Survey: Weighted Extended Top-down Tree Transducers Part III — Composition[⋆]

Aurélie Lagoutte[1] and Andreas Maletti[2,⋆⋆]

[1] École normale supérieure de Cachan
Département Informatique
61, avenue du Président Wilson, 94235 Cachan cedex, France
`aurelie.lagoutte@ens-cachan.fr`
[2] Universität Stuttgart
Institute for Natural Language Processing
Azenbergstraße 12, 70174 Stuttgart, Germany
`andreas.maletti@ims.uni-stuttgart.de`

Abstract. In this survey (functional) compositions of weighted tree transformations computable by weighted extended top-down tree transducers are investigated. The existing results in the literature are explained and illustrated. It is argued, why certain compositions are not possible in the general case, and 3 informed conjectures provide an insight into potentially 3 new composition results that extend and complement the existing results. In particular, if all were true, then the beautiful symmetry in the composition results for weighted top-down and bottom-up tree transducers would be recovered.

Keywords: weighted tree transducer, top-down tree transducer, composition, deletion, copying

1 Motivation

Weighted tree transducers [32,14,19] (also called 'tree series transducers') are a joint generalization of the unweighted tree transducer (such as the top-down tree transducer [41,42] or the bottom-up tree transducer [43]) and the weighted tree automaton [6,9,1,30,17,8,7]. A good overview over both predecessors is presented in [20]. For a more detailed historic account and an in-depth introduction into weighted extended top-down tree transducers, we refer the reader to the first part [36] of this survey. A popular application area that has driven tree transducer research in the past few years is (syntax-based) machine translation [28,27]. The second part [37] of this survey attempts to present a high-level perspective on some of the essential problems and algorithms used in this application domain.

⋆ The work was carried out while the first author was an intern at *Universität Stuttgart.*
⋆⋆ Financially supported by the German Research Foundation (DFG) grant MA/4959/1-1.

In this part of the survey, we will investigate compositions of weighted extended (top-down) tree transducers. Such a tree transducer computes a weighted relation between input and output trees (i.e., it assigns a weight to each pair of input and output trees). Two such relations can be composed in the usual manner with the only difference that a weight (for example, a confidence or a probability) is returned each time "membership is tested". Using the real numbers as weight structure, we compose two weighted relations $\tau_1 \colon A \times B \to \mathbb{R}$ and $\tau_2 \colon B \times C \to \mathbb{R}$ by requiring that

$$(\tau_1 \,;\, \tau_2)(a, c) = \sum_{b \in B} \tau_1(a, b) \cdot \tau_2(b, c) \tag{1}$$

for all $a \in A$ and $c \in C$. For the sake of simplicity, let us assume that B is finite. Equation (1) can be imagined in an operational manner. The first process transforms the input a into an intermediate product b at a certain cost $\tau_1(a, b)$. This intermediate product is then fed into the second process, which transforms it into a final product at cost $\tau_2(b, c)$. Thus, the components are executed in a sequential manner. Traditionally, the multiplicative operation of the weight structure (typically, a semiring [25,23]) is used to combine weights of processes that are executed in series (or in sequence). Consequently, we multiply the weights $\tau_1(a, b) \cdot \tau_2(b, c)$ to obtain the cost of producing c from a via the intermediate product b. Naturally, there might be a choice of intermediate products that are all suitable to some degree to produce the output c. Thus, we sum over all the possibilities of producing c from a.

Compositions have been and are used in a number of application areas as diverse as machine translation [46] and functional program optimization [29]. The complexity of a given tree transformation problem can be tackled and broken down into smaller pieces with the help of (de-)composition in a *divide-and-conquer* approach. Once all the subproblems have been solved, the individual solutions can be recombined with the help of composition. This approach is used in [46], where a translation model is broken into 3 smaller pieces:

– a reordering component, which changes the order of subtrees but keeps the trees otherwise intact,
– an insertion component, which has the ability to spontaneously add subtrees to the output of the translation, and
– a translation component, which just translates the words (or phrases) occurring in the input tree.

These components can now be trained and optimized individually (even from different resources). However, since the evaluation of composition chains can be very inefficient [40], automatic procedures that "compose" finite representations of such weighted relations are desirable. Naturally, the obtained finite representation should compute the composition of the weighted relations computed by the input representations. As expected, the finite representation discussed in this survey is the weighted extended top-down tree transducer (xtt) together with its variants.

In contrast to the first part [36] of this survey, we do not require a complete semiring [25,23] here, which yields that we have to avoid infinite summations. This change prompts a minor change in the definition of the model because we have to disallow rules that contain no input and output symbol at all. In fact, infinite sums also restrict the potential compositions because we have to guarantee that the sum in (1) is finite. This is achieved by two simple conditions, of which each is sufficient to guarantee the finiteness of the sum in (1). Moreover, we introduce a simple variant of our main model that has rule identifiers in order to simplify the composition constructions. The additional indirection via identifiers allows us to construct the same rule several times under different names. In this way we can obtain a closer and more direct relationship between the rules of the input xtt and the composed xtt.

In the main part of this survey, we investigate compositions of xtt. In other words, given two xtt M and N, we want to construct another xtt that computes the composition of the weighted tree transformations computed by the xtt M and N. It is known that already in the unweighted setting, this cannot always be achieved, and we will consider two important cases:

- compositions of an xtt with a top-down tree transducer, and
- compositions of selected xtt with top-down tree transducers that can additionally have ε-rules.

The former case has been investigated in the unweighted case by [12,5] and these results were partially lifted to the weighted setting in [14,33,34]. We recall all the relevant results and complement them by three conjectured results that handle the missing cases. More precisely, we conjecture:

- that a constant xtt can be composed with a linear top-down tree transducer (see Conjecture 11), where the property 'constant' will be introduced here,
- that a deterministic xtt can be composed with a nondeleting top-down tree transducer (see Conjecture 13), and
- that a constant and deterministic xtt can be composed with any top-down tree transducer (see Conjecture 14).

We explain why these conjectured cases cause additional problems, which are due to the presence of weights. While we will not present a formal construction for each case, we present a generic composition construction and then indicate how to modify it to obtain a formal construction for the individual cases.

The latter case, in which we compose an xtt with a top-down tree transducer with ε-rules, was investigated in [39] in the unweighted setting. Here, we extend the results of [39] to the weighted setting and conjecture a new result (see Conjecture 23), which is again based on the new property 'constant'. Overall, our conjectured results complement the existing results nicely, and if all were true, then we would obtain the beautiful symmetry in the weighted setting that is known from compositions [12,5] for unweighted top-down and bottom-up tree transducers [43].

2 Notation

The set of all nonnegative integers is $\mathbb{N}$, and we let $[n] = \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$ for every $n \in \mathbb{N}$. We fix the set $\mathrm{X} = \{\mathrm{x}_i \mid i \in \mathbb{N}\}$ of (formal) variables. The set of all finite words (or sequences) over a set S is S^*, where ε is the empty word. The concatenation of the words $v, w \in S^*$ is $v.w$ or simply vw. The length of a word $w \in S^*$ is denoted by $|w|$. An *alphabet* Σ is a nonempty and finite set, of which the elements are called *symbols*. For every alphabet Q, we let $Q(S) = \{q(s) \mid q \in Q, s \in S\}$. The set $T_\Sigma(S)$ of *Σ-trees*[3] *with leaf labels* S is the smallest set T such that $S \subseteq T$ and $\sigma(t_1, \ldots, t_k) \in T$ for every $k \in \mathbb{N}$, $\sigma \in \Sigma$, and $t_1, \ldots, t_k \in T$. We often omit qualifications like 'for all $k \in \mathbb{N}$' if it is obvious from the context that k is a nonnegative integer. Moreover, we generally assume that $\Sigma \cap S = \emptyset$, and thus we write $\sigma()$ simply as σ for every $\sigma \in \Sigma$. Given another alphabet Δ, we treat elements of $T_\Delta(T_\Sigma(S))$ and $Q(T_\Sigma(S))$ as particular trees of $T_{Q \cup \Sigma \cup \Delta}(S)$.[4] Finally, we write T_Σ for $T_\Sigma(\emptyset)$.

Next, we define a few operations on trees. The set $\mathrm{pos}(t) \subseteq \mathbb{N}^*$ of *positions* of a tree $t \in T_\Sigma(S)$ is inductively defined by $\mathrm{pos}(s) = \{\varepsilon\}$ for every $s \in S$ and

$$\mathrm{pos}(\sigma(t_1, \ldots, t_k)) = \{\varepsilon\} \cup \{i.w \mid i \in [k], w \in \mathrm{pos}(t_i)\}$$

for every $\sigma \in \Sigma$ and $t_1, \ldots, t_k \in T_\Sigma(S)$. The set $\mathrm{pos}(t)$ of positions is (totally) ordered by the lexicographic order on $\mathbb{N}^*$. Let $t, t' \in T_\Sigma(S)$ and $w \in \mathrm{pos}(t)$. The *label* of t at position w is $t(w)$, and the *w-rooted subtree* of t is $t|_w$. Formally, these notions can be defined by $s(\varepsilon) = s|_\varepsilon = s$ for every $s \in S$ and

$$\sigma(t_1, \ldots, t_k)(\varepsilon) = \sigma \qquad \sigma(t_1, \ldots, t_k)(i.v) = t_i(v)$$
$$\sigma(t_1, \ldots, t_k)|_\varepsilon = \sigma(t_1, \ldots, t_k) \qquad \sigma(t_1, \ldots, t_k)|_{i.v} = t_i|_v$$

for every $\sigma \in \Sigma$, $t_1, \ldots, t_k \in T_\Sigma(S)$, $i \in [k]$, and $v \in \mathrm{pos}(t_i)$. For every subset $L \subseteq \Sigma \cup S$ of labels and $s \in S$, we let $\mathrm{pos}_L(t) = \{w \in \mathrm{pos}(t) \mid t(w) \in L\}$ and $\mathrm{pos}_s(t) = \mathrm{pos}_{\{s\}}(t)$. The expression $t[u]_w$ denotes the tree that is obtained from $t \in T_\Sigma(S)$ by replacing the subtree $t|_w$ at position w by $u \in T_\Delta(S)$.

The following operations implicitly always use the fixed set X of variables. We let $\mathrm{var}(t) = \{x \in \mathrm{X} \mid \mathrm{pos}_x(t) \neq \emptyset\}$. The tree t is *linear* if every $x \in \mathrm{X}$ occurs at most once in t. A *substitution* $\theta \colon \mathrm{X} \to T_\Sigma(S)$ can be applied to a tree $t \in T_\Sigma(S)$, and returns the tree $t\theta$ that is obtained by replacing (in parallel) all occurrences of each variable $x \in \mathrm{X}$ by $\theta(x)$. Formally, (i) $x\theta = \theta(x)$ for every $x \in \mathrm{X}$, (ii) $s\theta = s$ for every $s \in S \setminus \mathrm{X}$, and (iii) $\sigma(t_1, \ldots, t_k)\theta = \sigma(t_1\theta, \ldots, t_k\theta)$ for every $\sigma \in \Sigma$ and $t_1, \ldots, t_k \in T_\Sigma(S)$.

A *(commutative) semiring* [25,23] is an algebraic structure $(A, +, \cdot, 0, 1)$ consisting of two commutative monoids $(A, +, 0)$ and $(A, \cdot, 1)$ such that $\cdot$ distributes

[3] These are actually unranked trees, but our operational tree transformation model will only have finitely many rules that prescribe (and limit) the ranks of symbols, so that we could have used a ranked alphabet as well.

[4] A benefit of our approach without explicit ranks for symbols is that we can always take the union of two alphabets.

over finite sums (including the empty sum, which yields $a \cdot 0 = 0$ for all $a \in A$). The semiring A is *idempotent* if $1 + 1 = 1$.[5] Examples of semirings include

- the Boolean semiring $(\{0, 1\}, \max, \min, 0, 1)$, which is idempotent,
- the powerset[6] semiring $(\mathcal{P}(S), \cup, \cap, \emptyset, S)$ for some set S, which is idempotent,
- the tropical semiring $(\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$, which is also idempotent,
- the nonnegative integers $(\mathbb{N}, +, \cdot, 0, 1)$, and
- the semiring of real numbers $(\mathbb{R}, +, \cdot, 0, 1)$.

Let S and T be sets, and let $(A, +, \cdot, 0, 1)$ be a semiring. A *weighted relation r from S to T* is a mapping $r \colon S \times T \to A$. Moreover, for every mapping $f \colon S \to A$, we let $\operatorname{supp}(f) = \{s \in S \mid f(s) \neq 0\}$. Thus, $\operatorname{supp}(r) \subseteq S \times T$.

From now on, let $(A, +, \cdot, 0, 1)$ *be an arbitrary semiring such that* $0 \neq 1$.

Let us recall the weighted extended (top-down) tree transducer [11,2,26,24]. We essentially follow the definitions of [35,38], in which the corresponding unweighted device is discussed in detail. An in-depth presentation of the weighted device can be found in the first part [36] of this survey. A *(weighted) extended (top-down) tree transducer* (xtt) is a tuple $(Q, \Sigma, \Delta, I, R)$, where

- Q is a finite set of *states*,
- Σ and Δ are alphabets of *input* and *output symbols* such that $Q \cap (\Sigma \cup \Delta) = \emptyset$,
- $I \subseteq Q$ is a set of *initial states*, and
- $R \colon Q(T_\Sigma(\mathrm{X})) \times T_\Delta(Q(\mathrm{X})) \to A$ assigns *rule weights* such that $\operatorname{supp}(R)$ is finite and for every $(l, r) \in \operatorname{supp}(R)$ we have that $\{l, r\} \not\subseteq Q(\mathrm{X})$, l is linear, and $\operatorname{var}(r) \subseteq \operatorname{var}(l)$.[7]

For the following discussion, let $M = (Q, \Sigma, \Delta, I, R)$ be an xtt. The elements of $\operatorname{supp}(R)$ are called *rules* (of M), and we often write them as $l \to r$ instead of (l, r). We call l and r of a rule $l \to r$ the *left-* and *right-hand side*, respectively. Moreover, we write $l \to r \in R$ instead of $(l, r) \in \operatorname{supp}(R)$, and we write $l \xrightarrow{a} r \in R$ instead of $R(l, r) = a$. A rule $l \to r \in R$ is (i) *linear* if r is linear, (ii) *nondeleting* if $\operatorname{var}(r) = \operatorname{var}(l)$, and (iii) *simple* if $|\operatorname{pos}_\Sigma(l)| = 1$. In addition, the rule $l \to r$ is

- *consuming* if $|\operatorname{pos}_\Sigma(l)| \geq 1$, and an *$\varepsilon$-rule* otherwise, and
- *producing* if $|\operatorname{pos}_\Delta(r)| \geq 1$, and *erasing* otherwise.[8]

The xtt M is (i) linear, (ii) nondeleting, and (iii) a top-down tree transducer (tdtt) [32,14] if every rule $l \to r \in R$ is (i) linear, (ii) nondeleting, and (iii) simple, respectively. Moreover, the xtt M is Boolean if $R(l, r) = 1$ for every $l \to r \in R$.

The semantics of the xtt M is given by term rewriting [13,4,38]. To simplify our composition constructions later on, we immediately present a semantics that

[5] By distributivity, this yields $a + a = a$ for all $a \in A$.

[6] The powerset $\mathcal{P}(S)$ of a set S is the set of its subsets; i.e., $\mathcal{P}(S) = \{U \mid U \subseteq S\}$.

