International Journal of Foundations of Computer Science

Compositions of Tree-to-Tree Statistical Machine Translation Models*

Andreas Maletti†

Universität Leipzig, Institute of Computer Science
PO box 100 920, 04009 Leipzig, Germany
maletti@informatik.uni-leipzig.de

Received (23 January 2017)
Revised (3 March 2018)
Accepted (21 February 2018)
Communicated by (Alexandre Blondin Massé and Srečko Brlek and Christophe Reutenauer)

The well-known synchronous context-free grammars (SCFGs) and synchronous tree-substitution grammars (STSGs), both of which are used as tree-to-tree translation models in statistical machine translation are investigated. Their composition hierarchies are established in both the unweighted as well as the weighted case. More precisely, it is shown that SCFGs are closed under composition in both cases and that there is a close connection between compositions of STSGs and compositions of certain tree transducers. With the help of the close ties, the composition closure of STSGs is identified in both cases as well. The results for the weighted case utilize a new lifting technique that might prove useful also in similar setups.

1. Introduction

Several different translation models are nowadays used in syntax-based statistical machine translation [16]. The translation model is the main component responsible for the transformation of the input into the translated output, and thus the expressive power of the translation model limits the possible translations. For example, the framework 'Moses' [17] provides implementations of synchronous context-free grammars (SCFGs) [1] and several variants of synchronous tree-substitution grammars (STSGs) [7]. The expressive power of SCFGs and STSGs is reasonably well-understood, and in particular, knowledge of the limitations of the models has helped many authors to pre-process [27, 6, 19] or post-process [5, 26] their data and to achieve better translation results. Together with pre- or post-processing steps,

*This article is an extended and revised version of [Maletti: *Compositions of Tree-to-Tree Statistical Machine Translation Models.* In Proc. Developments in Language Theory, LNCS 9840, pages 293–305, 2016].

†The author gratefully acknowledges the financial support of the DFG graduate research training group 1763: "Quantitative Logics and Automata".

the translation model is no longer solely responsible for the transformation process, but we rather obtain a composition of several models or simply a composition chain [22]. Occasionally, composition chains also appear because they ideally support a modular development of components for specific translation tasks [4] (e.g., translating numerals or geographic locations). However, it is often difficult to evaluate such composition chains efficiently especially when the pre- or post-processing steps are not realized by functions (i.e., are nondeterministic).

In the string-to-string setting the phrase-based models are essentially finite-state transducers and chains of them can be collapsed into a single transducer [23] because their translations are closed under composition. However, this is not true for several tree-to-tree models. Efficient on-the-fly evaluations for composition chains are presented in [22] along with the observation that the straightforward sequential evaluation of composition chains is terribly inefficient. Even in the on-the-fly evaluation the chains should be as short as possible. In this contribution, we will investigate the expressive power of composition chains of the established tree-to-tree translation models. The symmetric tree-to-tree setting, although typically worse in terms of translation quality than the string-to-tree or tree-to-string setting [24, 25], is particularly convenient since it allows a clean notion of composition.

We first demonstrate that (unweighted and weighted) composition chains of SCFGs can always be reduced to just a single SCFG. In addition, we demonstrate how to utilize results for unweighted extended tree transducers [21] to obtain results for STSGs. The main insight in this part is that even local models like STSGs obtain a finite-state behavior in composition chains. Thus, a composition of two STSGs is as powerful as a composition of two corresponding tree transducers. This close connection allows us to show that two STSGs are necessary and sufficient for arbitrary composition chains of certain simple, yet commonly used STSGs. These results hold in the absence of weights. However, all translation models used in statistical machine translation are weighted, so as a second contribution we demonstrate how to lift the unweighted results into the weighted setting. Our novel lifting procedure, which we believe will be useful also in other setups, relies on a separation of the weights and several normalization procedures. Overall, we achieve the same results on the power of compositions also in the weighted setting, which essentially shows that short chains of certain STSGs suffice.

2. Preliminaries

We use $\mathbb{N}$ for the set of nonnegative integers (including 0), and set $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$. For every $k \in \mathbb{N}$, we let $[k] = \{i \in \mathbb{N}_+ \mid 1 \leq i \leq k\}$. For a set A and $k \in \mathbb{N}$, we let $A^k = A \times \cdots \times A$ containing k factors A. Note that $A^0 = \{()\}$, and we also write ε for $()$. Moreover, we let $A^* = \bigcup_{k \in \mathbb{N}} A^k$ be the set of all finite sequences over A. The concatenation of sequences $v, w \in A^*$ is denoted by $v.w$ or simply by the juxtaposition vw. For concatenations of the same sequence $w \in A^*$, we use exponents as usual, so $w^2 = ww$. For every $w \in A^*$, the length $|w|$ of w is the unique $k \in \mathbb{N}$ such

that $w \in A^k$. Given a relation $\tau \subseteq A \times B$, we let $\mathrm{dom}(\tau) = \{a \in A \mid (a,b) \in \tau\}$ and $\mathrm{ran}(\tau) = \{b \in B \mid (a,b) \in \tau\}$. Given another $\nu \subseteq B \times C$, their composition $\tau\,;\nu$ is given by $\tau\,;\nu = \{(a,c) \in A \times C \mid \exists b \in B\colon (a,b) \in \tau,\ (b,c) \in \nu\}$.

For finite sets Σ and V, the set $T_\Sigma(V)$ of Σ-*trees indexed by* V is the smallest set T such that $V \subseteq T$ and $\sigma(t_1,\dots,t_k) \in T$ for all $k \in \mathbb{N}$, $\sigma \in \Sigma$, and $t_1,\dots,t_k \in T$. We simply write T_Σ for $T_\Sigma(\emptyset)$, and for all $k \in \mathbb{N}$, $\gamma \in \Sigma$, and $t \in T_\Sigma(V)$ we write $\gamma^k(t)$ for the tree $\gamma(\cdots\gamma(\gamma(t))\cdots)$ containing k times the symbol γ atop t. We will often omit the quantification and simply write '$t = \sigma(t_1,\dots,t_k)$' to express that $t = \sigma(t_1,\dots,t_k)$ for some $k \in \mathbb{N}$, $\sigma \in \Sigma$, and $t_1,\dots,t_k \in T_\Sigma(V)$. Given $t \in T_\Sigma(V)$, its *positions* $\mathrm{pos}(t)$ are inductively defined by

$$\mathrm{pos}(t) = \begin{cases} \{\varepsilon\} & \text{if } t \in V \\ \{\varepsilon\} \cup \{i.p \mid i \in [k],\ p \in \mathrm{pos}(t_i)\} & \text{if } t = \sigma(t_1,\dots,t_k)\ . \end{cases}$$

The positions are totally ordered by the lexicographic order $\sqsubseteq$ on $\mathbb{N}^*$. For every tree $t \in T_\Sigma(V)$, its size $|t|$ is defined to be $|t| = |\mathrm{pos}(t)|$, and the tree t is *shallow* if $\{1\} \subseteq \mathrm{pos}(t) \subseteq \{\varepsilon\} \cup \mathbb{N}_+$. Moreover, for every $p \in \mathrm{pos}(t)$, the label of t at p is $t(p)$ and inductively defined by

$$t(p) = \begin{cases} t & \text{if } t \in V \text{ and } p = \varepsilon \\ \sigma & \text{if } t = \sigma(t_1,\dots,t_k) \text{ and } p = \varepsilon \\ t_i(p') & \text{if } t = \sigma(t_1,\dots,t_k) \text{ and } p = i.p' \text{ for some } i \in \mathbb{N} \text{ and } p' \in \mathrm{pos}(t_i)\ . \end{cases}$$

For a subset $Q \subseteq V$ we let $\mathrm{leaves}_Q(t) = \{p \in \mathrm{pos}(t) \mid p.1 \notin \mathrm{pos}(t),\ t(p) \in Q\}$ be the Q-labeled leaf positions. Given a position $p \in \mathrm{pos}(t)$ and another tree $u \in T_\Sigma(V)$, we let $t[u]_p$ be the tree obtained from t by replacing the subtree at position p by u. Formally,

$$t[u]_\varepsilon = u$$
$$\sigma(t_1,\dots,t_k)[u]_{i.p'} = \sigma(t_1,\dots,t_{i-1},t_i[u]_{p'},t_{i+1},\dots,t_k)$$

for all $k \in \mathbb{N}$, $\sigma \in \Sigma$, $t_1,\dots,t_k \in T_\Sigma(V)$, $i \in [k]$, and $p' \in \mathrm{pos}(t_i)$. These notations are illustrated in Figure 1, and we refer to [12] for an in-depth exposition.

