The Power of Regularity-Preserving Multi Bottom-up Tree Transducers

Andreas Maletti*

Institute of Computer Science, Universität Leipzig
Augustusplatz 10–11, 04109 Leipzig, Germany
maletti@informatik.uni-leipzig.de

Abstract. The expressive power of regularity-preserving multi bottom-up tree transducers (MBOT) is investigated. These MBOT have very attractive theoretical and algorithmic properties. However, their expressive power is not well understood. It is proved that despite the restriction their power still exceeds that of composition chains of linear extended top-down tree transducers with regular look-ahead (XTOP^{R}), which are a natural super-class of STSG. In particular, topicalization can be modeled by such MBOT, whereas composition chains of XTOP^{R} cannot implement it. However, the inverse of topicalization cannot be implemented by any MBOT. An interesting, promising, and widely applicable proof technique is used to prove those statements.

1 Introduction

Statistical machine translation [15] deals with the automatic translation of natural language texts. A central component of each statistical machine translation model is the *translation model*, which is the model that actually performs the translation. Various other models support the translation (such as language models), but the type of transformations computable by the system is essentially determined by the translation model. Various different translation models are currently in use: (i) Phrase-based [23] systems essentially use a finite-state transducer [13]. (ii) Hierarchical phrase-based systems [4] use a synchronous context-free grammar (SCFG), and (iii) syntax-based systems use a form of synchronous tree grammar such as synchronous tree substitution grammars (STSG) [5], synchronous tree-adjoining grammars (STAG) [24], or synchronous tree-sequence substitution grammars (STSSG) [25]. In this contribution, we will focus on syntax-based systems. Since machine translation systems are trained on large data, the used translation model must meet two contradictory goals. Its expressive power should be large in order to be able to model all typical phenomena that occur in translation. On the other hand, the model should have nice algorithmic properties and important operations should have low computational complexity. The mentioned models cover a wide spectrum along this axis with SCFG and STSG

* The author was financially supported of the German Research Foundation (DFG) grant MA / 4959 / 1-1.

as the weakest models with the best parsing complexities. It is thus essential for the evaluation of a translation model to accurately determine its expressive power and the complexities of its principal operations [14].

A relatively recent proposal for another translation model suggests the multi bottom-up tree transducer (MBOT) [18, 19]. It can be understood as an extension of STSG that allows discontinuity on the output side or as a restriction of STSSG that disallows discontinuity on the input side. MBOT are thus a natural (half-way) model in between STSG and STSSG. In addition, [18, 19] demonstrated that MBOT have very good theoretical and algorithmic properties in comparison to both STSG and STSSG. They have been implemented [2] in the machine translation framework MOSES [16] and were successfully evaluated in an English-to-German translation task, in which they significantly outperformed the STSG baseline. However, MBOT also have a feature called *finite copying* [7], which on the positive side yields that the output string language can be a multiple context-free language (or equivalently a linear context-free rewriting system language) [12]. Since this class of languages is much more powerful than context-free languages, its algorithmic properties are not as nice as those of the regular tree languages [10, 11], which can be used to represent the parse trees of context-free grammars. It is not clear whether this added complexity is necessary to model common discontinuities like topicalization.

In this contribution we demonstrate that the regularity-preserving MBOT (i.e., those whose output is always a regular tree language) retain the power to compute discontinuities such as topicalization. Moreover, these MBOT remain more powerful than arbitrary composition chains [22] of STSG. In particular, no chain of STSG can implement topicalization. However, whereas STSG can trivially be inverted, neither MBOT nor regularity-preserving MBOT can be inverted in general. In fact, we show that the inverse of topicalization cannot be implemented by any MBOT, which confirms the bottom-up nature of MBOT. Overall, these results allow us to relate the expressive power of regularity-preserving MBOT to the other classes (see Figure 6). Secondly, we want to promote the use of explicit links as a tool for analyses. Links naturally record which parts of the input and output tree have to develop synchronously in a derivation step. However, once expanded, the "used" links are typically dropped [3]. Here we retain all links in a special component of the sentential form in the spirit of [20, 9]. We investigate the properties of these links and then use them to prove our main results. With the links the proofs split into a standard technical part that establishes certain mandatory links [9] and a rather straightforward high-level argumentation that refutes that the obtained link ensemble is well-formed. We believe that this proof method holds much potential and can successfully be applied to many additional setups.

