Every sensible extended top-down tree transducer is a multi bottom-up tree transducer

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

Abstract

A tree transformation is sensible if the size of each output tree is uniformly bounded by a linear function in the size of the corresponding input tree. Every sensible tree transformation computed by an arbitrary weighted extended top-down tree transducer can also be computed by a weighted multi bottom-up tree transducer. This further motivates weighted multi bottom-up tree transducers as suitable translation models for syntax-based machine translation.

1 Introduction

Several different translation models are used in syntax-based statistical machine translation. Koehn (2010) presents an introduction to statistical machine translation, and Knight (2007) presents an overview of syntax-based statistical machine translation. The oldest and best-studied tree transformation device is the top-down tree transducer of Rounds (1970) and Thatcher (1970). Gécseg and Steinby (1984) and Fülöp and Vogler (2009) present the existing results on the unweighted and weighted model, respectively. Knight (2007) promotes the use of weighted extended top-down tree transducers (XTOP), which have also been implemented in the toolkit TIBURON by May and Knight (2006) [more detail is reported by May (2010)]. In the context of bimorphisms, Arnold and Dauchet (1976) investigated XTOP, and Lilin (1978) and Arnold and Dauchet (1982) investigated multi bottom-up tree transducers (MBOT) [as k-morphisms]. Recently, weighted XTOP and MBOT, which are the central devices in this contribution, were investigated by Maletti (2011a) in the context of statistical machine translation.

Several tree transformation devices are used as translation models in statistical machine translation. Chiang (2007) uses synchronous context-free grammars, which force translations to be very similar as observed by Eisner (2003) and Shieber (2004). This deficiency is overcome by synchronous tree substitution grammars, which are state-less linear and nondeleting XTOP. Recently, Maletti (2010b) proposed MBOT, and Zhang et al. (2008b) and Sun et al. (2009) proposed the even more powerful synchronous tree-sequence substitution grammars. Those two models allow certain translation discontinuities, and the former device also offers computational benefits over linear and nondeleting XTOP as argued by Maletti (2010b).

The simplicity of XTOP makes them very appealing as translation models. In 2010 the ATANLP participants [workshop at ACL] identified 'copying' as the most exciting and promising feature of XTOP, but unrestricted copying can lead to an undesirable explosion of the size of the translation. According to Engelfriet and Maneth (2003) a tree transformation has linear size-increase if the size of each output tree is linearly bounded by the size of its corresponding input tree. The author believes that this is a very sensible restriction that intuitively makes sense and at the same time suitably limits the copying power of XTOP.

We show that every sensible tree transformation

[*] The author was supported by the German Research Foundation (DFG) grant MA 4959/1-1.

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 263–273,
Montréal, Canada, June 3–8, 2012. ©2012 Association for Computational Linguistics

that can be computed by an XTOP can also be computed by an MBOT. For example, linear XTOP (i.e., no copying) compute only sensible tree transformations, and Maletti (2008) shows that for each linear XTOP there exists an equivalent MBOT. Here, we do not make any restrictions on the XTOP besides some sanity conditions (see Section 3). In particular, we consider copying XTOP. If we accept the restriction to linear size-increase tree transformation, then our main result further motivates MBOT as a suitable translation model for syntax-based machine translation because MBOT can implement each reasonable (even copying) XTOP. In addition, our result allows us to show that each reasonable XTOP preserves regularity under backward application. As demonstrated by May et al. (2010) backward application is the standard application of XTOP in the machine translation pipeline, and preservation of regularity is the essential property for several of the evaluation algorithms of May et al. (2010).

2 Notation

We start by introducing our notation for trees, whose nodes are labeled by elements of an alphabet Σ and a set V. However, only leaves can be labeled by elements of V. For every set T, we let

$$\Sigma(T) = \{\sigma(t_1, \dots, t_k) \mid \sigma \in \Sigma, t_1, \dots, t_k \in T\} \enspace ,$$

which contains all trees with a Σ-labeled root and direct successors in T. The set $T_\Sigma(V)$ of *Σ-trees with V-leaves* is the smallest set T such that $V \cup \Sigma(T) \subseteq T$. We use $X = \{x_1, x_2, \dots\}$ as a set of formal *variables*.

Each node of the tree $t \in T_\Sigma(V)$ is identified by a *position* $p \in \mathbb{N}_+$, which is a sequence of positive integers. The root is at position ε (the empty string), and the position ip with $i \in \mathbb{N}_+$ and $p \in \mathbb{N}_+^*$ is the position p in the i-th direct subtree. The set $\mathrm{pos}(t)$ contains all positions of t, and the *size* of t is $|t| = |\mathrm{pos}(t)|$. For each $p \in \mathrm{pos}(t)$, the *label* of t at p is $t(p)$. Given a set $L \subseteq \Sigma \cup V$ of labels, we let $\mathrm{pos}_L(t) = \{p \in \mathrm{pos}(t) \mid t(p) \in L\}$ be the positions with L-labels. We write $\mathrm{pos}_l(t)$ for $\mathrm{pos}_{\{l\}}(t)$ for each $l \in L$. Finally, we write $t[u]_p$ for the tree obtained from t by replacing the subtree at position p by the tree $u \in T_\Sigma(V)$.

The following notions refer to the variables X. The tree $t \in T_\Sigma(V)$ [potentially $V \cap X = \emptyset$] is *linear* if every $x \in X$ occurs at most once in t (i.e., $|\mathrm{pos}_x(t)| \leq 1$). Moreover,

$$\mathrm{var}(t) = \{x \in X \mid \mathrm{pos}_x(t) \neq \emptyset\}$$

contains the variables that occur in t. A *substitution* θ is a mapping $\theta \colon X \to T_\Sigma(V)$. When applied to t, it returns the tree $t\theta$, which is obtained from t by replacing all occurrences of $x \in X$ in t by $\theta(x)$. Our notions for trees are illustrated in Figure 1.

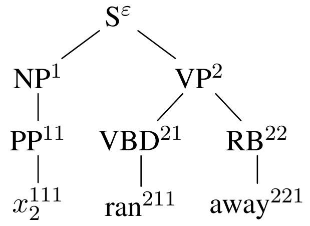

Figure 1: The tree t (with positions indicated as superscripts) is linear and $\mathrm{var}(t) = \{x_2\}$. The tree $t[\mathrm{He}]_{111}$ is the same tree with x_2 replaced by 'He'.