[7] The restriction $\{l, r\} \not\subseteq Q(\mathrm{X})$, which is not present in [36], disallows rules of the form $(q(\mathrm{x}_i), p(\mathrm{x}_i))$. This additional restriction is necessary because we do require complete semirings [25,23], which yields that we have to avoid infinite summations.

[8] The name 'erasing' is justified by the fact that each erasing rule is consuming.

can handle “foreign” symbols, which are symbols that are not in $Q \cup \Sigma \cup \Delta$.[9] Let Σ' and Δ' be such that $\Sigma \subseteq \Sigma'$ and $\Delta \subseteq \Delta'$. The elements of $T_{\Delta'}(Q(T_{\Sigma'}(\mathrm{X})))$ are called *sentential forms*. A position $w \in \mathrm{pos}_Q(\xi)$ in a sentential form ξ is *reducible (for M)* if there exists a rule $l \to r \in R$ and a substitution $\theta \colon \mathrm{X} \to T_{\Sigma'}(\mathrm{X})$ such that $\xi|_w = l\theta$. Let $\xi, \zeta \in T_{\Delta'}(Q(T_{\Sigma'}(\mathrm{X})))$ be sentential forms and $l \to r \in R$ be a rule. We say that ξ *rewrites to* ζ *using* $l \to r$, denoted by $\xi \Rightarrow_M^{(l,r)} \zeta$, if there exists a substitution $\theta \colon \mathrm{X} \to T_{\Sigma'}(\mathrm{X})$ such that $\xi|_w = l\theta$ and $\zeta = \xi[r\theta]_w$ where w is the least reducible position in $\mathrm{pos}_Q(\xi)$ with respect to the lexicographic total ordering on $\mathbb{N}^*$.[10] As usual, we use ‘;’ for relation composition, thus for example,

$$(\Rightarrow_M^{\rho_1} ; \Rightarrow_M^{\rho_2}) = \{(\xi, \zeta) \mid \exists \xi' \colon \xi \Rightarrow_M^{\rho_1} \xi' \Rightarrow_M^{\rho_2} \zeta\} \enspace .$$

The *(extended) weighted relation* τ'_M (or weighted tree transformation) *computed by* M is given by

$$\tau'_M(\xi, \zeta) = \sum_{\substack{\rho_1, \dots, \rho_k \in \mathrm{supp}(R) \\ \xi \Rightarrow_M^{\rho_1} ; \cdots ; \Rightarrow_M^{\rho_k} \zeta}} \Bigl(\prod_{i=1}^{k} R(\rho_i)\Bigr)$$

for every $\xi, \zeta \in T_{\Delta'}(Q(T_{\Sigma'}(\mathrm{X})))$. The *semantics* τ_M of the xtt M is the weighted relation $\tau_M \colon T_\Sigma \times T_\Delta \to A$ such that $\tau_M(t, u) = \sum_{q \in I} \tau'_M(q(t), u)$ for every $t \in T_\Sigma$ and $u \in T_\Delta$.[11] Finally, the xtt M is *deterministic*[12] (respectively, *total*) if for all $q \in Q$ and $t \in T_\Sigma$ there exists at most (respectively, at least) one $u \in T_\Delta$ such that $(q(t), u) \in \mathrm{supp}(\tau'_M)$.[13]

3 An Example Composition

We start our investigation into compositions of xtt with an example to illustrate the problem and the general principle used to solve it. Roughly speaking, given two xtt M and N we want to construct a single xtt that behaves like the two xtt M and N in sequence. Before we move to the formal definition of composition, let us introduce two example xtt and demonstrate derivations (i.e., term rewrite steps).

[9] A definition of the semantics without this extension can be found in [36].

[10] Given a sentential form ξ and a rule $\rho \in R$, there exists at most one sentential form ζ such that $\xi \Rightarrow_M^{\rho} \zeta$.

[11] Since the xtt M cannot consume symbols from $\Sigma' \setminus \Sigma$ and cannot produce symbols from $\Delta' \setminus \Delta$, the semantics τ_M does not depend on the particular choice of Σ' and Δ'.

[12] This property should correctly be called ‘unambiguous’, but for historical reasons we use ‘deterministic’ in the following.

[13] For top-down tree transducers these properties are typically defined using syntactic restrictions [12,5], which imply our corresponding semantic conditions. It requires a significant technical overhead to generalize the syntactic conditions faithfully to xtt, so we chose to present only the semantic property.

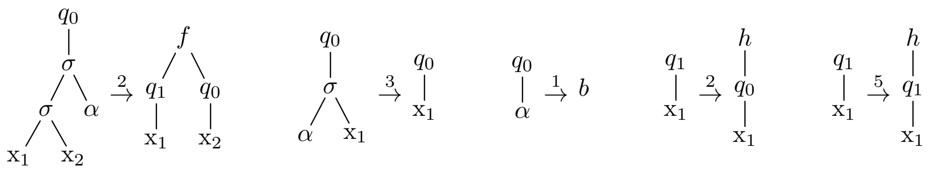

Fig. 1. The rules of the xtt M (see Example 1).

Example 1. We assume that the used semiring is the semiring $(\mathbb{R}, +, \cdot, 0, 1)$ of real numbers. Let $M = (\{q_0, q_1\}, \Sigma, \Gamma, \{q_0\}, R)$ be the xtt with input alphabet $\Sigma = \{\sigma, \alpha\}$, output alphabet $\Gamma = \{f, h, b\}$, and the following rules:

$$\begin{aligned}
\rho_1 :&\quad q_0(\sigma(\sigma(\mathrm{x}_1, \mathrm{x}_2), \alpha)) \xrightarrow{2} f(q_1(\mathrm{x}_1), q_0(\mathrm{x}_2)) &\qquad \rho_4 :&\quad q_1(\mathrm{x}_1) \xrightarrow{2} h(q_0(\mathrm{x}_1)) \\
\rho_2 :&\quad q_0(\sigma(\alpha, \mathrm{x}_1)) \xrightarrow{3} q_0(\mathrm{x}_1) &\qquad \rho_5 :&\quad q_1(\mathrm{x}_1) \xrightarrow{5} h(q_1(\mathrm{x}_1)) \\
\rho_3 :&\quad q_0(\alpha) \xrightarrow{1} b \ .
\end{aligned}$$

We illustrate these rules in Fig. 1 and demonstrate their properties in the following table. Since all rules are linear and nondeleting, the xtt M is linear and nondeleting.

ε-rule	consuming	erasing	producing	linear	nondeleting
ρ_4, ρ_5	ρ_1–ρ_3	ρ_2	ρ_1, ρ_3–ρ_5	ρ_1–ρ_5	ρ_1–ρ_5

Next, let us demonstrate a derivation using M. As input and output tree we consider

$$s = \sigma(\sigma(\alpha, \sigma(\alpha, \sigma(\alpha, \alpha))), \alpha) \qquad \text{and} \qquad t = f(h(h(b)), b) \ .$$

Figure 2 shows a derivation from $q_0(s)$ to t. Its weight is

$$R(\rho_1) \cdot R(\rho_5) \cdot R(\rho_4) \cdot R(\rho_3) \cdot R(\rho_2) \cdot R(\rho_2) \cdot R(\rho_3) = 2 \cdot 5 \cdot 2 \cdot 1 \cdot 3 \cdot 3 \cdot 1 = 180$$

It is easy to verify that this is the only possible derivation from $q_0(s)$ to t. Since q_0 is the only initial state, we can conclude that $\tau_M(s, t) = 180$. □

Example 2. We keep the semiring of real numbers as our used semiring. A second xtt is given by $N = (\{p\}, \Gamma, \Delta, \{p\}, R')$, where $\Gamma = \{f, h, b\}$, $\Delta = \{\lambda, \gamma, \delta, \beta\}$, and R' contains the rules:

$$\begin{aligned}
\mu_1 :&\quad p(\mathrm{x}_1) \xrightarrow{2} \gamma(p(\mathrm{x}_1)) &\qquad \mu_4 :&\quad p(h(\mathrm{x}_1)) \xrightarrow{8} \delta(p(\mathrm{x}_1)) \\
\mu_2 :&\quad p(f(\mathrm{x}_1, \mathrm{x}_2)) \xrightarrow{5} \lambda(p(\mathrm{x}_1), p(\mathrm{x}_2)) &\qquad \mu_5 :&\quad p(b) \xrightarrow{1} \beta \\
\mu_3 :&\quad p(f(\mathrm{x}_1, \mathrm{x}_2)) \xrightarrow{5} \lambda(\beta, \lambda(p(\mathrm{x}_1), p(\mathrm{x}_2))) \ .
\end{aligned}$$

Again, the properties of the rules are documented in the following table. We observe that the xtt N is linear and nondeleting.

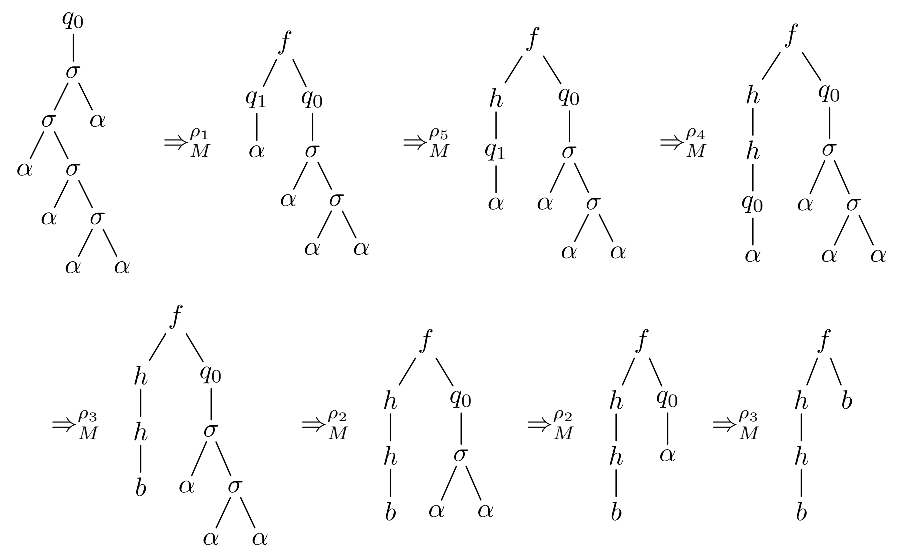

Fig. 2. A derivation from $q_0(s)$ to t in M (see Example 1).

ε-rule	consuming	erasing	producing	linear	nondeleting
μ_1	μ_2–μ_5		μ_1–μ_5	μ_1–μ_5	μ_1–μ_5

This time we illustrate a derivation for the input and output tree

$$t = f(h(h(b)), b) \qquad \text{and} \qquad u = \lambda(\beta, \lambda(\delta(\delta(\beta)), \beta)) \enspace .$$

Figure 3 shows the unique derivation from $p(t)$ to u. It has the weight

$$R'(\mu_3) \cdot R'(\mu_4) \cdot R'(\mu_4) \cdot R'(\mu_5) \cdot R'(\mu_5) = 5 \cdot 8 \cdot 8 \cdot 1 \cdot 1 = 320 \enspace . \qquad \square$$

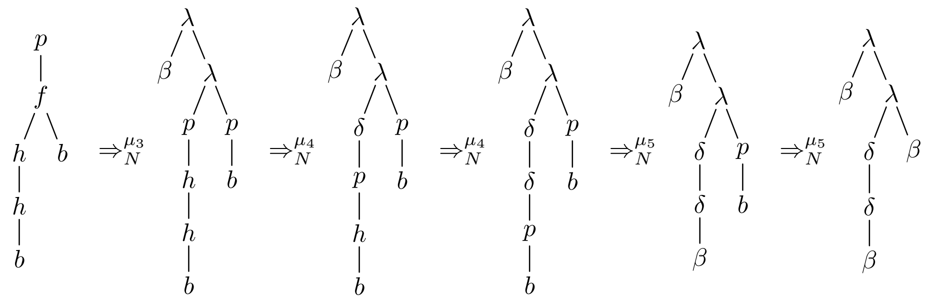

Fig. 3. A derivation from $p(t)$ to u in N (see Example 2).

Composition is the process of running two xtt one after the other. In this way, the output tree of the first xtt becomes the input tree of the second xtt.

For example, the output tree t generated by the xtt M in Example 1 can be processed by the xtt N of Example 2, which is demonstrated in Example 2. Thus, in a composition of M and N (see Examples 1 and 2) we can transform the input tree s of Example 1 immediately to the output tree u of Example 2 by first running the xtt M on s to produce the intermediate tree t, which can then be transformed into u by N.

Formally, two weighted tree relations (of suitable type) are composed as follows. Let $\tau_1 \colon T_\Sigma \times T_\Gamma \to A$ and $\tau_2 \colon T_\Gamma \times T_\Delta \to A$ be weighted relations. Their *composition* $(\tau_1 \mathbin{;} \tau_2) \colon T_\Sigma \times T_\Delta \to A$ is the weighted relation such that

$$(\tau_1 \mathbin{;} \tau_2)(s, u) = \sum_{t \in T_\Gamma} \tau_1(s, t) \cdot \tau_2(t, u) \tag{2}$$

for every $s \in T_\Sigma$ and $u \in T_\Delta$. Clearly, this definition generalizes the classical definition of composition for relations. Whereas the infinite sum is not a problem in the unweighted case (where it is an infinite disjunction that becomes true once one element is true), we have to address it in the weighted case. There are essentially two options:

(i) to permit infinite sums and require that the semiring is suitably rich to handle infinite sums [25,23], which was done in [14,18,19,33,34,20] and also in the first part [36] of this survey, or
(ii) to avoid the infinite sums by restricting the weighted relations (and thus the xtt) that we allow in compositions.

In this part of the survey, we will follow the second approach by requiring that in a composition $\tau_1 \mathbin{;} \tau_2$ we have that

$$\{t \mid (s, t) \in \operatorname{supp}(\tau_1)\} \qquad \text{or} \qquad \{t \mid (t, u) \in \operatorname{supp}(\tau_2)\} \tag{3}$$

is finite for every $s \in T_\Sigma$ and $u \in T_\Delta$. It is clear that in both cases the sum (2) in the definition of the composition $\tau_1 \mathbin{;} \tau_2$ is finite.

Let us illustrate the general approach that is used in most composition constructions. To construct an xtt that computes the composition $\tau_M \mathbin{;} \tau_N$ of the weighted relations computed by two xtt M and N, we need to make sure that the intermediate tree t (in Examples 1 and 2) is never constructed explicitly. To achieve this, the second xtt has to immediately consume every intermediate symbol that is produced by the first xtt M. Let us illustrate this approach by combining the two derivations of Figs. 2 and 3 such that intermediate symbols (from Γ) are consumed as soon as possible. The obtained derivation that now uses rules of both M and N is displayed in Fig. 4.