2 Notation

We write $\mathbb{N}$ for the set of all nonnegative integers (including 0). Given a relation $R \subseteq S_1 \times S_2$ and $S \subseteq S_1$, we let $R(S) = \{s_2 \mid \exists s_1 \in S \colon (s_1, s_2) \in R\}$

and $R^{-1} = \{(s_2, s_1) \mid (s_1, s_2) \in R\}$ be the elements of S_2 related to elements of S (via R) and the inverse relation of R, respectively. Instead of $R(\{s\})$ with $s \in S_1$ we also write $R(s)$. The composition of two relations $R_1 \subseteq S_1 \times S_2$ and $R_2 \subseteq S_2 \times S_3$ is the relation $R_1 \mathbin{;} R_2 \subseteq S_1 \times S_3$ given by

$$R_1 \mathbin{;} R_2 = \{(s_1, s_3) \mid R_1(s_1) \cap R_2^{-1}(s_3) \neq \emptyset\} \enspace .$$

As usual, S^* denotes the set of all (finite) words over a set S with the empty word ε. We simply write $v.w$ or vw for the concatenation of the words $v, w \in S^*$, and the length of a word $w \in S^*$ is $|w|$. Given languages $L, L' \subseteq S^*$, we let $L \cdot L' = \{v.w \mid v \in L, w \in L'\}$. An alphabet Σ is a nonempty and finite set of symbols. Given an alphabet Σ and a set S, the set $T_\Sigma(S)$ of Σ-trees indexed by S is the smallest set such that $S \subseteq T_\Sigma(S)$ and $\sigma(t_1, \dots, t_k) \in T_\Sigma(S)$ for all $k \in \mathbb{N}$, $\sigma \in \Sigma$, and $t_1, \dots, t_k \in T_\Sigma(S)$. We write T_Σ for $T_\Sigma(\emptyset)$.

Whenever we need to address specific parts of a tree, we use positions. Each position is a word of $\mathbb{N}^*$. The root of a tree has position ε and the position $i.p$ with $i \in \mathbb{N}$ and $p \in \mathbb{N}^*$ addresses the position p in the i^{th} direct child of the root. The set $\text{pos}(t)$ denotes the set of all positions in a tree $t \in T_\Sigma(S)$. We note that positions are totally ordered by the lexicographic order $\sqsubseteq$ on $\mathbb{N}^*$ and partially ordered by the prefix order $\leq$ on $\mathbb{N}^*$. The total order $\sqsubseteq$ allows us to turn a finite set $P \subseteq \mathbb{N}^*$ into a vector $\boldsymbol{P}$ by letting $\boldsymbol{P} = (w_1, \dots, w_m)$ if $P = \{w_1, \dots, w_m\}$ with $w_1 \sqsubset \cdots \sqsubset w_m$. Given a tree $t \in T_\Sigma(S)$ its size $|t|$ is the number of its nodes (i.e., $|t| = |\text{pos}(t)|$), and its height $\text{ht}(t)$ coincides with the length of the longest position (i.e., $\text{ht}(t) = \max_{w \in \text{pos}(t)} |w|$).

We conclude this section with some essential operations on trees. To this end, let $t, u \in T_\Sigma(S)$ be trees and $w \in \text{pos}(t)$ be a position in t. The label of t at w is $t(w)$. For every $s \in S$, we let $\text{pos}_s(t) = \{w \in \text{pos}(t) \mid t(w) = s\}$ be those positions in t that are labeled by s. The tree $t \in T_\Sigma(S)$ is linear if $|\text{pos}_s(t)| \leq 1$ for every $s \in S$. We let $\text{idx}(t) = \{s \in S \mid \text{pos}_s(t) \neq \emptyset\}$. Finally, the expression $t[u]_w$ denotes the tree that is obtained from t by replacing the subtree at w by u. We also extend this notation to sequences $\boldsymbol{u} = (u_1, \dots, u_m)$ of trees and positions $\boldsymbol{w} = (w_1, \dots, w_m)$ of t that are pairwise incomparable with respect to $\leq$. Thus, $t[\boldsymbol{u}]_{\boldsymbol{w}}$ denotes the tree obtained from t by replacing the subtree at w_i by u_i for all $1 \leq i \leq m$. Formally, $t[\boldsymbol{u}]_{\boldsymbol{w}} = (\cdots (t[u_1]_{w_1}) \cdots)[u_m]_{w_m}$.

3 Formal models

The main transformational grammar formalism under discussion is the multi bottom-up tree transducer (MBOT), which was introduced by [17, 1]. An English theoretical treatment can be found in [6]. In general, MBOT are synchronous grammars [3] with potentially discontinuous output sides, which makes them more powerful than the commonly used STSG [5]. Thus, each rule of an MBOT specifies potentially several parts of the output tree. We essentially recall the definition of [20].

Definition 1. *A* multi bottom-up tree transducer *(for short:* MBOT*) is a tuple* $G = (N, \Sigma, S, P)$*, where*

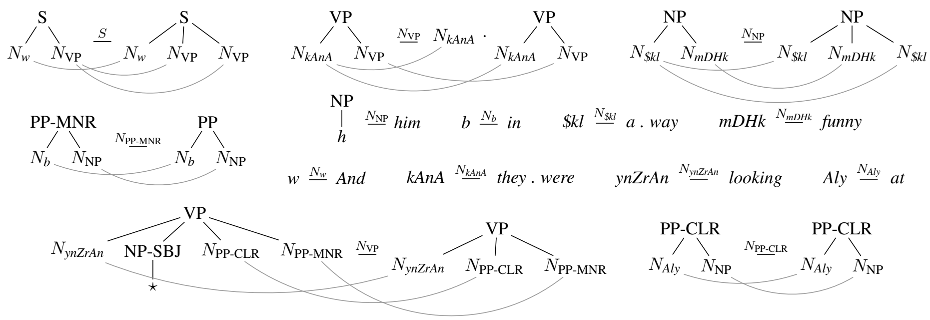

Fig. 1. Example productions.