Finally, we present *weighted tree grammars* (WTG) as defined by Fülöp and Vogler (2009), who defined it for arbitrary semirings as weight structures. In contrast, our weights are always nonnegative reals, which form the semiring $(\mathbb{R}_+, +, \cdot, 0, 1)$ and are used in probabilistic grammars. For each weight assignment $f \colon T \to \mathbb{R}_+$, we let

$$\mathrm{supp}(f) = \{t \in T \mid f(t) \neq 0\} \enspace .$$

WTG offer an efficient representation of weighted forests (i.e., set of weighted trees), which is even more efficient than the packed forests of Mi et al. (2008) because they can be minimized efficiently using an algorithm of Maletti and Quernheim (2011). In particular, WTG can share more than equivalent subtrees and can even represent infinite sets of trees. A WTG is a system $G = (Q, \Sigma, q_0, P, \mathrm{wt})$ with

- a finite set Q of *states* (nonterminals),
- an alphabet Σ of *symbols*,
- a *starting state* $q_0 \in Q$,
- a finite set P of *productions* $q \to r$, where $q \in Q$ and $r \in T_\Sigma(Q) \setminus Q$, and
- a mapping $\mathrm{wt} \colon P \to \mathbb{R}_+$ that assigns *production weights*.

Without loss of generality, we assume that we can distinguish states and symbols (i.e., $Q \cap \Sigma = \emptyset$). For all $\xi, \zeta \in T_\Sigma(Q)$ and a production $\rho = q \to r$,

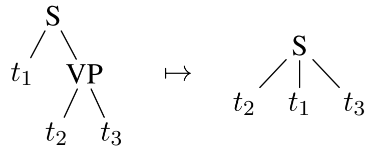

Figure 2: Example rotation. In principle, such rotations are required in the translation from English to Arabic.

we write $\xi \Rightarrow_G^\rho \zeta$ if $\xi = \xi[q]_p$ and $\zeta = \xi[r]_p$, where p is the lexicographically least element of $\mathrm{pos}_Q(\xi)$. The WTG G *generates* the weighted tree language $L_G \colon T_\Sigma \to \mathbb{R}_+$ such that

$$L_G(t) = \sum_{\substack{n \in \mathbb{N}, \rho_1, \dots, \rho_n \in P \\ q_0 \Rightarrow_G^{\rho_1} \cdots \Rightarrow_G^{\rho_n} t}} \mathrm{wt}(\rho_1) \cdot \ldots \cdot \mathrm{wt}(\rho_n)$$

for every $t \in T_\Sigma$. Each such language is *regular*, and $\mathrm{Reg}(\Sigma)$ contains all those languages over the alphabet Σ. A thorough introduction to tree languages is presented by Gécseg and Steinby (1984) and Gécseg and Steinby (1997) for the unweighted case and by Fülöp and Vogler (2009) for the weighted case.

3 Extended top-down tree transducers

We start by introducing the main model of this contribution. Extended top-down tree transducers (XTOP) are a generalization of the *top-down tree transducers* (TOP) of Rounds (1970) and Thatcher (1970). XTOP allow rules with several (non-state and non-variable) symbols in the left-hand side (as in the rule of Figure 3), whereas a TOP rule contains exactly one symbol in the left-hand side. Shieber (2004) and Knight (2007) identified that this extension is essential for many NLP applications because without it linear (i.e., non-copying) cannot compute rotations (see Figure 2). In the form of bimorphisms XTOP were investigated by Arnold and Dauchet (1976) and Arnold and Dauchet (1982) in the 1970s, and Knight (2007) invigorated research.

As demonstrated by Graehl et al. (2009) the most general XTOP model includes copying, deletion, and regular look-ahead in the spirit of Engelfriet (1977). More powerful models (such as synchronous tree-sequence substitution grammars and multi bottom-up tree transducers) can handle translation discontinuities naturally as evidenced by Zhang et al. (2008a) and Maletti (2011b), but XTOP need copying and deletion to handle them. Copying essentially allows an XTOP to translate certain parts of the input several times and was identified by the ATANLP 2010 participants as one of the most interesting and promising features of XTOP. Currently, the look-ahead feature is not used in machine translation, but we need it later on in the theoretical development.

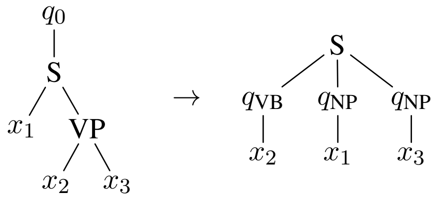

Figure 3: Example XTOP rule by Graehl et al. (2008).

Given an alphabet Q and a set T, we let

$$Q[T] = \{q(t) \mid q \in Q, t \in T\},$$

in which the root always has exactly one successor from T in contrast to $Q(T)$. We treat elements of $Q[T_\Sigma(V)]$ as special trees of $T_{\Sigma \cup Q}(V)$. Moreover, we let $\widetilde{1}_\Sigma(t) = 1$ for every $t \in T_\Sigma$. XTOP with regular look-ahead (XTOP$^{\mathrm{R}}$) were also studied by Knight and Graehl (2005) and Graehl et al. (2008). Formally, an XTOP$^{\mathrm{R}}$ is a system

$$M = (Q, \Sigma, \Delta, q_0, R, c, \mathrm{wt})$$

with

- a finite set Q of *states*,
- alphabets Σ and Δ of *input* and *output symbols*,
- a *starting state* $q_0 \in Q$,
- a finite set R of *rules* of the form $\ell \to r$ with linear $\ell \in Q[T_\Sigma(X)]$ and $r \in T_\Delta(Q[\mathrm{var}(\ell)])$,
- $c \colon R \times X \to \mathrm{Reg}(\Sigma)$ assigns a regular *look-ahead* to each deleted variable of a rule [i.e., $c(\ell \to r, x) = \widetilde{1}_\Sigma$ for all $\ell \to r \in R$ and $x \in X \setminus (\mathrm{var}(\ell) \setminus \mathrm{var}(r))$], and
- $\mathrm{wt} \colon R \to \mathbb{R}_+$ assigns *rule weights*.