Once we have reordered the rule applications as indicated in the previous paragraph, we "glue" all rule applications that produce intermediate symbols together with the rule applications that consume these symbols. In this step, we also interpret two adjacent states (one of M and one of N) as in $p(q(s))$ as a single state $\langle p, q \rangle$. For example, based on Fig. 4 we glue the first two rule applications (of the rules ρ_1 and μ_3) together to obtain a single rule application

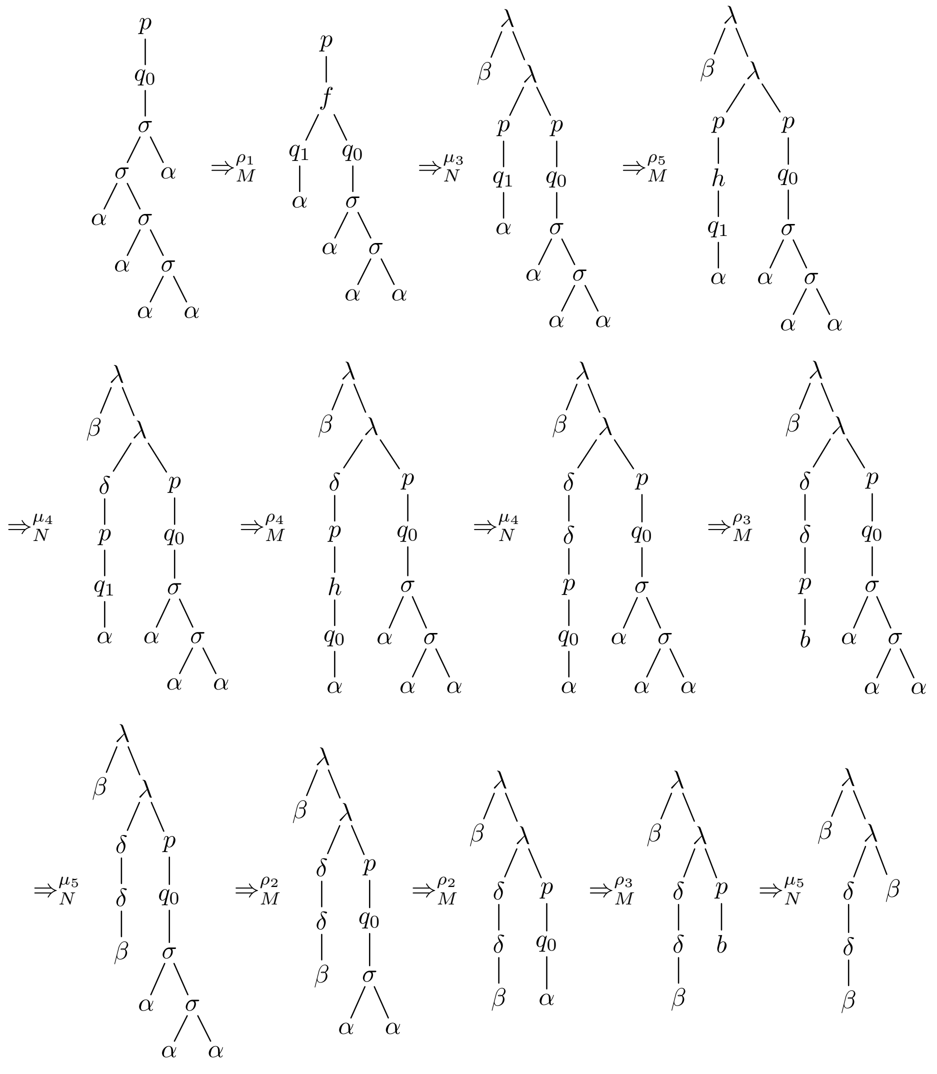
Fig. 4. Intertwined derivation from $p(q_0(s))$ to u in M and N (see Examples 1 and 2).

of the rule

$$\langle p, q_0 \rangle(\sigma(\sigma(\mathrm{x}_1, \mathrm{x}_2), \alpha)) \xrightarrow{2 \cdot 5} \lambda(\beta, \lambda(\langle p, q_1 \rangle(\mathrm{x}_1), \langle p, q_0 \rangle(\mathrm{x}_2))) \enspace ,$$

which is displayed in Fig. 5. For the rest of the discussion, we drop the distinction between rule applications and rules. In general, several rules of the first xtt need to be "glued" with several rules of the second xtt. However, it was already shown in [3] (and in the second part [37] of this survey) that this strategy does not work in general (even if both xtt M and N are linear and nondeleting). In the rest of this survey, we will thus focus on simpler cases, in which the left-hand sides of the rules of the second xtt N contain at most one input symbol. In Sect. 5 we consider compositions of an xtt M with a top-down tree transducer N. Thus, in Sect. 5 the second xtt N is such that every rule has exactly one input symbol in its left-hand side. We relax this requirement slightly in Sect. 6, where we investigate compositions of an xtt M with a top-down tree transducer N with ε-rules [39], which is an xtt in which each rule contains at most one input symbol in its left-hand side. However, before we proceed with the mentioned composition constructions we first introduce a modification of our xtt model that will prove to be useful in Sections 5 and 6.

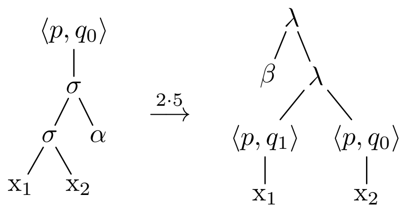

Fig. 5. Composed rule constructed from ρ_1 and μ_3 of Examples 1 and 2.

4 An Equivalent Model

In this section, we introduce an alternative description for weighted extended top-down tree transducers that will be useful for our composition constructions. Essentially, we introduce explicit rule identifiers (like ρ_1–ρ_5 used in Example 1) that stand for a specific rule. A mapping that becomes part of the specification assigns weighted rules to identifiers. This indirection allows us to use multiple rules with the same left- and right-hand side and even the same weight. In our composition constructions we use this facility to establish a more concise and simpler relation between the constructed rules of the composed xtt and the original rules of the input xtt.

Definition 3. *A* weighted extended (top-down) tree transducer with rule identifiers *is a system* $(Q, \Sigma, \Delta, I, \mathcal{R}, \chi)$*, where*

- *Q, Σ, Δ, and I are the same as the corresponding elements of an xtt,*
- *$\mathcal{R}$ is a finite set of* rule identifiers, *and*
- *$\chi\colon \mathcal{R} \to Q(T_\Sigma(\mathrm{X})) \times A \times T_\Delta(Q(\mathrm{X}))$ is a* rule assignment *that maps each rule identifier $\rho \in \mathcal{R}$ to its content $\chi(\rho) = (l, a, r)$ such that $\{l, r\} \nsubseteq Q(\mathrm{X})$, l is linear, and* $\mathrm{var}(r) \subseteq \mathrm{var}(l)$.

In accordance with our notation for rules, we often write $l \xrightarrow{a} r$ for elements $(l, a, r) \in Q(T_\Sigma(\mathrm{X})) \times A \times T_\Delta(Q(\mathrm{X}))$. Moreover, we let $\mathrm{wt}\colon \mathcal{R} \to A$ be such that $\mathrm{wt}(\rho) = a$ for all $\rho \in \mathcal{R}$ with $\chi(\rho) = l \xrightarrow{a} r$. Intuitively, 'wt' maps a rule identifier to the weight of its identified rule.

The semantics of the xtt $M = (Q, \Sigma, \Delta, I, \mathcal{R}, \chi)$ with rule identifiers $\mathcal{R}$ is given by rewriting in essentially the same way as before. Let Σ' and Δ' be two alphabets such that $\Sigma \subseteq \Sigma'$ and $\Delta \subseteq \Delta'$ and $Q \cap (\Sigma' \cup \Delta') = \emptyset$. Again, we call a position $w \in \mathrm{pos}_Q(\xi)$ in a sentential form $\xi \in T_{\Delta'}(Q(T_{\Sigma'}(\mathrm{X})))$ *reducible (for M)* if there exists a rule $\rho \in \mathcal{R}$ with $\chi(\rho) = l \xrightarrow{a} r$ and a substitution $\theta\colon \mathrm{X} \to T_{\Sigma'}(\mathrm{X})$ such that $\xi|_w = l\theta$. Now, let $\xi, \zeta \in T_{\Delta'}(Q(T_{\Sigma'}(\mathrm{X})))$, $\rho \in \mathcal{R}$, and $\chi(\rho) = l \xrightarrow{a} r$. We say that ξ *rewrites to* ζ *using* ρ, denoted by $\xi \Rightarrow_M^\rho \zeta$, if there exists a substitution $\theta\colon \mathrm{X} \to T_{\Sigma'}(\mathrm{X})$ such that $\xi|_w = l\theta$ and $\zeta = \xi[r\theta]_w$ where w is the least reducible position in $\mathrm{pos}_Q(\xi)$ with respect to the lexicographic total order on $\mathbb{N}^*$. The *(extended) weighted relation τ'_M computed by M* is given by

$$\tau'_M(\xi, \zeta) = \sum_{\substack{\rho_1, \dots, \rho_k \in \mathcal{R} \\ \xi \Rightarrow_M^{\rho_1} ; \cdots ; \Rightarrow_M^{\rho_k} \zeta}} \Big(\prod_{i=1}^{k} \mathrm{wt}(\rho_i) \Big)$$

for every $\xi, \zeta \in T_{\Delta'}(Q(T_{\Sigma'}(\mathrm{X})))$. As for xtt, the *semantics* τ_M of the xtt M with rule identifiers is the weighted relation $\tau_M\colon T_\Sigma \times T_\Delta \to A$ such that $\tau_M(t, u) = \sum_{q \in I} \tau'_M(q(t), u)$ for every $t \in T_\Sigma$ and $u \in T_\Delta$. The properties of rules and xtt defined in Sect. 2 generalize straightforwardly to xtt with rule identifiers.

Example 4. Let $N = (\{p\}, \Gamma, \Delta, \{p\}, \mathcal{R}, \chi)$ be the xtt with rule identifiers such that

- $\Gamma = \{f, h, b\}$ and $\Delta = \{\lambda, \gamma, \delta, \beta\}$,
- $\mathcal{R} = \{\mu_1, \dots, \mu_7\}$, and
- the rule assignment χ is given by

$$\begin{aligned}
\chi(\mu_1) &= p(\mathrm{x}_1) \xrightarrow{2} \gamma(p(\mathrm{x}_1)) & \chi(\mu_5) &= p(h(\mathrm{x}_1)) \xrightarrow{4} \delta(p(\mathrm{x}_1)) \\
\chi(\mu_2) &= p(f(\mathrm{x}_1, \mathrm{x}_2)) \xrightarrow{2} \lambda(p(\mathrm{x}_1), p(\mathrm{x}_2)) & \chi(\mu_6) &= p(h(\mathrm{x}_1)) \xrightarrow{4} \delta(p(\mathrm{x}_1)) \\
\chi(\mu_3) &= p(f(\mathrm{x}_1, \mathrm{x}_2)) \xrightarrow{3} \lambda(p(\mathrm{x}_1), p(\mathrm{x}_2)) & \chi(\mu_7) &= p(b) \xrightarrow{1} \beta \\
\chi(\mu_4) &= p(f(\mathrm{x}_1, \mathrm{x}_2)) \xrightarrow{5} \lambda(\beta, \lambda(p(\mathrm{x}_1), p(\mathrm{x}_2))) \enspace .
\end{aligned}$$

ε-rule	consuming	erasing	producing	linear	nondeleting
μ_1	μ_2–μ_7		μ_1–μ_7	μ_1–μ_7	μ_1–μ_7

Let $t = f(h(h(b)), b)$ and $u = \lambda(\beta, \lambda(\delta(\delta(\beta)), \beta))$ as in Example 2. Figure 6 shows a derivation from $p(t)$ to u with weight

$$\mathrm{wt}(\mu_4) \cdot \mathrm{wt}(\mu_5) \cdot \mathrm{wt}(\mu_5) \cdot \mathrm{wt}(\mu_7) \cdot \mathrm{wt}(\mu_7) = 5 \cdot 4 \cdot 4 \cdot 1 \cdot 1 = 80 \enspace .$$

We can construct exactly three other derivations from $p(t)$ to u using the rule sequences $\mu_4\mu_5\mu_6\mu_7\mu_7$, $\mu_4\mu_6\mu_5\mu_7\mu_7$, and $\mu_4\mu_6\mu_6\mu_7\mu_7$. All of these derivations have the same weight. Consequently, $\tau_N(t, u) = 80 + 80 + 80 + 80 = 320$. □

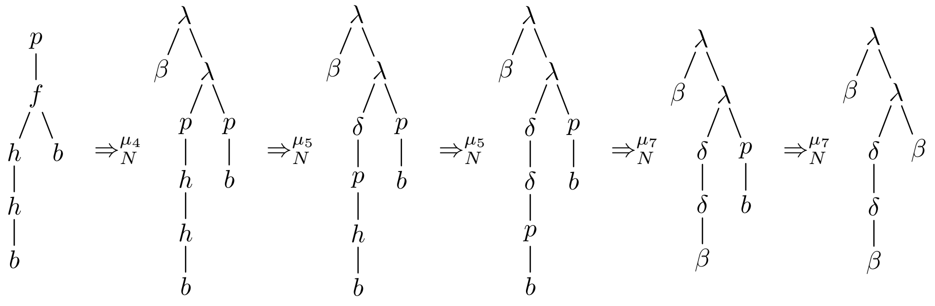

Fig. 6. One possible derivation from $p(t)$ to u in N of Example 4.

We can easily see that the two models of xtt are equally expressive. For every xtt $(Q, \Sigma, \Delta, I, R)$ we can construct an equivalent xtt $(Q, \Sigma, \Delta, I, R, \chi)$ with rule identifiers by setting $\chi = \mathrm{id}_R$, where id_R is the identity on R. Conversely, given an xtt $M = (Q, \Sigma, \Delta, I, \mathcal{R}, \chi)$ with rule identifiers we can obtain an equivalent xtt $(Q, \Sigma, \Delta, I, R)$ by setting

$$R(l, r) = \sum_{\rho \in \mathcal{R} \colon \chi(\rho) = (l, a, r)} \mathrm{wt}(\rho) \tag{4}$$

for all $l \in Q(T_\Sigma(\mathrm{X}))$ and $r \in T_\Delta(Q(\mathrm{X}))$.[14] The construction is illustrated in Example 5.

Example 5. Let N be the xtt with rule identifiers of Example 4. To obtain an equivalent xtt $N' = (Q, \Gamma, \Delta, I, R)$, we

- merge the rules μ_2 and μ_3 to form the rule $p(f(\mathrm{x}_1, \mathrm{x}_2)) \xrightarrow{2+3} \lambda(p(\mathrm{x}_1), p(\mathrm{x}_2))$,
- merge the rules μ_5 and μ_6 to form the rule $p(h(\mathrm{x}_1)) \xrightarrow{4+4} \delta(p(\mathrm{x}_1))$, and
- keep the rules μ_1, μ_4, and μ_7 with their original weight.

In this manner, we obtain exactly the xtt of Example 2, for which we only have one derivation from $p(t)$ to u, which is shown in Fig. 3. Naturally, its weight is 320. □

[14] The sum (4) returns 0, as desired, if M has no rules with left-hand side l and right-hand side r.

5 Composition with a Top-down Tree Transducer

In this section, we will discuss compositions $\tau_M ; \tau_N$ for xtt $M = (Q, \Sigma, \Gamma, I_1, R_1)$ and $N = (P, \Gamma, \Delta, I_2, R_2)$, in which the xtt N is actually a top-down tree transducer (tdtt). Moreover, we require that the xtt M does not have any ε-rules. This restriction ensures that the set $\{t \mid (s, t) \in \mathrm{supp}(\tau_M)\}$ is finite for every $s \in T_\Sigma$ [see (3)] because each rule application consumes at least one input symbol. Hence there can only be finitely many rule applications to $q(s)$ given a state $q \in Q$ and an input tree s, which yields an upper bound on the number of potential derivations, which in turn limits the number of output symbols in each output tree. We already demonstrated in Sect. 3 that this restriction is sufficient to ensure that the sum over all intermediate trees t occurring in (2):

$$(\tau_M ; \tau_N)(s, u) = \sum_{t \in T_\Gamma} \tau_M(s, t) \cdot \tau_N(t, u)$$

is finite for all $s \in T_\Sigma$ and $u \in T_\Delta$. Thus, composition is well-defined in all cases discussed in this section.