Our weights will be taken from *commutative semirings* [15, 13], which are algebraic structures $(C,+,\cdot,0,1)$ such that $(C,+,0)$ and $(C,\cdot,1)$ are commutative monoids and $(\sum_{i=1}^k c_i)\cdot c = \sum_{i=1}^k (c_i \cdot c)$ for all $k \in \mathbb{N}$ and $c, c_1,\dots,c_k \in C$. Typical examples of such semirings include the Boolean semiring $\mathbb{B} = (\{0,1\},\max,\min,0,1)$, the Viterbi semiring $([0,1],\max,\cdot,0,1)$ on the unit interval $[0,1]$, and the semiring $(\mathbb{Q},+,\cdot,0,1)$ of rational numbers. In the following, let $\mathcal{C} = (C,+,\cdot,0,1)$ be an arbitrary commutative semiring.

A *weighted (linear, nondeleting extended top-down) tree transducer* [14, 9] is a tuple $T = (Q,\Sigma,Q_0,R,\mathrm{wt})$ consisting of (i) a finite set Q of *states*, (ii) a finite set Σ of labels for the trees generated, (iii) designated *initial states* $Q_0 \subseteq Q \times Q$, (iv) a finite set R of *rules* of the form $(q,t) \xrightarrow{\varphi} (q',t')$ consisting of states $q, q' \in Q$,

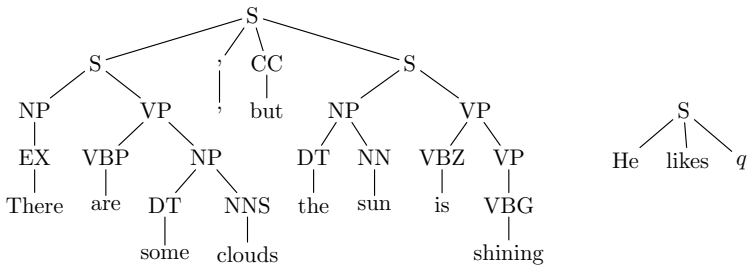

Figure 1. The left tree t is not shallow, whereas the right tree u is. If $Q = \{q\}$, then $\mathrm{leaves}_Q(t) = \emptyset$ and $\mathrm{leaves}_Q(u) = \{3\}$. Obviously, its label is $u(3) = q$. Moreover, $u[\mathrm{her}]_3 = \mathrm{S}(\mathrm{He}, \mathrm{likes}, \mathrm{her})$. Note that the q-labeled node vanishes in the replacement.

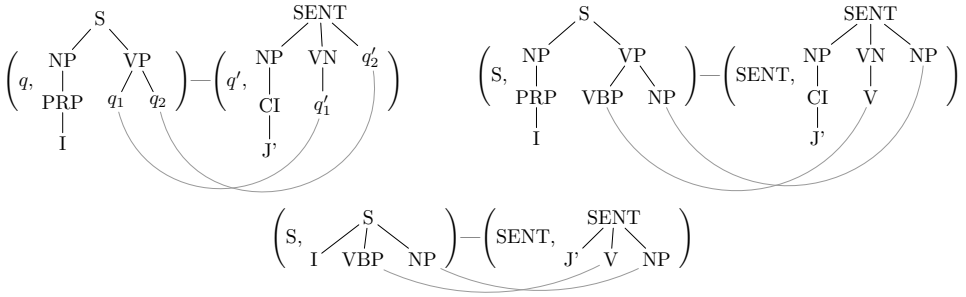

Figure 2. Example rules of a tree transducer [top left], an STSG [top right], and an SCFG [bottom].

input and output tree fragments $t, t' \in T_\Sigma(Q)$, and a bijective *alignment* (partial) mapping $\varphi\colon \mathrm{leaves}_Q(t) \dashrightarrow \mathrm{leaves}_Q(t')$ such that $\varphi \neq \{(\varepsilon, \varepsilon)\}$, and (v) a *rule weight* assignment $\mathrm{wt}\colon R \to C$. The transducer T is a *synchronous tree-substitution grammar* (STSG) [7] if $Q = \Sigma$ and $t(\varepsilon) = q$ and $t'(\varepsilon) = q'$ in each rule $(q, t) \xrightarrow{\varphi} (q', t') \in R$; i.e., the root labels of t and t' are q and q', respectively. Finally, T is a *synchronous context-free grammar* (SCFG) [1] if it is an STSG and in each rule $(q, t) \xrightarrow{\varphi} (q', t') \in R$ the trees t and t' are shallow. In an SCFG, the input and the output tree are assembled like derivation trees of a context-free grammar (i.e., one level at a time). We recall two restrictions on tree transducer rules. A rule $(q, t) \xrightarrow{\varphi} (q', t')$ is an *ε-rule* if $\varepsilon \in \mathrm{dom}(\varphi)$. Similarly, it is *non-strict* if $\varepsilon \in \mathrm{ran}(\varphi)$. The tree transducer T is *ε-free* if it does not contain any ε-rules in R, and it is *strict* provided that it has no non-strict rules in R. Finally, *simple* tree transducers are both ε-free and strict. Note that an SCFG is always simple. We show a few example rules of each type in Figure 2.

Next, we recall the derivation tree semantics [10] of a weighted tree transducer $T = (Q, \Sigma, Q_0, R, \mathrm{wt})$. For all $q, q' \in Q$, we define the set $D_T(q, q')$ of (q, q')-rooted derivation trees by simultaneous induction to be the smallest sets $D(q, q') \subseteq T_R$ such that $\rho(d_1, \ldots, d_k) \in D(q, q')$ for every rule $\rho = (q, t) \xrightarrow{\varphi} (q', t') \in R$ and $d_i \in D(q_i, q'_i)$