The XTOP$^{\mathrm{R}}$ M is *linear* [respectively, *nondeleting*] if r is linear [respectively, $\mathrm{var}(\ell) = \mathrm{var}(r)$] for every rule $\ell \to r \in R$. It has *no look-ahead* (XTOP) if $c(\rho, x) = \widetilde{1}_\Sigma$ for all $\rho \in R$ and $x \in X$. Figure 3 shows a rule of a linear and nondeleting XTOP.

The look-ahead can be used to restrict rule applications. It can inspect subtrees that are deleted by a

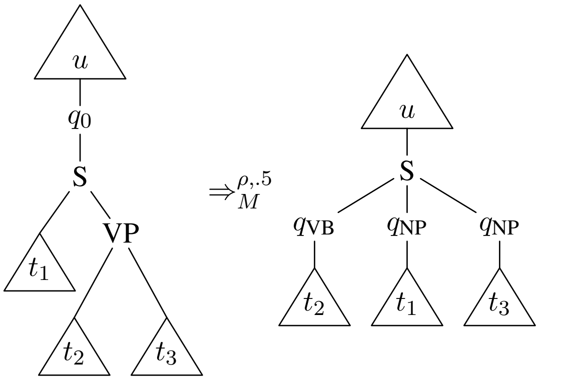

Figure 4: Rewrite step using rule ρ of Figure 3.

rule application, so for each rule $\rho = \ell \to r$, we let $\mathrm{del}(\rho) = \mathrm{var}(\ell) \setminus \mathrm{var}(r)$ be the set of deleted variables in ρ. If we suppose that a variable $x \in \mathrm{del}(\rho)$ matches to an input subtree t, then the weight of the look-ahead $c(\rho, x)(t)$, which we also write $c_{\rho,x}(t)$, is applied to the derivation. If it is 0, then this look-ahead essentially prohibits the application of ρ. It is important that the look-ahead is regular (i.e., there exists a WTG accepting it). The toolkit TIBURON by May and Knight (2006) implements XTOP together with a number of essential operations. Look-ahead is not implemented in TIBURON, but it can be simulated using a composition of two XTOP, in which the first XTOP performs the look-ahead and marks the results, so that the second XTOP can access the look-ahead information.

As for WTG the semantics for the XTOP$^{\text{R}}$ $M = (Q, \Sigma, \Delta, I, R, c, \mathrm{wt})$ is presented using rewriting. Without loss of generality, we again suppose that $Q \cap (\Sigma \cup \Delta) = \emptyset$. Let $\xi, \zeta \in T_\Delta(Q[T_\Sigma])$, $w \in \mathbb{R}_+$, and $\rho = \ell \to r$ be a rule of R. We write $\xi \Rightarrow_M^{\rho,w} \zeta$ if there exists a substitution $\theta\colon X \to T_\Sigma$ such that

- $\xi = \xi[\ell\theta]_p$,
- $\zeta = \xi[r\theta]_p$, and
- $w = \mathrm{wt}(\rho) \cdot \prod_{x\in\mathrm{del}(\rho)} c_{\rho,x}(x\theta)$,

where $p \in \mathrm{pos}_Q(\xi)$ is the lexicographically least Q-labeled position in ξ. Figure 4 illustrates a derivation step.

The XTOP$^{\text{R}}$ M computes a weighted tree transformation by applying rewrite steps to the tree $q_0(t)$, where $t \in T_\Sigma$ is the input tree, until an output tree $u \in T_\Delta$ has been produced. The weight of a particular derivation is obtained by multiplying the weights of the rewrite steps. The weight of the transformation from t to u is obtained by summing all weights of the derivations from $q_0(t)$ to u. Formally[1], the *weighted tree transformation computed by M* in state $q \in Q$ is

$$\tau_M^q(t,u) = \sum_{\substack{n\in\mathbb{N}, \rho_1,\dots,\rho_n\in R \\ q(t)\Rightarrow_M^{\rho_1,w_1}\cdots\Rightarrow_M^{\rho_n,w_n} u}} w_1 \cdot \ldots \cdot w_n \tag{1}$$

for every $t \in T_\Sigma$ and $u \in T_\Delta$. The XTOP$^{\text{R}}$ M computes the weighted tree transformation $\tau_M^{q_0}$. Two XTOP$^{\text{R}}$ M and N are equivalent, if $\tau_M = \tau_N$.

The sum (1) can be infinite, which we avoid by simply requiring that all our XTOP$^{\text{R}}$ are *producing*, which means that $r \notin Q[X]$ for every rule $\ell \to r \in R$.[2] In a producing XTOP$^{\text{R}}$ each rule application produces at least one output symbol, which limits the number n of rule applications to the size of the output tree u. A detailed exposition to XTOP$^{\text{R}}$ is presented by Arnold and Dauchet (1982) and Graehl et al. (2009) for the unweighted case and by Fülöp and Vogler (2009) for the weighted case.

Example 1. Let $M_{\text{ex}} = (Q, \Sigma, \Sigma, q, R, c, \mathrm{wt})$ be the nondeleting XTOP with

- $Q = \{q\}$,
- $\Sigma = \{\sigma, \gamma, \alpha\}$,
- the two rules

$$q(\alpha) \to \alpha \tag{ρ}$$

$$q(\gamma(x_1)) \to \sigma(q(x_1), q(x_1)) \tag{ρ'}$$

- trivial look-ahead (i.e., $c(\rho, x) = \widetilde{1}_\Sigma$), and
- $\mathrm{wt}(\rho) = 2$ and $\mathrm{wt}(\rho') = 1$.

The XTOP$^{\text{R}}$ M_{ex} computes the tree transformation that turns the input tree $\gamma^n(\alpha)$ into the fully balanced binary tree u of the same height with weight $2^{(2^n)}$. An example derivation is presented in Figure 5. □

Unrestricted copying (as in Example 1) yields very undesirable phenomena and is most likely not needed in the machine translation task. In fact, it is almost universally agreed that a translation model should be "linear-size increase", which means that

[1] There is an additional restriction that is discussed in the next paragraph.

[2] This is a convenience requirement. We can use other conditions on the XTOP$^{\text{R}}$ or the used weight structures to guarantee a well-defined semantics.