5.1 Construction

Now we are ready to present the generic composition construction. For the sake of uniformity, we will construct more rules than strictly necessary. As already indicated in Fig. 4, the states of the composed xtt will be pairs of states with one state from each input xtt. Next, let us fix an important constant m.

– Let $c \geq |\mathrm{pos}_x(r)|$ for all $l \to r \in R_2$ and $x \in \mathrm{X}$. Roughly speaking, c is larger than the maximal *copying degree* of N, which is the maximal number of times a variable occurs on some right-hand side of a rule of N. To keep the presentation simple, we assume that $c \geq 1$.
– Let $s \geq |\mathrm{pos}_\Gamma(r)|$ for all $l \to r \in R_1$. Consequently, s is larger than the maximal number of output symbols in a right-hand side of a rule of M.
– Finally, let $m \geq c^s$. The constant m provides an upper bound to the number of steps required by N to process a right-hand side of a rule of M.

Recall that given a sentential form $\xi \in T_\Delta(P(T_\Gamma(Q(T_\Sigma(\mathrm{X})))))$ and a rule $\rho \in R_1$ there exists at most one $\zeta \in T_\Delta(P(T_\Gamma(Q(T_\Sigma(\mathrm{X})))))$ such that $\xi \Rightarrow_M^\rho \zeta$.[15] Naturally, the same property holds for the tdtt N. To avoid an explicit conversion, we identify elements of $T_\Delta(P(Q(T_\Sigma(\mathrm{X}))))$ with elements of $T_\Delta((P \times Q)(T_\Sigma(\mathrm{X})))$ in the obvious manner. Finally, we let

$$\Rightarrow_N^w = (\Rightarrow_N^{\mu_1} ; \cdots ; \Rightarrow_N^{\mu_k})$$

if $w = \mu_1 \cdots \mu_k$ with $\mu_1, \ldots, \mu_k \in R_2$.

[15] To match this statement to the earlier one, we have to set $\Delta' = \Delta \cup P \cup \Gamma$.

Definition 6. *The composed xtt $M \,;\, N$ is the xtt $(P \times Q, \Sigma, \Delta, I_2 \times I_1, \mathcal{R}, \chi)$ with rule identifiers*

$$\mathcal{R} = \{\langle \rho, p, w\rangle \mid \rho \in R_1, p \in P, w \in R_2^*, |w| \leq m\}$$

such that $\chi(\langle l \to r, p, \mu_1 \cdots \mu_k\rangle) = (p(l), a, r')$ for every $l \to r \in R_1$, $p \in P$, and rule sequence $\mu_1, \ldots, \mu_k \in R_2$ with $k \leq m$, where $r' \in T_\Delta(P(Q(\mathrm{X})))$ and

$$a = \begin{cases} R_1(l \to r) \cdot \prod_{i=1}^{k} R_2(\mu_i) & \text{if } p(l) \Rightarrow_M^{(l,r)} ; \Rightarrow_N^{\mu_1 \cdots \mu_k} r' \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, the construction might return a lot of rule identifiers whose associated rules have weight 0. These rules are useless, and we typically will not report them in our examples. Moreover, we can easily see that the constructed rules never have the forbidden shape $l \to r$ with $\{l, r\} \subseteq P(Q(\mathrm{X}))$ because the left-hand side l equals $p(l')$ for some left-hand side l' of a rule of M, which does not have ε-rules. Let us illustrate the construction on two example xtt, which we will use throughout this section.

Example 7. We again use the semiring of real numbers in this example. Moreover, let us consider the xtt M and N, which are given as follows:

$$M = (\{q\}, \Sigma, \Sigma, \{q\}, R_1) \qquad \text{and} \qquad N = (\{p_0, p\}, \Sigma, \Delta, \{p_0\}, R_2) \enspace ,$$

where

– $\Sigma = \{\gamma, \alpha\}$ and $\Delta = \{\sigma\} \cup \Sigma$,
– R_1 contains the rules

$$\rho_1\colon \quad q(\gamma(\mathrm{x}_1)) \xrightarrow{2} \gamma(\gamma(q(\mathrm{x}_1))) \qquad\qquad \rho_2\colon \quad q(\alpha) \xrightarrow{2} \alpha \enspace ,$$

– and R_2 contains the rules

$$\begin{aligned} &\mu_1\colon \quad p_0(\gamma(\mathrm{x}_1)) \xrightarrow{4} \sigma(p_0(\mathrm{x}_1), p_0(\mathrm{x}_1)) && \mu_6\colon \quad p(\gamma(\mathrm{x}_1)) \xrightarrow{1} \gamma(p(\mathrm{x}_1)) \\ &\mu_2\colon \quad p_0(\gamma(\mathrm{x}_1)) \xrightarrow{2} \sigma(p_0(\mathrm{x}_1), p(\mathrm{x}_1)) && \mu_7\colon \quad p(\gamma(\mathrm{x}_1)) \xrightarrow{3} \alpha \\ &\mu_3\colon \quad p_0(\gamma(\mathrm{x}_1)) \xrightarrow{2} \sigma(p(\mathrm{x}_1), p_0(\mathrm{x}_1)) && \mu_8\colon \quad p(\alpha) \xrightarrow{1} \alpha \\ &\mu_4\colon \quad p_0(\gamma(\mathrm{x}_1)) \xrightarrow{1} \sigma(p(\mathrm{x}_1), p(\mathrm{x}_1)) && \\ &\mu_5\colon \quad p_0(\alpha) \xrightarrow{1} \alpha \enspace . && \end{aligned}$$

ε-rule	consuming	erasing	producing	linear	nondeleting
	ρ_1, ρ_2		ρ_1, ρ_2	ρ_1, ρ_2	ρ_1, ρ_2
	μ_1–μ_8		μ_1–μ_8	μ_5–μ_8	μ_1–μ_6, μ_8

Both M and N are tdtt, M is linear and nondeleting, whereas N is neither linear nor nondeleting. Additionally, the xtt M is deterministic and total. We can set $c = 2$ and $s = 2$, and thus, we can select $m = 4$. To increase readability,

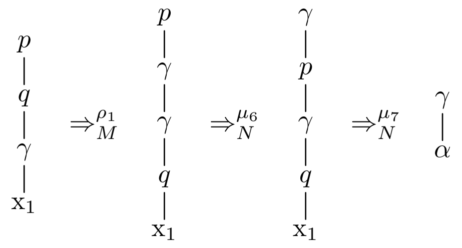

Fig. 7. Derivation for rule $\langle \rho_1, p, \mu_6\mu_7 \rangle$ (see Example 7).

let $\mathcal{R}^\gamma_{p_0} = \{\mu_1, \mu_2, \mu_3, \mu_4\}$ and $\mathcal{R}^\gamma_p = \{\mu_6, \mu_7\}$. Intuitively, $\mathcal{R}^\gamma_{p_0}$ and $\mathcal{R}^\gamma_p$ are the sets of rules that consume the input symbol γ in state p_0 and p, respectively. Now let us construct the composition $M \,;\, N$. It is the xtt

$$M \,;\, N = (Q', \Sigma, \Delta, I', \mathcal{R}, \chi)$$

with rule identifiers such that

– $Q' = \{\langle p_0, q\rangle, \langle p, q\rangle\}$ and $I' = \{\langle p_0, q\rangle\}$,
– $\mathcal{R} = \{\langle \rho_2, p_0, \mu_5\rangle, \langle \rho_2, p, \mu_8\rangle, \langle \rho_1, p, \mu_6\mu_6\rangle, \langle \rho_1, p, \mu_6\mu_7\rangle, \langle \rho_1, p, \mu_7\rangle\} \cup \mathcal{R}'$ with

$$\begin{aligned}\mathcal{R}' = \{\langle \rho_1, p_0, \mu_1\mu\mu'\rangle \mid \mu, \mu' \in \mathcal{R}^\gamma_{p_0}\} \cup \\ \cup \{\langle \rho_1, p_0, \mu_2\mu\mu'\rangle \mid \mu \in \mathcal{R}^\gamma_{p_0}, \mu' \in \mathcal{R}^\gamma_p\} \cup \\ \cup \{\langle \rho_1, p_0, \mu_3\mu\mu'\rangle \mid \mu \in \mathcal{R}^\gamma_p, \mu' \in \mathcal{R}^\gamma_{p_0}\} \cup \\ \cup \{\langle \rho_1, p_0, \mu_4\mu\mu'\rangle \mid \mu, \mu' \in \mathcal{R}^\gamma_p\} \enspace .\end{aligned}$$

In total we have $5 + 16 + 8 + 8 + 4 = 41$ (meaningful) rule identifiers. We will not present all 41 corresponding rules, but we will show two example rules to demonstrate the construction. Let us first construct the rule for the identifier $\langle \rho_1, p, \mu_6\mu_7\rangle$. To this end, we need to build a derivation starting at $p(q(\gamma(x_1)))$ using the rule sequence $\rho_1\mu_6\mu_7$. This derivation is illustrated in Fig. 7. We obtain the rule

$$\chi(\langle \rho_1, p, \mu_6\mu_7\rangle) = \Big(\langle p, q\rangle(\gamma(x_1)), 2 \cdot 1 \cdot 3, \gamma(\alpha)\Big) \enspace .$$

Secondly, let us construct the rule for the identifier $\langle \rho_1, p_0, \mu_3\mu_7\mu_2\rangle$. This time we need to build a derivation that starts with $p_0(q(\gamma(x_1)))$ and uses the rule sequence $\rho_1\mu_3\mu_7\mu_2$. We illustrate the derivation in Fig. 8. Consequently, we obtain the rule

$$\chi(\langle \rho_1, p_0, \mu_3\mu_7\mu_2\rangle) = \Big(\langle p_0, q\rangle(\gamma(x_1)), 2 \cdot 2 \cdot 3 \cdot 2, \sigma(\alpha, \sigma(\langle p_0, q\rangle(x_1), \langle p, q\rangle(x_1)))\Big) \enspace .$$
□

Our general composition construction allows us to compose an xtt with a tdtt. As we have seen, it closely follows the intuition provided in Sect. 3 and uses the xtt with rule identifiers that we introduced in Sect. 4. This has the benefit

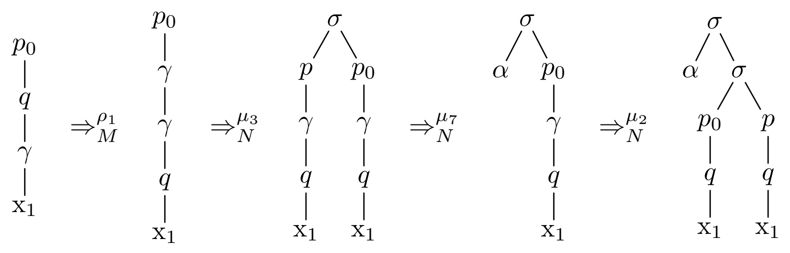
p_0
q
γ
x_1
$\Rightarrow^{\rho_1}_{M}$
$\Rightarrow^{\mu_3}_{N}$
$\Rightarrow^{\mu_7}_{N}$
$\Rightarrow^{\mu_2}_{N}$
σ
p
α

Fig. 8. Derivation for rule $\langle \rho_1, p_0, \mu_3\mu_7\mu_2 \rangle$ (see Example 7).

that we can obtain a direct correspondence between rule sequences of the xtt M and N and a rule in the composed xtt $M \,;N$. In the standard xtt (without rule identifiers) this direct correspondence is lost since several derivations might create the same rule.[16]

Naturally, we would expect that the composed xtt $M \;; N$ computes the weighted tree transformation $\tau_M \;; \tau_N$; i.e., the composition of the weighted tree transformations computed by M and N. In other words, we hope that $\tau_{M;N} = \tau_M \;; \tau_N$. Although the rule aggregation and intertwining approach followed in the construction (and shown in Sect. 3) is reasonable, it fails to produce a correct xtt (i.e., an xtt that computes $\tau_M \;; \tau_N$) in a number of cases. This is already true in the unweighted case [12,5] and the presence of weights adds a few more problematic cases, which we will discuss in the next section.

5.2 Correctness

In this section, we will investigate in which cases the composition construction (see Definition 6) actually produces an xtt that computes the composition $\tau_M;\tau_N$. In principle, the xtt M need not be a tdtt, but for the following discussion we assume that it is. The generalization to the general case is simple in almost all cases (see [38,16] for a few notable differences). Consequently, let us look at compositions of tdtt. Top-down tree transducers have been studied quite extensively in the unweighted case (see [21,22,10] for an overview). The following two slogans are known to represent properties that are unavailable in a single tdtt [12,5]:

– Nondeterminism followed by copying (non-linearity), and
– Checking (non-totality) followed by deletion.

A composition $\tau_M \;; \tau_N$ of two tdtt M and N can implement both properties mentioned in the slogans. Thus, these properties already restrict the potential successful compositions of tdtt. In fact, in all remaining cases shown in Table 1 the composition of the tdtt M and N is possible in the unweighted case [5, Theorem 1]. A detailed explanation of those restrictions on compositions is presented in [12,5]. Here we will focus on the particular problems that occur in

[16] The interested reader can compare our construction to [33,34].

Table 1. Cases for unweighted tdtt composition.

Case	M	N
(a)		linear and nondeleting
(b)	total	linear
(c)	deterministic	nondeleting
(d)	deterministic and total	

the generalization of those results to the weighted case because the limitations on compositions in the unweighted case transfer immediately to our setting.[17] Thus, we will not investigate compositions that do not fulfill the requirements in Table 1.[18]

Case (a) has been partially generalized in [31, Theorem 2.4] to weighted tdtt. More precisely, it was shown that the composition succeeds if both M and N are linear and nondeleting.[19] This result was further (partially) generalized in [14, Theorem 5.18], which covers the case in which M and N are deterministic and only N is linear and nondeleting. Finally, [33, Theorem 26] presents the full generality and matches Case (a) of the unweighted setting exactly.

Theorem 8 (see [33, Theorem 26]). *If the tdtt N is linear and nondeleting, then $\tau_{M;N} = \tau_M \, ; \tau_N$.*

Case (b) is slightly problematic in the weighted setting, and the only known generalizations are actually instances of Case (d). Let us illustrate the problem. The tdtt N can delete an intermediate subtree t' that was output by M as the result of processing an input subtree s'. In the composed tdtt, the input subtree s' is deleted right away without processing it. This phenomenon is abstractly illustrated in Fig. 9. In addition, we showcase a derivation using our example xtt of Example 7 in Fig. 10 (note that only linear rules of N are used in this derivation). Thus, the actual input subtree s' and the intermediate subtree t' are not relevant in the composed tdtt. In the unweighted setting, this independence is guaranteed by the totality of M, which yields that for each input tree s there exists a translation t of it. In other words, for each input tree $s \in T_\Sigma$ and state $q \in Q$, we have

$$\sum_{t \in T_\Gamma} \Big(\sum_{\substack{\rho_1, \dots, \rho_k \in R_1 \\ q(s) \Rightarrow_M^{\rho_1} ; \cdots ; \Rightarrow_M^{\rho_k} t}} \Big(\prod_{i=1}^{k} R_1(\rho_i) \Big) \Big) = 1 \enspace .$$

[17] More precisely, the restrictions only transfer to xtt over non-rings due to a result by Wang [44,45]. A ring is a semiring that has additive inverses; i.e., there exists an element -1 such that $1 + (-1) = 0$.

[18] Although such compositions can, in principle, succeed. Mind that the counter-examples of [12,5] only generalize to non-rings. In fact, given a suitably strong ring, any composition might become possible.

[19] In fact, [31] proves closure under composition for a slightly more general class, but the mentioned result can be obtained easily by instantiating the more general construction to our weighted tdtt model.