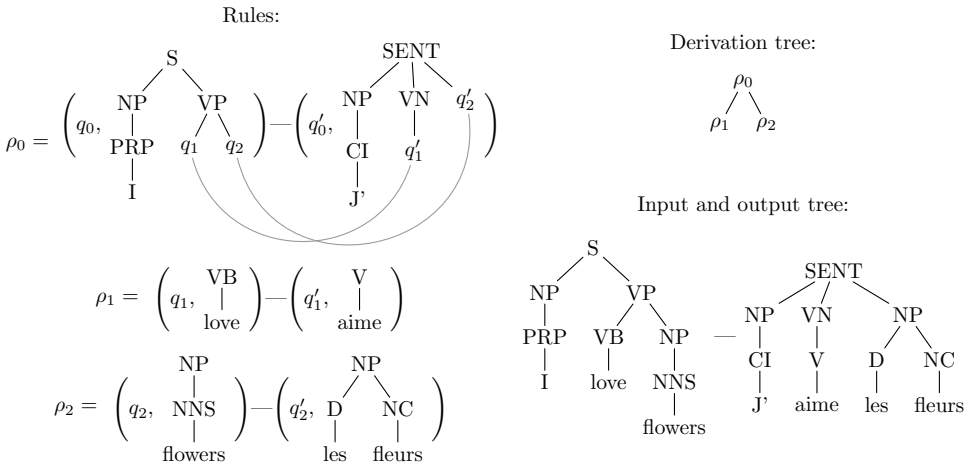

Figure 3. Illustration of a derivation tree and its input and output tree.

for all $i \in [k]$ such that $\{w_1, \ldots, w_k\} = \mathrm{dom}(\varphi)$ with $w_1 \sqsubset \cdots \sqsubset w_k$ and $t(w_i) = q_i$ and $t'(\varphi(w_i)) = q'_i$ for all $i \in [k]$. In other words, derivation trees are trees of rules that respect the state behavior. Similarly the input tree $\mathrm{in}(d) \in T_\Sigma(Q)$, the output tree $\mathrm{out}(d) \in T_\Sigma(Q)$, and the weight $\mathrm{wt}(d)$ of a derivation tree $d \in D_T(q, q')$ are defined inductively. Given a (q, q')-rooted derivation tree $d = \rho(d_1, \ldots, d_k)$ as above we let

$$\begin{aligned} \mathrm{in}(d) &= (\cdots(t[\mathrm{in}(d_1)]_{w_1})\cdots)[\mathrm{in}(d_k)]_{w_k} \\ \mathrm{out}(d) &= (\cdots(t'[\mathrm{out}(d_1)]_{\varphi(w_1)})\cdots)[\mathrm{out}(d_k)]_{\varphi(w_k)} \\ \mathrm{wt}(d) &= \mathrm{wt}(\rho) \cdot \prod_{i \in [k]} \mathrm{wt}(d_i) \ . \end{aligned}$$

We illustrate derivation trees and the associated notions in Figure 3. Additionally, we note that given trees $t, t' \in T_\Sigma$ and $q, q' \in Q$, there are only finitely many $d \in D_T(q, q')$ such that $\mathrm{in}(d) = t$ and $\mathrm{out}(d) = t'$ because each rule contributes to the input or output tree. For all trees $t, t' \in T_\Sigma$, we hence can define the weight assigned by T to (t, t') to be

$$T(t, t') = \sum_{(q,q') \in Q_0} \sum_{\substack{d \in D_T(q,q') \\ \mathrm{in}(d)=t,\, \mathrm{out}(d)=t'}} \mathrm{wt}(d) \ .$$

In this manner, the transducer T computes a mapping $T \colon T_\Sigma \times T_\Sigma \to C$. We use $\mathrm{SCFG}(\mathcal{C})$, $\mathrm{STSG}(\mathcal{C})$, and $\mathrm{TT}(\mathcal{C})$ to denote the classes of mappings computed by SCFGs, STSGs, and tree transducers over the commutative semiring $\mathcal{C}$. It is evident from the formal definitions that each SCFG is a special STSG, which in turn is a special tree transducer. Hence, the expressive power necessarily increases from SCFGs to STSGs to tree transducers (TTs); i.e., $\mathrm{SCFG}(\mathcal{C}) \subseteq \mathrm{STSG}(\mathcal{C}) \subseteq \mathrm{TT}(\mathcal{C})$.

Finally, let us formally introduce compositions of weighted tree translations. For each alphabet Σ, a mapping $\tau\colon T_\Sigma \times T_\Sigma \to C$ is *finitary*, if for every $t \in T_\Sigma$ there exist only finitely many $u \in T_\Sigma$ such that $\tau(t,u) \neq 0$. Similarly, it is *co-finitary*, if for every $u \in T_\Sigma$ there exist only finitely many $t \in T_\Sigma$ such that $\tau(t,u) \neq 0$. Now let $\tau, \tau'\colon T_\Sigma \times T_\Sigma \to C$ be such that τ is finitary or τ' is co-finitary. Then the composition τ followed by τ', written $\tau\,;\tau'$, is defined for all $t, v \in T_\Sigma$ by

$$(\tau\,;\tau')(t,v) = \sum_{u \in T_\Sigma} \tau(t,u) \cdot \tau'(u,v) \ .$$

Note that this sum is well-defined because of the finitary or co-finitary restriction, which yields that only finitely many choices of u yield non-zero products. Roughly speaking, we sum over all potential intermediate trees u and take the product of the weights for the translation from t to u and the translation from u to v, which shows that composition corresponds to executing the second transducer on the output of the first transducer. Composition extends to classes $\mathcal{K}$ of weighted translations in the usual manner, and we use $\mathcal{K}^n$ for the composition $\mathcal{K};\cdots;\mathcal{K}$ containing the class $\mathcal{K}$ exactly n times. Finally, the inverse of a weighted tree translation $\tau\colon T_\Sigma \times T_\Sigma \to C$ is $\tau^{-1}\colon T_\Sigma \times T_\Sigma \to C$ given by $\tau^{-1}(t,u) = \tau(u,t)$ for all $t, u \in T_\Sigma$. We note that, as usual, $(\tau\,;\tau')^{-1} = (\tau')^{-1}\,;\tau^{-1}$.

3. Classification of unweighted composition closures

We start with the unweighted case, in which the Boolean semiring $\mathbb{B}$ is the weight structure. However, several results in this section will immediately be proved in the weighted setting. Weighted tree-to-tree translations are essentially relations on trees in the unweighted case, and we omit the rule weight assignment from the specification of tree transducers. The key property that separates unweighted SCFGs and STSGs was already identified in [7], where it was observed that the relations computed by SCFGs only contain pairs of isomorphic trees (in the graph-theoretic sense disregarding the labels and the order of the children, so in particular the input and output tree always have the same size). The separation between unweighted STSGs and simple tree transducers can easily be achieved by the following relation $\tau = \big\{\big(\gamma^i(\delta(\gamma^j(\alpha))), \gamma^{i+j+1}(\alpha)\big) \mid i \in \mathbb{N}_+, j \in \mathbb{N}\big\}$, which can easily be computed by a simple tree transducer, but cannot be computed by an STSG. More precisely, the simple tree transducer $(\{q_0, q_1, q_2\}, \Sigma, \{(q_0, q_0)\}, R)$ with the rules

$$\big(q_0, \gamma(q_0)\big) \overset{\{(1,1)\}}{\longrightarrow} \big(q_0, \gamma(q_0)\big) \qquad \big(q_1, \delta(q_2)\big) \overset{\{(1,1)\}}{\longrightarrow} \big(q_1, \gamma(q_2)\big) \qquad \big(q_2, \alpha\big) \overset{\emptyset}{\longrightarrow} \big(q_2, \alpha\big)$$
$$\big(q_0, \gamma(q_1)\big) \overset{\{(1,1)\}}{\longrightarrow} \big(q_0, \gamma(q_1)\big) \qquad \big(q_2, \gamma(q_2)\big) \overset{\{(1,1)\}}{\longrightarrow} \big(q_2, \gamma(q_2)\big)$$

obviously computes τ. To prove that τ cannot be computed by an STSG, assume that an STSG $G = (\Sigma, \Sigma, \Sigma_0, R)$ computes τ, and let $n \in \mathbb{N}$ with $n > \max(|t|, |t'|)$ for every rule $(\sigma, t) \overset{\varphi}{\longrightarrow} (\sigma', t') \in R$. Since $(\gamma^{3n}(\delta(\gamma^{3n}(\alpha))), \gamma^{6n+1}(\alpha)) \in \tau$, we must have a derivation tree d for this pair of input and output trees. This derivation tree must have a subtree of the form $d' = \rho_3(\rho_2(\rho_1))$ with $\rho_1, \rho_2, \rho_3 \in R$. We

Table 1. Known results on unweighted composition closures.

