Survey: Weighted Extended Top-down Tree Transducers Part I — Basics and Expressive Power

Andreas Maletti[*]

Abstract

Weighted extended top-down tree transducers (transducteurs généralisés descendants [Arnold, Dauchet: *Bi-transductions de forêts.* ICALP'76. Edinburgh University Press. 1976]) received renewed interest in the field of Natural Language Processing, where they are used in syntax-based machine translation. This survey presents the foundations for a theoretical analysis of weighted extended top-down tree transducers. In particular, it discusses essentially complete semirings, which are a novel concept that can be used to lift incomparability results from the unweighted case to the weighted case even in the presence of infinite sums. In addition, several equivalent ways to define weighted extended top-down tree transducers are presented and the individual benefits of each presentation is shown on a small result.

Keywords: tree transducer, weighted tree transducer, expressive power

1 Introduction

Tree automata theory [24, 25] and computational linguistics [42, 30] were tightly intertwined at their inception. In particular, top-down tree transducers were devised by Thatcher [47] and Rounds [44] for applications in natural language processing (NLP). However, this tight connection was lost early on and the two fields went separate ways. NLP research focussed on the algorithmic and scaling issues, whereas the tree automata theory research focussed on refining and extending models of automata and transducers. In particular, the following devices were investigated:

- bottom-up tree transducers [48] and attributed tree transducers [18],
- macro tree transducers [7, 15] and modular tree transducers [16],

[*] Universitat Rovira i Virgili, Departament de Filologies Romàniques, Avinguda Catalunya 35, 43002 Tarragona, Spain. E-mail: `andreas.maletti@urv.cat`. The author now is at: Universität Stuttgart, Institut for Natural Language Processing, Azenbergstraße 12, 70174 Stuttgart, Germany

- tree bimorphisms [3], and various models with synchronization (e.g., [43]).

Due to the technical difficulties and algorithmic and scaling complexities encountered with tree automata, computational linguists had reverted to finite-state string transducers [29], which are simple to understand, easy to train even on large amounts of data, and have nice theoretical properties. However, finite-state string transducers are not expressive enough for many applications in natural language processing [32]. This realization recently sparked a revival of tree automata in NLP research.

Shieber [46] and others have argued that the classical top-down tree transducers [47, 44] are generally inadequate for linguistic tasks without the use of copying and deletion. In general, copying causes many operations to become intractable or impossible, which severely limits even the use of copying top-down tree transducers.

A promising alternative is the *extended top-down tree transducer*, which was originally conceived by [44] and has been further pursued by [8, 2, 27, 33, 41]. In this survey we provide an in-depth review of some of the results for weighted extended top-down tree transducers. In fact, we assume that the reader has some fundamental understanding of unweighted tree automata theory. The goal is to present the common definitions and provide an overview of the various techniques used in the weighted setting. In this aspect this survey differs significantly from [41], which provides a detailed explanation of unweighted extended top-down tree transducers, their expressive power, and their essential features. In addition, [23] surveys results on weighted top-down tree transducers. Contrary to traditional research papers, we do not aim for the most general results here, but rather try to keep the material accessible.

In this survey, we introduce the theoretical foundations and showcase a general method that allows to lift inequalities from the unweighted case to the weighted setting. This is prepared in the first section with a detailed review of semirings and some required properties. Then we introduce the basic notions and notations for handling trees before we finally introduce the main model, the weighted extended top-down tree transducer, in Section 3. Our reference semantics is based on rewriting, but we provide an alternative semantics that corresponds to the initial-algebra semantics [23] of weighted top-down tree transducers. In addition, we relate the linear and nondeleting version of weighted extended top-down tree transducers to linear and complete bimorphisms [3]. The benefits of those additional representations and semantics are illustrated on typical constructions in Section 4. Moreover, in Section 4 we demonstrate a general method to lift known results from the unweighted setting to the weighted setting. We use a part of the Hasse diagram of [41] for classes of tree transformations computed by unweighted extended top-down tree transducers and show how to translate it to the weighted setting using the semiring basics introduced in Section 2.

Overall, we provide proof details whenever instructive, but it is generally safe to skip them. However, the main purpose of this survey is to showcase the techniques, so the proofs generally contain important information. Moreover, we provide examples for some interesting concepts and constructions. We refer to the original research papers for the details and the most general forms of the statements reproduced or reported here. In addition, we refer to [23] for a survey of weighted top-down tree transducers and to the forth-coming paper [21], which discusses an even more general weighted model and contains some of the results reported here.

2 The Basics

In this section, we first recall the definition and cover some basic properties of our weight structures: semirings [28, 26]. Recently, more general weight structures such as bi-monoids have been proposed [10] for weighted automata and transducers on strings and trees, but we will focus exclusively on semirings in this survey. To keep the presentation self-contained, we try to present proof details whenever possible.

2.1 Semirings

Let $\cdot \colon A^2 \to A$ be a binary operation on a set A, which we write using juxtaposition (i.e., ab stands for $a \cdot b$). It is *associative* if $(ab)c = a(bc)$ for every $a, b, c \in A$. Moreover, the operation $\cdot$ is *commutative* if $ab = ba$ for every $a, b \in A$. An element $1 \in A$ is a *neutral element* if $1a = a = a1$ for every $a \in A$. Finally, an element $0 \in A$ is *absorbing* if $a0 = 0 = 0a$ for every $a \in A$. In general, operations that use "multiplicative" operation signs (like $\cdot$, $\times$, $\otimes$, $\prod$, etc.) have precedence over operations written using "additive" signs (like $+$, $\oplus$, $\sum$, etc.). Thus, $ab + c$ stands for $(a \cdot b) + c$ and $\sum_{i \in I} a_i b_i$ stands for $\sum_{i \in I} (a_i \cdot b_i)$. As with $\cdot$ we often drop multiplicative symbols altogether and write a product $a \otimes b$ simply as the juxtaposition ab. As usual, the product $a \cdots a$ containing $n \in \mathbb{N}$ factors $a \in A$ is abbreviated by a^n.