– *N is its finite set of* nonterminals*,*
– *Σ is its alphabet of* input *and* output symbols*,*
– *$S \in N$ is its* initial *nonterminal, and*
– *$P \subseteq T_\Sigma(N) \times N \times T_\Sigma(N)^*$ is its finite set of* productions *such that $\ell \notin N$, ℓ is linear, and $\bigcup_{1 \le i \le |\boldsymbol{r}|} \mathrm{idx}(r_i) \subseteq \mathrm{idx}(\ell)$ for every $\langle \ell, n, \boldsymbol{r} \rangle \in P$.*

If all productions $\langle \ell, n, \boldsymbol{r} \rangle \in P$ obey $|\boldsymbol{r}| \le 1$, then G is a (linear) extended top-down tree transducer with regular look-ahead *(XTOP$^{\mathrm{R}}$) [21], and if they even fulfill $|\boldsymbol{r}| = 1$, then it is a (linear)* nondeleting extended top-down tree transducer *(n-XTOP).*

In comparison to [20] we added the requirement of $\ell \notin N$, which could be called *input ε-freeness*. To avoid repetition, we henceforth let $G = (N, \Sigma, S, P)$ be an arbitrary MBOT. As usual, ℓ and $\boldsymbol{r}$ of a production $\langle \ell, n, \boldsymbol{r} \rangle \in P$ are called *left-* and *right-hand side*, respectively. We also write $\ell \xrightarrow{n} \boldsymbol{r}$ instead of $\langle \ell, n, \boldsymbol{r} \rangle$. The productions of our running example MBOT are displayed in Figure 1. The initial nonterminal is S, and we omit an explicit representation of the set N of nonterminals (containing the various slanted N_x and S) and the set Σ of symbols because they can be deduced from the productions. For completeness' sake, the leftmost production on the first line in Figure 1 can be written as

$$\langle \mathrm{S}(N_w, N_{\mathrm{VP}}),\ S,\ \mathrm{S}(N_w, N_{\mathrm{VP}}, N_{\mathrm{VP}}) \rangle \ .$$

In contrast to [19, 2], which present the semantics of MBOT using a bottom-up process based on pre-translations, we present a top-down semantics in the style of [20] here. As usual [3], the top-down semantics requires us to keep track of the positions that are supposed to develop synchronously in the input and output. Such related positions are called *linked positions*, and the additional data structure recording the linked positions is called the *link structure*. Although the link structure might at first be seen as an overhead (since it is not required for the bottom-up semantics), it will be an essential tool later on. In fact, all our later arguments will be based on the link structure, so we explicitly want to promote the use of link structures and an investigation into their detailed properties.

We start with the introduction of the link structure resulting from a single production. In fact, the link structure of a production is implicit because we

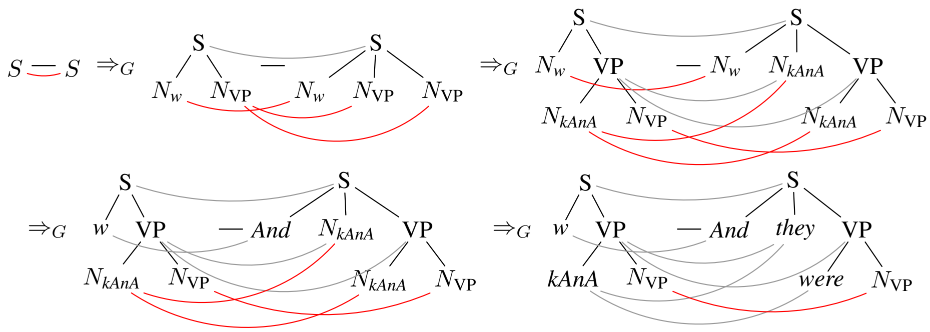

Fig. 2. Partial derivation using the productions of Figure 1. The active links are clearly marked, whereas disabled links are light.

assume that an occurrence of a nonterminal n in the left-hand side is linked to all its occurrences in the right-hand side. We usually depict these links as (light) splines in graphical illustrations of productions (see Figure 1). However, once we move to the derivation process, an explicit representation of these links is required to keep track of synchronously developing nonterminals.

