitution grammars* (SFSG) are a natural generalization of the non-contiguous synchronous tree sequence substitution grammars of [19] to include full grammar nonterminals (or states). They naturally coincide with the binary rational relations studied by [17,16]. To keep the presentation simple, we assume a global ranked alphabet Σ of input and output terminal symbols. Moreover, we immediately present it in a form inspired by tree bimorphisms [1] and tree grammars with multi-variables [17].

Definition 1. *A* (stateful) synchronous forest substitution grammar *(SFSG) is a tuple $G = (N, \Sigma, I, R, B)$, where*

- *(N, Σ, I, R) is a regular tree grammar, and*
- *$B \subseteq (\bigcup_{n\in I} R_n \times R_n) \cup (\bigcup_{n\in N\setminus I} R_n^* \times R_n^*)$ is a finite set of* aligned rules.

It is a multi bottom-up tree transducer *(MBOT) if $B \subseteq \bigcup_{n\in N} R_n \times R_n^*$.*

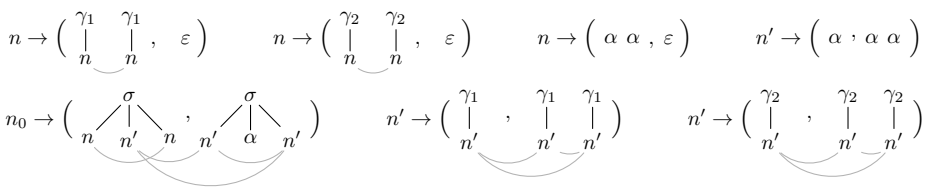

Fig. 1. Aligned example rules of the SFSG of Example 2.

Roughly speaking, we have a regular tree grammar containing all the potentially used rules. However, potentially several rules with the same left-hand side are applied at the same time on both the input and the output side. This dependence is expressed by the set B of aligned rules. For all initial nonterminals, only one rule is applied to the input and output side as we want to compute a tree translation. For the remaining nonterminals we can use arbitrarily many rules on the input and the output side. The alignment in the rules is established implicitly by occurrences of the same nonterminal in the right-hand sides. To make aligned rules more readable, we also write $n \to (\ell_1 \cdots \ell_k, r_1 \cdots r_{k'})$ or $n \to (\boldsymbol{\ell}, \boldsymbol{r})$ for a rule $(n \to \ell_1 \cdots n \to \ell_k, n \to r_1 \cdots n \to r_{k'}) \in B$, where $n \to \ell_1, \ldots, n \to \ell_k, n \to r_1, \ldots, n \to r_{k'} \in R_n$ are rules for the same nonterminal $n \in N$. In short, we write the common nonterminal only once on the left-hand side and then group all the right-hand sides of the rules of R_n. We assume that the nonterminals N of each SFSG are totally ordered by $\leq_N$. Finally, we let $\operatorname{var}(\chi) = \operatorname{var}(\boldsymbol{\ell}) \cup \operatorname{var}(\boldsymbol{r})$ for every rule $\chi = n \to (\boldsymbol{\ell}, \boldsymbol{r})$, where $\boldsymbol{\ell}$ and $\boldsymbol{r}$ contain only the right-hand sides of rules of R (as per the previous declaration).

Example 2. Let $(N, \Sigma, \{n_0\}, R)$ be the regular tree grammar such that
- $N = \{n_0, n, n'\}$ with $n_0 <_N n <_N n'$ and $\Sigma = \{\alpha^{(0)}, \gamma_1^{(1)}, \gamma_2^{(1)}, \sigma^{(3)}\}$, and
- the following rules are in R:

$$\begin{array}{llll} \rho_0 \colon n_0 \to \sigma(n, n', n) & \rho_2 \colon n \to \gamma_1(n) & \rho_4 \colon n \to \gamma_2(n) & \rho_6 \colon n \to \alpha \\ \rho_1 \colon n_0 \to \sigma(n', \alpha, n') & \rho_3 \colon n' \to \gamma_1(n') & \rho_5 \colon n' \to \gamma_2(n') & \rho_7 \colon n' \to \alpha \ . \end{array}$$

Based on this RTG we construct the SFSG $G = (N, \Sigma, \{n_0\}, R, B)$ with

$$B = \{(\rho_0, \rho_1),\ (\rho_2\rho_2, \varepsilon),\ (\rho_4\rho_4, \varepsilon),\ (\rho_6\rho_6, \varepsilon),\ (\rho_3, \rho_3\rho_3),\ (\rho_5, \rho_5\rho_5)\,(\rho_7, \rho_7\rho_7)\} \ .$$

We illustrate these aligned rules in Fig. 1, where we indicate the implicit links by splines. Clearly, the SFSG G is (syntactically) not an MBOT.