A *semiring* [28, 26] is an algebraic structure $(A, +, \cdot, 0, 1)$ with two binary operations $+, \cdot \colon A^2 \to A$ and two constants $0, 1 \in A$ such that

- $+$ and $\cdot$ are associative operations, of which $+$ is also commutative,
- $\cdot$ distributes over $+$ from both sides, which means that $(a + b)c = ac + bc$ and $a(b + c) = ab + ac$ for every $a, b, c \in A$,
- 0 and 1 are the neutral elements for $+$ and $\cdot$, respectively, and
- 0 is absorbing for $\cdot$.

In other words, the semiring $(A, +, \cdot, 0, 1)$ consists of the commutative (additive) monoid $(A, +, 0)$ and the (multiplicative) monoid $(A, \cdot, 1)$ such that $\cdot$ distributes (both-sided) over finite sums $\sum_{i=1}^{k} a_i$. The absorption property

$$a \cdot 0 = a \cdot \sum_{i=1}^{0} a_i = \sum_{i=1}^{0} a a_i = 0$$

corresponds to distributivity over the empty sum $\sum_{i=1}^{0} a_i = 0$. If $\cdot$ is also commutative, then the semiring is *commutative*. It is a *ring* if there exists an element $-1 \in A$ such that $(-1) + 1 = 0$. By distributivity this yields that in a ring every element $a \in A$ has an *additive inverse* $-a \in A$. A ring is a *field* if for every $a \in A$ there exists a *multiplicative inverse* $a^{-1} \in A$ such that $aa^{-1} = 1 = a^{-1}a$. Finally, given two semirings $(A, +, \cdot, 0, 1)$ and $(S, \oplus, \odot, \mathbf{0}, \mathbf{1})$ and a mapping $h\colon A \to S$, the mapping h is a *semiring homomorphism* if $h(0) = \mathbf{0}$, $h(1) = \mathbf{1}$, $h(a + b) = h(a) \oplus h(b)$, and $h(a \cdot b) = h(a) \odot h(b)$ for every $a, b \in A$. Consequently, a semiring homomorphism is compatible with finite sums and products. For every mapping $f\colon B \to A$, we let $\operatorname{supp}(f) = \{b \in B \mid f(b) \neq 0\}$.

Commonly used semirings include

- the commutative BOOLEAN semiring $(\{0, 1\}, \max, \min, 0, 1)$ where 0 can be understood as *false* and 1 as *true*,
- the commutative semiring of natural numbers $(\mathbb{N}, +, \cdot, 0, 1)$ with the usual addition and multiplication,
- the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$, which is also commutative,
- the commutative field $(\mathbb{R}, +, \cdot, 0, 1)$ of real numbers, and
- the (typically non-commutative) semiring $(A^{Q \times Q}, \underline{+}, \underline{\cdot}, \underline{0}, \underline{1})$ of $Q \times Q$-matrices over a semiring $(A, +, \cdot, 0, 1)$ for every finite set Q where $\underline{+}$ and $\underline{\cdot}$ are the usual operations of matrix addition and multiplication, respectively, $\underline{0}$ is the zero matrix, and $\underline{1}$ is the unit matrix.

A semiring $(S, \oplus, \odot, \mathbf{0}, \mathbf{1})$ is a *subsemiring* of a semiring $(A, +, \cdot, 0, 1)$ if $S \subseteq A$ and the identity mapping $\mathrm{id}\colon S \to A$ such that $\mathrm{id}(s) = s$ for every $s \in S$ is a semiring homomorphism. In other words, $\mathbf{0} = 0$, $\mathbf{1} = 1$, $a \oplus b = a + b$, and $a \odot b = ab$ for every $a, b \in S$, which states that the subsemiring shares the constants 0 and 1 and uses the same operations restricted to its (smaller) carrier set. Given a set $S \subseteq A$, we denote by $\langle S \rangle$ the carrier set of the subsemiring *generated by* S; i.e., the carrier set of the smallest subsemiring of $(A, +, \cdot, 0, 1)$ that contains S. The subsemiring $(S, \oplus, \odot, 0, 1)$ is *finitely generated* if there exists a finite set $C \subseteq A$ such that $S = \langle C \rangle$. Due to the distributivity law, every element $a \in \langle S \rangle$ with $S \subseteq A$ can be presented as $\sum_{i=1}^{k} \bigl(\prod_{j=1}^{n_i} a_{ij}\bigr)$ with $k, n_1, \dots, n_k \in \mathbb{N}$ and $a_{ij} \in S$ for every $1 \leq i \leq k$ and $1 \leq j \leq n_i$. This representation as a sum of products will be important later on.