Model	Composition closure	Reference
top-down tree transducer	1	Theorem 1 of [3]
simple tree transducer	2	Théorème 6.2 of [2]
other tree transducer	∞	Theorem 45 of [8]

immediately observe that $\mathrm{in}(d') = \gamma^j(\alpha)$ and $\mathrm{out}(d') = \gamma^\ell(\alpha)$ for some $j, \ell \in \mathbb{N}$. We can now distinguish several cases. If $j, \ell \in \mathbb{N}_+$, then $d' \in D_G(\gamma, \gamma)$ and hence $(\gamma^j(\alpha), \gamma^\ell(\alpha)) \in \tau$, which is a contradiction. If $j = 0$ (and $\ell \neq 0$), then we can replace d' by just $\rho_2(\rho_1)$ in d. Since ρ_3 must contain at least one additional occurrence of γ besides the root, the newly obtained derivation tree generates the same input tree, but the output tree γ^{6n+1-m} for some $m \geq 1$, which is again a contradiction. The case of $\ell = 0$ is analogous. Consequently, we obtain the strict hierarchies

$$\mathrm{SCFG}(\mathbb{B}) \subset \mathrm{STSG}(\mathbb{B}) \subset \mathrm{TT}(\mathbb{B}) \quad \text{and} \quad \mathrm{SCFG}(\mathbb{B}) \subset \text{s-STSG}(\mathbb{B}) \subset \text{s-TT}(\mathbb{B}) \ . \quad (1)$$

Composition essentially corresponds to running two translations consecutively, where the first translation translates the input into intermediate results and the second translation translates those intermediate results into the final results. Compositions of tree translations have been extensively investigated (see [11] for a survey). All classes $\mathcal{K}$ of translations discussed here contain the identity relation, so compositions of our classes $\mathcal{K}$ form a natural hierarchy; i.e., $\mathcal{K} \subseteq \mathcal{K}^2 \subseteq \mathcal{K}^3 \subseteq \cdots$. Such a hierarchy collapses at level n if $\mathcal{K}^n = \mathcal{K}^{n+1}$. We also say that the composition closure is obtained at level n provided that n is the least integer, for which the hierarchy collapses. Intuitively, if the closure is obtained at level n, then compositions of n translations of $\mathcal{K}$ are necessary and sufficient to generate any translation computable by any composition of translations from $\mathcal{K}$. Provided that the composition closure for $\mathcal{K}$ is n, we thus have $\mathcal{K} \subset \cdots \subset \mathcal{K}^n = \mathcal{K}^{n+1} = \cdots$. We use ∞ to indicate that the hierarchy never collapses. We summarize the known results [3, 2, 8] on the composition closure in Table 1.

We start our investigation with SCFGs. Given two SCFGs T and T' we can simply "join" rules of T and T' that coincide on the intermediate tree. We illustrate this approach in Figure 4. Such rules can certainly be executed consecutively in the on-the-fly approach [22]. A refined version of this approach taking the finite-state information and the non-shallow output into account is used to prove that (our linear and nondeleting) top-down tree transducers are closed under composition [3].

Theorem 1. *The composition closure of* $\mathrm{SCFG}(\mathcal{C})$ *is achieved at the first level.*

Proof. Given two SCFGs $T = (\Sigma, \Sigma, \Sigma_0, R, \mathrm{wt})$ and $T' = (\Sigma, \Sigma, \Sigma'_0, R', \mathrm{wt}')$, we construct the SCFG $T'' = (\Sigma, \Sigma, \Sigma_0 \,;\, \Sigma'_0, R'', \mathrm{wt}'')$, in which the rules R'' and their weights wt'' are defined as follows: For every rule $\rho = (\sigma, s) \xrightarrow{\varphi} (\delta, t)$ of R and rule $\rho' = (\delta, t) \xrightarrow{\psi} (\gamma, u)$ of R' such that $\mathrm{ran}(\varphi) = \mathrm{dom}(\psi)$ we construct the rule

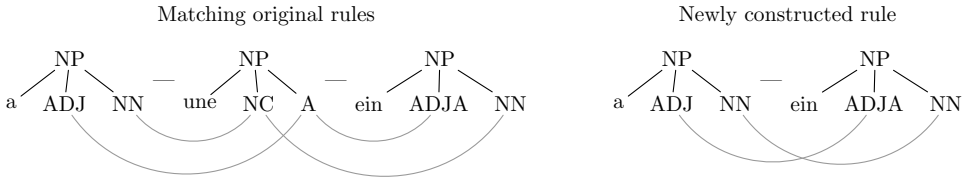

Figure 4. Rule matching and joining in the composition of SCFGs. The left and middle part form a rule of T and the middle and right part form a rule of T'. The newly constructed rule will simply avoid the intermediate tree fragment.

$(\rho\,;\rho') = (\sigma, s) \overset{\varphi;\psi}{\text{---}} (\gamma, u)$ of R'' and set $\mathrm{wt}''(\rho\,;\rho') = \mathrm{wt}(\rho) \cdot \mathrm{wt}'(\rho')$. No other rules are in R''.

To prove correctness, we define the function $f\colon T_R \times T_{R'} \to T_{R''}$ inductively for all rules ρ and ρ' as above and $d_1, \ldots, d_k \in T_R$ and $d'_1, \ldots, d'_k \in T_{R'}$ by

$$f(\rho(d_1, \ldots, d_k), \rho'(d'_1, \ldots, d'_k)) = (\rho\,;\rho')(f(d_1, d'_{\pi(1)}), \ldots, f(d_k, d'_{\pi(k)}))\ ,$$

where $\pi\colon [k] \to [k]$ is the permutation that φ realizes on the $\sqsubseteq$-ordered positions (e.g., in Figure 4 the permutation would be $\pi(1) = 2$ and $\pi(2) = 1$ because the first "nonterminal" ADJ is linked to the second "nonterminal" A in the intermediate tree). Otherwise, f is undefined. It is straightforward to show that for all $q, q'' \in \Sigma$, the partial function f is a bijection between $\{(d, d') \in D_T(q, q') \times D_{T'}(q', q'') \mid \mathrm{out}(d) = \mathrm{in}(d')\}$ and $D_{T''}(q, q'')$ that additionally also preserves input- and output trees and weights in the sense that $\mathrm{in}(d) = \mathrm{in}(f(d, d'))$, $\mathrm{out}(d') = \mathrm{out}(f(d, d'))$, and $\mathrm{wt}(d) \cdot \mathrm{wt}(d') = \mathrm{wt}(f(d, d'))$. Hence $T\,;T' = T''$. □

Next, we will show that the composition closure for simple STSGs can be obtained from the results for simple tree transducers via a small insight. Recall that SCFGs and STSGs are both local, so they are missing the finite-state behavior of tree transducers. However, we can simulate the finite-state behavior for both models in compositions with the help of the intermediate trees. Namely, we can annotate the desired finite-state information on the intermediate trees in the spirit of the representation of a regular tree language as the image of a local tree language under a relabeling [12]. We illustrate the approach in Figure 5. Note that the first STSG encodes the states in its output (i.e., the intermediate tree), whereas the second STSG would encode them in its input (i.e., also the intermediate tree). We use the prefix 's-' to restrict to simple STSG or tree transducers, so 's-TT($\mathcal{C}$)' denotes the class of tree translations computed by simple tree transducers. We immediately prove the next lemma also in the general weighted setting.