Next, we introduce the (bottom-up) semantics of an SFSG G. It works on pre-translations, which are pairs of input and output tree sequences together with a governing nonterminal. The pre-translations computed by G are inductively defined, and each pre-translation is obtained from an aligned rule $\chi = n \to (\boldsymbol{\ell}, \boldsymbol{r})$ of G by replacing each nonterminal $n \in \operatorname{var}(\chi)$ by a pre-translation computed by G that is governed by n. Alongside, we introduce the derivation tree, which records how the aligned rules combined.

Definition 3. *Let $G = (N, \Sigma, I, R, B)$ be an SFSG. A* pre-translation *for G is a triple $\langle \boldsymbol{t}, n, \boldsymbol{u} \rangle$ consisting of a nonterminal $n \in N$ and input and output tree sequences $\boldsymbol{t}, \boldsymbol{u} \in T_\Sigma^*$. The set* $\mathrm{PT}(G)$ *of* pre-translations generated by G *is the smallest set T such that (†): $\langle \boldsymbol{\ell}\theta\,,\, n\,,\, \boldsymbol{r}\theta' \rangle \in \mathrm{PT}(G)$ for all aligned rules $\chi = n \to (\boldsymbol{\ell}, \boldsymbol{r}) \in B$, all mappings $\theta, \theta' \colon \mathrm{var}(\chi) \to T_\Sigma^*$, and for all $n' \in \mathrm{var}(\chi)$*

- *$|\theta(n')| = |\mathrm{pos}_{n'}(\boldsymbol{\ell})|$ and $|\theta'(n')| = |\mathrm{pos}_{n'}(\boldsymbol{r})|$, and*
- *$\langle \theta(n'), n', \theta'(n') \rangle \in T$ is a pre-translation generated by G.*

The derivation tree corresponding to the pre-translation (†) is $\chi(d_{n_1}, \ldots, d_{n_k})$, where $\mathrm{var}(\chi) = \{n_1, \ldots, n_k\}$ with $n_1 <_N \cdots <_N n_k$ and d_n is the derivation tree corresponding to the pre-translation $\langle \theta(n), n, \theta'(n) \rangle$ for every $n \in \mathrm{var}(\chi)$.

Example 4. Recall the SFSG G of Example 2. The aligned rules $\chi_6 = (\rho_6\rho_6, \varepsilon)$ and $\chi_7 = (\rho_7, \rho_7\rho_7)$ immediately yield the pre-translations $\langle (\alpha, \alpha)\,,\, n\,,\, \varepsilon \rangle$ and $\langle \alpha\,,\, n'\,,\, (\alpha, \alpha) \rangle$ with derivation trees χ_6 and χ_7, respectively. The former pre-translation (and the pre-translations obtained) can be used with the aligned rules $\chi_2 = (\rho_2\rho_2, \varepsilon)$ and $\chi_4 = (\rho_4\rho_4, \varepsilon)$ to obtain the pre-translations

$$\begin{array}{ll} \langle (\gamma_1(\alpha), \gamma_1(\alpha))\,,\, n\,,\, \varepsilon \rangle & \text{with derivation tree } \chi_2(\chi_6), \text{ or more generally,} \\ \{ \langle (t, t)\,,\, n\,,\, \varepsilon \rangle \mid t \in T_{\{\gamma_1, \gamma_2, \alpha\}} \} & \text{with derivation trees } d \in T_{\{\chi_2, \chi_4, \chi_6\}}, \end{array}$$

where the rules χ_2 and χ_4 have rank 1 in the derivation trees. Similarly, with the help of the rules $\chi_3 = (\rho_3, \rho_3\rho_3)$ and $\chi_5 = (\rho_5, \rho_5\rho_5)$ we can obtain the pre-translations $\{ \langle (t, t)\,,\, n'\,,\, t \rangle \mid t \in T_{\{\gamma_1, \gamma_2, \alpha\}} \}$ with derivation trees $d \in T_{\{\chi_3, \chi_5, \chi_7\}}$. Plugging those pre-translations into the rule $\chi_1 = (\rho_0, \rho_1)$, we obtain

$$\{ \langle \sigma(t, u, t)\,,\, n_0\,,\, \sigma(u, \alpha, u) \rangle \mid t, u \in T_{\{\gamma_1, \gamma_2, \alpha\}} \} \subseteq \mathrm{PT}(G)$$

with derivation trees $\{ \chi_1(d_1, d_2) \mid d_1 \in T_{\{\chi_2, \chi_4, \chi_6\}}, d_2 \in T_{\{\chi_3, \chi_5, \chi_7\}} \}$. We illustrate the last step of the process in Fig. 2.