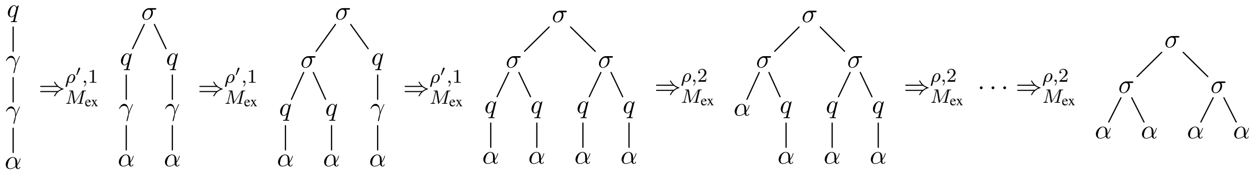
q γ γ α
$\Rightarrow^{\rho',1}_{M_{\text{ex}}}$
σ q q γ γ α α
$\Rightarrow^{\rho',1}_{M_{\text{ex}}}$
σ σ q q q γ α α α
$\Rightarrow^{\rho',1}_{M_{\text{ex}}}$
σ σ σ q q q q α α α α
$\Rightarrow^{\rho,2}_{M_{\text{ex}}}$
σ σ σ α q q q α α α
$\Rightarrow^{\rho,2}_{M_{\text{ex}}} \cdots \Rightarrow^{\rho,2}_{M_{\text{ex}}}$
σ σ σ α α α α

Figure 5: Example derivation using the XTOP M_{ex} with weight $1^3 \cdot 2^4 = 16$.

the size of each output tree should be linearly bounded in the size of the corresponding input tree according to Aho and Ullman (1971) and Engelfriet and Maneth (2003).

Definition 2. A mapping $\tau\colon T_\Sigma \times T_\Delta \to \mathbb{R}_+$ is *linear-size increase* if there exists an integer $n \in \mathbb{N}$ such that $|u| \leq n \cdot |t|$ for all $(t, u) \in \text{supp}(\tau)$. An XTOP$^{\text{R}}$ M is *sensible* if τ_M is linear-size increase. □

'Sensible' is not a syntactic property of an XTOP$^{\text{R}}$ as it does not depend on the actual rules, but only on its computed weighted tree transformation. The XTOP M_{ex} of Example 1 is not sensible because $|u| = 2^{|t|} - 1$ for every $(t, u) \in \tau_{M_{\text{ex}}}$. Intuitively, the number of times that M_{ex} can use the copying rule ρ' is not uniformly bounded.

We need an auxiliary result in the main part. Let $\tau\colon T_\Sigma \times T_\Delta \to \mathbb{R}_+$ be a weighted tree transformation. We need the weighted tree language $\tau^{-1}(u)\colon T_\Sigma \to \mathbb{R}_+$ of input trees weighted by their translation weight to a given output tree $u \in T_\Delta$. Formally, $\bigl(\tau^{-1}(u)\bigr)(t) = \tau(t, u)$ for every $t \in T_\Sigma$.

Theorem 3. For every producing XTOP$^{\text{R}}$ M and output tree $u' \in T_\Delta$, the weighted tree language $\tau_M^{-1}(u')$ is regular.

Proof sketch. We use some properties that are only defined in the next sections (for proof economy). It is recommended to skip this proof on the first reading and revisit it later. Maletti (2010a) shows that we can construct an XTOP$^{\text{R}}$ M' such that

$$\tau_{M'}(t, u) = \begin{cases} \tau_M(t, u) & \text{if } u' = u \\ 0 & \text{otherwise} \end{cases}$$

for every $t \in T_\Sigma$ and $u \in T_\Delta$. This operation is called 'output product' by Maletti (2010a). The obtained XTOP$^{\text{R}}$ M' is also producing, so we know that M' can take at most $|u'|$ rewrite steps to derive u'. Since M' can only produce the output tree u', this also limits the total number of rule applications in any successful derivation. Consequently, M' can only apply a copying rule at most $|u'|$ times, which shows that M' is finitely copying (see Definition 8). By Theorem 11 we can implement M' by an equivalent MBOT M'' (i.e., $\tau_{M''} = \tau_{M'}$; see Section 5), for which we know by Theorem 14 of Maletti (2011a) that $\tau_{M''}^{-1}(u) = \tau_{M'}^{-1}(u)$ is regular. □

Finally, let us illustrate the overall structure of our arguments to show that every sensible XTOP$^{\text{R}}$ can be implemented by an equivalent MBOT. We first normalize the given XTOP$^{\text{R}}$ such that the semantic property 'sensible' yields a syntactic property called 'finitely copying' (see Section 4). In a second step, we show that each finitely copying XTOP$^{\text{R}}$ can be implemented by an equivalent MBOT (see Section 5). Figure 6 illustrates these steps towards our main result. In the final section, we derive some consequences from our main result (see Section 6).

4 From sensible to finite copying

First, we adjust a normal form of Engelfriet and Maneth (2003) to our needs. This section borrows heavily from Aho and Ullman (1971) and Engelfriet and Maneth (2003), where "sensible" (unweighted) deterministic macro tree transducers (MAC) [see Engelfriet and Vogler (1985)] are considered. Our setting is simpler on the one hand because XTOP$^{\text{R}}$ do not have context parameters as MAC, but more difficult on the other hand because we consider nondeterministic and weighted transducers.

Intuitively, a sensible XTOP$^{\text{R}}$ cannot copy a lot since the size of each output tree is linearly bounded in the size of the corresponding input tree. However, the actual presentation of the XTOP$^{\text{R}}$ M might con-

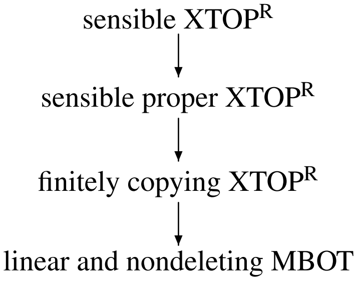
sensible XTOP$^{\mathrm{R}}$
sensible proper XTOP$^{\mathrm{R}}$
finitely copying XTOP$^{\mathrm{R}}$
linear and nondeleting MBOT

Figure 6: Overview of the proof steps.

tain rules that allow unbounded copying. This unbounded copying might not manifest due to the look-ahead restrictions or due to the fact that those rules cannot be used in a successful derivation. The purpose of the normal form is the elimination of those artifacts. To this end, we eliminate all states (except the initial state) that can only produce finitely many outputs. Such a state can simply be replaced by one of the output trees that it can produce and an additional look-ahead that checks whether the current input tree indeed allows that translation (and inserts the correct translation weight).