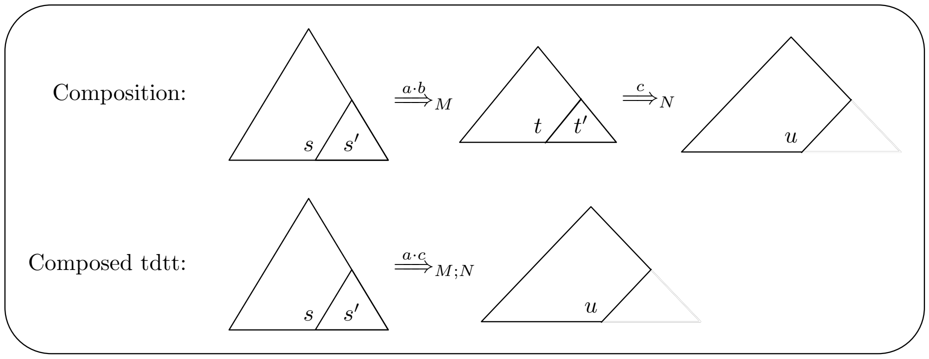

Fig. 9. Difference between composition and the composed tdtt. Atop the arrows we mark the weight and next to it the tdtt, in which the derivation happens. More precisely, weight a is charged for processing s (without s'), weight b is charged for processing s', and weight c is charged for processing t (without t').

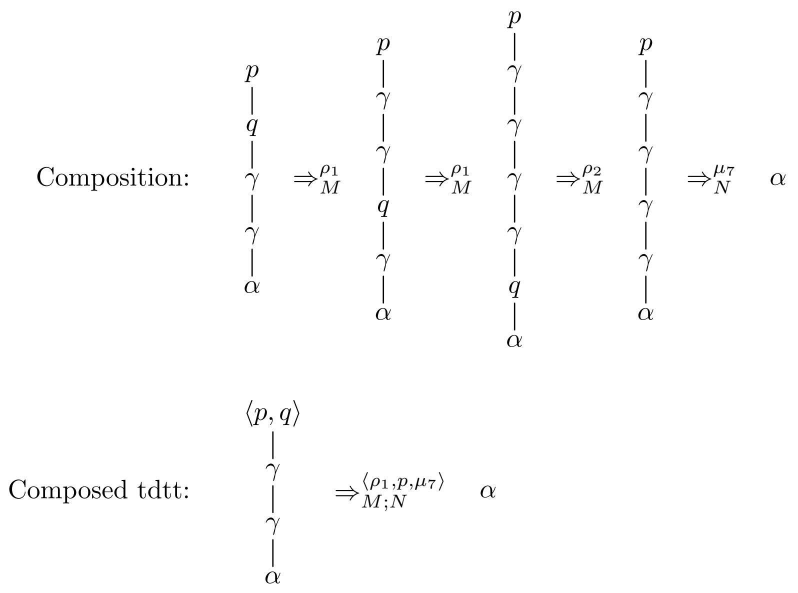

Fig. 10. Difference between composition and the composed tdtt on the tdtt M and N of Example 7. The composition charges weight $R_1(\rho_1)\cdot R_1(\rho_1)\cdot R_1(\rho_2)\cdot R_2(\mu_7) = 2^2\cdot 2\cdot 3$, whereas the composed tdtt only charges $2 \cdot 3$, which is the weight of the rule $\langle \rho_1, p, \mu_7\rangle$. The charge $R_1(\rho_1) \cdot R_1(\rho_2) = 2 \cdot 2$ for processing the input subtree $\gamma(\alpha)$ is lost.

Clearly, in the Boolean semiring, the previous equation is fulfilled if there is at least one derivation from $q(s)$ to some t. To obtain a generalization of the requirement for the weighted setting, we observe that the composed tdtt also ignores the input subtree s.

Definition 9. *A state $q \in Q$ is* constant *if there exists a semiring element $a \in A$ such that for every $s \in T_\Sigma$ we have*

$$\sum_{t \in T_\Gamma} \Bigl(\sum_{\substack{\rho_1, \dots, \rho_k \in R_1 \\ q(s) \Rightarrow_M^{\rho_1} ; \cdots ; \Rightarrow_M^{\rho_k} t}} \Bigl(\prod_{i=1}^{k} R_1(\rho_i) \Bigr) \Bigr) = a \enspace . \tag{5}$$

*We also say that q is a-*constant *if q is constant using the semiring element a. The xtt M is* constant *if all its states $q \in Q$ are constant.*

Note that the sums in (5) are always finite, which we already showed at the beginning of Sect. 5. Let us demonstrate some constant tdtt, in which all states are 1-constant. In general, different states of a constant tdtt can have different semiring elements for which they are constant.

Example 10. All of the following tdtt have only 1-constant states:

- every total tdtt over the Boolean semiring,
- every Boolean and total tdtt over an idempotent semiring, and
- every deterministic, total, and Boolean tdtt over any semiring.

Clearly, the total tdtt M of Example 7 is not constant, which is also shows that a total tdtt is not necessarily constant. If the tdtt M is constant, then we can perfectly predict the missing weight b in the derivation of the composed tdtt in Fig. 9 and charge it for the rule that actually performs the deletion.[20] Note that our presented composition construction (see Definition 6) might fail, but the authors believe that it can be modified as indicated to obtain the following result.

Conjecture 11. If the xtt M is constant and the tdtt N is linear, then $\tau_M ; \tau_N$ can be computed by an xtt.

Note that Conjecture 11 covers all the cases (for M) mentioned in Example 10. It remains to be determined whether the indicated adjustment actually works. Moreover, depending on the semiring, it might be difficult to determine whether a state is constant, so additional syntactic requirements that lead to constant states (potentially with a weight different from 0 and 1) would be desirable.

Case (c) is also problematic and has not been addressed in the literature. This is due to the fact that an intermediate output tree t' can be copied by N.

[20] In fact, we can only predict the aggregated weight (as opposed to the weights of single deleted derivations), but that is sufficient.

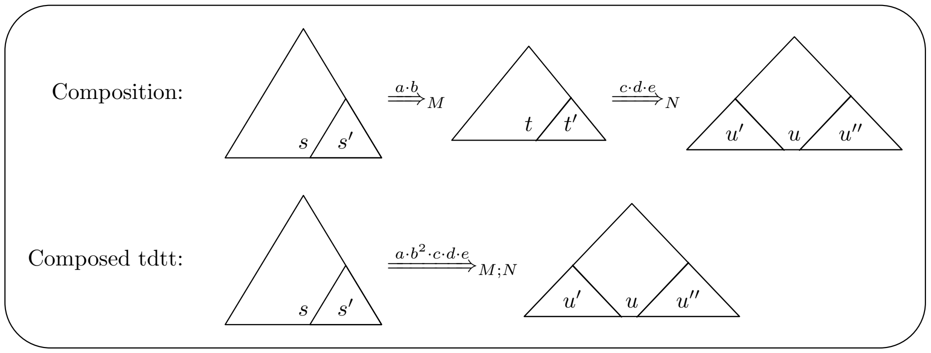

Fig. 11. Another difference between composition and the composed tdtt. Atop the arrow we mark the weight and next to it the tdtt, in which the derivation happens. More precisely, weight a is charged for processing s (without s'), weight b is charged for processing s', weight c is charged for processing t (without t'), and weights d and e are charged for processing t' (producing u' and u'', respectively).

In the composition, the weight charged for generating the tree t' from an input subtree s' is charged once, but in the composed tdtt this weight is charged twice since the input subtree s' will be copied and processed twice. The process is illustrated in Fig. 11. In addition, we provide derivations using the xtt of Example 7 that demonstrate the phenomenon in Fig. 12 (note that the tdtt M of Example 7 is deterministic and we only used nondeleting rules of N in these derivations).

So again our generic composition construction (see Definition 6) might fail, but contrary to the previous case, the authors believe that this can be addressed without any further requirement. Instead of using the original state of M in all copies, the authors propose to use the corresponding state from an unweighted copy of M in all but one copies. Thus, the weight that M charges for processing the input tree would only be charged in the single copy and the other copies, which run using the unweighted copy of M, do not cause additional charges for processing the input. Since the input tdtt M is deterministic, we know that the copies will behave equally in all aspects besides the weight that they charge. Let us provide some detail.

Definition 12. *An* unweighted copy *of M is a* Boolean *xtt* $(Q, \Sigma, \Gamma, I_1, R'_1)$ *such that*

$$l \to r \in R'_1 \iff l \to r \in R_1 \enspace .$$

We will not formalize the modified construction, but we will present the essential steps. First, we take the (disjoint) union of M and an unweighted copy M' of M (by renaming all states of the copy from q to $\overline{q}$). Let us assume that a state $q \in Q$ corresponds to a state $\overline{q}$ in M', and similarly, a rule $\rho \in R_1$ corresponds to a rule $\overline{\rho}$ in M'. When processing a rule in which the tdtt N copies, we modify all but one copies to use the corresponding state from M'. Let

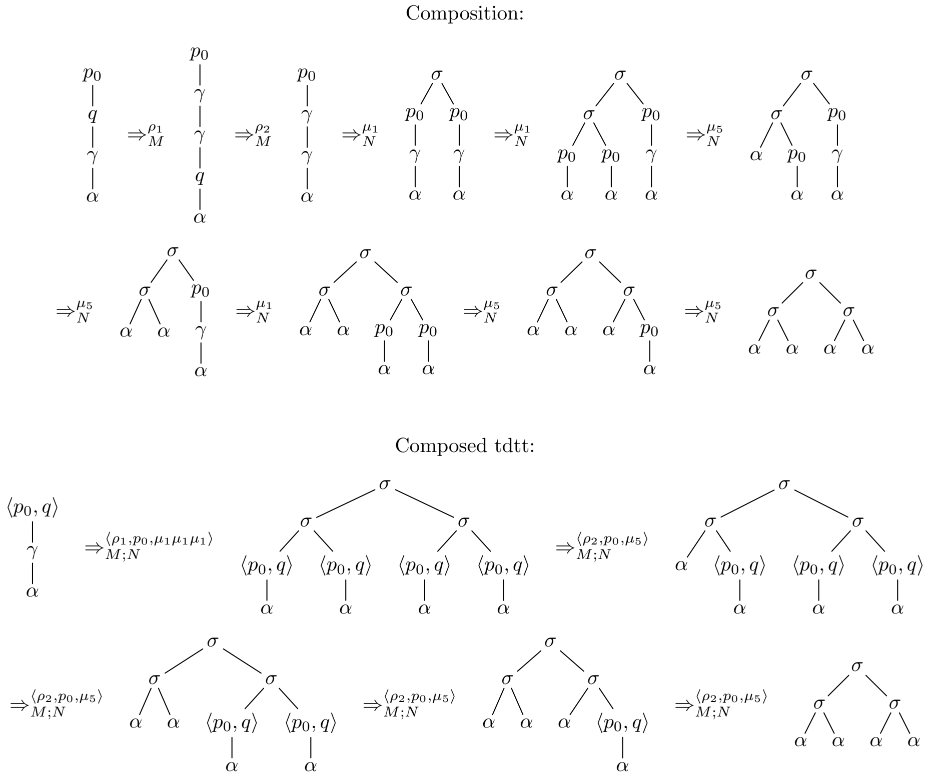

Fig. 12. Difference between composition and the composed tdtt on the tdtt M and N of Example 7. The composition (upper display) charges the weight $R_1(\rho_1) \cdot R_1(\rho_2) \cdot R_2(\mu_1)^3 \cdot R_2(\mu_5)^4 = 2 \cdot 2 \cdot 4^3 \cdot 1^4$, whereas the composed tdtt charges $2 \cdot 4^3 \cdot (2 \cdot 1)^4$. The additional weight $R_1(\rho_2)^3 = 2^3$ is charged by $M\,;N$ for processing the input subtree α (using ρ_2) three more times.

us illustrate this adjustment on an example rule of Example 7. Figure 13 shows the original and the modified derivation that lead to the rule $\nu = \langle \rho_1, p_0, \mu_3\mu_6\mu_1 \rangle$ of $M \,;\, N$ and our new rule

$$\chi(\nu) = \Big(\langle p_0, q\rangle(\gamma(\mathrm{x}_1)), 16, \sigma(\gamma(\langle p, q\rangle(\mathrm{x}_1)), \sigma(\langle p_0, q\rangle(\mathrm{x}_1), \langle p_0, q\rangle(\mathrm{x}_1))) \Big)$$
$$\chi'(\nu) = \Big(\langle p_0, q\rangle(\gamma(\mathrm{x}_1)), 16, \sigma(\gamma(\langle p, \overline{q}\rangle(\mathrm{x}_1)), \sigma(\langle p_0, \overline{q}\rangle(\mathrm{x}_1), \langle p_0, q\rangle(\mathrm{x}_1))) \Big) \enspace .$$

Figure 14 shows the modified derivation corresponding to the derivation of the composed tdtt, which is displayed in Fig. 12.

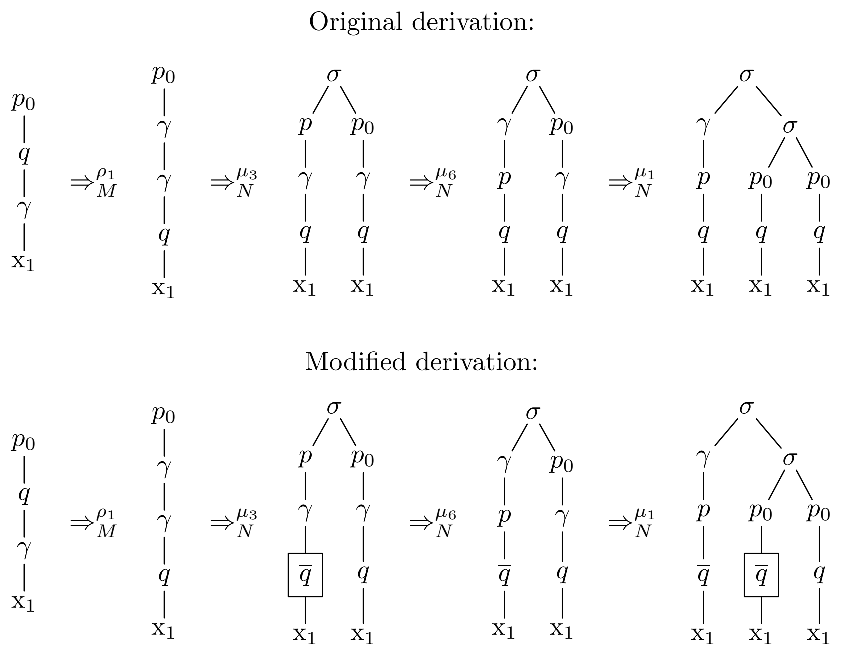

Fig. 13. Two derivations that yield rules. The upper one follows our composition construction, whereas the lower one is adjusted to address the problem of Case (c). We boxed the states that are adjusted due to copying of N. We selected to mark the left copies, but the choice is arbitrary.

Conjecture 13. If the xtt M is deterministic and the tdtt N is nondeleting, then $\tau_M \,;\, \tau_N$ can be computed by an xtt.