Now we are ready to define the tree translation computed by an SFSG. Intuitively all pre-translations governed by initial nonterminals are translations.

Definition 5. *Let $G = (N, \Sigma, I, R, B)$ be an SFSG. It computes the tree translation $\tau_G \subseteq T_\Sigma \times T_\Sigma$ defined by $\tau_G = \bigcup_{n \in I} \{ (t, u) \mid \langle t, n, u \rangle \in \mathrm{PT}(G) \}$. The derivation tree language $D(G)$ contains all derivation trees for the pre-translations $\langle t, n, u \rangle \in \mathrm{PT}(G)$ with $n \in I$. As usual, two SFSG are equivalent if their computed tree translations coincide. Finally, we denote the classes of tree translations computable by SFSG and MBOT by* SFSG *and* MBOT*, respectively.*

In the rest of this section, we present a normal form for MBOT, which allows us to relate our notion of MBOT to that of [4]. Moreover, we present some simple properties of SFSG. Let us start with classic MBOT [4].

Definition 6. *The MBOT (N, Σ, I, R, B) is* classic *if ℓ is linear in N and $\mathrm{var}(\boldsymbol{r}) \subseteq \mathrm{var}(\ell)$ for every $n \to (\ell, \boldsymbol{r}) \in B$.*

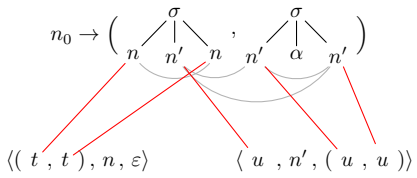

Fig. 2. Illustration of the combination of an aligned rule with pre-translations.

Proposition 7. *For every MBOT there exists an equivalent classic MBOT.*

Proof. Let $G = (N, \Sigma, I, R, B)$ be the given MBOT. We construct the MBOT $G' = (N, \Sigma, I, R, B')$ with $B' = \{n \to (\ell, \boldsymbol{r}) \in B \mid \ell \text{ linear in } N, \operatorname{var}(\boldsymbol{r}) \subseteq \operatorname{var}(\ell)\}$ that is obviously classic. It remains to prove that G and G' are equivalent. To this end, we observe that $|\boldsymbol{t}| = 1$ for all $\langle \boldsymbol{t}, n, \boldsymbol{u}\rangle \in \operatorname{PT}(G)$ due to the rule shape of G. Now, let $\chi = n \to (\ell, \boldsymbol{r}) \in B$ be a rule and $n' \in \operatorname{var}(\boldsymbol{r}) \setminus \operatorname{var}(\ell)$. To build a pre-translation of $\operatorname{PT}(G)$ with χ, we need an existing pre-translation $\langle \varepsilon, n', \boldsymbol{u}\rangle \in \operatorname{PT}(G)$ because $n' \in \operatorname{var}(\chi)$, but $n' \notin \operatorname{var}(\ell)$. Such pre-translations do not exist, hence the rule χ is useless (i.e., there are no derivation trees that contain χ), which proves that deleting it does not affect the semantics. In the same manner, rules whose left-hand side is not linear in N can be deleted (because they would require a pre-translation $\langle \boldsymbol{t}, n, \boldsymbol{u}\rangle \in \operatorname{PT}(G)$ with $|\boldsymbol{t}| \geq 2$). □

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

Definition 8. *The classic MBOT (N, Σ, I, R, B) is in* one-symbol (input) normal form *if $|\operatorname{pos}_\Sigma(\ell)| \leq 1$ for every aligned rule $n \to (\ell, \boldsymbol{r})$.*

Lemma 9 (see [4, Lemma 14]). *For every MBOT there exists an equivalent classic MBOT in one-symbol (input) normal form.*

Proof. By Proposition 7 we can construct an equivalent classic MBOT for every MBOT. With the help of [4, Lemma 14] we can then construct an equivalent MBOT in one-symbol normal form. □

Given one-symbol normal form, we can now define deterministic MBOT, which we use instead of k-morphisms [1] to avoid another concept. It should be noted that deterministic MBOT are slightly more expressive than k-morphisms.