Next, we discuss infinite sums in a semiring $(A, +, \cdot, 0, 1)$. Since we will only encounter sums with countably many summands, we only define *countably complete semirings* [11, 36, 28, 26, 31]. Recall that a set I is *countable* if $|I| \leq \aleph_0$ where $\aleph_0 = |\mathbb{N}|$ (i.e., if it has at most as many elements as the natural numbers). Moreover, a *partition* of a set I is a set $\mathcal{J}$ of nonempty, pairwise disjoint subsets of I such that their union is I. Formally, $\mathcal{J}$ is a partition of I if (i) $\emptyset \notin \mathcal{J}$, (ii) $J \cap J' = \emptyset$ for every $J, J' \in \mathcal{J}$ with $J \neq J'$, and (iii) $I = \bigcup_{J \in \mathcal{J}} J$. Note that any partition of a countable set is itself countable. An infinitary sum operation $\sum$ is a family $(\sum_I)_I$ of mappings $\sum_I\colon A^I \to A$ for every countable index set I. We generally write $\sum_{i \in I} f(i)$ instead of $\sum_I f$. The semiring $(A, +, \cdot, 0, 1)$ together with the infinitary sum operation $\sum$ is *countably complete* if

(B) $\sum_{i \in \{j, j'\}} a_i = a_j + a_{j'}$ for all $j \neq j'$ and $a_j, a_{j'} \in A$,

(P) $\sum_{i \in I} a_i = \sum_{J \in \mathcal{J}} \bigl(\sum_{i \in J} a_i\bigr)$ for every countable index set I, partition $\mathcal{J}$ of I, and element $a_i \in A$ with $i \in I$, and

(D) $a\bigl(\sum_{i \in I} a_i\bigr) = \sum_{i \in I} a a_i$ and $\bigl(\sum_{i \in I} a_i\bigr)a = \sum_{i \in I} a_i a$ for every $a \in A$, countable index set I, and $a_i \in A$ with $i \in I$.

The axioms (B), (P), and (D) are also called binary sum, partition, and distributivity axiom, respectively, and they guarantee that the usual laws of associativity, commutativity and distributivity also hold for sums of countably many summands. The literature often also lists the following two additional axioms (E) $\sum_{i\in\emptyset} a_i = 0$ and (U) $\sum_{i\in\{j\}} a_i = a_j$. Next we show that they are valid in any countably complete semiring, and we will use them freely after the proof. In addition, we present another interesting known property [28, 26] of such semirings.

Proposition 1. *If $(A, +, \cdot, 0, 1)$ is countably complete with respect to $\sum$, then*

- $\sum_{i\in I} 0 = 0$ *for every countable index set I,*
- $\sum_{i\in\{j\}} a_i = a_j$ *for every j and $a_j \in A$, and*
- $a + b = 0$ *implies $a = 0 = b$ for every $a, b \in A$.*

Proof. To illustrate the handling of infinite sums, we prove these statements formally. For the third item, we present the proof of [26, Proposition 22.28]. First, the distributivity axiom immediately yields

$$\sum_{i\in I} 0 = \sum_{i\in I} 0a_i \overset{\text{(D)}}{=} 0 \cdot \sum_{i\in I} a_i = 0$$

for every countable index set I and $a_i \in A$ with $i \in I$. This proves the first item and Axiom (E), which is a special case of the first item. Next, let j be an arbitrary element and $a_j \in A$. We prove the unary sum axiom (U) by adding the neutral element 0 to a. In this way, we can form binary sums. Let j' be such that $j' \neq j$ and $a_{j'} = 0$. Then

$$\begin{aligned}\sum_{i\in\{j\}} a_i &= \Big(\sum_{i\in\{j\}} a_i\Big) + 0 \overset{\dagger}{=} \Big(\sum_{i\in\{j\}} a_i\Big) + \Big(\sum_{i\in\{j'\}} a_i\Big) \overset{\text{(B)}}{=} \sum_{J\in\{\{j\},\{j'\}\}} \Big(\sum_{i\in J} a_i\Big) \\ &\overset{\text{(P)}}{=} \sum_{i\in\{j,j'\}} a_i \overset{\text{(B)}}{=} a_j + a_{j'} = a_j + 0 = a_j\end{aligned}$$

where we used the first item in the step marked †. For the last statement, let $a, b \in A$ be such that $a + b = 0$. Consider the following two partitions of $\mathbb{N}$:

$$\begin{aligned}\mathcal{J} &= \{\{2j, 2j+1\} \mid j \in \mathbb{N}\} \\ \mathcal{J}' &= \{\{0\}\} \cup \{\{2j+1, 2j+2\} \mid j \in \mathbb{N}\} \ .\end{aligned}$$

Moreover, for every $i \in \mathbb{N}$, let $a_i = a$ if i is even and $a_i = b$ otherwise. Intuitively, consider the sum $\sum_{i\in\mathbb{N}} a_i$. The partition $\mathcal{J}$ always pairs a_{2j} and a_{2j+1}, which are a and b, respectively. Thus, all subsums under this partition will be $a + b = 0$. The partition $\mathcal{J}'$ is similar. It pairs a_{2j+1} and a_{2j+2}, which are b and a, respectively, but it keeps $a_0 = a$ separate. Thus, the subsum for $\{0\}$ will be a and the remaining subsums will be $b + a = 0$. Using the first item, we can then prove that $\sum_{i\in\mathbb{N}} a_i = 0$ using the partition $\mathcal{J}$ because all the subsums are 0, but we can also prove that $\sum_{i\in\mathbb{N}} a_i = a$ using the partition $\mathcal{J}'$. Clearly, this proves that $a = 0$. Formally, let

$\mathcal{J}'' = \mathcal{J}' \setminus \{\{0\}\}$. Then

$$\begin{aligned} a = a + 0 &\overset{(\mathrm{U}),\dagger}{=} \Big(\sum_{i \in \{0\}} a_i\Big) + \sum_{J \in \mathcal{J}''} 0 = \Big(\sum_{i \in \{0\}} a_i\Big) + \sum_{J \in \mathcal{J}''} (b + a) \\ &\overset{(\mathrm{B})}{=} \Big(\sum_{i \in \{0\}} a_i\Big) + \sum_{J \in \mathcal{J}''} \Big(\sum_{i \in J} a_i\Big) \overset{(\mathrm{P})}{=} \Big(\sum_{i \in \{0\}} a_i\Big) + \sum_{i \in \mathbb{N}_+} a_i \overset{(\mathrm{B})}{=} \sum_{J \in \{\{0\}, \mathbb{N}_+\}} \Big(\sum_{i \in J} a_i\Big) \\ &\overset{(\mathrm{P})}{=} \sum_{i \in \mathbb{N}} a_i \overset{(\mathrm{P})}{=} \sum_{J \in \mathcal{J}} \Big(\sum_{i \in J} a_i\Big) \overset{(\mathrm{B})}{=} \sum_{J \in \mathcal{J}} (a + b) = \sum_{J \in \mathcal{J}} 0 \overset{\dagger}{=} 0 \enspace , \end{aligned}$$

where we used the first item in the steps marked †. Clearly, this also yields that $b = 0 + b = a + b = 0$. □