Finally, Case (d) is essentially a combination of Cases (b) and (c). This case was first addressed by [14, Theorem 5.18], in which it was shown that a Boolean, deterministic, and total tdtt M can be composed with a deterministic tdtt N. A similar statement was obtained in [33, Theorem 30], where (i) the same

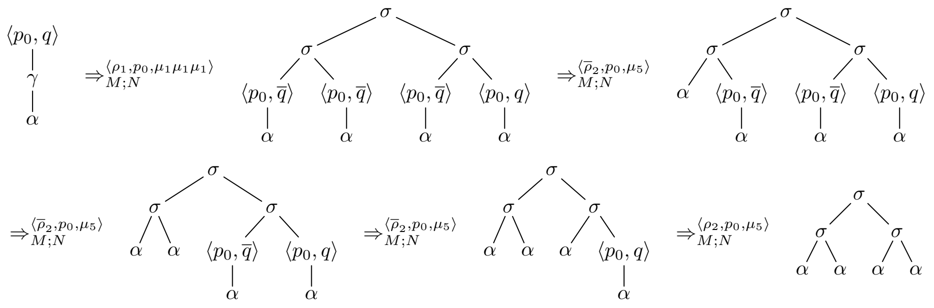

Fig. 14. Derivation using the new rules (see Fig. 12). The derivation now correctly charges the weight $R'_1(\rho_1) \cdot R'_2(\mu_1)^3 \cdot R_2(\mu_5)^4 \cdot R'_1(\rho_2) \cdot R'_1(\overline{\rho_2})^3 = 2 \cdot 4^3 \cdot 1^4 \cdot 2 \cdot 1^3$ because overlined rules charge weight 1.

restrictions are placed on M and (ii) N is required to be linear. The result of [33] clearly avoids the problematic Case (c) by requiring N to be linear. The result of [14] allows non-linear tdtt N, and it could thus be reasoned that they also had to handle the problematic Case (c). However, the requirement that the tdtt M is BOOLEAN already enforces that the additional weights (see Figs. 11 and 12) charged by the composed tdtt (constructed according to the general composition construction of Definition 6) are all 1. Thus, no modification was necessary under their assumptions. Using the indicated improvements suggested in Cases (b) and (c), the authors conjecture the following result, which covers both known results. Essentially, the authors believe that a constant and deterministic xtt M can be composed with any tdtt N.

Conjecture 14. If the xtt M is constant and deterministic and N is a tdtt, then $\tau_M \mathbin{;} \tau_N$ can be computed by an xtt.

This concludes our investigation of compositions of tdtt. Table 2 shows the various results obtained in the weighted case. It is interesting that if Conjectures 11, 13, and 14 were true, then we would recover the beautiful symmetry that is present in the composition results [12,5] for unweighted top-down and bottom-up tree transducers [43] also in the weighted case. A summary of the composition results for weighted bottom-up tree transducers [14] can be found in Table 3, but the reader is referred to [14,33,34] for the detailed results.

6 Allowing ε-rules

This section is devoted to compositions of tree transformations computed by xtt M and N, of which the xtt N is a top-down tree transducer with ε-rules [39]. Roughly speaking, a top-down tree transducer with ε-rules is an xtt, in which simple and ε-rules are allowed. In other words, this section investigates the effect of ε-rules in N to the results of Sect. 5. In the unweighted setting, this scenario was investigated in [39], and we essentially report the results of [39], which we

Table 2. Composition results for weighted tdtt ('nondel.' abbreviates 'nondeleting' and 'det.' abbreviates 'deterministic').

Case	M	N	Reference
(a)	linear and nondel.	linear and nondel.	[31, Theorem 2.4]
	deterministic	det., linear, nondel.	[14, Theorem 5.18]
		linear and nondel.	[33, Theorem 26]
(b)	constant	linear	Conjecture 11
(c)	deterministic	nondeleting	Conjecture 13
(d)	Boolean, det., total	deterministic	[14, Theorem 5.18]
	Boolean, det., total	linear	[33, Theorem 30]
	constant and det.		Conjecture 14

adjusted to our weighted setting. Let us start with the formal definition of the requirements of this section. For the rest of this section, let $M = (Q, \Sigma, \Gamma, I_1, R_1)$ and $N = (P, \Gamma, \Delta, I_2, R_2)$ be the xtt that we want to compose.

Definition 15 (cf. [15, Definition 4] and [39, Definition 1]).

- *The xtt M is* shallow *if $|\text{pos}_\Gamma(r)| \leq 1$ for every $l \to r \in R_1$.*
- *The xtt N is a* tdtt with ε-rules *if $|\text{pos}_\Gamma(l)| \leq 1$ for every $l \to r \in R_2$.*

Clearly, each tdtt is a tdtt with ε-rules, but a tdtt need not be shallow. Let us examine these properties for the xtt in our examples.

xtt	tdtt with ε-rules	shallow
M of Example 1	no (due to rule ρ_1)	yes
N of Example 2	yes	no (due to rule μ_3)
M of Example 7	yes (because it is a tdtt)	no (due to rule ρ_1)
N of Example 7	yes (because it is a tdtt)	yes

Now we can formally define the goal of this section. We will investigate compositions of xtt M and N such that M is shallow and N is a tdtt with

Table 3. Composition results for weighted bottom-up tree transducers [14] ('nondel.' abbreviates 'nondeleting' and 'det.' abbreviates 'deterministic') for comparison. Note that every weighted bottom-up tree transducer can be made total.

Case	M	N	Reference
(a)	linear, nondel.	linear and nondel.	[31, Theorem 2.4]
	linear, nondel.	homomorphism	[14, Corollary 5.5]
	linear, nondel.		[33, Theorem 13]
(b)	linear	[total]	[33, Theorem 20]
(c)	nondeleting	Boolean, deterministic	[33, Theorem 24]
	nondeleting	constant, deterministic	conjectured
(d)		Boolean, homomorphism	[14, Corollary 5.5]
		Boolean, det., [total]	[33, Theorem 24]
		constant, det., [total]	conjectured

ε-rules. To show that the condition that ensures well-definedness of the sum in the definition (2) of composition does not influence the results much, we additionally assume here that N does only have producing rules. In this case, there can only be finitely many rule applications generating the output tree u, which limits the size of the intermediate tree [see (3)]. Thus, all compositions are well-defined in the cases of this section.

6.1 Construction

Before we present an adaptation of the generic construction in Sect. 5.1, let us demonstrate that the generic construction fails to handle ε-rules of N in a meaningful manner.

Example 16. Let M and N be the xtt of Examples 1 and 2. Using the notions of Sect. 5, we can select $c = 1$ and $s = 1$. Consequently, we consider $m = 1$, which yields that all rule identifiers constructed in Definition 6 use at most one rule of N.[21] A derivation like the one depicted in Fig. 15, which starts with a rule of N, cannot be simulated by $M \,;N$ because the rules constructed for $M \,;N$ always trigger a rule of M first. □

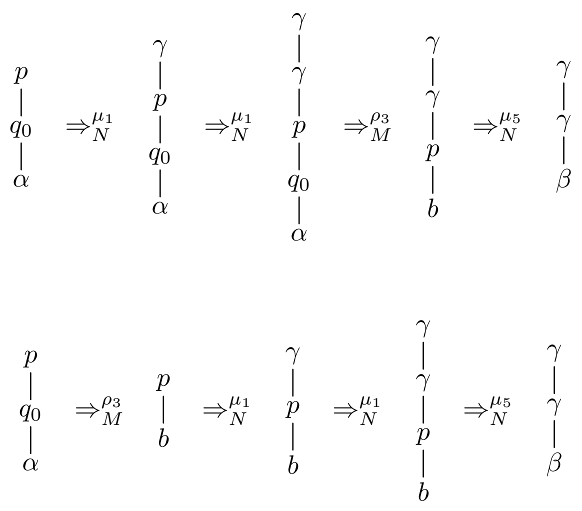
$p(q_0(\alpha)) \Rightarrow_N^{\mu_1} \gamma(p(q_0(\alpha))) \Rightarrow_N^{\mu_1} \gamma(\gamma(p(q_0(\alpha)))) \Rightarrow_M^{\rho_3} \gamma(\gamma(p(b))) \Rightarrow_N^{\mu_5} \gamma(\gamma(\beta))$
$p(q_0(\alpha)) \Rightarrow_M^{\rho_3} p(b) \Rightarrow_N^{\mu_1} \gamma(p(b)) \Rightarrow_N^{\mu_1} \gamma(\gamma(p(b))) \Rightarrow_N^{\mu_5} \gamma(\gamma(\beta))$

Fig. 15. Two derivations using the xtt M and N of Examples 1 and 2. The upper derivation cannot be simulated by $M\,;N$ since it starts with a rule of N. In principle, an unbounded number of rule applications of rule μ_1 could happen before the intermediate symbol b is consumed in the lower derivation. Thus, such derivations can, in general, also not be simulated by $M \,;N$.

[21] We could not avoid the problem, even if we would consider larger values for m.

Thus, we need to adjust our construction. To avoid the problem in the lower derivation of Fig. 15, we restrict the rules of N that can be used when processing the right-hand side r of a rule $\rho \in R_1$. As in [39] we require that r is processed only with consuming rules of N. The ε-rules of N need to fire either before ρ or after all intermediate symbols of r are fully consumed by N. This creates a problem, if the rule ρ creates 2 intermediate symbols at the same time, and the original derivation uses ε-rules after consuming one intermediate symbol but before consuming the second intermediate symbol. To avoid this problem, we already assumed in this section that M is shallow. Consequently, $m = 1$ provides an upper bound to the number of consuming rules required by N to process the right-hand side of a rule of M. This is due to the fact that there is at most one intermediate symbol in any right-hand side of a rule of M, and we can only use consuming rules of N to process it. As before, for any sentential form $\xi \in T_\Delta(P(T_\Gamma(Q(T_\Sigma(\mathrm{X})))))$ and rule $\rho \in R$ there exists at most one $\zeta \in T_\Delta(P(T_\Gamma(Q(T_\Sigma(\mathrm{X})))))$ such that $\xi \Rightarrow_M^\rho \zeta$, which also holds for the xtt N. Similarly, we recall that we identify elements of $T_\Delta(P(Q(T_\Sigma(\mathrm{X}))))$ with elements of $T_\Delta((P \times Q)(T_\Sigma(\mathrm{X})))$ in the obvious manner.

Definition 17 (cf. [39, Definition 9]). *The* ε-composition $M ;_\varepsilon N$ *of* M *and* N *is the xtt* $(P \times Q, \Sigma, \Delta, I_2 \times I_1, \mathcal{R}, \chi)$ *with rule identifiers*

$$\begin{aligned} \mathcal{R} = \{\langle \rho, p, \varepsilon\rangle \mid \textit{erasing } \rho \in R_1, p \in P\} \cup \\ \cup \{\langle \rho, p, \mu\rangle \mid \textit{producing } \rho \in R_1, p \in P,\ \textit{consuming } \mu \in R_2\} \cup \\ \cup \{\langle \varepsilon, q, \mu\rangle \mid q \in Q,\ \varepsilon\textit{-rule } \mu \in R_2\} \end{aligned}$$

such that

- $\chi(\langle l \to r, p, \varepsilon\rangle) = (p(l), R_1(l \to r), p(r))$ *for every erasing rule* $l \to r \in R_1$ *and* $p \in P$,
- $\chi(\langle l \to r, p, \mu\rangle) = (p(l), a, r')$, *where*

$$a = \begin{cases} R_1(l \to r) \cdot R_2(\mu) & \textit{if } p(l) \Rightarrow_M^{(l,r)} ; \Rightarrow_N^\mu r' \\ 0 & \textit{otherwise} \end{cases}$$

  for every producing $l \to r \in R_1$, $p \in P$, *and consuming* $\mu \in R_2$, *and*
- $\chi(\langle \varepsilon, q, l \to r\rangle) = (l\theta, R_2(l \to r), r\theta)$, *where* $\theta(x) = q(x)$ *for every* $x \in \mathrm{X}$, $q \in Q$, *and* ε-*rule* $l \to r \in R_2$.

Note that the only differences to the construction of [39] are the presence of (i) non-simple left-hand sides in rules of M and (ii) weights. Let us discuss the three sets of rule identifiers mentioned in Definition 17. Rule identifiers of the form $\langle \rho, p, \varepsilon\rangle$ refer to variants of an erasing rule ρ of R_1. For each state $p \in P$, we obtain a variant by annotating the two states (in the left- and right-hand side) by p. In other words, we perform a step using M, but since no intermediate symbol is produced, we do not perform a step using N. Second, the rule identifiers of the form $\langle \rho, p, \mu\rangle$ contain rules that are obtained in the usual way by processing the right-hand side of a producing rule of M by consuming rules of N. Since M is shallow and N is a tdtt with ε-rules, each producing rule of M contains exactly one intermediate symbol and each consuming rule of N contains exactly one intermediate symbol. Thus, the derivation only succeeds if the producing rule of M produces exactly the symbol that the consuming rule of N consumes. These two types of rules were also present in the generic composition construction of Sect. 5.1. Finally, rule identifiers of the form $\langle \varepsilon, q, \mu \rangle$ refer to a variant of an ε-rule μ of N that is annotated with the state $q \in Q$.

Let us quickly check whether the obtained rules $l \to r$ are admissible; i.e., whether $\{l, r\} \not\subseteq P(Q(\mathrm{X}))$. Clearly, identifiers of the form $\langle \rho, p, \varepsilon \rangle$ yield admissible rules because they contain just copies of rules of M. The same reasoning applies to rules with identifiers of the form $\langle \varepsilon, q, \mu \rangle$, which are copies of rules of N. Finally, rules with identifiers like $\langle \rho, p, \mu \rangle$ are always producing because each rule $\mu \in R_2$ is producing. Clearly, producing rules are admissible. Next, let us illustrate the construction.

Example 18. Let $M = (\{q_0, q_1\}, \Sigma, \Gamma, \{q_0\}, R)$ and $N = (\{p\}, \Gamma, \Delta, \{p\}, R')$ be the xtt of Examples 1 and 2, respectively. The composition construction of Definition 17 yields the xtt $M ;_\varepsilon N = (P \times Q, \Sigma, \Delta, \{\langle p, q_0 \rangle\}, \mathcal{R}, \chi)$ with rule identifiers

$$\begin{aligned}\mathcal{R} = \{&\langle \rho_2, p, \varepsilon \rangle, \langle \rho_1, p, \mu_2 \rangle, \langle \rho_1, p, \mu_3 \rangle, \langle \rho_3, p, \mu_5 \rangle, \langle \rho_4, p, \mu_4 \rangle, \langle \rho_5, p, \mu_4 \rangle,\\ &\langle \varepsilon, q_0, \mu_1 \rangle, \langle \varepsilon, q_1, \mu_1 \rangle\}\end{aligned}$$

such that

$$\begin{aligned}
\chi(\langle \rho_2, p, \varepsilon \rangle) &= \langle p, q_0 \rangle(\sigma(\alpha, \mathrm{x}_1)) \xrightarrow{3} \langle p, q_0 \rangle(\mathrm{x}_1)\\
\chi(\langle \rho_1, p, \mu_2 \rangle) &= \langle p, q_0 \rangle(\sigma(\sigma(\mathrm{x}_1, \mathrm{x}_2), \alpha)) \xrightarrow{2 \cdot 5} \lambda(\langle p, q_1 \rangle(\mathrm{x}_1), \langle p, q_0 \rangle(\mathrm{x}_2))\\
\chi(\langle \rho_1, p, \mu_3 \rangle) &= \langle p, q_0 \rangle(\sigma(\sigma(\mathrm{x}_1, \mathrm{x}_2), \alpha)) \xrightarrow{2 \cdot 5} \lambda(\beta, \lambda(\langle p, q_1 \rangle(\mathrm{x}_1), \langle p, q_0 \rangle(\mathrm{x}_2)))\\
\chi(\langle \rho_3, p, \mu_5 \rangle) &= \langle p, q_0 \rangle(\alpha) \xrightarrow{1 \cdot 1} \beta\\
\chi(\langle \rho_4, p, \mu_4 \rangle) &= \langle p, q_1 \rangle(\mathrm{x}_1) \xrightarrow{2 \cdot 8} \delta(\langle p, q_0 \rangle(\mathrm{x}_1))\\
\chi(\langle \rho_5, p, \mu_4 \rangle) &= \langle p, q_1 \rangle(\mathrm{x}_1) \xrightarrow{5 \cdot 8} \delta(\langle p, q_1 \rangle(\mathrm{x}_1))\\
\chi(\langle \varepsilon, q_0, \mu_1 \rangle) &= \langle p, q_0 \rangle(\mathrm{x}_1) \xrightarrow{2} \gamma(\langle p, q_0 \rangle(\mathrm{x}_1))\\
\chi(\langle \varepsilon, q_1, \mu_1 \rangle) &= \langle p, q_1 \rangle(\mathrm{x}_1) \xrightarrow{2} \gamma(\langle p, q_1 \rangle(\mathrm{x}_1)) \enspace .
\end{aligned}$$

The construction of the rule corresponding to the rule identifier $\langle \rho_1, p, \mu_2 \rangle$ is illustrated in Fig. 16. □

6.2 Correctness

Let us start by recalling the two cases, in which the composition construction of [39], of which our construction in Definition 17 is an adaptation, is successful in the unweighted setting. Recall that M is shallow and N is a tdtt with ε-rules. If

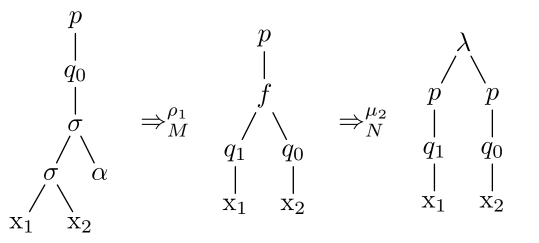

Fig. 16. Construction of the rule with identifier $\langle \rho_1, p, \mu_2 \rangle$ in Example 18.