Definition 10. *A classic MBOT (N, Σ, I, R, B) in one-symbol normal form is* deterministic *if (i) I is a singleton, (ii) $\ell \notin N$ for every $n \to (\ell, \boldsymbol{r}) \in B$, and (iii) for every $n \in N$ and $\sigma \in \Sigma$ there exists at most one aligned rule $n \to (\ell, \boldsymbol{r}) \in B$ such that $\ell(\varepsilon) = \sigma$.*

Theorem 11. *The following simple properties can easily be observed:*

1. $\mathrm{SFSG} = \mathrm{SFSG}^{-1}$.
2. *The domain* $\mathrm{dom}(\tau)$ *and the range* $\mathrm{ran}(\tau)$ *of a tree translation* $\tau \in \mathrm{SFSG}$ *are not necessarily regular.*
3. $\mathrm{MBOT} \subsetneq \mathrm{SFSG}$.

Proof. The first property is immediate because the syntactic definition of SFSG is completely symmetric. For the second property we observe that the tree translation τ_G computed by the SFSG G of Example 2 is such that both its domain and its range are not regular. Finally, the inclusion in the third item is obvious. Moreover, we know that $\mathrm{dom}(\tau)$ is regular for every $\tau \in \mathrm{MBOT}$ by Proposition 7 and [4, Theorem 25], so the tree translation τ_G is not in MBOT. □

4 Composition and Decomposition

In this section, we develop a characterization of SFSG in terms of MBOT in order to better understand the expressive power of SFSG. Since we already showed $\mathrm{MBOT} \subsetneq \mathrm{SFSG}$ in Theorem 11, we will use compositions of MBOT to characterize the expressive power of SFSG. To this end, we need a decomposition (see Theorem 12) and a composition (see Theorem 15) result.

Theorem 12 (see [17, Proposition 4.5]). *For every SFSG* G*, there exist two deterministic MBOT* G_1 *and* G_2 *such that* $\tau_G = \tau_{G_1}^{-1} ; \tau_{G_2}$.

Proof. Let $G = (N, \Sigma, I, R, B)$ be the original SFSG. Without loss of generality, we can assume that I is a singleton. Whenever we explicitly list nonterminals like $\{n_1, \ldots, n_k\}$, we assume that $n_1 <_N \cdots <_N n_k$. We construct the two MBOT $G_1 = (N, \Sigma \cup B, I, R \cup R', B')$ and $G_2 = (N, \Sigma \cup B, I, R \cup R', B'')$ with

- $R' = \{n \to \chi(n_1, \ldots, n_k) \mid \chi = n \to (\boldsymbol{\ell}, \boldsymbol{r}) \in B, \mathrm{var}(\chi) = \{n_1, \ldots, n_k\}\}$,
- $B' = \{n \to (\chi(n_1, \ldots, n_k), \boldsymbol{\ell}) \mid \chi = n \to (\boldsymbol{\ell}, \boldsymbol{r}) \in B, \mathrm{var}(\chi) = \{n_1, \ldots, n_k\}\}$, and
- $B'' = \{n \to (\chi(n_1, \ldots, n_k), \boldsymbol{r}) \mid \chi = n \to (\boldsymbol{\ell}, \boldsymbol{r}) \in B, \mathrm{var}(\chi) = \{n_1, \ldots, n_k\}\}$.

Obviously, both G_1 and G_2 are classic MBOT in one-symbol normal form, and moreover, they are deterministic. It only remains to prove that $\tau_G = \tau_{G_1}^{-1} ; \tau_{G_2}$. A straightforward induction can be used to prove that G_1 and G_2 translate derivation trees of $D(G)$ to the corresponding input and output tree, respectively. Since each derivation tree $d \in D(G)$ uniquely determines the corresponding input and the output tree, we immediately obtain the statement. A more detailed proof can be found in [17]. □

Corollary 13 (of Theorem 12). *The derivation tree language* $D(G)$ *of an SFSG* G *is regular.*

Proof. By the proof of Theorem 12, there exist classic MBOT that translate the derivation trees to the corresponding input and output tree. Moreover, by [4, Theorem 25] the domain of each MBOT is regular, which yields the result. □

Note that in the proof of Theorem 12 the rule χ uniquely determines the nonterminal n. Nevertheless, the constructed MBOT have (potentially) several nonterminals as we need to check that the behavior of the original SFSG is properly matched. In fact, it follows straightforwardly from the proof of Theorem 12 that each SFSG can be characterized by a regular derivation tree language and two deterministic MBOT mapping the derivation trees to the input and output trees. This view essentially coincides with the bimorphism approach of [1] (essentially, SFSG are equally expressive the bimorphisms of [1], in which both the input and output morphisms are allowed to be k-morphisms). We will reuse this characterization, so let us make it more explicit.