Definition 2. *Given* $\ell \overset{n'}{—} \boldsymbol{r} \in P$ *and positions* v *and* $\boldsymbol{w} = (w_1, \ldots, w_m)$ *with* $m = |\boldsymbol{r}|$ *to which the production should be applied, we define the* link structure $\mathrm{links}_{v,\boldsymbol{w}}(\ell \overset{n'}{—} \boldsymbol{r})$ *by* $\bigcup_{n \in N, 1 \leq i \leq m} \big(\{v\} \cdot \mathrm{pos}_n(\ell)\big) \times \big(\{w_i\} \cdot \mathrm{pos}_n(r_i)\big)$.

In other words, besides linking occurrences of the same nonterminal as already mentioned, we prefix the positions by the corresponding position given as a parameter. These argument positions will hold the positions to which the production shall be applied to. Now we are ready to present the semantics. Simply said, we select an input position, its actively linked output positions, and a production that has the right number of right-hand sides. Then we disable the selected links, substitute the production components into the corresponding selected positions, and add the link structure of the production to the set of active links. Formally, a sentential form is simply a tuple $\langle t, A, D, u\rangle$ consisting of an input and an output tree $t, u \in T_\Sigma(N)$ and two sets of links $A, D \subseteq \mathrm{pos}(t) \times \mathrm{pos}(u)$ containing *active* and *disabled* links, respectively. We let $\mathcal{SF}(G)$ be the set of all sentential forms, and $\mathcal{D}(G)$ is the set $\{\langle t, D, u\rangle \mid \langle t, \emptyset, D, u\rangle \in \mathcal{SF}(G), t, u \in T_\Sigma\}$ of all potential dependencies for nonterminal-free input and output trees. In graphical representations we only present the input and output trees and illustrate the links of A and D as clear and light splines, respectively.

Definition 3. *We write* $\langle t, A, D, u\rangle \Rightarrow_G \langle t', A', D', u'\rangle$ *for two sentential forms* $\langle t, A, D, u\rangle, \langle t', A', D', u'\rangle \in \mathcal{SF}(G)$*, if there exist a nonterminal* $n \in N$*, an input position* $v \in \mathrm{pos}_n(t)$ *labeled by* n*, actively linked output positions* $A(v)$*, and a production* $\ell \overset{n}{—} \boldsymbol{r} \in P$ *such that*

– $|\boldsymbol{r}| = |A(v)|$ *and* $\boldsymbol{w} = \boldsymbol{A(v)}$*,*

– $t' = t[\ell]_v$ *and* $u' = u[\boldsymbol{r}]_{\boldsymbol{w}}$*, and*
– $A' = (A \setminus L) \cup \text{links}_{v,\boldsymbol{w}}(\ell \stackrel{n}{—} \boldsymbol{r})$ *and* $D' = D \cup L$ *with* $L = \{(v, w) \mid w \in A(v)\}$.

As usual, $\Rightarrow_G^*$ *is the reflexive and transitive closure of* $\Rightarrow_G$*. The* MBOT G *computes the dependencies* $\text{dep}(G) \subseteq \mathcal{D}(G)$ *given by*

$$\{\langle t, D, u\rangle \in \mathcal{D}(G) \mid \langle S, \{(\varepsilon, \varepsilon)\}, \emptyset, S\rangle \Rightarrow_G^* \langle t, \emptyset, D, u\rangle\} \enspace .$$

Finally, the MBOT G *computes the relation* $G \subseteq T_\Sigma \times T_\Sigma$*, which is given by* $G = \{\langle t, u\rangle \mid \langle t, D, u\rangle \in \text{dep}(G)\}$.

Note that disabled links are often not preserved in the sentential forms in the literature [3], but we want to investigate and reason about those links as in [20, 9], so we preserve them. The first steps of a derivation using the productions of Figure 1 are presented in Figure 2.

In the remaining part of this section, we recall the notion of regular tree languages [10, 11] and some properties on dependencies [20, 9]. Any subset $L \subseteq T_\Sigma$ is a *tree language*, and a tree language $L \subseteq T_\Sigma$ is *regular* [6] if and only if there exists an MBOT $G = (N, \Sigma, S, P)$ such that $L = G^{-1}(T_\Sigma)$ (i.e., L is the domain of G). A relation $R \subseteq T_\Sigma \times T_\Sigma$ *preserves regularity* if $R(L)$ is regular for every regular tree language $L \subseteq T_\Sigma$.

Next, we recall the properties on dependencies of [20, 9]. We only define them for the input side, but assume that they are also defined (in the same manner) for the output side.

Definition 4. *A dependency* $\langle t, D, u\rangle \in \mathcal{D}(G)$ *is*
- input hierarchical *if* $w_2 \not< w_1$ *and there exists* $(v_1, w_1') \in D$ *with* $w_1' \leq w_2$ *for all* $(v_1, w_1), (v_2, w_2) \in D$ *with* $v_1 < v_2$,
- strictly input hierarchical *if (i)* $v_1 < v_2$ *implies* $w_1 \leq w_2$ *and (ii)* $v_1 = v_2$ *implies* $w_1 \leq w_2$ *or* $w_2 \leq w_1$ *for all* $(v_1, w_1), (v_2, w_2) \in D$,
- input link-distance bounded by $b \in \mathbb{N}$ *if for all links* $(v_1, w_1), (v_1v', w_2) \in D$ *with* $|v'| > b$ *there exists* $(v_1v, w_3) \in D$ *such that* $v < v'$ *and* $1 \leq |v| \leq b$,
- strict input link-distance bounded by b *if for all positions* $v_1, v_1v' \in \text{pos}(t)$ *with* $|v'| > b$ *there exists* $(v_1v, w_3) \in D$ *such that* $v < v'$ *and* $1 \leq |v| \leq b$.