Table 4. Cases for unweighted xtt composition of M and N, where M is shallow and N is a tdtt with ε-rules.

Case	M	N	Reference
(a)		linear and nondeleting	[39, Theorem 17]
(b)	total	linear	[39, Theorem 17]

– N is linear, and
– M is total or N is nondeleting,

then $\tau_M \,;\, \tau_N$ can be computed by an xtt [39, Theorem 17]. Table 4 shows these two cases, which correspond to the equally named cases in Sect. 5.

Let us start with Case (a). As in the previous section, this case does not cause further problems in the weighted setting, and we will sketch the correctness proof for our composition construction of Definition 17.

Theorem 19. *If M is shallow and N is a linear and nondeleting tdtt with ε-rules, then $\tau_{M ;_\varepsilon N} = \tau_M \,;\, \tau_N$.*

Proof (sketch). Let $\xi \in P(Q(T_\Sigma))$ and $u \in T_\Delta$. We claim that there is a weight-preserving bijection between the derivations of the form

$$\xi \; (\Rightarrow_M^{\rho_1} ; \cdots ; \Rightarrow_M^{\rho_k}) \,;\, (\Rightarrow_N^{\mu_1} ; \cdots ; \Rightarrow_N^{\mu_n}) \; u \enspace ,$$

and the derivations of the form $\xi \Rightarrow_{M;_\varepsilon N}^{\nu_1} ; \cdots ; \Rightarrow_{M;_\varepsilon N}^{\nu_\ell} u$.

We construct the bijection by induction on k. Let $s \in T_\Sigma$, $p \in P$, and $q \in Q$ be such that $\xi = p(q(s))$. Next, we distinguish whether the first applied rule ρ_1 is erasing. If it is, then we start the derivation using $M ;_\varepsilon N$ with the rule $\langle \rho_1, p, \varepsilon \rangle$, which has the same weight as ρ_1. Otherwise, the rule ρ_1 produces exactly one intermediate symbol $\gamma \in \Gamma$ that will be consumed by exactly one rule μ_i for some $i \in \mathbb{N}$. The symbol γ is consumed by exactly one rule because N is linear and nondeleting. Clearly, all rules $\mu_1, \ldots, \mu_{i-1}$ before μ_i must be ε-rules because otherwise they would consume the symbol γ. In $M ;_\varepsilon N$ we simulate this derivation by starting with the ε-rules $\langle \varepsilon, q, \mu_1 \rangle, \ldots, \langle \varepsilon, q, \mu_{i-1} \rangle$ followed by the consuming rule $\langle \rho_1, p', \mu_i \rangle$, where p' is the (unique) state that occurs in the right-hand side of the rule μ_{i-1}.[22] Clearly, this part of the derivation has the same weight as the

[22] If $i = 1$, then we let $p' = p$.

product of the weight of rule ρ_1 and the weights of the rules $\mu_1, \dots, \mu_i$. Now we covered all three cases and shortened the derivation using M. The remainder of the derivation can then be processed using the induction hypothesis. Thus, our construction relates derivations bijectively and preserves the weight. Given this bijective and weight-preserving relation, the main statement follows trivially. □

Let us illustrate the construction used in the proof of Theorem 19.

Example 20. Let M and N be the xtt of Examples 1 and 2, and recall that $M \mathbin{;_\varepsilon} N$ is shown in Example 18. Moreover, let

$$s = \sigma(\sigma(\alpha, \sigma(\alpha, \sigma(\alpha, \alpha))), \alpha) \qquad \text{and} \qquad t = f(h(h(b)), b)$$

be an input and output tree for M as in Example 1. Figure 2 shows a derivation d_M with weight 180 from $q_0(s)$ to t using M. Moreover, let

$$u = \lambda(\beta, \lambda(\delta(\delta(\beta)), \gamma(\gamma(\beta)))) \ .$$

Figure 17 shows a derivation d_N with weight $5 \cdot 8^2 \cdot 1 \cdot 2^2 \cdot 1 = 1\,280$ from $p(t)$ to u using N. The concatenation of the two derivations gives us a derivation d from $p(q_0(s))$ to u using rules of M and N. Clearly, the weight of this derivation is $180 \cdot 1\,280 = 230\,400$.

The image of the derivations d_M and d_N by the bijection constructed in the proof of Theorem 19 is shown in Fig. 18. The first four rules in the derivation d_M are producing, and the produced symbol is immediately consumed in the corresponding step in the derivation d_N. The fifth and sixth rules in the derivation d_M are erasing rules, which are simulated by the corresponding erasing rules in the derivation d. The last rule in the derivation d_M is another producing rule, whose produced symbol b is not immediately consumed in the current step of the derivation d_N. Rather the ε-rule μ_1 is applied twice before rule μ_5 consumes the symbol b. Consequently, we have to defer the application of the rule $\langle \rho_3, p, \mu_5 \rangle$ to first allow the applications of the rule $\langle \varepsilon, q_0, \mu_1 \rangle$. The following table lists the rule applications for all three derivations and shows the correspondence.

Step	1	2	3	4	5	6	7
d_M:	ρ_1	ρ_5	ρ_4	ρ_3	ρ_2	ρ_2	ρ_3
d_N:	μ_3	μ_4	μ_4	μ_5			μ_1 μ_1 μ_5
d:	$\langle \rho_1, p, \mu_3 \rangle$	$\langle \rho_5, p, \mu_4 \rangle$	$\langle \rho_4, p, \mu_4 \rangle$	$\langle \rho_3, p, \mu_5 \rangle$	$\langle \rho_2, p, \varepsilon \rangle$	$\langle \rho_2, p, \varepsilon \rangle$	$\langle \varepsilon, q_0, \mu_1 \rangle$ $\langle \varepsilon, q_0, \mu_1 \rangle$ $\langle \rho_3, p, \mu_5 \rangle$

The weight of the derivation d is

$$(2 \cdot 5) \cdot (5 \cdot 8) \cdot (2 \cdot 8) \cdot (1 \cdot 1) \cdot 3 \cdot 3 \cdot 2 \cdot 2 \cdot (1 \cdot 1) = 230\,400 \ ,$$

which coincides with the expected result. □

Let us move on to Case (b), in which we experience the same problem with deleted subtrees as in Sect. 5. We refer the reader to the discussion of Case (b) in Sect. 5 for an illustration of the problem and examples. Here, we avoid the problem by requiring (i) that the xtt M is Boolean and (ii) that the semiring A is idempotent (i.e., $1 + 1 = 1$). This yields that the xtt M is essentially

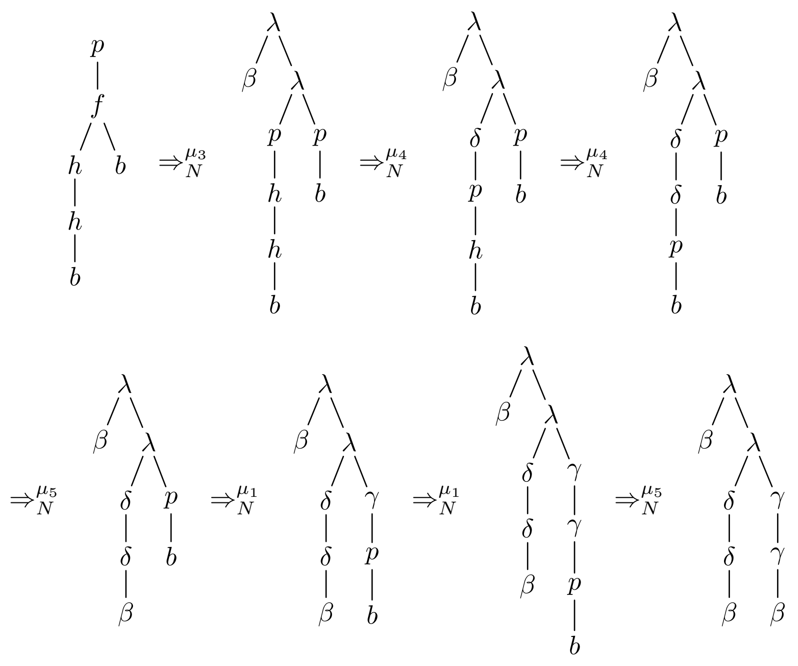

Fig. 17. A derivation from $p(t)$ to u using N (see Example 20).

unweighted and constant (with weight 1). We observe that $\tau_M(s,t) = 1$ for all $(s,t) \in \text{supp}(\tau_M)$ because M is Boolean and A is idempotent, and for every $s \in T_\Sigma$ there exists $t \in T_\Gamma$ such that $(s,t) \in \text{supp}(\tau_M)$ due to the totality of M.

Theorem 21. *If the shallow xtt M is total and* Boolean, *the tdtt N with ε-rules is linear, and the semiring A is idempotent, then $\tau_{M ;_\varepsilon N} = \tau_M ; \tau_N$.*

Proof (sketch). Let $\xi \in P(Q(T_\Sigma))$ and $u \in T_\Delta$. We claim that there is a weight-preserving surjective mapping f from the derivations of the form

$$\xi \; (\Rightarrow_M^{\rho_1} ; \cdots ; \Rightarrow_M^{\rho_k}) ; (\Rightarrow_N^{\mu_1} ; \cdots ; \Rightarrow_N^{\mu_n}) \; u \enspace ,$$

and the derivations of the form $\xi \Rightarrow_{M;_\varepsilon N}^{\nu_1} ; \cdots ; \Rightarrow_{M;_\varepsilon N}^{\nu_\ell} u$.

Clearly, the derivation sequence $\rho_1 \cdots \rho_k \mu_1 \cdots \mu_n$ is successful. Let $\perp$ be a fresh symbol, and let $l \to r \in R_1$ be a rule of M. The *mutilated copy* of $l \to r$ is the rule $l \to \perp(r)$. We denote the mutilated copy of $\rho \in R_1$ by $\overline{\rho}$. Next, we obtain a rule sequence $\rho'_1 \cdots \rho'_k$ from $\rho_1 \cdots \rho_k$ by replacing maximally many rules ρ_i by their mutilated copy $\overline{\rho_i}$ such that

$$\xi \; (\Rightarrow_M^{\rho'_1} ; \cdots ; \Rightarrow_M^{\rho'_k}) ; (\Rightarrow_N^{\mu_1} ; \cdots ; \Rightarrow_N^{\mu_n}) \; u \enspace .$$

In order words, the new rule sequence is still a successful derivation from ξ to u. Clearly, this derivation can only be successful if N ignores (i.e., deletes) the subtrees created by mutilated rules because N cannot process the symbol $\perp$. In the

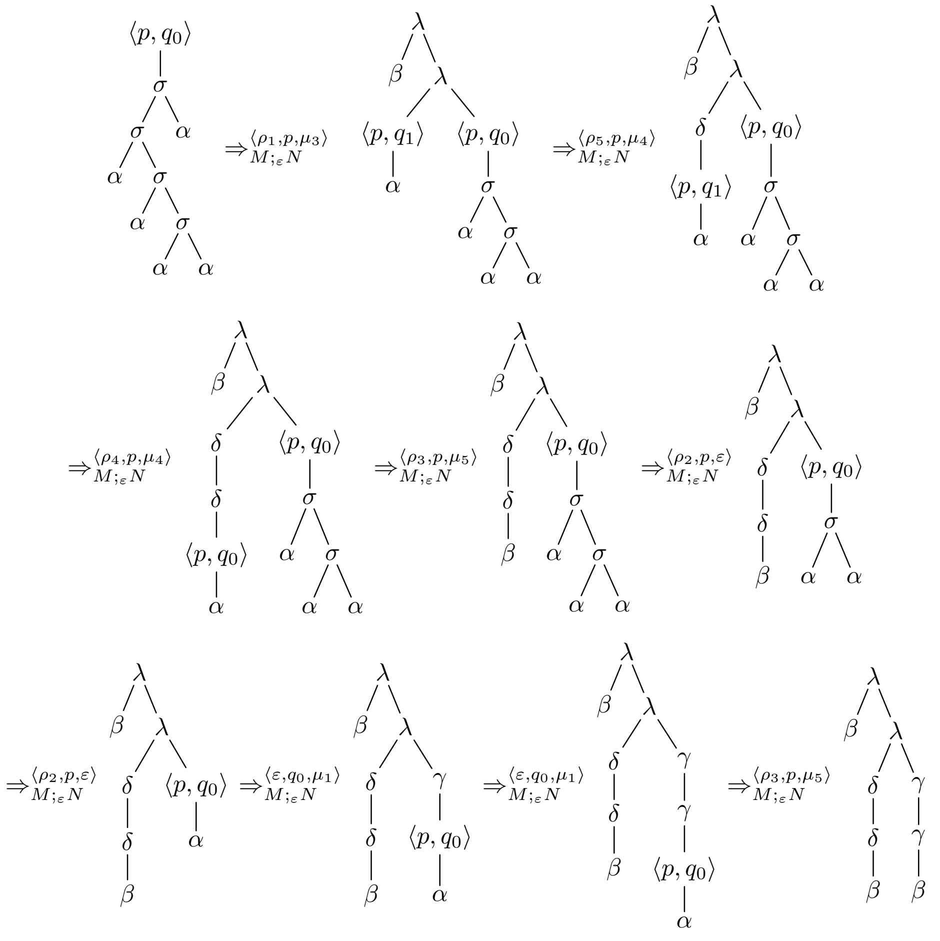
$\langle p, q_0 \rangle$
$\Rightarrow^{\langle \rho_1, p, \mu_3 \rangle}_{M ;_\varepsilon N}$
$\Rightarrow^{\langle \rho_5, p, \mu_4 \rangle}_{M ;_\varepsilon N}$
$\Rightarrow^{\langle \rho_4, p, \mu_4 \rangle}_{M ;_\varepsilon N}$
$\Rightarrow^{\langle \rho_3, p, \mu_5 \rangle}_{M ;_\varepsilon N}$
$\Rightarrow^{\langle \rho_2, p, \varepsilon \rangle}_{M ;_\varepsilon N}$
$\Rightarrow^{\langle \rho_2, p, \varepsilon \rangle}_{M ;_\varepsilon N}$
$\Rightarrow^{\langle \varepsilon, q_0, \mu_1 \rangle}_{M ;_\varepsilon N}$
$\Rightarrow^{\langle \varepsilon, q_0, \mu_1 \rangle}_{M ;_\varepsilon N}$
$\Rightarrow^{\langle \rho_3, p, \mu_5 \rangle}_{M ;_\varepsilon N}$

Fig. 18. The matching derivation from $\langle p, q_0 \rangle(s)$ to u using $M ;_\varepsilon N$ (see Example 20).

next step we drop all mutilated rules from the rule sequence $\rho'_1 \cdots \rho'_k$ and relate the obtained rule sequence in the same way as in the proof of Theorem 19 to the derivation of the composed xtt. The obtained derivation using the composed xtt has the same weight as the original derivation because we only dropped rules of R_1, which have weight 1 because M is Boolean. It is not difficult to see that this mapping is surjective because we can always recover one subderivation for parts that we dropped due to the totality of M. This approach is illustrated in an example following the proof.