Lemma 2.

$$\text{s-STSG}(\mathcal{C})\,;\text{s-TT}(\mathcal{C}) = \text{s-TT}(\mathcal{C})\,;\text{s-TT}(\mathcal{C}) \tag{i}$$

$$\text{s-STSG}(\mathcal{C})\,;\text{s-STSG}(\mathcal{C}) = \text{s-TT}(\mathcal{C})\,;\text{s-STSG}(\mathcal{C})\ . \tag{ii}$$

Proof. Clearly, the left-to-right inclusions are trivial. For the converse inclusion in (i), let $T = (Q, \Sigma, Q_0, R, \mathrm{wt})$ and $T' = (Q', \Sigma, Q'_0, R', \mathrm{wt}')$ be simple tree transducers. We construct the STSG $U = (\Sigma', \Sigma', \Sigma'_0, R_U, \mathrm{wt}_U)$ and the tree transducer $U' = (Q', \Sigma', Q'_0, R'', \mathrm{wt}'')$ such that $\Sigma' = \Sigma \cup (\Sigma \times Q \times Q)$, which allows us to use combinations of internal symbols together with two states. For every rule $\rho = (q, t) \xrightarrow{\varphi} (q', t') \in R$ of the first simple tree transducer T and all mappings $a, b\colon \mathrm{leaves}_Q(t) \to \Sigma$, we construct the rule

$$\rho_{a,b} = \big(t(\varepsilon), t[a(w_1)]_{w_1} \dots [a(w_k)]_{w_k}\big) \xrightarrow{\varphi} \big(\langle \sigma, q, q' \rangle, \langle \sigma, q, q' \rangle(u'_1, \dots, u'_k)\big) \in R_U \ ,$$

where $u'_i = t'_i[\langle b(w_1), t(w_1), t'(\varphi(w_1))\rangle]_{\varphi(w_1)} \cdots [\langle b(w_k), t(w_k), t'(\varphi(w_k))\rangle]_{\varphi(w_k)}$ for every $i \in [k]$, we have $t(\varepsilon) \in \Sigma$ due to the ε-freeness, $t' = \sigma(t'_1, \dots, t'_k)$ for some internal symbol $\sigma \in \Sigma$ and subtrees $t'_1, \dots, t'_k$ due to strictness, and $\mathrm{leaves}_Q(t) = \{w_1, \dots, w_k\}$. Moreover, we set $\mathrm{wt}_U(\rho_{a,b}) = \mathrm{wt}(\rho)$. In other words, we guess the internal symbols that will replace a state leaf in the input and output fragment t and t' and replace the state leaf by the guessed internal symbol in the input fragment and the triple containing the guessed internal symbol and the two synchronized states. No other rules are in R_U. Finally, we set $\Sigma'_0 = \{(\sigma, \langle \delta, q, q' \rangle) \mid (q, q') \in Q_0, \sigma, \delta \in \Sigma\}$. In preparation for the definition of U', we let $\imath\colon \Sigma' \to \Sigma$ be such that $\imath(\sigma) = \sigma$ and $\imath(\langle \sigma, q, q' \rangle) = \sigma$ for all $\sigma \in \Sigma$ and $q, q' \in Q$. We extend $\imath$ to a mapping $\imath\colon T_{\Sigma'}(Q') \to T_\Sigma(Q')$ by $\imath(q') = q'$ for every $q' \in Q'$ and $\imath(\sigma'(t_1, \dots, t_k)) = \imath(\sigma')(\imath(t_1), \dots, \imath(t_k))$ for all $k \in \mathbb{N}$, $\sigma' \in \Sigma'$, and $t_1, \dots, t_k \in T_{\Sigma'}(Q')$. For every rule $\rho' = (q, t) \xrightarrow{\varphi} (q', t') \in R'$ of the second tree transducer, we have that $\rho'_u = (q, u) \xrightarrow{\varphi} (q', t') \in R''$ for every $u \in \imath^{-1}(t)$. Moreover, $\mathrm{wt}''(\rho'_u) = \mathrm{wt}'(\rho')$. No other rules are in R''. In other words, the second tree transducer simply ignores the additional annotation of the intermediate tree.

Next, we prove the correctness of the construction. First, we observe that $U'(u, t') = T'(\imath(u), t')$ for all $u \in T_{\Sigma'}$ and $t' \in T_\Sigma$. For every $q, q' \in Q$ and every derivation tree $d \in D_T(q, q')$ we let $\ell(d) = \mathrm{in}(d)(\varepsilon)$ and $r(d) = \mathrm{out}(d)(\varepsilon)$ be the root labels of the input and output tree generated by the derivation tree. For every such derivation tree d there exists a derivation tree $d' \in D_U(\ell(d), \langle r(d), q, q' \rangle)$, which is inductively defined by $d' = \rho_{a,b}(d'_1, \dots, d'_k)$, where $d = \rho(d_1, \dots, d_k)$ with $\rho = (q, t) \xrightarrow{\varphi} (q, t')$, the derivation tree d'_i corresponds to the derivation tree d_i for every $i \in [k]$, and $a, b\colon \mathrm{leaves}_Q(t) \to \Sigma$ are defined by $a(w_i) = \ell(d_i)$ and $b(w_i) = r(d_i)$ for every $i \in [k]$ and $\{w_1, \dots, w_k\} = \mathrm{leaves}_Q(t)$ with $w_1 \sqsubset \cdots \sqsubset w_k$. Moreover, $\mathrm{in}(d) = \mathrm{in}(d')$, $\mathrm{out}(d) = \imath(\mathrm{out}(d'))$, and $\mathrm{wt}(d) = \mathrm{wt}_U(d')$. This correspondence actually establishes weight-preserving bijections between the sets $\{d \in D_T(q, q') \mid \ell(d) = \sigma, r(d) = \gamma\}$ and $D_U(\sigma, \langle \gamma, q, q' \rangle)$ for all $\sigma, \gamma \in \Sigma$ and $q, q' \in Q$. Given $t, u \in T_\Sigma$ we thus have $T(t, u) = \sum_{u' \in \imath^{-1}(u)} U(t, u')$ and

$$\begin{aligned}(T\,;U)(t, v) &= \sum_{u \in T_\Sigma} T(t, u) \cdot T'(u, v) = \sum_{u \in T_\Sigma} \sum_{u' \in \imath^{-1}(u)} U(t, u') \cdot U'(u', v) \\ &= \sum_{u' \in T_{\Sigma'}} U(t, u') \cdot U'(u', v) = (U\,;U')(t, v)\end{aligned}$$

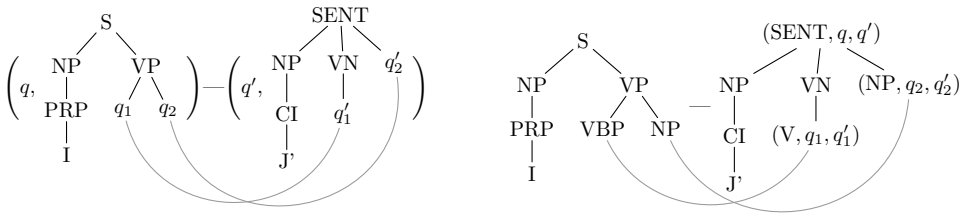

Figure 5. Illustration of the state annotation on the intermediate tree. The left part shows the original tree transducer rule and the right STSG rule shows how the state annotation is performed on the output tree using the guessed nonterminal pairs (VBP, V) and (NP, NP) for (q_1, q'_1) and (q_2, q'_2), respectively.

for all $t, v \in T_\Sigma$.