Theorem 14. SFSG $=$ d-MBOT^{-1} ; FTA ; d-MBOT*, where* d-MBOT *is the class of all tree translations computed by deterministic MBOT.*

Now we are ready to state our composition result. We first prove it using several known results on decompositions and compositions together with a few new results. However, for the reader's benefit, we will present an fully integrated construction and an example after the next theorem.

Theorem 15. MBOT^{-1} ; MBOT $\subseteq$ SFSG.

Proof. Let G_1 and G_2 be the given MBOT. By Lemma 9 we can assume without loss of generality that G_1 and G_2 are classic MBOT in one-symbol normal form. By the construction of [4, Lemma 6] applied to both G_1 and G_2 we obtain that

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

for some delabelings d_1 and d_2, regular tree languages $L_1, L_2 \in \mathrm{Reg}$, and deterministic MBOT G_1' and G_2'. Our approach is displayed in Fig. 3. Consequently,

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

Now we show that $d_1 ; d_2^{-1} = e_2^{-1} ; e_1$ for some delabelings e_1 and e_2 in the spirit of [3, Sect. II-1-4-2-1]. Let $\Sigma' = \{\underline{\sigma} \mid \sigma \in \Sigma, d_1(\sigma) = \square\}$ be the ranked alphabet containing (same-rank) copies of the elements of Σ that are erased by d_1. Similarly, let $\Sigma'' = \{\overline{\sigma} \mid \sigma \in \Sigma, d_2(\sigma) = \square\}$ contain copies of those elements that are erased by d_2. Moreover, let

$$\Sigma''' = \{\langle \sigma, \sigma' \rangle \mid \sigma, \sigma' \in \Sigma, d_1(\sigma) = d_2(\sigma') \neq \square\}$$

and $\Delta = \Sigma' \cup \Sigma'' \cup \Sigma'''$. Then we construct delabelings $e_1, e_2 \colon T_\Delta \to T_\Sigma$ as follows:

$$\begin{aligned} e_2(\underline{\sigma}) &= \sigma & e_2(\overline{\sigma}) &= \square & e_2(\langle \sigma, \sigma' \rangle) &= \sigma \\ e_1(\underline{\sigma}) &= \square & e_2(\overline{\sigma}) &= \sigma & e_2(\langle \sigma, \sigma' \rangle) &= \sigma' \end{aligned}$$

for all $\sigma, \sigma' \in \Sigma$ provided that the listed elements belong to Σ', Σ'', and Σ''', respectively. We omit the formal proof of $d_1 ; d_2^{-1} = e_2^{-1} ; e_1$, but it can be achieved by a simple induction. So far we thus obtained

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

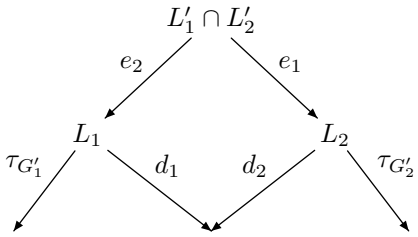

Fig. 3. Illustration of the approach used in the proof of Theorem 15.

by the exchange of the delabelings. Now let $L_1' = e_2^{-1}(L_1)$ and $L_2' = e_1^{-1}(L_2)$. Clearly, both L_1' and L_2' are regular, and also $L_1' \cap L_2'$ is regular [7,8]. Thus

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

which can be simplified to $\tau_{G_1''}^{-1} ; \mathrm{id}_{L_1' \cap L_2'} ; \tau_{G_2''}$ because we can compose the delabelings e_1 and e_2 with the deterministic MBOT G_1' and G_2' to obtain the deterministic MBOT G_1'' and G_2'', respectively, using [4, Theorem 23]. With this final step, we obtain a form suitable for Theorem 14, so $\tau_{G_1}^{-1} ; \tau_{G_2} \in \mathrm{SFSG}$. □

Corollary 16 (of Theorems 12 and 15). $\mathrm{SFSG} = \mathrm{MBOT}^{-1} ; \mathrm{MBOT}$.

As mentioned, we provide an explicit construction for the composition of an inverse MBOT with an MBOT into an SFSG. Our construction follows the general approach of translating the output of the first MBOT with the help of the second MBOT as also demonstrated in [4].