Normalized XTOP$^{\mathrm{R}}$ are called 'proper', and we define this property next. For the rest of this section, let $M = (Q, \Sigma, \Delta, q_0, R, c, \mathrm{wt})$ be the considered sensible XTOP$^{\mathrm{R}}$. Without loss of generality, we assume that the state q_0 does not occur in the right-hand sides of rules. Moreover, we write $\xi \Rightarrow_M^* \zeta$ if there exist nonzero weights $w_1, \dots, w_n \in \mathbb{R}_+ \setminus \{0\}$ and rules $\rho_1, \dots, \rho_n \in R$ with

$$\xi \Rightarrow_M^{\rho_1, w_1} \cdots \Rightarrow_M^{\rho_n, w_n} \zeta \enspace .$$

In essence, $\xi \Rightarrow_M^* \zeta$ means that M can transform ξ into ζ (in the unweighted setting).

Definition 4. A state $q \in Q$ is *proper* if there are infinitely many $u' \in T_\Delta$ such that there exists a derivation

$$q_0(t) \Rightarrow_M^* \xi[q(s)]_p \Rightarrow_M^* u[u']_p$$

where $s, t \in T_\Sigma$ are input trees, $\xi \in T_\Delta(Q[T_\Sigma])$, $p \in \mathrm{pos}(\xi)$, and $u \in T_\Delta$ is an output tree. □

The derivation in Definition 4 is illustrated in Figure 7. In other words, a proper state is reachable from the initial state and can transform infinitely many input trees into infinitely many output trees. The latter is an immediate consequence of Definition 4 since each input tree can be transformed into only finitely many output trees due to sensibility. The restriction includes the look-ahead (because we require that the rewrite step weights are nonzero), which might further restrict the input trees.

Example 5. The state q of the XTOP M_{ex} is proper because we already demonstrated that it can transform infinitely many input trees into infinitely many output trees. □

The XTOP$^{\mathrm{R}}$ M is *proper* if all its states except the initial state q_0 are proper. Next, we show that each XTOP$^{\mathrm{R}}$ can be transformed into an equivalent proper XTOP$^{\mathrm{R}}$ using a simplified version of the construction of Lemma 5.4 by Engelfriet and Maneth (2003). Mind that we generally assume that all considered XTOP$^{\mathrm{R}}$ are producing.

Theorem 6. For every XTOP$^{\mathrm{R}}$ there exists an equivalent proper XTOP$^{\mathrm{R}}$.

Proof sketch. The construction is iterative. Suppose that M is not yet proper. Then there exists a state $q \in Q$, which can produce only finitely many outputs U. It can be decided whether a state is proper using Theorem 4.5 of Drewes and Engelfriet (1998), and in case it is proper, the set U can also be computed effectively. The cited theorem applies to unweighted XTOP$^{\mathrm{R}}$, but it can be applied also in our setting because $\Rightarrow_M^*$ in Definition 4 disregards weights. Now we consider each $u \in U$ individually. Clearly, $(\tau_M^q)^{-1}(u)$ is regular by Theorem 3. For each u and each occurrence of q in the right-hand side of a rule $\rho \in R$ of M, we create a copy ρ' of ρ, in which the selected occurrence of $q(x)$ is replaced by u and the new look-ahead is $c(\rho', x) = c(\rho, x) \cdot (\tau_M^q)^{-1}(u)$, which restricts the input tree appropriately and includes the adjustment of the weights. Since regular weighted tree languages are closed under HADAMARD products [see Fülöp and Vogler (2009)], the look-ahead $c(\rho, x) \cdot (\tau_M^q)^{-1}(u)$ is again regular.

Essentially, we precompute the action of q as much as possible, and immediately output one of the finitely many output trees, check that the input tree has the required shape using the look-ahead, and charge the weight for the precomputed transformation again using the look-ahead. This process is done for each occurrence, so if a rule contains two occurrences of q, then the process must be

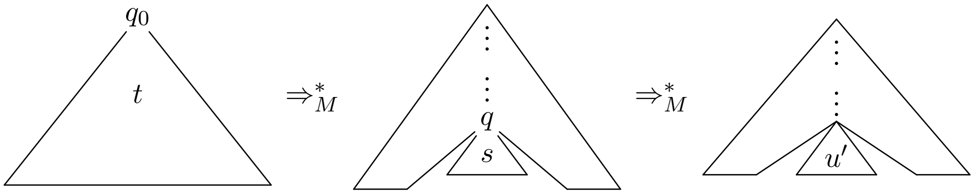

Figure 7: Illustration of the derivation in Definition 4.

done twice to this rule. In this way, we eventually purge all occurrences of q from the right-hand sides of rules of M without changing the computed transformation. Since $q \neq q_0$ and q is now unreachable, it is useless and can be deleted, which removes one non-proper state. This process is repeated until all states except the initial state q_0 are proper. □

Clearly, the construction of Theorem 6 applied to a sensible XTOP$^{\mathrm{R}}$ M yields a sensible proper XTOP$^{\mathrm{R}}$ M' since the property 'sensible' refers to the computed transformation and $\tau_M = \tau_{M'}$. Let us illustrate the construction on a small example.

Example 7. Let ρ be the rule displayed in Figure 3, and let us assume that the state q_{VB} is not proper. Moreover, suppose that q_{VB} can yield the output tree u and that we already computed the translation options that yield u. Let $t_1, \dots, t_n \in T_\Sigma$ be those translation options. Then we create the copy ρ'

$$q_0(\mathrm{S}(x_1, \mathrm{VP}(x_2, x_3))) \to \mathrm{S}(u, q_{\mathrm{NP}}(x_1), q_{\mathrm{NP}}(x_3))$$

of the rule ρ with look-ahead $c'(\rho', x)$ such that

$$c'_{\rho',x}(t) = \begin{cases} c_{\rho,x}(t) & \text{if } x \neq x_2 \\ \tau_M^{q_{\mathrm{VB}}}(t, u) & \text{if } x = x_2 \ . \end{cases}$$ □

In general, there can be infinitely many input trees t_i that translate to a selected output tree u, so we cannot simply replace the variable in the left-hand side by all the options for the input tree. This is the reason why we use the look-ahead because the set $\tau_M^{-1}(u)$ is a regular weighted tree language.