For statement (ii), where T' is an STSG $(\Sigma, \Sigma, \Sigma_0, R', \mathrm{wt}')$, we use the same construction but instead construct the STSG $U' = (\Sigma', \Sigma', \Sigma''_0, R'', \mathrm{wt}'')$ with $\Sigma''_0 = \{(\kappa, \sigma) \in \Sigma' \times \Sigma' \mid (\iota(\kappa), \sigma) \in \Sigma_0\}$ and for every rule $\rho' = (t(\varepsilon), t) \xrightarrow{\varphi} (t'(\varepsilon), t')$ of R' of the STSG T', we have that the rule $\rho'_u = (u(\varepsilon), u) \xrightarrow{\varphi} (t'(\varepsilon), t')$ belongs to R'' for every $u \in \iota^{-1}(t)$. As before, $\mathrm{wt}''(\rho_u) = \mathrm{wt}'(\rho)$, and no other rules are in R''. The proof remains valid also in this case. □

Corollary 3. *For all $n \geq 2$, compositions of n simple STSGs are as expressive as compositions of n simple tree transducers.*

Proof. Since the classes s-STSG($\mathcal{C}$) and s-TT($\mathcal{C}$) are obviously closed under inverses, we can conclude from Lemma 2(ii) that

$$\text{s-STSG}(\mathcal{C}) \,;\, \text{s-TT}(\mathcal{C}) = \text{s-STSG}(\mathcal{C})^{-1} \,;\, \text{s-TT}(\mathcal{C})^{-1} = \big(\text{s-TT}(\mathcal{C}) \,;\, \text{s-STSG}(\mathcal{C})\big)^{-1}$$
$$= \big(\text{s-STSG}(\mathcal{C}) \,;\, \text{s-STSG}(\mathcal{C})\big)^{-1} = \text{s-STSG}(\mathcal{C})^{-1} \,;\, \text{s-STSG}(\mathcal{C})^{-1} = \text{s-STSG}(\mathcal{C})^2 \ .$$

Consequently, $\text{s-TT}^n = \text{s-STSG}^n$ by $n - 1$ applications of Lemma 2(i) and an application of the previous conclusion. □

Theorem 4. *The composition closure of* s-STSG($\mathbb{B}$) *is obtained at the second level.*

Proof. Unweighted simple tree transducers achieve the composition closure at the second level [2]. Since the second levels of the composition hierarchy for s-TT($\mathbb{B}$) and s-STSG($\mathbb{B}$) coincide by Corollary 3, and unweighted simple STSGs are strictly less expressive by (1), the composition closure of unweighted simple STSGs is achieved at the second level as well. □

In the remaining unweighted cases, which consist of (i) strict STSGs, (ii) ε-free STSGs, and (iii) general STSGs, the composition hierarchy of the corresponding unweighted tree transducers is infinite. We summarize the results in Table 2.

Table 2. Composition closure results for unweighted and weighted SCFGs and STSGs. They mirror the corresponding results for tree transducers.

Model	(Un-)weighted composition closure	Results
SCFGs	1	Theorem 1
simple STSGs	2	Theorems 4 and 8
other STSGs	∞	Theorems 5 and 9

Theorem 5. *The composition hierarchy of unweighted strict STSGs, unweighted ε-free STSGs, and unweighted general STSGs is infinite.*

Proof. If we re-examine the counterexample translation τ provided in Example 43 of [8], then we can easily see that it does not utilize its states and can be generated by an unweighted ε-free STSG as well. Hence for every $n \geq 1$ we obtain a translation τ^{n+1} that can be computed by $(n+1)$ unweighted ε-free STSGs, but not by n unweighted tree transducers according to Lemma 44 of [8]. Since by (1) we have $\mathrm{STSG}(\mathbb{B}) \subseteq \mathrm{TT}(\mathbb{B})$, it follows that $\mathrm{STSG}(\mathbb{B})^n \subseteq \mathrm{TT}(\mathbb{B})^n$ and thus n unweighted STSGs also cannot implement τ^{n+1}, which proves the infiniteness of the composition hierarchies for unweighted ε-free and general STSGs. The analogous arguments using the inverse translation τ^{-1} can be used to prove the infiniteness of the composition hierarchy for unweighted strict STSGs. □

4. Classification of weighted composition closures

In the weighted setting, which is more relevant in statistical machine translation, the models assign a weight to each rule. The weights of the rules occurring in a derivation tree are multiplied, and if there are several derivation trees for the same input- and output-tree pair, then their weights are summed up. To avoid infinite summations in compositions, we restrict ourselves to ε-free or strict models to enforce that all translations are finitary or co-finitary.

The goal of this section is to lift the unweighted results of the previous section into the weighted setting. In Theorem 1 we already proved that SCFGs are closed under composition. Moreover, Corollary 3 shows that the composition closure of simple STSGs coincides with that of simple tree transducers, so it only remains to establish the composition closure for simple tree transducers. Roughly speaking, we will reduce this problem to the unweighted setting by removing the weights from the tree transducer and moving them into a particularly simple type of translation, called weighted relabeling, which we define next.

For a given alphabet Σ, a *(weighted) relabeling* is a mapping $\kappa\colon \Sigma \times \Sigma \to C$. In other words, it is a weighted association between symbols. It extends to pairs of trees such that (i) it assigns weight 0 to all pairs of trees of different shape and (ii) it simply takes the product of the symbol-to-symbol weights given by κ for all corresponding nodes in trees of the same shape. Formally, each such relabeling κ

extends to a weighted tree translation $\kappa\colon T_\Sigma(Q) \times T_\Sigma(Q) \to C$, which is defined for every $t, u \in T_\Sigma(Q)$ by

$$\kappa(t,u) = \begin{cases} \prod_{w \in \mathrm{pos}(t) \setminus \mathrm{leaves}_Q(t)} \kappa\big(t(w), u(w)\big) & \text{if } \mathrm{pos}(t) = \mathrm{pos}(u) \text{ and} \\ & t(w) = u(w) \text{ for all } w \in \mathrm{leaves}_Q(t) \\ 0 & \text{otherwise.} \end{cases}$$

We use $\mathrm{REL}(\mathcal{C})$ for the class of all weighted translations computable by relabelings.