Definition 17. *Let $G_1 = (N_1, \Sigma, I_1, R_1, B_1)$ and $G_2 = (N_2, \Sigma, I_2, R_2, B_2)$ be classic MBOT such that $N_1 \cap N_2 = \emptyset$. Moreover, let $G_1' = (N_1, \Sigma, I_1, R_1)$ and $G_2' = (N_2, \Sigma, I_2, R_2)$ be the underlying regular tree grammars, respectively. We construct the composed SFSG $(G_1^{-1} ; G_2) = (N_1 \times N_2, \Sigma, I_1 \times I_2, R, B)$ such that*

- *the set R of rules is given by:*
  - *$\langle n_1, n_2 \rangle \to \langle n_1, n_2' \rangle \in R$ for every $n_1 \in N_1$ and $n_2, n_2' \in N_2$,*
  - *$\langle n_1, n_2 \rangle \to \langle n_1', n_2 \rangle \in R$ for every $n_1, n_1' \in N_1$ and $n_2 \in N_2$,*
  - *$\langle n_1, n_2 \rangle \to r(f_1)$ with $r(f_1) = r[n \leftarrow \langle n, f_1(n) \rangle \mid n \in \mathrm{var}(r)] \in R$ for every rule $\rho = n_1 \to r \in R_1$, $n_2 \in N_2$, and injection $f_1 \colon \mathrm{var}(r) \to N_2$,*
  - *$\langle n_1, n_2 \rangle \to r(f_2)$ with $r(f_2) = r[n \leftarrow \langle f_2(n), n \rangle \mid n \in \mathrm{var}(r)] \in R$ for every rule $\rho = n_2 \to r \in R_2$, $n_1 \in N_1$, and injection $f_2 \colon \mathrm{var}(r) \to N_1$,*
  - *and no further rules are in R, and*
- *the set B of aligned rules is given by:*
  - *$\langle n_1, n_2 \rangle \to (\boldsymbol{r}[n_1' \leftarrow \langle n_1', n_2 \rangle], \langle n_1', n_2 \rangle) \in B$ for every aligned rule $n_1 \to (n_1', \boldsymbol{r}) \in B_1$ with $n_1' \in N_1$ and $n_2 \in N_2$,*
  - *$\langle n_1, n_2 \rangle \to (\langle n_1, n_2' \rangle, \boldsymbol{r}[n_2' \leftarrow \langle n_1, n_2' \rangle]) \in B$ for every aligned rule $n_2 \to (n_2', \boldsymbol{r}) \in B_2$ with $n_2' \in N_2$ and $n_1 \in N_1$,*

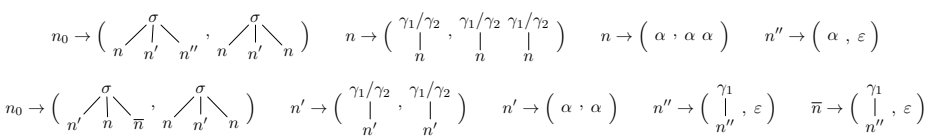

Fig. 4. Rules of the classic MBOT G_1 used in Example 18.

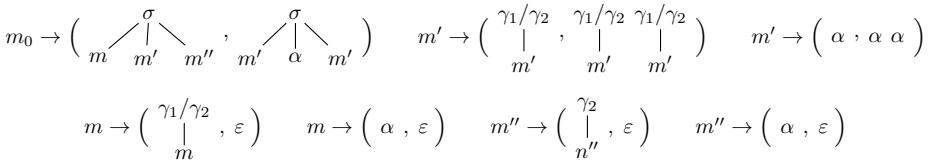

Fig. 5. Rules of the classic MBOT G_2 used in Example 18.

- $\chi = \langle n_1, n_2\rangle \to (\boldsymbol{\ell}(f_1)\,,\, \boldsymbol{r}(f_2)) \in B$ *for all aligned rules* $n_1 \to (r, \boldsymbol{\ell}) \in B_1$ *and* $n_2 \to (r', \boldsymbol{r}) \in B_2$*, and injective mappings* $f_1 \colon \mathrm{var}(r) \to N_2$ *and* $f_2 \colon \mathrm{var}(r') \to N_1$ *such that* $r(f_1) = r'(f_2)$ *and* $L(G_1')_{n_1'} \cap L(G_2')_{n_2'} \neq \emptyset$ *for all omitted nonterminals* $\langle n_1', n_2'\rangle \in \mathrm{var}(r(f_1)) \setminus \mathrm{var}(\chi)$*,*[1]
- *and no further aligned rules are in* B*.*

Let us illustrate the construction on an example.