Now let Q be a finite set and $(A, +, \cdot, 0, 1)$ be a countably complete semiring with respect to $\sum$. Then the matrix semiring $(A^{Q \times Q}, \underline{+}, \underline{\cdot}, \underline{0}, \underline{1})$ is a (typically non-commutative) semiring that is countably complete with respect to the usual generalization of matrix addition to countable sums. Consequently, we can define a $Q \times Q$-matrix M^* for every $Q \times Q$-matrix $M \in A^{Q \times Q}$ over A by $M^* = \sum_{n \in \mathbb{N}} M^n$. Recall that $M^0 = \underline{1}$ and $M^{n+1} = M \,\underline{\cdot}\, M^n$ for every $n \in \mathbb{N}$. If we interpret M as the incidence matrix of a weighted graph, then the entry $M^*_{p,q}$ with $p, q \in Q$ equals the sum of the weights of all paths leading from p to q where the weight of a path is obtained by multiplying the weights of the edges. For example, in the tropical semiring, $M^*_{p,q}$ equals the smallest weight of path from p to q.

Proposition 2. *Every ring $(A, +, \cdot, 0, 1)$ with $0 \neq 1$ is not countably complete with respect to any $\sum$.*

Proof. Suppose that it is countably complete with respect to some $\sum$. Since there exists an element $-1 \in A$ such that $(-1) + 1 = 0$, we conclude by Proposition 1 that $-1 = 0 = 1$, which contradicts the assumption $0 \neq 1$. □

Finally, we consider semiring homomorphisms that preserve certain countably infinite sums. Let $(A, +, \cdot, 0, 1)$ be a countably complete semiring with respect to $\sum$ and $(S, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be a countably complete semiring with respect to $\bigoplus$. In addition, let $h \colon A \to S$ be a semiring homomorphism and $B \subseteq A$. Then h is *B-complete* if $h(\sum_{i \in I} a_i) = \bigoplus_{i \in I} h(a_i)$ for every countable index set I and every $a_i \in B$ with $i \in I$. A semiring homomorphism is *essentially complete* [21] if it is $\langle B \rangle$-complete for every finite set $B \subseteq A$. The traditional notion of a *complete* semiring homomorphism [34, 17] requires that it is A-complete. In other words, an essentially complete homomorphism preserves countable sums, of which the summands all belong to a finitely-generated subsemiring, whereas the traditional notion requires that all countable sums need to be preserved. The relaxed notion of 'essential completeness' is typically sufficient for weighted finite-state devices because their finitely many transitions carry only finitely many weights. Thus, most weighted finite-state devices [36, 9] (whether over trees, strings, pictures, etc.) compute in a finitely-generated subsemiring.

We say that a semiring is *proper* if it is not a ring. For example, the BOOLEAN semiring is proper. WANG proved in [49, Theorem 2.1] and [50, Lemma 3.1] that for every proper commutative semiring there exists a semiring homomorphism from it to the BOOLEAN semiring. This important result essentially yields that weighted devices over proper commutative semirings behave as the corresponding unweighted (i.e., BOOLEAN weighted) devices. Here, we extend this result to include infinite summation, which is present in some finite-state models [9]. However, we first recall the original construction of the semiring homomorphism by [49, 50]. To this end, we introduce some additional notions for a commutative semiring $(A, +, \cdot, 0, 1)$. A set $C \subseteq A$ is a *co-ideal* if

- $cc' \in C$ for all $c, c' \in C$ and
- $a + c \in C$ for every $a \in A$ and $c \in C$.

In other words, co-ideals are closed under multiplication of its elements and closed under addition of one of its elements with any semiring element. Dually, an *ideal* $I \subseteq A$ is such that

- $ai \in I$ for every $a \in A$ and $i \in I$ and
- $i + i' \in I$ for all $i, i' \in I$.

Note that if $0 \in C$ for a co-ideal C, then $C = A$. Now consider the smallest co-ideal $C(\{1\})$ that contains 1. An easy exercise (using distributivity) shows that $C(\{1\}) = \{1 + a \mid a \in A\}$. More generally, for every $S \subseteq A$, the smallest co-ideal containing S is

$$C(S) = \{s_1 \cdots s_k + a \mid k \in \mathbb{N}, s_1, \ldots, s_k \in S, a \in A\} \enspace .$$