From now on, we assume that the XTOP$^{\mathrm{R}}$ M is proper. Next, we want to invoke Theorem 7.1 of Engelfriet and Maneth (2003) to show that a proper sensible XTOP$^{\mathrm{R}}$ is finitely copying. Engelfriet and Maneth (2003) present a formal definition of finite copying, but we only present a high-level description of it.

Definition 8. The XTOP$^{\mathrm{R}}$ M is *finitely copying* if there is a copying bound $n \in \mathbb{N}$ such that no input subtree is copied more than n times in any derivation $q(t) \Rightarrow_M^* u$ with $q \in Q$, $t \in T_\Sigma$, and $u \in T_\Delta$. □

Example 9. The XTOP of Example 1 is not finitely copying as the input subtree α is copied 2^n times if the input tree is $\gamma^n(\alpha)$. Clearly, this shows that there is no uniform bound on the number of copies. □

It is worth noting that the properties 'sensible' and 'finitely copying' are essentially unweighted properties. They largely disregard the weights and a weighted XTOP$^{\mathrm{R}}$ does have one of those properties if and only if its associated unweighted XTOP$^{\mathrm{R}}$ has it. We now use this tight connection to lift Theorem 7.1 of Engelfriet and Maneth (2003) from the unweighted (and deterministic) case to the weighted (and nondeterministic) case.

Theorem 10. If a proper XTOP$^{\mathrm{R}}$ is sensible, then it is finitely copying.

Proof. Let M be the input XTOP$^{\mathrm{R}}$. Since M is sensible, its associated unweighted XTOP$^{\mathrm{R}}$ N, which is obtained by setting all weights to 1 and computing in the BOOLEAN semiring, is sensible. Consequently, N is finitely copying by Theorem 7.1 of Engelfriet and Maneth (2003). Thus, also M is finitely copying, which concludes the proof. We remark that Theorem 7.1 of Engelfriet and Maneth (2003) only applies to deterministic XTOP$^{\mathrm{R}}$, but the essential pumping argument, which is Lemma 6.2 of Engelfriet and Maneth (2003) also works for nondeterministic XTOP$^{\mathrm{R}}$. Essentially, the pumping argument shows the contraposition. If M is not finitely copying, then M can copy a certain subtree an arbitrarily often. Due to the properness of M, all these copies have an impact on the output tree, which yields that its size grows beyond any uniform linear bound, which in turn demonstrates that M is not sensible. □

We showed that each sensible XTOP$^{\mathrm{R}}$ can be implemented by a finitely copying XTOP$^{\mathrm{R}}$ via the construction of the proper normal form. This approach actually yields a characterization because finitely copying XTOP$^{\mathrm{R}}$ are trivially sensible by Theorem 4.19 of Engelfriet and Maneth (2003).

5 From finite copying to an MBOT

We complete the argument by showing how to implement a finitely copying XTOP$^{\mathrm{R}}$ by a weighted multi bottom-up tree transducer (MBOT). First, we recall the MBOT, which was introduced by Arnold and Dauchet (1982) and Lilin (1978) in the unweighted case. Engelfriet et al. (2009) give an English presentation. We present the linear and nondeleting MBOT of Engelfriet et al. (2009).

A *weighted multi bottom-up tree transducer* is a system $M = (Q, \Sigma, \Delta, F, R, \mathrm{wt})$ with

- an alphabet Q of *states*,
- alphabets Σ and Δ of *input and output symbols*,
- a set $F \subseteq Q$ of *final states*,
- a finite set R of *rules* of the form $\ell \to r$ where $\ell \in T_\Sigma(Q(X))$ and $r \in Q(T_\Delta(X))$ are linear and $\mathrm{var}(\ell) = \mathrm{var}(r)$, and
- $\mathrm{wt}\colon R \to \mathbb{R}_+$ assigning *rule weights*.

We now use $T_\Sigma(Q(X))$ and $Q(T_\Delta(X))$ instead of $T_\Sigma(Q[X])$ and $Q[T_\Delta(X)]$, which highlights the difference between XTOP$^{\mathrm{R}}$ and MBOT. First, MBOT are a bottom-up device, which yields that Σ and Δ as well as ℓ and r exchange their place. More importantly, MBOT can use states with more than 1 successor (e.g, $Q(X)$ instead of $Q[X]$). An example rule is displayed in Figure 8.

Let $M = (Q, \Sigma, \Delta, F, R, \mathrm{wt})$ be an MBOT such that $Q \cap (\Sigma \cup \Delta) = \emptyset$.[3] We require that $r \notin Q(X)$ for each rule $\ell \to r \in R$ to guarantee finite derivations and thus a well-defined semantics.[4] As before, we present a rewrite semantics. Let $\xi, \zeta \in T_\Sigma(Q(T_\Delta))$, and let $\rho = \ell \to r$ be a rule. We write $\xi \Rightarrow_M^\rho \zeta$ if there exists a substitution $\theta\colon X \to T_\Delta$ such that $\xi = \xi[\ell\theta]_p$ and $\zeta = \xi[r\theta]_p$, where $p \in \mathrm{pos}(\xi)$ be is the lexicographically least reducible position in ξ. A rewrite step is illustrated in Figure 8.

The *weighted tree transformation computed by* M in state $q \in Q$ is

$$\tau_M^q(t, u_1 \cdots u_k) = \sum_{\substack{n \in \mathbb{N}, \rho_1, \ldots, \rho_n \in R \\ t \Rightarrow_M^{\rho_1} \cdots \Rightarrow_M^{\rho_n} q(u_1, \ldots, u_k)}} \mathrm{wt}(\rho_1) \cdot \ldots \cdot \mathrm{wt}(\rho_n)$$

for all $t \in T_\Sigma$ and $u_1, \ldots, u_k \in T_\Delta$. The semantics of M is $\tau_M(t, u) = \sum_{q \in F} \tau_M^q(t, u)$ for all $t \in T_\Sigma$ and $u \in T_\Delta$.