Example 18. Let $G_1 = (N, \Sigma, \{n_0\}, R_1, B_1)$ be the classic MBOT with nonterminals $N = \{n_0, n, n', n'', \overline{n}\}$, $\Sigma = \{\alpha^{(0)}, \gamma_1^{(1)}, \gamma_2^{(1)}, \sigma^{(3)}\}$, and the rules R_1 and aligned rules B_1 that are depicted in Fig. 4. Let $G_2 = (M, \Sigma, \{m_0\}, R_2, B_2)$ be the classic MBOT with nonterminals $M = \{m_0, m, m', m''\}$ and the rules R_2 and aligned rules B_2 depicted in Fig. 5. The SFSG $G_1^{-1}\,;G_2$ is essentially the SFSG of Example 2, but we will explain the construction of two aligned rules. The aligned rule $\langle n_0, m_0\rangle \to \big(\sigma(\langle n, m\rangle, \langle n', m'\rangle, \langle n, m\rangle)\,,\, \sigma(\langle n', m'\rangle, \alpha, \langle n', m'\rangle)\big)$ is constructed from the first aligned rule of G_1 (left, top row in Fig. 4) and the first aligned rule of G_2 (left, top row in Fig. 5). During the overlay of the left-hand sides also the state $\langle n'', m''\rangle$ is created. Since the languages of n'' and m'' both contain the tree α, the previous aligned rule can be constructed. The process is illustrated in Fig. 6. However, if we want to use the left rule in the second row in Fig. 4 instead, then we can construct

$$\langle n_0, m_0\rangle \to \big(\sigma(\langle n, m'\rangle, \langle n', m\rangle, \langle n, m'\rangle)\,,\, \sigma(\langle n, m'\rangle, \alpha, \langle n, m'\rangle)\big)\ ,$$

but it is not in the composition because the state $\langle \overline{n}, m''\rangle$ combines the states $\overline{n}$ and m'', which have an empty intersection.

We conclude with some further properties of SFSG and their consequences for MBOT using our main result of Corollary 16. In particular, it is known [9] that the output string language of an MBOT is an LCFRS [20,18]. Using Corollary 16,

[1] As usual $\boldsymbol{\ell}(f_1) = \ell_1(f_1) \cdots \ell_k(f_1)$ provided that $\boldsymbol{\ell} = \ell_1 \cdots \ell_k$.

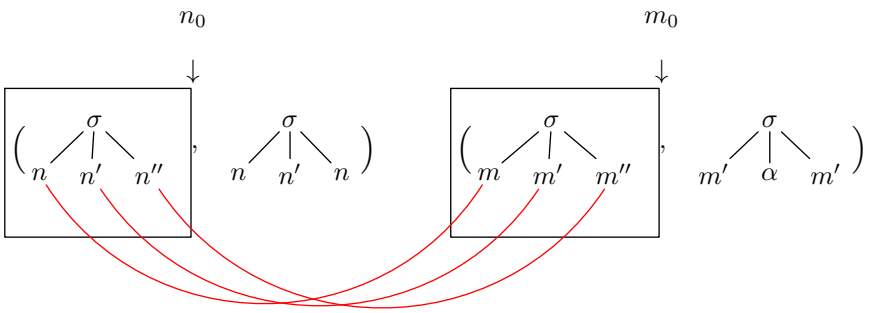

Fig. 6. Illustration of the composition construction (see Example 18). The matching happens inside the boxes and the obtained linked states are paired in the left-hand and right-hand side outside the box.

Table 1. Complexity results for a SFSG G and input strings (w_1, w_2) and trees (t_1, t_2), where $\mathrm{rk}(G)$ is the length of the longest sequence in an aligned rule of G.

problem	string level	tree level
Parsing	$\mathcal{O}(\lvert G\rvert \cdot (\lvert w_1\rvert \cdot \lvert w_2\rvert)^{2\,\mathrm{rk}(G)+2})$	$\mathcal{O}(\lvert G\rvert \cdot \lvert t_1\rvert \cdot \lvert t_2\rvert)$
Translation	$\mathcal{O}(\lvert G\rvert \cdot \lvert w_1\rvert^{2\,\mathrm{rk}(G)+2})$	$\mathcal{O}(\lvert G\rvert \cdot \lvert t_1\rvert)$

we can conclude that both the input and the output string language of an SFSG are LCFRS. Moreover, we can import several complexity results from MBOT [14] to SFSG as indicated in Table 1.