The set $\text{dep}(G)$ *has those properties if each dependency* $\langle t, D, u\rangle \in \text{dep}(G)$ *has them.*

We also say that $\text{dep}(G)$ is *input link-distance bounded* if there exists an integer $b \in \mathbb{N}$ such that it is input link-distance bounded by b. We summarize the known properties in Table 1.

4 Main results

In this contribution, we want to investigate the expressive power of regularity-preserving MBOT, which constitute the class of all MBOT whose computed relation preserves regularity. This class has very nice (algorithmic) properties (see Table 1). It was already argued by [19] that regularity should be preserved by

Table 1. Summary of the properties of the dependencies dep(G) for grammars G belonging to the various grammar formalisms [20, 9].

Model \ Property	hierarchical		link-distance bounded		regular		
	input	output	input	output	domain	range	pres.
n-XTOP	strictly	strictly	strictly	strictly	✓	✓	✓
XTOP^{R}	strictly	strictly	✓	strictly	✓	✓	✓
MBOT	✓	strictly	✓	strictly	✓	✗	✗
reg.-pres. MBOT	✓	strictly	✓	strictly	✓	✓	✓

any grammar formalism (used in syntax-based machine translation) in order to obtain an efficient representation of the output tree language. In fact, several (syntactic) ways to obtain regularity preserving MBOT are discussed there, but these all yield subclasses of the class of all regularity-preserving MBOT. On the other hand, XTOP^{R} and n-XTOP, which are both slightly more powerful than the commonly used STSG [5] but strictly less powerful than regularity-preserving MBOT, are not closed under composition [21], but always preserve regularity. Consequently, [22] consider the efficient evaluation of (composition) chains of n-XTOP, and their approach can easily be extended to XTOP^{R}. Obviously, every chain of XTOP^{R} can be simulated by a regularity-preserving MBOT because each individual XTOP^{R} can be simulated and MBOT are closed under composition [6]. However, the exact relation between these two classes remained open. This question is interesting because it solves whether the (non-copying) features of MBOT (such as discontinuity) can be achieved by chains of XTOP^{R}. In particular, it settles the question whether chains of XTOP^{R} can handle discontinuities, which, in general, cannot be handled by a single XTOP^{R}.

The author assumes that the question remained open because both possible answers require deep insight. If the classes coincide, then we should be able to simulate each regularity-preserving MBOT by a chain of XTOP^{R}, which is complicated due to the fact that "regularity-preserving" is a semantic property on the computed relation. Such a construction would (most likely) shed light on the exact (syntactic) consequences of the restriction to regularity-preserving MBOT. On the other hand, if regularity-preserving MBOT are more powerful than chains of XTOP^{R} (which we prove in this contribution), then we need to exhibit a relation that cannot be computed by any chain of XTOP^{R}, which requires deep insight into the relations computable by chains of XTOP^{R}. Fortunately, there was recent progress in the latter area. In [8] it was shown that a chain of 3 XTOP^{R} can simulate any chain of XTOP^{R}. Together with the linking technique of [20, 9], this will allow us to present a regularity-preserving MBOT that cannot be simulated by any chain of XTOP^{R}. The counterexample is even linguistically motivated in the sense that it abstractly represents *topicalization* (see Figure 3), which is a common form of discontinuity.

Example 5. Let Tpc $= (N, \Sigma, S, P)$ be the MBOT with $N = \{S, T, T', T'', U\}$, symbols $\Sigma = \{\sigma, \delta, \gamma, \alpha\}$, and the productions P illustrated in Figure 3. It is clearly regularity-preserving because it is straightforward to develop two n-

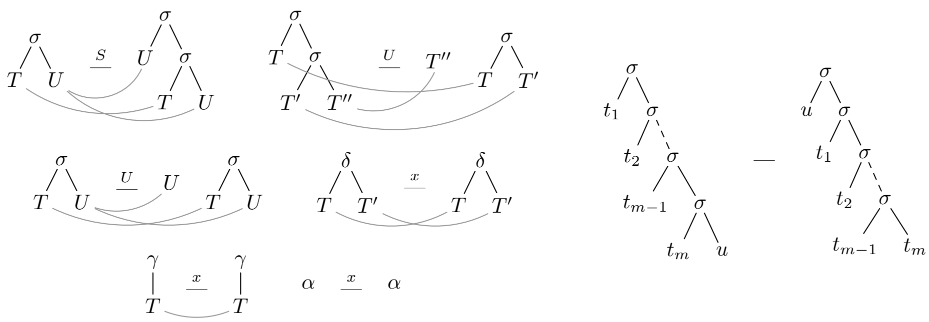

Fig. 3. Productions of the counterexample MBOT Tpc with $x \in \{T, T', T''\}$ and relation (topicalization) computed by it for all $m \in \mathbb{N}$ and arbitrary trees $u, t_1, \ldots, t_m$, which can contain binary δ-symbols, unary γ-symbols, and nullary α-symbols.