If $(A, +, \cdot, 0, 1)$ is a ring, then $0 \in C(\{1\})$, and thus $C(\{1\}) = A$. However, if it is proper, then clearly $0 \notin C(\{1\})$, and thus $C(\{1\}) \neq A$. Now ZORN's Lemma guarantees that in the latter case there exists a maximal co-ideal C such that $C(\{1\}) \subseteq C \subseteq A \setminus \{0\}$. The remaining elements $A \setminus C$ form an ideal that contains 0. This is verified as follows. Let $a \in A$ and $i, i' \in A \setminus C$. Since $i, i' \notin C$ there exist $c, c' \in C$, $n, n' \in \mathbb{N}$, and $b, b' \in A$ such that $i^n c + b = 0$ and $(i')^{n'} c' + b' = 0$ by maximality of C. Indeed, if such elements do not exist, then i or i' can be added to C to induce an even larger, proper co-ideal $C(C \cup \{i\})$ or $C(C \cup \{i'\})$. We show that $ai \notin C$ and $i + i' \notin C$, which proves that $A \setminus C$ is an ideal. Since $(ai)^n c + a^n b = a^n(i^n c + b) = 0$, we have $(ai)^n \notin C$ because $c \in C$. In fact, if $(ai)^n \in C$, then also $(ai)^n c \in C$ and $(ai)^n c + a^n b \in C$, which contradicts $0 \notin C$ because $(ai)^n c + a^n b = 0$. Consequently, $ai \notin C$ because if $ai \in C$, then $(ai)^n \in C$ for every $n \in \mathbb{N}$. Similarly, $(i + (i'))^m cc'$ where $m = n + n'$ can be presented as a sum of elements of the form $a_j = i^j (i')^{m-j} cc'$ where $j \in [m]$. Mind that the same summand can occur multiple times in the sum. Let $j \geq n$. Then $i^j (i')^{m-j} cc' + b i^{j-n} (i')^{m-j} c' = (i^n c + b) i^{j-n} (i')^{m-j} c' = 0$. Let $b_j = b i^{j-n} (i')^{m-j} c'$. An analogous argument applies to the case $j < n$, which yields that $m - j \geq n'$. Thus, for every summand a_j there exists an element $b_j \in A$ such that $a_j + b_j = 0$, which proves that also for $(i + (i'))^m cc'$, which is finite sum of summands a_j, there exists an element $b'' \in A$ such that $(i + (i'))^m cc' + b'' = 0$, which proves that $i + i' \notin C$.

Now we can define the semiring homomorphism h by

$$h(a) = \begin{cases} 1 & \text{if } a \in C \\ 0 & \text{otherwise} \end{cases}$$

for every $a \in A$. Since C is a co-ideal and $A \setminus C$ is an ideal, this mapping h is a semiring homomorphism.

Theorem 1. *For every countably complete semiring* $(A, +, \cdot, 0, 1)$ *with respect to* $\sum$ *there exists an essentially complete semiring homomorphism to the* Boolean *semiring, which is countably complete with respect to* max.

Proof. We start with the semiring homomorphism h that we constructed above. Recall that $C \subseteq A \setminus \{0\}$ is maximal co-ideal with $1 \in C$. It remains to prove that h is essentially complete. Let I be an index set, $S \subseteq A$ be finite, and $a_i \in \langle S \rangle$ for every $i \in I$. We have to prove that

$$h\Bigl(\sum_{i \in I} a_i\Bigr) = \max_{i \in I} h(a_i) \enspace , \tag{1}$$

which yields two (mutually exclusive) cases:

- there exists $i \in I$ such that $h(a_i) = 1$ (i.e., $a_i \in C$) or
- $h(a_i) = 0$ (i.e., $a_i \in I$) for all $i \in I$.

In the former case, the right-hand side of (1) evaluates to 1. This yields that we have to prove that $\sum_{i \in I} a_i \in C$. Let $j \in I$ be such that $a_j \in C$. Then $\sum_{i \in I} a_i = a_j + \sum_{i \in I \setminus \{j\}} a_i$ using the axioms (P), (B), and (U). Since $a_j \in C$, also $a_j + \sum_{i \in I \setminus \{j\}} a_i \in C$ because C is a co-ideal, which proves that $\sum_{i \in I} a_i \in C$.

In the second case, the right-hand side of (1) evaluates to 0. Thus, we need to prove that $\sum_{i \in I} a_i \notin C$. Since each a_i is in $\langle S \rangle$, we can represent it as a (finite) sum $\sum_{j=1}^{n_i} b_{ij}$ of products b_{ij} of elements of S as already remarked. Clearly, $b_{ij} \notin C$ for every $1 \leq j \leq n_i$ because otherwise $a_i \in C$. Consequently, we can write

$$\sum_{i \in I} a_i = \sum_{i \in I} \Bigl(\sum_{j=1}^{n_i} b_{ij}\Bigr) = \sum_{i \in I'} b_i a'_i$$

for some index set I', and $b_i \in S \setminus C$ and $a'_i \in A$ for every $i \in I'$. The existence of the factors $b_i \notin C$ follows from the fact that $a_i \notin C$. The set $B = \{b_i \mid i \in I'\}$ is finite because $B \subseteq S$. Let $B = \{e_1, \ldots, e_n\}$. Since C is maximal, we know that for each $b \in B$ there exist $\ell_b \in \mathbb{N}$, $c_b \in C$, and $d_b \in A$ such that $b^{\ell_b} c_b + d_b = 0$. Otherwise, the co-ideal $C(C \cup \{b\})$ would still be different from A, which contradicts the maximality of C. By Proposition 1, we know that $b^{\ell_b} c_b = 0$ (and $d_b = 0$). Let $\ell = \sum_{b \in B} \ell_b$ and $c = \prod_{b \in B} c_b$. Then

$$\Bigl(\sum_{i \in I} a_i\Bigr)^{\ell} c = \Bigl(\sum_{i \in I'} b_i a'_i\Bigr)^{\ell} \cdot \prod_{b \in B} c_b = \sum_{i \in I''} e_1^{\ell_{i1}} \cdots e_n^{\ell_{in}} a''_i c_{e_1} \cdots c_{e_n}$$

for some index set I'', $\ell_{i1}, \dots, \ell_{in} \in \mathbb{N}$ and $a''_i \in A$ for every $i \in I''$ such that $\sum_{j=1}^{n} \ell_{ij} = \ell$. From the last expression it is clear that each summand contains a factor $b^m c_b = 0$ for some $b \in B$ and $m \geq \ell_b$. Thus, each summand is 0 and the sum is 0 by Proposition 1, which proves that $(\sum_{i \in I} a_i)^\ell c = 0$. However, since $c \in C$, we proved that $\sum_{i \in I} a_i$ cannot be in C because it would yield $0 \in C$, which is a contradiction. Consequently, $\sum_{i \in I} a_i \notin C$, which proves the statement.