Now we complete the proof as follows:

$$(\tau'_M \mathbin{;} \tau'_N)(\xi, u) = \sum_{\substack{\rho_1, \ldots, \rho_k \in R_1 \\ \mu_1, \ldots, \mu_n \in R_2 \\ \xi (\Rightarrow^{\rho_1}_M ; \cdots ; \Rightarrow^{\rho_k}_M) ; (\Rightarrow^{\mu_1}_N ; \cdots ; \Rightarrow^{\mu_n}_N) u}} \Bigl(\prod_{i=1}^k R_1(\rho_i) \cdot \prod_{i=1}^n R_2(\mu_i) \Bigr)$$

$$= \sum_{\substack{\nu_1, \ldots, \nu_\ell \in \mathcal{R} \\ d \colon \xi \Rightarrow^{\nu_1}_{M ;_\varepsilon N} ; \cdots ; \Rightarrow^{\nu_\ell}_{M ;_\varepsilon N} u}} \Bigl(\sum_{d' \in f^{-1}(d)} \Bigl(\prod_{i=1}^\ell \mathrm{wt}(\nu_i) \Bigr) \Bigr)$$

(because the second sum is never empty due to surjectivity of f)

$$= \sum_{\substack{\nu_1, \ldots, \nu_\ell \in \mathcal{R} \\ \xi \Rightarrow^{\nu_1}_{M ;_\varepsilon N} ; \cdots ; \Rightarrow^{\nu_\ell}_{M ;_\varepsilon N} u}} \Bigl(\prod_{i=1}^\ell \mathrm{wt}(\nu_i) \Bigr) = \tau'_{M ;_\varepsilon N}(\xi, u)$$

because A is idempotent and $f^{-1}(d) \neq \emptyset$. Thus, we conclude that $M ;_\varepsilon N$ computes $\tau_M \mathbin{;} \tau_N$. □

Let us illustrate the construction in the proof of Theorem 21 on an example.

Example 22. Let us consider Fig. 19, which is a minor variation of Fig. 10. Figure 19 displays two derivations that we want to relate. Obviously, the rule sequence of M is $\rho_1 \rho_1 \rho_2$. Now we need to mutilate the rules in the rule sequence. We start with the most aggressive attempt and mutilate every rule in the sequence to obtain $\overline{\rho_1 \rho_1 \rho_2}$. Figure 20 shows that the derivation is no longer successful for this rule sequence. Thus, we try the sequence $\rho_1 \overline{\rho_1 \rho_2}$, which indeed still delivers a successful derivation as depicted in Fig. 20. Next, we reduce the sequence by taking out all mutilated rules. We obtain just ρ_1. Now we combine the producing rule ρ_1 as usual with the consuming rule μ_7 of the second rule sequence and relate them to the rule $\langle \rho_1, p, \mu_7 \rangle$ of the composed tdtt. □

Following the ideas of Sect. 5, the authors suspect that instead of idempotence and a total and Boolean xtt M, we can simply require that M is constant. This leads to our final conjecture, which would generalize Theorem 21. We collect all obtained results of this section in Table 5.

Conjecture 23. If the shallow xtt M is constant and the tdtt N with ε-rules is linear, then $\tau_M \mathbin{;} \tau_N$ can be computed by an xtt.

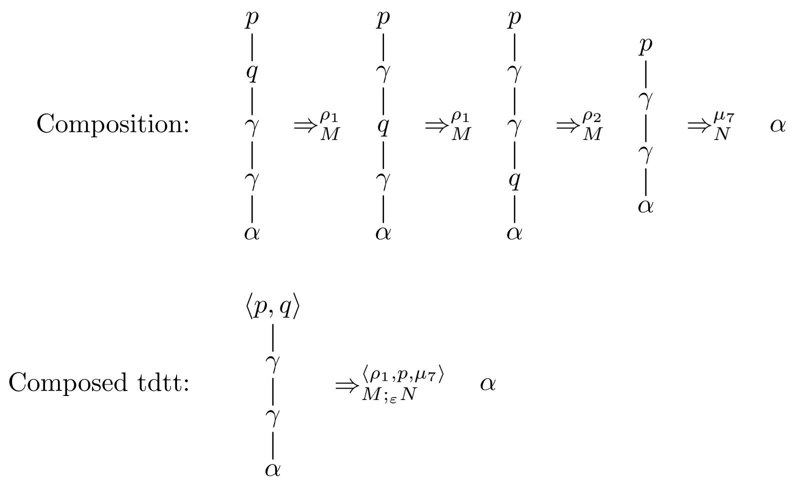

Fig. 19. Relating rule sequences.

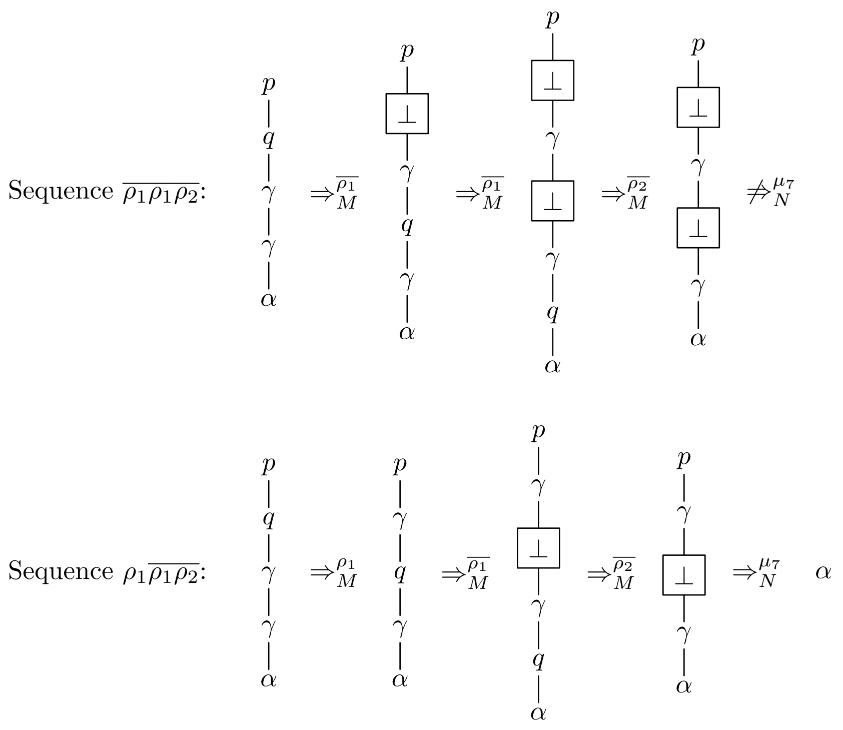
Sequence $\overline{\rho_1}\,\overline{\rho_1}\,\overline{\rho_2}$:
Sequence $\rho_1\overline{\rho_1}\,\overline{\rho_2}$:

Fig. 20. Testing two mutilated sequences. The upper one is too aggressive and the rule μ_7 of N is not applicable anymore. The lower sequence represents the sought sequence because the derivation is still successful. After the deletion of the mutilated rules, we thus obtain just ρ_1.

Table 5. Cases for weighted xtt composition, where M is a shallow xtt and N is a tdtt with ε-rules. In the second line (which uses Theorem 21), we additionally need to require that the semiring is idempotent.

Case	M	N	Reference
(a)		linear and nondeleting	Theorem 19
(b)	total and BOOLEAN	linear	Theorem 21
	constant	linear	Conjecture 23

References

1. Alexandrakis, A., Bozapalidis, S.: Weighted grammars and Kleene's theorem. Inf. Process. Lett. 24(1), 1–4 (1987)
2. Arnold, A., Dauchet, M.: Bi-transductions de forêts. In: Michaelson, S., Milner, R. (eds.) ICALP 1976. pp. 74–86. Edinburgh University Press (1976)
3. Arnold, A., Dauchet, M.: Morphismes et bimorphismes d'arbres. Theoret. Comput. Sci. 20(4), 33–93 (1982)
4. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press (1998)
5. Baker, B.S.: Composition of top-down and bottom-up tree transductions. Inform. and Control 41(2), 186–213 (1979)
6. Berstel, J., Reutenauer, C.: Recognizable formal power series on trees. Theoret. Comput. Sci. 18(2), 115–148 (1982)
7. Borchardt, B.: The Theory of Recognizable Tree Series. Ph.D. thesis, Technische Universität Dresden (2005)
8. Borchardt, B., Vogler, H.: Determinization of finite state weighted tree automata. J. Autom. Lang. Combin. 8(3), 417–463 (2003)
9. Bozapalidis, S., Louscou-Bozapalidou, O.: The rank of a formal tree power series. Theoret. Comput. Sci. 27(1–2), 211–215 (1983)
10. Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Löding, C., Tison, S., Tommasi, M.: Tree automata techniques and applications. Available at: `http://tata.gforge.inria.fr` (2007)
11. Dauchet, M.: Transductions inversibles de forêts. Thèse 3ème cycle, Université de Lille (1975)
12. Engelfriet, J.: Bottom-up and top-down tree transformations: A comparison. Math. Systems Theory 9(3), 198–231 (1975)
13. Engelfriet, J.: Top-down tree transducers with regular look-ahead. Math. Systems Theory 10(1), 289–303 (1976)
14. Engelfriet, J., Fülöp, Z., Vogler, H.: Bottom-up and top-down tree series transformations. J. Autom. Lang. Combin. 7(1), 11–70 (2002)
15. Engelfriet, J., Lilin, E., Maletti, A.: Extended multi bottom-up tree transducers. In: Ito, M., Toyama, F.M. (eds.) DLT 2008. LNCS, vol. 5257, pp. 289–300. Springer, Heidelberg (2008)
16. Engelfriet, J., Lilin, E., Maletti, A.: Composition and decomposition of extended multi bottom-up tree transducers. Acta Inform. 46(8), 561–590 (2009)
17. Ésik, Z., Kuich, W.: Formal tree series. J. Autom. Lang. Combin. 8(2), 219–285 (2003)
18. Fülöp, Z., Gazdag, Z., Vogler, H.: Hierarchies of tree series transformations. Theoret. Comput. Sci. 314(3), 387–429 (2004)

19. Fülöp, Z., Vogler, H.: Tree series transformations that respect copying. Theory Comput. Systems 36(3), 247–293 (2003)
20. 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, Heidelberg (2009)
21. Gécseg, F., Steinby, M.: Tree Automata. Akadémiai Kiadó, Budapest (1984)
22. Gécseg, F., Steinby, M.: Tree languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, chap. 1, pp. 1–68. Springer, Heidelberg (1997)
23. Golan, J.S.: Semirings and their Applications. Kluwer Academic, Dordrecht (1999)
24. Graehl, J., Knight, K., May, J.: Training tree transducers. Comput. Linguist. 34(3), 391–427 (2008)
25. Hebisch, U., Weinert, H.J.: Semirings — Algebraic Theory and Applications in Computer Science. No. 5 in Series in Algebra, World Scientific, Singapore (1998)
26. Knight, K., Graehl, J.: An overview of probabilistic tree transducers for natural language processing. In: Gelbukh, A.F. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 1–24. Springer, Heidelberg (2005)
27. Koehn, P.: Statistical Machine Translation. Cambridge University Press (2010)
28. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: NAACL 2003. pp. 48–54. Association for Computational Linguistics (2003)
29. Kühnemann, A.: Benefits of tree transducers for optimizing functional programs. In: Arvind, V., Ramanujam, R. (eds.) FSTTCS 1998. LNCS, vol. 1530, pp. 146–157. Springer, Heidelberg (1998)
30. Kuich, W.: Formal power series over trees. In: Bozapalidis, S. (ed.) DLT 1997. pp. 61–101. Aristotle University of Thessaloniki (1998)
31. Kuich, W.: Full abstract families of tree series I. In: Karhumäki, J., Maurer, H.A., Paun, G., Rozenberg, G. (eds.) Jewels are Forever, pp. 145–156. Springer, Heidelberg (1999)
32. Kuich, W.: Tree transducers and formal tree series. Acta Cybernet. 14(1), 135–149 (1999)
33. Maletti, A.: Compositions of tree series transformations. Theoret. Comput. Sci. 366(3), 248–271 (2006)
34. Maletti, A.: The Power of Tree Series Transducers. Ph.D. thesis, Technische Universität Dresden (2006)
35. Maletti, A.: Compositions of extended top-down tree transducers. Inform. and Comput. 206(9–10), 1187–1196 (2008)
36. Maletti, A.: Survey: Weighted extended top-down tree transducers — Part I: Basics and expressive power. Acta Cybernet. (2011), preprint available at: `http://www.ims.uni-stuttgart.de/$\sim$maletti/pub/mal11.pdf`
37. Maletti, A.: Survey: Weighted extended top-down tree transducers — Part II: Application in machine translation. Fund. Inform. (2011)
38. Maletti, A., Graehl, J., Hopkins, M., Knight, K.: The power of extended top-down tree transducers. SIAM J. Comput. 39(2), 410–430 (2009)
39. Maletti, A., Vogler, H.: Compositions of top-down tree transducers with ε-rules. In: Yli-Jyrä, A., Kornai, A., Sakarovitch, J., Watson, B. (eds.) FSMNLP 2009. LNAI, vol. 6062, pp. 69–80. Springer, Heidelberg (2010)
40. May, J., Knight, K., Vogler, H.: Efficient inference through cascades of weighted tree transducers. In: ACL 2010. pp. 1058–1066. Association for Computational Linguistics (2010)
41. Rounds, W.C.: Mappings and grammars on trees. Math. Systems Theory 4(3), 257–287 (1970)

42. Thatcher, J.W.: Generalized^2 sequential machine maps. J. Comput. System Sci. 4(4), 339–367 (1970)
43. Thatcher, J.W.: Tree automata: An informal survey. In: Aho, A.V. (ed.) Currents in the Theory of Computing, chap. 4, pp. 143–172. Prentice Hall, Englewood Cliffs, NJ (1973)
44. Wang, H.: On characters of semirings. Houston J. Math. 23(3), 391–405 (1997)
45. Wang, H.: On rational series and rational languages. Theoret. Comput. Sci. 205(1–2), 329–336 (1998)
46. Yamada, K., Knight, K.: A decoder for syntax-based statistical MT. In: ACL 2002. pp. 303–310. Association for Computational Linguistics (2002)