XTOP G_1 and G_2 that compute transformations similar to topicalization (see Figure 3), but just preserving u and just preserving $t_1, \ldots, t_m$, respectively. Thus, the language Tpc(L) for a regular tree language L is obtained as $G_1(L) \cap G_2(L)$. Since n-XTOP preserve regularity [21], $G_1(L)$ and $G_2(L)$ are regular tree languages, and regular tree languages are closed under intersection [10, 11]. The relation computed by Tpc is depicted in Figure 3.

Theorem 6. *The relation Tpc cannot be computed by any chain of* XTOP$^{\mathrm{R}}$.

Proof (Sketch). We already remarked that 3 XTOP$^{\mathrm{R}}$ suffice to simulate any chain of XTOP$^{\mathrm{R}}$ according to [8]. Consequently, in order to derive a contradiction we assume that there exist 3 XTOP$^{\mathrm{R}}$ G_1, G_2, G_3 such that Tpc $= G_1 ; G_2 ; G_3$. We know that dep(G_1), dep(G_2), dep(G_3) are input and output link-distance bounded (see Table 1), so let $b \in \mathbb{N}$ be such that all link-distances (for all 3 XTOP$^{\mathrm{R}}$) are bounded by b. Using an involved technical argumentation based on the link properties and size arguments [9] (using only the symbols γ and α for the trees $u, t_1, \ldots, t_m$), we can deduce the existence of the light dependencies depicted in Figure 4 (for the input and output tree and two intermediate trees without the clearly marked links), in which $m \gg b^3$. Consequently, the ellipsis (clearly marked dots) in the output tree (last tree in Figure 4) hides at least b^2 links that point to this part of the output because there must be a link every b positions by the link-distance bound. Let $(v''_1, w''_1), \ldots, (v''_{m''}, w''_{m''})$ with $m'' \gg b^2$ be those links such that $w''_1 < \cdots < w''_{m''}$. These links are marked with (1) in Figure 4. Clearly, $w''_{m''}$ dominates (via $\leq$) the positions of the subtrees t_{m-1} and t_m, but it does not dominate that of the subtree u. The input positions of those links, which point to positions inside the third tree in Figure 4, automatically fulfill $v''_1 \leq \cdots \leq v''_{m''}$ since dep(G_3) is strictly output hierarchical. A straightforward induction can be used to show that (for any XTOP$^{\mathrm{R}}$) all links sharing the same input positions must be incomparable with respect to the prefix order $\leq$ [9], which uses the restriction that $\ell \notin N$ for each production $\langle \ell, n, \boldsymbol{r} \rangle \in P$. Consequently, $v''_1 < \cdots < v''_{m''}$. Similarly, we can conclude $v''_{m''} < w'_{t_{m-1}}$, $v''_{m''} < w'_{t_m}$, and $v''_1 \not\leq w'_u$, where the

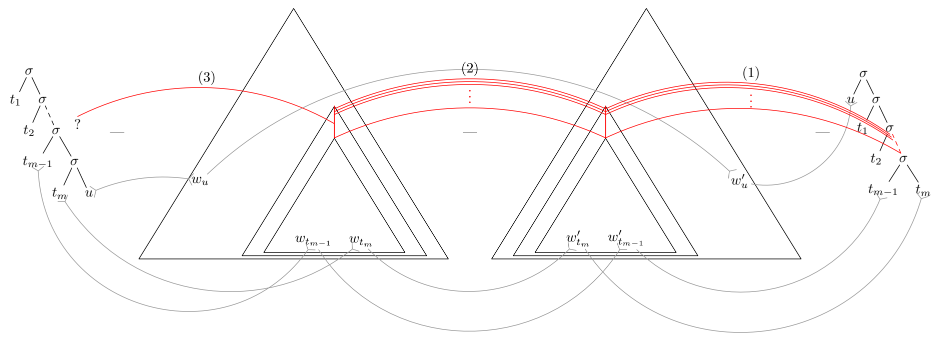

Fig. 4. Illustration of the dependencies discussed in the proof of Theorem 6. Inverted arrow heads indicate that the link points to a position below the one indicated by the spline. The links relating the roots of the trees are omitted.