Consequently, we proved that h is $\langle S \rangle$-complete, and since S was chosen arbitrarily, we also proved that h is essentially complete. □

For the rest of this paper, let $(A, +, \cdot, 0, 1)$ be an arbitrary nontrivial (i.e., $0 \neq 1$) commutative semiring.

2.2 Sets, relations, and trees

We denote the set of all nonnegative integers (including 0) by $\mathbb{N}$. For every $n \in \mathbb{N}$, the subset $\{i \in \mathbb{N} \mid 1 \leq i \leq n\}$ is denoted by $[n]$. We fix the set $\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, \dots\}$ of (formal) variables and let $\mathrm{X}_n = \{\mathrm{x}_i \mid i \in [n]\}$ for every $n \in \mathbb{N}$.

Now, let S, T, and U be countable sets. A *relation from S to T* is a subset of $S \times T$. Let $R \subseteq S \times T$ and $R' \subseteq T \times U$. The *inverse relation of* R, denoted by R^{-1}, is $\{(t, s) \mid (s, t) \in R\}$ and the *composition of R and R'*, denoted by $R ; R'$, is $\{(s, u) \mid \exists t \in T \colon (s, t) \in R, (t, u) \in R'\}$. These notions extend to classes of relations in the standard manner. A *relation on* S is a subset of $S \times S$. For every set $L \subseteq S$ we denote by id_L the relation $\{(s, s) \mid s \in L\}$. The reflexive and transitive closure of a relation $R \subseteq S \times S$ is denoted by R^*.

Next, we extend these notions to the weighted setting. A *weighted relation from S to T* is a mapping of $\rho \colon S \times T \to A$. Let $\rho \colon S \times T \to A$ and $\rho' \colon T \times U \to A$. The *inverse relation of* ρ, denoted by ρ^{-1}, is such that $\rho^{-1}(t, s) = \rho(s, t)$ for every $s \in S$ and $t \in T$, and the *composition of ρ and ρ'*, denoted by $\rho \mathbin{;} \rho'$, is such that $(\rho \mathbin{;} \rho')(s, u) = \sum_{t \in T} \rho(s, t) \rho'(t, u)$ for every $s \in S$ and $u \in U$. Depending on the set T and the weighted relations ρ and ρ', the sum in the definition of $\rho ; \rho'$ might be infinite. If it is, then we typically assume that $(A, +, \cdot, 0, 1)$ is countably complete with respect to $\sum$. We will discuss this issue in more detail later on. Again, these notions extend to classes of weighted relations in the standard manner. A *weighted relation on* S is a mapping $\rho \colon S \times S \to A$. For every weighted set $\varphi \colon S \to A$, we denote by id_φ the weighted relation such that $\mathrm{id}_\varphi(s, s) = \varphi(s)$ and $\mathrm{id}_\varphi(s, s') = 0$ for every $s, s' \in S$ such that $s \neq s'$. If $\varphi(s) = 1$ for every $s \in S$, then we also just write id instead of id_φ. Given that $(A, +, \cdot, 0, 1)$ is countably complete with respect to $\sum$, the reflexive and transitive closure of a weighted relation $\rho \colon S \times S \to A$ is denoted by ρ^* and is defined by $\rho^*(s, s') = \sum_{n \in \mathbb{N}} \rho^n(s, s')$ where $\rho^0 = \mathrm{id}$ and $\rho^{k+1} = \rho \mathbin{;} \rho^k$ for every $k \in \mathbb{N}$. Note that all weighted notions over the Boolean semiring correspond to the unweighted notions via the mapping 'supp'. For example, the weighted relation $\rho \colon S \times T \to \{0, 1\}$ corresponds to the (unweighted) relation $\mathrm{supp}(\rho)$.

The set of all finite sequences (words) over S is denoted by S^*, of which ε denotes the empty sequence (the empty word). The concatenation of the words $v, w \in S^*$

is denoted by $v.w$ or simply by vw. The length of a word $w \in S^*$ (i.e., the number of occurrences of elements of S in w) is denoted by $|w|$.

An *alphabet* Σ is a nonempty and finite set, of which the elements are called *symbols*. Next, we define trees using only alphabets. In contrast to many definitions in the literature [12, 24, 25, 23], we do not assume a ranked alphabet, which means that a symbol can have different numbers of children in a tree. This is only a simplification because our automata model will still have only a finite number of rules, which thus determine a maximal rank for each used symbol. Let Q be an alphabet and L a countable set of leaf labels. For every set T, we let

$$Q(T) = \{q(t) \mid q \in Q, t \in T\} \enspace .$$

The set $T_\Sigma(L)$ of *Σ-trees with leaf labels L* is the smallest set T such that $L \subseteq T$ and $\sigma(t_1, \dots, t_k) \in T$ for every $k \in \mathbb{N}$, $\sigma \in \Sigma$, and $t_1, \dots, t_k \in T$. We generally assume that $\Sigma \cap L = \emptyset$, and thus we write $\alpha()$ simply as α for every $\alpha \in \Sigma$. Given another alphabet Δ and $T \subseteq T_\Delta(L)$, we treat elements of $T_\Sigma(T)$ and $Q(T)$ as particular trees of $T_{Q \cup \Sigma \cup \Delta}(L)$. For every $\gamma \in \Sigma$, we abbreviate the tree $\gamma(\gamma(\cdots \gamma(t) \cdots))$ with n symbols γ on top of $t \in T_\Sigma(L)$ simply by $\gamma^n(t)$. Finally, we write T_Σ for $T_\Sigma(\emptyset)$. Note that $T_\Sigma(L)$ is countable and that the elements of L can only appear as leaves in trees of $T_\Sigma(L)$.