Theorem 19 (see [16, Example 5]). SFSG *is not closed under composition.*

Corollary 20. $\mathrm{MBOT}\,;\mathrm{MBOT}^{-1} \not\subseteq \mathrm{SFSG}$.

Proof. Let us assume that (†): $\mathrm{MBOT}\,;\mathrm{MBOT}^{-1} \subseteq \mathrm{SFSG}$. Then

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

using Corollary 16, (†), Corollary 16, the closure under composition for MBOT [4, Theorem 23], and Corollary 16 once more. However, the result contradicts Theorem 19, thus (†) is false, proving the result. □

References

1. Arnold, A., Dauchet, M.: Morphismes et bimorphismes d'arbres. Theor. Comput. Sci. 20(1), 33–93 (1982)

2. Chiang, D.: An introduction to synchronous grammars. In: Proc. 44th ACL. Association for Computational Linguistics (2006), part of a tutorial given with K. Knight
3. Dauchet, M.: Transductions de forêts — Bimorphismes de magmoïdes. Première thèse, Université de Lille (1977)
4. Engelfriet, J., Lilin, E., Maletti, A.: Composition and decomposition of extended multi bottom-up tree transducers. Acta Inf. 46(8), 561–590 (2009)
5. Fülöp, Z., Kühnemann, A., Vogler, H.: A bottom-up characterization of deterministic top-down tree transducers with regular look-ahead. Inf. Process. Lett. 91(2), 57–67 (2004)
6. Fülöp, Z., Kühnemann, A., Vogler, H.: Linear deterministic multi bottom-up tree transducers. Theor. Comput. Sci. 347(1–2), 276–287 (2005)
7. Gécseg, F., Steinby, M.: Tree Automata. Akadémiai Kiadó, Budapest, Hungary (1984)
8. Gécseg, F., Steinby, M.: Tree languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, chap. 1, pp. 1–68. Springer (1997)
9. Gildea, D.: On the string translations produced by multi bottom-up tree transducers. Computational Linguistics 38(3), 673–693 (2012)
10. Knight, K., Graehl, J.: An overview of probabilistic tree transducers for natural language processing. In: Proc. 6th CICLing. LNCS, vol. 3406, pp. 1–24. Springer (2005)
11. Lilin, E.: Propriétés de clôture d'une extension de transducteurs d'arbres déterministes. In: Proc. 6th CAAP. LNCS, vol. 112, pp. 280–289. Springer (1981)
12. Maletti, A.: Compositions of extended top-down tree transducers. Inform. and Comput. 206(9–10), 1187–1196 (2008)
13. Maletti, A.: Why synchronous tree substitution grammars? In: Proc. HLT-NAACL. pp. 876–884. Association for Computational Linguistics (2010)
14. Maletti, A.: An alternative to synchronous tree substitution grammars. J. Nat. Lang. Engrg. 17(2), 221–242 (2011)
15. Maletti, A.: How to train your multi bottom-up tree transducer. In: Proc. 49th ACL. pp. 825–834. Association for Computational Linguistics (2011)
16. Radmacher, F.G.: An automata theoretic approach to rational tree relations. In: Proc. 34th SOFSEM. LNCS, vol. 4910, pp. 424–435. Springer (2008)
17. Raoult, J.C.: Rational tree relations. Bull. Belg. Math. Soc. Simon Stevin 4(1), 149–176 (1997)
18. Seki, H., Matsumura, T., Fujii, M., Kasami, T.: On multiple context-free grammars. Theor. Comput. Sci. 88(2), 191–229 (1991)
19. Sun, J., Zhang, M., Tan, C.L.: A non-contiguous tree sequence alignment-based model for statistical machine translation. In: Proc. 47th ACL. pp. 914–922. Association for Computational Linguistics (2009)
20. Vijay-Shanker, K., Weir, D.J., Joshi, A.K.: Characterizing structural descriptions produced by various grammatical formalisms. In: Proc. 25th ACL. pp. 104–111. Association for Computational Linguistics (1987)
21. Zhang, M., Jiang, H., Aw, A., Li, H., Tan, C.L., Li, S.: A tree sequence alignment-based tree-to-tree translation model. In: Proc. 46th ACL. pp. 559–567. Association for Computational Linguistics (2008)
22. Zhang, M., Jiang, H., Li, H., Aw, A., Li, S.: Grammar comparison study for translational equivalence modeling and statistical machine translation. In: Proc. 22nd CoLing. pp. 1097–1104. Association for Computational Linguistics (2008)