Lemma 6. *For every composition of a simple tree transducer and a relabeling in either order, we can present an equivalent simple tree transducer.*

$$\text{s-TT}(\mathcal{C})\,;\mathrm{REL}(\mathcal{C}) \subseteq \text{s-TT}(\mathcal{C}) \qquad \text{and} \qquad \mathrm{REL}(\mathcal{C})\,;\text{s-TT}(\mathcal{C}) \subseteq \text{s-TT}(\mathcal{C})$$

Proof. The first statement is obtained by combining the decomposition in Lemma 4.1 of [9] and the composition results in Theorem 2.4 of [18]. Moreover, since all the involved models are closed under inverses, we also immediately obtain the second statement. □

The next lemma shows that we can separate the weights from a simple tree transducer. A tree transducer $T = (Q, \Sigma, Q_0, R, \mathrm{wt})$ is *Boolean* if $\mathrm{wt}\colon R \to \{0,1\}$; i.e., it only uses the rule weights 0 and 1. Moreover, it is *unambiguous* if for every $t, u \in T_\Sigma$ there exists at most one $(q, q') \in Q_0$ and $d \in D_T(q, q')$ such that $\mathrm{in}(d) = t$ and $\mathrm{out}(d) = u$. When we strip the weights from a tree transducer we construct a composition of an unambiguous and Boolean simple tree transducer T' and a relabeling. Moreover, the tree translation computed by $T'\colon T_\Sigma \times T_\Sigma \to C$ will be *injective*, which means that for every output tree $u \in T_\Sigma$ there exists at most one input tree $t \in T_\Sigma$ such that $T'(t,u) \neq 0$. We use '$\text{su-TT}^{0,1}_{\mathrm{inj}}(\mathcal{C})$' for the class of injective translations computed by unambiguous and Boolean simple tree transducers. Note that these translations are essentially the characteristic functions of the relations computed by the corresponding unweighted tree transducers and enjoy the same properties as $\text{su-TT}_{\mathrm{inj}}(\mathbb{B})$.

Lemma 7. *Every simple tree transducer T can be equivalently represented by a composition of an unambiguous and Boolean simple tree transducer T' computing an injective translation followed by a relabeling κ.*

$$\text{s-TT}(\mathcal{C}) \subseteq \text{su-TT}^{0,1}_{\mathrm{inj}}(\mathcal{C})\,;\mathrm{REL}(\mathcal{C})$$

Proof. Let $T = (Q, \Sigma, Q_0, R, \mathrm{wt})$ be the simple tree transducer. We construct the tree transducer $T' = (Q, \Sigma', Q_0, R', \mathrm{wt}')$ with $\Sigma' = \Sigma \cup (\Sigma \times R)$ and the set R' of rules and their weights wt' as follows. For every rule $\rho = (q,t) \xrightarrow{\varphi} (q', t') \in R$ with $\mathrm{wt}(\rho) \neq 0$, we have $t' = \sigma(t'_1, \dots, t'_k)$ for some integer k, symbol $\sigma \in \Sigma$, and subtrees $t'_1, \dots, t'_k \in T_\Sigma(Q)$ because T is strict. Then the rule $\rho' = (q,t) \xrightarrow{\varphi} (q', \langle \sigma, \rho \rangle(t'_1, \dots, t'_k)) \in R'$ belongs to R', which essentially records

the rule application in the root of the output tree fragment. We set $\mathrm{wt}'(\rho') = 1$. No other rules are in R'. Finally, the relabeling $\kappa\colon \Sigma' \times \Sigma' \to C$ is defined for every $\omega, \omega' \in \Sigma'$ such that

$$\kappa(\omega, \omega') = \begin{cases} \mathrm{wt}(\rho) & \text{if } \omega = \langle \omega', \rho \rangle \text{ for some } \rho \in R \\ 1 & \text{if } \omega = \omega' \in \Sigma \\ 0 & \text{otherwise.} \end{cases}$$

In other words, the relabeling removes the annotation and charges the weight of the annotated rule.

Obviously, the constructed tree transducer T' is Boolean. In addition, for every output tree $u \in T_{\Sigma'}$ there is at most one pair $(q, q') \in Q \times Q$ and at most one derivation tree $d \in D_{T'}(q, q')$ such that $\mathrm{out}(d) = u$ because the derivation tree is completely encoded in the output. Consequently, T' is unambiguous. Moreover, the derivation tree d uniquely determines the input tree $\mathrm{in}(d)$, so the translation is also injective. Clearly, for all $q, q' \in Q$, we obtain a bijection between $D_T(q, q')$ and $D_{T'}(q, q')$ by simply replacing each label $\rho \in R$ of a derivation tree $d \in D_T(q, q')$ by the corresponding rule label $\rho' \in R'$ to obtain the derivation tree $d' \in D_{T'}(q, q')$. Moreover, for such related derivation trees d and d' we have $\mathrm{in}(d) = \mathrm{in}(d')$, $\kappa(\mathrm{out}(d'), \mathrm{out}(d)) = \mathrm{wt}(d)$, and $\mathrm{wt}'(d') = 1$. Overall, for all $t, u \in T_\Sigma$ this proves

$$\begin{aligned} T(t,u) &= \sum_{(q,q') \in Q_0} \sum_{\substack{d \in D_T(q,q') \\ \mathrm{in}(d)=t,\, \mathrm{out}(d)=u}} \mathrm{wt}(d) = \sum_{\substack{(q,q') \in Q_0 \\ d \in D_T(q,q') \\ \mathrm{in}(d')=t,\, \mathrm{out}(d)=u}} \kappa(\mathrm{out}(d'), \mathrm{out}(d)) \\ &= \sum_{t' \in T_{\Sigma'}} \left(\sum_{\substack{(q,q') \in Q_0 \\ d' \in D_{T'}(q,q') \\ \mathrm{in}(d')=t,\, \mathrm{out}(d')=t'}} \mathrm{wt}'(d') \right) \cdot \kappa(t', u) = \sum_{t' \in T_{\Sigma'}} T'(t, t') \cdot \kappa(t', u) \\ &= (T' \,;\, \kappa)(t, u) \ . \end{aligned}$$

□

Using Lemmas 6 and 7 we can now separate the weights from a composition chain because

$$\begin{aligned} &\text{s-TT}(\mathcal{C}) \,;\, \text{s-TT}(\mathcal{C})^2 \subseteq \text{su-TT}^{0,1}_{\text{inj}}(\mathcal{C}) \,;\, \text{REL}(\mathcal{C}) \,;\, \text{s-TT}(\mathcal{C})^2 && \text{(Lemma 7)} \\ \subseteq\ &\text{su-TT}^{0,1}_{\text{inj}}(\mathcal{C}) \,;\, \text{s-TT}(\mathcal{C})^2 \subseteq \text{su-TT}^{0,1}_{\text{inj}}(\mathcal{C})^2 \,;\, \text{REL}(\mathcal{C}) \,;\, \text{s-TT}(\mathcal{C}) && \text{(Lemmas 6 and 7)} \\ \subseteq\ &\text{su-TT}^{0,1}_{\text{inj}}(\mathcal{C})^2 \,;\, \text{s-TT}(\mathcal{C}) \subseteq \text{su-TT}^{0,1}_{\text{inj}}(\mathcal{C})^3 \,;\, \text{REL}(\mathcal{C}) \ . && \text{(Lemmas 6 and 7)} \end{aligned}$$

Next we identify the neutral elements $\{0, 1\}$ of $\mathcal{C}$ with the corresponding elements of $\mathbb{B}$, which yields $\text{su-TT}^{0,1}_{\text{inj}}(\mathcal{C}) = \text{su-TT}_{\text{inj}}(\mathbb{B})$. Since the composition closure of $\text{s-TT}(\mathbb{B})$ is achieved at the second level [2] and since the composition of injective translations is injective, we obtain

$$\text{su-TT}^{0,1}_{\text{inj}}(\mathcal{C})^3 \,;\, \text{REL}(\mathcal{C}) \subseteq \underbrace{\text{s-TT}(\mathbb{B})^2}_{\text{injective}} \,;\, \text{REL}(\mathcal{C})$$