We move to the last step for our main result, in which we show how to implement each finitely copying XTOP$^{\mathrm{R}}$ by an MBOT using a weighted version of the construction in Lemma 15 of Maletti (2008). The computational benefits (binarization, composition, efficient parsing, etc.) of MBOT over XTOP$^{\mathrm{R}}$ are described by Maletti (2011a).

Theorem 11. Every finitely copying XTOP$^{\mathrm{R}}$ can be implemented by an MBOT.

Proof sketch. We plan to utilize Theorem 18 of Engelfriet et al. (2009), which proves the same statement in the unweighted and deterministic case. Again, the weights are not problematic, but we need to remove the nondeterminism before we can apply it. This is achieved by a decomposition into two XTOP$^{\mathrm{R}}$. The first XTOP$^{\mathrm{R}}$ annotates the input tree with the rules that the second XTOP$^{\mathrm{R}}$ is supposed to use. Thus, the first XTOP$^{\mathrm{R}}$ remains nondeterministic, but the second XTOP$^{\mathrm{R}}$, which simply executes the annotated rules, is now deterministic. This standard approach due to Engelfriet (1975) is used in many similar constructions.

Suppose that n is a copying bound for the input XTOP$^{\mathrm{R}}$ M, which means that no more than n rules are applied to each input symbol. The first XTOP$^{\mathrm{R}}$ is actually a nondeterministic linear and nondeleting XTOP that annotates each input tree symbol with exactly n rules of M that are consistent with the state behavior of M. Moreover, the annotation also prescribes with which of n rules the processing should continue at each subtree. Since we know all the rules that will potentially be applied for a certain symbol, we can make the assignment such that no annotated rule is used twice in the same derivation. The details for this construction can be found in Lemma 15 of Maletti (2008).

In this way, we obtain a weighted linear and nondeleting XTOP M_1, which includes the look-ahead,

[3] This restriction can always be achieved by renaming the states.

[4] Again this could have been achieved with the help of other conditions on the MBOT or the used weight structure.

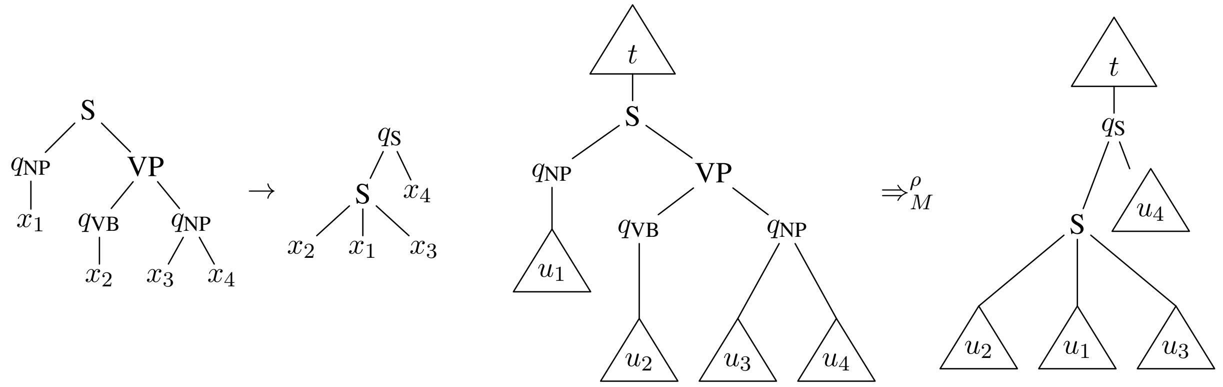

Figure 8: Example MBOT rule ρ [left] and its use in a rewrite step [right].

and an unweighted deterministic XTOP M_2. Only the weight and look-ahead of rules that are actually executed are applied (e.g., although we annotate n rules at the root symbol, we only execute the first rule and thus only apply its weight and look-ahead). The look-ahead of different rules is either resolved (i.e., pushed to the next rules) or multiplied using the HADAMARD product [see Fülöp and Vogler (2009)], which preserves regularity. This process is also used by Seemann et al. (2012). Now we can use Theorem 4 of Maletti (2011a) to obtain an MBOT N_1 that is equivalent to M_1. Similarly, we can use Theorem 18 of Engelfriet et al. (2009) to obtain an MBOT N_2 that is equivalent to M_2. Since MBOT are closed under composition by Theorem 23 of Engelfriet et al. (2009), we can compose N_1 and N_2 to obtain a single MBOT N that is equivalent to M. □

Corollary 12. For every sensible producing XTOP$^{\text{R}}$ there exists an equivalent MBOT.

Proof. Theorem 6 shows that there exists an equivalent proper XTOP$^{\text{R}}$, which must be finitely copying by Theorem 10. This last fact allows us to construct an equivalent MBOT by Theorem 11. □

6 Preservation of regularity

Finally, we present an application of Corollary 12 to solve an open problem. The translation model is often used in a backwards manner in a machine translation system as demonstrated, for example, by May et al. (2010), which means that an output tree is supplied and the corresponding input trees are sought. This starting output tree is typically the best parse of the string that we want to translate. However, instead of a single tree, we want to use all parses of this sentence together with their parse scores. Those parses form a regular weighted tree language, and applying them backwards to the translation model yields another weighted tree language L of corresponding input trees. For an efficient representation and efficient modification algorithms (such a k-best extraction) we would like L to be regular. However, Fülöp et al. (2011) demonstrate that the backward application of a regular weighted tree language to an XTOP$^{\text{R}}$ is not necessarily regular. The counterexample uses a variant of the XTOP of Example 1 and is thus not sensible. Theorem 14 of Maletti (2011a) shows that MBOT preserve regularity under backward application.

Corollary 13. Sensible XTOP$^{\text{R}}$ preserve regularity under backward application.

Conclusion

We demonstrated that each sensible XTOP$^{\text{R}}$ can be implemented by an MBOT. The latter formalism offers many computational advantages, so that the author believes that MBOT should be used instead of XTOP. We used real number weights, but the author believes that our results carry over to at least all zero-sum and zero-divisor free semirings [see Hebisch and Weinert (1998) and Golan (1999)], which are semirings such that (i) $a + b = 0$ implies $a = 0$ and (ii) $a \cdot b = 0$ implies $0 \in \{a, b\}$. Whether our results hold in other semirings (such as the semiring of all reals where $-1 + 1 = 0$) remains an open question.