last statement uses that $\text{dep}(G_3)$ is strictly input hierarchical. Repeating essentially the same arguments for $\text{dep}(G_2)$, we obtain links $(v'_1, w'_1), \ldots, (v'_{m'}, w'_{m'})$ with $m' \gg b$ such that $v''_1 \leq w'_1 < \cdots < w'_{m'} \leq v''_{m''}$ and $v'_1 < \cdots < v'_{m'}$. These links are labeled by (2) in Figure 4. Moreover, $v'_{m'} < w_{t_{m-1}}$, $v'_{m'} < w_{t_m}$, and $v'_1 \not\leq w_u$. Using the arguments once more for $\text{dep}(G_1)$, we obtain a link (v, w) such that $v'_1 \leq w \leq v'_{m'}$. This final link is marked (3) in Figure 4. Moreover, we have $v < v_{t_{m-1}}$ and $v < v_{t_m}$, but $v \not\leq v_u$. However, such a position does not exist in the input tree, which completes the desired contradiction. □

It is noteworthy that the proof can be achieved using high-level arguments based on the links and their properties. In fact, the whole proof is rather straightforward once the basic links (light in Figure 4) are established using [9]. Arguably, the omitted part of the proof that establishes those links is quite technical and involved (using size arguments and thus the particular shape of the trees $u, t_1, \ldots, t_m$), but it can be reused in similar setups as it generally establishes links in the presented way between identical subtrees (for which infinitely many trees are possible). The proof nicely demonstrates that the difficult argumentation via two unknown intermediate trees reduces to (relatively) simple arguments with the help of the links. The author believes that the links will provide a powerful and versatile tool in the future and have been neglected for too long. From Theorem 6 it follows that (some) topicalizations cannot be computed by any chain of XTOP$^{\text{R}}$ (or any chain of n-XTOP), and since Tpc is computed by a regularity-preserving MBOT, we can conclude that regularity-preserving MBOT are strictly more powerful than chains of XTOP$^{\text{R}}$.

Corollary 7. *Regularity-preserving* MBOT *are strictly more powerful than composition chains of* XTOP$^{\text{R}}$ *(and composition chains of n-*XTOP*).*

Our next result will limit the expressive power of MBOT. Using the linking technique [9] once more (this time for MBOT), we will prove that the inverse relation Tpc^{-1} cannot be computed by any MBOT. This confirms the bottom-up

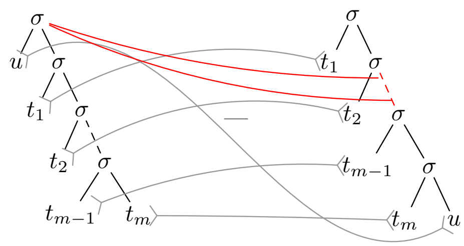

Fig. 5. Illustration of the dependencies discussed in the proof of Theorem 8. Inverted arrow heads indicate that the link points to a position below the one indicated by the spline. The links relating the roots of the trees are omitted.

nature of the device. It can "grab" deeply nested subtrees and transport them towards the root, but it cannot achieve the converse.

Theorem 8. *The relation Tpc^{-1} cannot be computed by any (composition chain of)* MBOT*.*

Proof (Sketch). Since MBOT are closed under composition [6], we need to consider only a single MBOT. In order to derive a contradiction, let $G = (N, \Sigma, S, P)$ be an MBOT such that $G = \text{Tpc}^{-1}$. As usual, we know that $\text{dep}(G)$ is input and output link-distance bounded (see Table 1), so let $b \in \mathbb{N}$ be a suitable bound. Moreover, let $a > |\boldsymbol{r}|$ for all productions $\langle \ell, n, \boldsymbol{r} \rangle \in P$. Hence a is an upper bound for the length of the right-hand sides. Finally, let $k > \max(a, b)$ be our main constant.

Again we need to use a (different, but similar) involved technical argumentation [9] based on the link properties and size and height arguments (that uses the symbols δ, γ, and α for the subtrees $u, t_1, \ldots, t_m$) to deduce the existence of the light dependencies shown in Figure 5, in which $m \gg 2k$. Consequently, the ellipsis (clearly marked dots) in the output tree hides at least 2 links that point to this part of the output because there must be a link every b positions by the link-distance bound. Let $(v, w), (v', w')$ be those links such that $w < w'$. These links are clearly indicated in Figure 5.

Clearly, w' dominates the positions of the subtrees t_m and u. Since $\text{dep}(G)$ is strictly output hierarchical (see Table 1), we obtain that (i) $v \leq v'$ and (ii) v' dominates the input positions of the (light) links pointing into the subtrees t_m and u. Obviously, the root ε is the only suitable position, so $v = v' = \varepsilon$ as indicated in Figure 5. Another straightforward induction can be used to show that (for any MBOT) all links sharing the same input positions must be incomparable with respect to the prefix order $\leq$ [9], which uses the restriction that $\ell \notin N$ for each production $\langle \ell, n, \boldsymbol{r} \rangle \in P$. However, (ε, w) and (ε, w') are two links with the same source and comparable target because $w < w'$, so we derived the desired contradiction. □