The set $\mathrm{pos}(t) \subseteq \mathbb{N}^*$ of *positions* of a tree $t \in T_\Sigma(L)$ is inductively defined by $\mathrm{pos}(\ell) = \{\varepsilon\}$ for every $\ell \in L$ and

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

for every $k \in \mathbb{N}$, $\sigma \in \Sigma$, and $t_1, \dots, t_k \in T_\Sigma(L)$. Let $t, t' \in T_\Sigma(L)$ and $w \in \mathrm{pos}(t)$. The label of t at position w is $t(w)$, and the subtree of t that is rooted at w is $t|_w$. We can define these notions inductively as follows: $\ell(\varepsilon) = \ell|_\varepsilon = \ell$ for every $\ell \in L$ and

$$t(w) = \begin{cases} \sigma & \text{if } w = \varepsilon \\ t_i(v) & \text{if } w = iv \text{ with } i \in [k] \end{cases} \qquad \text{and} \qquad t|_w = \begin{cases} t & \text{if } w = \varepsilon \\ t_i|_v & \text{if } w = iv \text{ with } i \in [k] \end{cases}$$

where $t = \sigma(t_1, \dots, t_k)$ for every $k \in \mathbb{N}$, $\sigma \in \Sigma$, and $t_1, \dots, t_k \in T_\Sigma(L)$. For every set of labels $S \subseteq \Sigma \cup L$, we let $\mathrm{pos}_S(t) = \{w \in \mathrm{pos}(t) \mid t(w) \in S\}$. For $S = \{s\}$ we abbreviate $\mathrm{pos}_S(t)$ simply by $\mathrm{pos}_s(t)$. We say that $s \in S$ *occurs* $|\mathrm{pos}_s(t)|$ times in t. Finally, $t[u]_w$ denotes the tree that is obtained from $t \in T_\Sigma(L)$ by replacing the subtree $t|_w$ at w by $u \in T_\Delta(L)$.

The *height* $\mathrm{ht}(t)$ of t is $\max\{|w| + 1 \mid w \in \mathrm{pos}(t)\}$, and the *size* $|t|$ of the tree t is $|t| = |\mathrm{pos}(t)|$. Recall the special set X of formal variables. We let $\mathrm{var}(t) = \{x \in \mathrm{X} \mid \mathrm{pos}_x(t) \neq \emptyset\}$ for every $t \in T_\Sigma(L \cup \mathrm{X})$. The tree t is *linear* (respectively, *nondeleting*) in $V \subseteq \mathrm{X}$, if every $x \in V$ occurs at most (respectively, at least) once in t. Every $t \in T_\Sigma(V)$ that is linear and nondeleting in V is a *V-context* of $T_\Sigma(V)$. The set of all V-contexts of $T_\Sigma(V)$ is denoted by $C_\Sigma(V)$. For every such context and $x \in \mathrm{var}(t)$, we identify the unique element of $\mathrm{pos}_x(t)$ with $\mathrm{pos}_x(t)$. If t is linear and nondeleting in X_k for some $k \in \mathbb{N}$ and the variables occur in order (i.e., $\mathrm{pos}_{x_i}(t) < \mathrm{pos}_{x_j}(t)$ in the usual lexicographic ordering

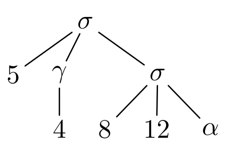

Figure 1: Graphical representation of the tree $\sigma(5, \gamma(4), \sigma(8, 12, \alpha))$.

for all $1 \leq i < j \leq k$), then t is called *normalized.* For every $V \subseteq \mathrm{X}$, a mapping $\theta\colon V \to T_\Sigma(L)$ is a *substitution.* The substitution θ can be applied to a tree $t \in T_\Sigma(L \cup \mathrm{X})$, written $t\theta$, which yields the tree that is obtained by replacing (in parallel) all occurrences of a variable $x \in V$ by $\theta(x)$. Formally, $x\theta = \theta(x)$ for every $x \in V$, and $\sigma(t_1, \ldots, t_k)\theta = \sigma(t_1\theta, \ldots, t_k\theta)$ for every $k \in \mathbb{N}$, $\sigma \in \Sigma$, and $t_1, \ldots, t_k \in T_\Sigma(L \cup \mathrm{X})$. To avoid explicitly defining a substitution $\theta\colon \mathrm{X}_k \to T_\Sigma(L)$, we sometimes write $t[\theta(\mathrm{x}_1), \ldots, \theta(\mathrm{x}_k)]$ for $t\theta$.

Example 1. Let $\Sigma = \{\sigma, \gamma, \alpha\}$. Then $t = \sigma(5, \gamma(4), \sigma(8, 12, \alpha))$ is a tree of $T_\Sigma(\mathbb{N})$. Its graphical representation is displayed in Figure 1. Its set $\mathrm{pos}(t)$ of positions is $\{\varepsilon, 1, 2, 2.1, 3, 3.1, 3.2, 3.3\}$ and $\mathrm{pos}_\sigma(t) = \{\varepsilon, 3\}$. The tree $\sigma(\mathrm{x}_1, \mathrm{x}_3, \mathrm{x}_2, \mathrm{x}_2)$ is linear and nondeleting in $\{\mathrm{x}_1, \mathrm{x}_3\}$, but not linear in X_3.

3 The Model

In this section, we will recall the main model of this survey: the weighted extended top-down tree transducer [2, 3, 38, 40]. However, we first recall the corresponding automaton model: the weighted tree automaton [6, 34, 17, 23]. A *weighted tree language* (or tree series) is simply a weighted set of trees; i.e., a mapping $\varphi\colon T_\Sigma \to A$ for some alphabet Σ.