References

Alfred V. Aho and Jeffrey D. Ullman. 1971. Translations on a context-free grammar. *Inform. and Control*, 19(5):439–475.

André Arnold and Max Dauchet. 1976. Bi-transductions de forêts. In *Proc. 3th Int. Coll. Automata, Languages and Programming*, pages 74–86. University of Edinburgh.

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

David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228.

Frank Drewes and Joost Engelfriet. 1998. Decidability of the finiteness of ranges of tree transductions. *Inform. and Comput.*, 145(1):1–50.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proc. 41st Ann. Meeting Association for Computational Linguistics*, pages 205–208. Association for Computational Linguistics.

Joost Engelfriet and Sebastian Maneth. 2003. Macro tree translations of linear size increase are MSO definable. *SIAM J. Comput.*, 32(4):950–1006.

Joost Engelfriet and Heiko Vogler. 1985. Macro tree transducers. *J. Comput. System Sci.*, 31(1):71–146.

Joost Engelfriet, Eric Lilin, and Andreas Maletti. 2009. Extended multi bottom-up tree transducers — composition and decomposition. *Acta Inform.*, 46(8):561–590.

Joost Engelfriet. 1975. Bottom-up and top-down tree transformations — a comparison. *Math. Systems Theory*, 9(3):198–231.

Joost Engelfriet. 1977. Top-down tree transducers with regular look-ahead. *Math. Systems Theory*, 10(1):289–303.

Zoltán Fülöp and Heiko Vogler. 2009. Weighted tree automata and tree transducers. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, EATCS Monographs on Theoret. Comput. Sci., chapter 9, pages 313–403. Springer.

Zoltán Fülöp, Andreas Maletti, and Heiko Vogler. 2011. Weighted extended tree transducers. *Fundam. Inform.*, 111(2):163–202.

Ferenc Gécseg and Magnus Steinby. 1984. *Tree Automata*. Akadémiai Kiadó, Budapest.

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

Jonathan S. Golan. 1999. *Semirings and their Applications*. Kluwer Academic, Dordrecht.

Jonathan Graehl, Kevin Knight, and Jonathan May. 2008. Training tree transducers. *Comput. Linguist.*, 34(3):391–427.

Jonathan Graehl, Mark Hopkins, Kevin Knight, and Andreas Maletti. 2009. The power of extended top-down tree transducers. *SIAM J. Comput.*, 39(2):410–430.

Udo Hebisch and Hanns J. Weinert. 1998. *Semirings—Algebraic Theory and Applications in Computer Science*. World Scientific.

Kevin Knight and Jonathan Graehl. 2005. An overview of probabilistic tree transducers for natural language processing. In *Proc. 6th Int. Conf. Computational Linguistics and Intelligent Text Processing*, volume 3406 of LNCS, pages 1–24. Springer.

Kevin Knight. 2007. Capturing practical natural language transformations. *Machine Translation*, 21(2):121–133.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Eric Lilin. 1978. *Une généralisation des transducteurs d'états finis d'arbres: les S-transducteurs*. Thèse 3ème cycle, Université de Lille.

Andreas Maletti and Daniel Quernheim. 2011. Pushing for weighted tree automata. In *Proc. 36th Int. Symp. Mathematical Foundations of Computer Science*, volume 6907 of LNCS, pages 460–471. Springer.

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

Andreas Maletti. 2010a. Input and output products for weighted extended top-down tree transducers. In *Proc. 14th Int. Conf. Developments in Language Theory*, volume 6224 of LNCS, pages 316–327. Springer.

Andreas Maletti. 2010b. Why synchronous tree substitution grammars? In *Proc. Human Language Technologies: Conf. North American Chapter of the ACL*, pages 876–884. Association for Computational Linguistics.

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

Andreas Maletti. 2011b. How to train your multi bottom-up tree transducer. In *Proc. 49th Ann. Meeting Association for Computational Linguistics: Human Language Technologies*, pages 825–834. Association for Computational Linguistics.

Jonathan May and Kevin Knight. 2006. Tiburon: A weighted tree automata toolkit. In *Proc. 11th Int. Conf. Implementation and Application of Automata*, volume 4094 of LNCS, pages 102–113. Springer.

Jonathan May, Kevin Knight, and Heiko Vogler. 2010. Efficient inference through cascades of weighted tree transducers. In *Proc. 48th Ann. Meeting Association for Computational Linguistics*, pages 1058–1066. Association for Computational Linguistics.

Jonathan May. 2010. *Weighted Tree Automata and Transducers for Syntactic Natural Language Processing*. Ph.D. thesis, University of Southern California, Los Angeles.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proc. 46th Ann. Meeting Association for Computational Linguistics*, pages 192–199. Association for Computational Linguistics.

William C. Rounds. 1970. Mappings and grammars on trees. *Math. Systems Theory*, 4(3):257–287.

Nina Seemann, Daniel Quernheim, Fabienne Braune, and Andreas Maletti. 2012. Preservation of recognizability for weighted linear extended top-down tree transducers. In *Proc. 2nd Workshop Applications of Tree Automata in Natural Language Processing*, pages 1–10. Association for Computational Linguistics.

Stuart M. Shieber. 2004. Synchronous grammars as tree transducers. In *Proc. 7th Int. Workshop Tree Adjoining Grammars and Related Formalisms*, pages 88–95.

Jun Sun, Min Zhang, and Chew Lim Tan. 2009. A non-contiguous tree sequence alignment-based model for statistical machine translation. In *Proc. 47th Ann. Meeting Association for Computational Linguistics*, pages 914–922. Association for Computational Linguistics.

James W. Thatcher. 1970. Generalized^2 sequential machine maps. *J. Comput. System Sci.*, 4(4):339–367.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008a. A tree sequence alignment-based tree-to-tree translation model. In *Proc. 46th Ann. Meeting Association for Computational Linguistics*, pages 559–567. Association for Computational Linguistics.

Min Zhang, Hongfei Jiang, Haizhou Li, Aiti Aw, and Sheng Li. 2008b. Grammar comparison study for translational equivalence modeling and statistical machine translation. In *Proc. 22nd Int. Conf. Computational Linguistics*, pages 1097–1104. Association for Computational Linguistics.