Again we note that the proof could be straightforwardly achieved using high-level arguments on the links and their interrelation after establishing the basic

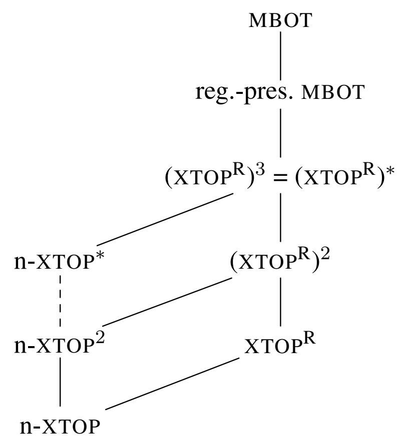

Fig. 6. Hasse diagram for the discussed classes, where $\mathcal{C}^*$ is the composition closure of class $\mathcal{C}$ and the dashed line just indicates that all powers in between form a chain and are thus strictly contained as well.

links (light in Figure 5). Then the link-distance can be used to conclude the existence of links and their input and output target can be related to existing links using the hierarchical properties. In this way, we could in both cases derive a contradiction in rather straightforward ways, which would not have been possible without the links. Typically, such (negative) statements are proved using the fooling technique (see [1] or [21] for examples), which requires a rather detailed case analysis of all possible intermediate trees and applied productions, which then individually have to be contradicted. In a scenario with 2 unknown intermediate trees such an approach becomes (nearly) impossible to handle. Thus, we strongly want to promote the use of links and their interrelations in the analysis of translation models.

Theorem 8 yields that regularity-preserving MBOT are not closed under inversion. In other words, there are regularity-preserving MBOT G (such as Tpc), whose inverted computed relation G^{-1} cannot be computed by any MBOT.

Corollary 9. *Regularity-preserving* MBOT *(and general* MBOT*) are not closed under inversion.*

Let us now collect these results together with some minor consequences in a Hasse diagram (see Figure 6).

References

1. Arnold, A., Dauchet, M.: Morphismes et bimorphismes d'arbres. Theoret. Comput. Sci. 20(1), 33–93 (1982)
2. Braune, F., Seemann, N., Quernheim, D., Maletti, A.: Shallow local multi bottom-up tree transducers in statistical machine translation. In: Proc. 51st ACL. pp. 811–821. Association for Computational Linguistics (2013)

3. Chiang, D.: An introduction to synchronous grammars. In: Proc. 44th ACL. Association for Computational Linguistics (2006), part of a tutorial given with Kevin Knight
4. Chiang, D.: Hierarchical phrase-based translation. Comput. Linguist. 33(2), 201–228 (2007)
5. Eisner, J.: Learning non-isomorphic tree mappings for machine translation. In: Proc. 41st ACL. pp. 205–208. Association for Computational Linguistics (2003)
6. Engelfriet, J., Lilin, E., Maletti, A.: Composition and decomposition of extended multi bottom-up tree transducers. Acta Inform. 46(8), 561–590 (2009)
7. Engelfriet, J., Rozenberg, G., Slutzki, G.: Tree transducers, L systems, and two-way machines. J. Comput. System Sci. 20(2), 150–202 (1980)
8. Fülöp, Z., Maletti, A.: Composition closure of ε-free linear extended top-down tree transducers. In: Proc. 17th DLT. LNCS, vol. 7907, pp. 239–251. Springer (2013)
9. Fülöp, Z., Maletti, A.: Linking theorems for tree transducers (2014), manuscript
10. Gécseg, F., Steinby, M.: Tree Automata. Akadémiai Kiadó, Budapest (1984)
11. 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)
12. Gildea, D.: On the string translations produced by multi bottom-up tree transducers. Comput. Linguist. 38(3), 673–693 (2012)
13. de Gispert, A., Iglesias, G., Blackwood, G.W., Banga, E.R., Byrne, W.: Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars. Comput. Linguist. 36(3), 505–533 (2010)
14. 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)
15. Koehn, P.: Statistical Machine Translation. Cambridge University Press (2010)
16. 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. 45th ACL. pp. 177–180. Association for Computational Linguistics (2007)
17. 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)
18. Maletti, A.: Why synchronous tree substitution grammars? In: Proc. HLT-NAACL 2010. pp. 876–884. Association for Computational Linguistics (2010)
19. Maletti, A.: How to train your multi bottom-up tree transducer. In: Proc. 49th ACL. pp. 825–834. Association for Computational Linguistics (2011)
20. Maletti, A.: Tree transformations and dependencies. In: Proc. 12th MOL. LNAI, vol. 6878, pp. 1–20. Springer (2011)
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. 48th ACL. pp. 1058–1066. Association for Computational Linguistics (2010)
23. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. Comput. Linguist. 30(4), 417–449 (2004)
24. Shieber, S.M., Schabes, Y.: Synchronous tree-adjoining grammars. In: Proc. 13th CoLing. vol. 3, pp. 253–258 (1990)
25. 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)