We cannot simply simulate those unweighted tree transducers directly by tree transducers over $\mathcal{C}$ since there might be different derivation trees d and d' for a given input-output tree pair (t, u); i.e., $\mathrm{in}(d) = \mathrm{in}(d') = t$ and $\mathrm{out}(d) = \mathrm{out}(d') = u$. In the Boolean semiring the obtained weight for (t, u) is still 1 because $\max(1, 1) = 1$. However, in the semiring $\mathcal{C}$ we might have $1 + 1 \neq 1$ (i.e., $\mathcal{C}$ might be non-idempotent), which yields that the existence of several derivation trees for the same tree pair would potentially yield wrong weights. It is well known that if $f ; g$ is injective for given mappings $f \colon A \to B$ and $g \colon B \to C$, then the mappings $f \cap (A \times \mathrm{dom}(g))$ and $g \cap (\mathrm{ran}(f) \times C)$ are injective as well. Moreover the translations $\tau \in \text{s-TT}(\mathbb{B})$ computed by unweighted simple tree transducers have regular tree languages as domain $\mathrm{dom}(\tau)$ and range $\mathrm{ran}(\tau)$ [21] and for every $\tau' \in \text{s-TT}(\mathbb{B})$ and all regular tree languages L and L' we also have $\tau' \cap (L \times L') \in \text{s-TT}(\mathbb{B})$ [21]. Thus we can use regular restrictions restricting the output of the first translation to the inputs of the second and vice versa to make both unweighted translations injective. Moreover each injective translation can be made unambiguous using the regular look-ahead [21], so we obtain

$$\underbrace{\text{s-TT}(\mathbb{B})^2}_{\text{injective}} ; \mathrm{REL}(\mathcal{C}) \subseteq \text{su-TT}_{\text{inj}}(\mathbb{B})^2 ; \mathrm{REL}(\mathcal{C}) \subseteq \text{s-TT}(\mathcal{C})^2 \ ,$$

where the last step uses Lemma 6. Note that unweighted unambiguous tree transducers can easily be simulated by the corresponding tree transducers over $\mathcal{C}$. Thus, we derived the difficult part of the composition closure.

Theorem 8. *The composition closure of* s-STSG($\mathcal{C}$) *is achieved at the second level.*

Proof. We showed that $\text{s-TT}(\mathcal{C})^3 \subseteq \text{s-TT}(\mathcal{C})^2$, so the composition hierarchy of the class s-TT($\mathcal{C}$) collapses at level 2. Moreover using the linking arguments of [20] we can also conclude that $\text{s-TT}(\mathcal{C}) \subset \text{s-TT}(\mathcal{C})^2$. □

Finally, for the remaining classes (i.e., strict STSGs and ε-free STSGs), we can essentially import the infinite composition hierarchy from the unweighted case using the linking technique of [20].

Theorem 9. *The composition hierarchy of strict STSGs and ε-free STSGs is infinite.*

Conclusion

We have investigated the expressive power of compositions of the well-established tree-to-tree translation models: SCFGs, STSGs, and tree transducers. In the unweighted case, the results for the local devices [i.e., SCFGs and STSGs] closely mirror the known composition results for tree transducers due to the fact that we can encode the finite-state information in the intermediate trees of a composition. The same picture presents itself in the weighted setting, for which we showed how to

lift the corresponding results from the unweighted setting to the weighted setting. This uses a novel decomposition separating the weights from simple tree transducers and then constructions for the obtained unambiguous and injective tree transducers. Overall, we demonstrated that in the relevant cases, short (length 1 or 2) composition chains are necessary and sufficient to simulate arbitrarily long composition chains.

Bibliography

[1] Aho, A.V., Ullman, J.D.: Syntax directed translations and the pushdown assembler. J. Comput. System Sci. 3(1), 37–56 (1969)
[2] Arnold, A., Dauchet, M.: Morphismes et bimorphismes d'arbres. Theoret. Comput. Sci. 20(1), 33–93 (1982)
[3] Baker, B.S.: Composition of top-down and bottom-up tree transductions. Inform. and Control 41(2), 186–213 (1979)
[4] Chen, S., Matsumoto, T.: Translation of quantifiers in Japanese-Chinese machine translation. In: Proc. JapTAL. LNAI, vol. 7614, pp. 11–22. Springer (2012)
[5] Clifton, A., Sarkar, A.: Combining morpheme-based machine translation with post-processing morpheme prediction. In: Proc. ACL. pp. 32–42. ACL (2011)
[6] Collins, M., Koehn, P., Kucerová, I.: Clause re-structuring for statistical machine translation. In: Proc. ACL. pp. 531–540. ACL (2005)
[7] Eisner, J.: Learning non-isomorphic tree mappings for machine translation. In: Proc. ACL. pp. 205–208. ACL (2003)
[8] Engelfriet, J., Fülöp, Z., Maletti, A.: Composition closure of linear extended top-down tree transducers. Theory Comput. Systems 60(2), 129–171 (2017)
[9] Fülöp, Z., Maletti, A., Vogler, H.: Weighted extended tree transducers. Fundam. Inform. 111(2), 163–202 (2011)
[10] Fülöp, Z., Vogler, H.: Weighted tree transducers. J. Autom. Lang. Combin. 9(1), 31–54 (2004)
[11] Fülöp, Z., Vogler, H.: Weighted tree automata and tree transducers. In: Droste, M., Kuich, W., Vogler, H. (eds.) Handbook of Weighted Automata, chap. 9, pp. 313–403. Springer (2009)
[12] Gécseg, F., Steinby, M.: Tree Automata. Akadémiai Kiadó, Budapest (1984), 2nd edition available at `https://arxiv.org/abs/1509.06233`
[13] Golan, J.S.: Semirings and their Applications. Kluwer Academic (1999)
[14] Graehl, J., Knight, K.: Training tree transducers. In: Proc. HLT-NAACL. pp. 105–112. ACL (2004)
[15] Hebisch, U., Weinert, H.J.: Semirings — Algebraic Theory and Applications in Computer Science. World Scientific (1998)
[16] Koehn, P.: Statistical Machine Translation. Cambridge University Press (2010)
[17] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proc. ACL. pp. 177–180. ACL (2007)
[18] Kuich, W.: Full abstract families of tree series I. In: Jewels Are Forever, pp. 145–156. Springer (1999)
[19] Lerner, U., Petrov, S.: Source-side classifier preordering for machine translation. In: Proc. EMNLP. pp. 513–523. ACL (2013)
[20] Maletti, A.: The power of weighted regularity-preserving multi bottom-up tree transducers. Int. J. Found. Comput. Sci. 26(7), 987–1005 (2015)

[21] Maletti, A., Graehl, J., Hopkins, M., Knight, K.: The power of extended top-down tree transducers. SIAM J. Comput. 39(2), 410–430 (2009)
[22] May, J., Knight, K., Vogler, H.: Efficient inference through cascades of weighted tree transducers. In: Proc. ACL. pp. 1058–1066. ACL (2010)
[23] Mohri, M.: Finite-state transducers in language and speech processing. Comput. Linguist. 23(2), 269–311 (1997)
[24] Seemann, N., Braune, F., Maletti, A.: String-to-tree multi bottom-up tree transducers. In: Proc. ACL. pp. 815–824. ACL (2015)
[25] Seemann, N., Braune, F., Maletti, A.: A systematic evaluation of MBOT in statistical machine translation. In: Proc. MT Summit. pp. 200–214. AMTA (2015)
[26] Stymne, S.: Text Harmonization Strategies for Phrase-Based Statistical Machine Translation. Ph.D. thesis, Linköping University (2012)
[27] Xia, F., McCord, M.C.: Improving a statistical MT system with automatically learned rewrite patterns. In: Proc. CoLing. pp. 508–514. ACL (2004)