Definition 1. *A* weighted (bottom-up) tree automaton *(wta) is a tuple* (Q, Σ, δ, F) *such that*

- Q *is an alphabet of* states*,*
- Σ *is an alphabet of* input symbols *such that* $Q \cap \Sigma = \emptyset$*,*
- $\delta\colon Q^* \times \Sigma \times Q \to A$ *is a* transition weight mapping *with finite* $\mathrm{supp}(\delta)$*, and*
- $F \subseteq Q$ *is a set of* final states.

The transition weight mapping δ of the wta $M = (Q, \Sigma, \delta, F)$ is extended to a mapping $\underline{\delta}\colon T_\Sigma \times Q \to A$ as follows:

$$\underline{\delta}(\sigma(t_1, \ldots, t_k), q) = \sum_{q_1, \ldots, q_k \in Q} \delta(q_1 \cdots q_k, \sigma, q) \cdot \prod_{i=1}^{k} \underline{\delta}(t_i, q_i)$$

for every $q \in Q$, $k \in \mathbb{N}$, $\sigma \in \Sigma$, and $t_1, \ldots, t_k \in T_\Sigma$. In the sequel, we will simply write δ instead of $\underline{\delta}$. The weighted tree language $\varphi_M\colon T_\Sigma \to A$ *recognized by* M is such that $\varphi_M(t) = \sum_{q \in F} \delta(t, q)$ for every $t \in T_\Sigma$. A weighted tree language $\varphi\colon T_\Sigma \to A$ is *recognizable* if there exists a wta M such that $\varphi_M = \varphi$.

The recognizable weighted tree languages are the natural generalization of the recognizable tree languages [24, 25]. For the BOOLEAN semiring the two notions coincide. An excellent introduction into the subject is presented in [23]. Here we just present a quick example.

Example 2. Let us consider the artic semiring $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$ and the wta (Q, Σ, δ, F) with

- $Q = \{z, h\}$ and $F = \{h\}$,
- $\Sigma = \{\sigma, \alpha\}$, and
- δ returns $-\infty$ except in the following cases:

$$\begin{array}{lll} \delta(\varepsilon, \alpha, z) = 0 & \delta(z, \alpha, z) = 0 & \delta(zz, \sigma, z) = 0 \\ \delta(\varepsilon, \alpha, h) = 1 & \delta(h, \alpha, h) = 1 & \delta(zh, \sigma, h) = 1 \\ & & \delta(hz, \sigma, h) = 1 \ . \end{array}$$

This wta rejects (i.e., assigns weight $-\infty$ to) all trees t that contain a symbol α with at least two children [i.e., trees t that have a position $w \in \mathrm{pos}_\alpha(t)$ such that $a.2$ is also in $\mathrm{pos}(t)$] or a symbol σ with anything but 2 children. Moreover, it assigns the height $\mathrm{ht}(t)$ to the remaining trees t.

Next, we recall the weighted extended top-down tree transducer [8, 2, 27, 33]. For simplicity, we will henceforth just call them "extended tree transducer" (xtt). We essentially follow the definitions of [38, 41], in which the corresponding unweighted device is discussed in detail. We will lift the results obtained in [38, 41] to the weighted setting either directly or using the semiring homomorphism introduced in Section 2. We start with the definition of the general device.

Definition 2. *A* (weighted) extended (top-down) tree transducer *(xtt) is a tuple* $M = (Q, \Sigma, \Delta, I, R)$ *where*

- Q *is an alphabet 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* $\mathrm{supp}(R)$ *is finite,* l *is linear in* X *and* $\mathrm{var}(r) \subseteq \mathrm{var}(l)$ *for every* $(l, r) \in \mathrm{supp}(R)$*.*

If for every $(l, r) \in \mathrm{supp}(R)$ *there exist* $k \in \mathbb{N}$*,* $q \in Q$*, and* $\sigma \in \Sigma$ *such that* $l = q(\sigma(\mathrm{x}_1, \dots, \mathrm{x}_k))$*, then* M *is a* top-down tree transducer *[35, 14].*

In the sequel, we often write $l \to r \in \mathrm{supp}(R)$ instead of $(l, r) \in \mathrm{supp}(R)$, and we write $l \xrightarrow{a} r \in R$ instead of $R(l, r) = a$. The xtt M is *linear* (respectively, *nondeleting* if r is linear (respectively, nondeleting) in $\mathrm{var}(l)$ for every $l \to r \in \mathrm{supp}(R)$. A rule of the form $q(x) \to r$ with $q \in Q$, $x \in \mathrm{X}$, and $r \in T_\Delta(Q(\mathrm{X}))$ is called *input ε-rule*, and any rule of the form $l \to q(x)$ with $q \in Q$, $x \in \mathrm{X}$, and $l \in Q(T_\Sigma(\mathrm{X}))$ is called *output ε-rule*. A rule that is both an input and an output ε-rule is called a *pure ε-rule*. The set of all pure ε-rules of $\mathrm{supp}(R)$ is denoted by R^ε. Any remaining rule contains at least one input or output symbol.

Example 3 (see [41, Section 3])**.** Let us consider the field $(\mathbb{R}, +, \cdot, 0, 1)$ of real numbers. The xtt $(Q, \Sigma, \Delta, \{q\}, R)$ with

$$\begin{aligned} Q &= \{q, q_{\mathrm{S}}, q_{\mathrm{V}}, q_{\mathrm{NP}}\} \\ \Sigma &= \Gamma \cup \{\mathit{saw}, \mathit{the}, \mathit{boy}, \mathit{door}\} \\ \Delta &= \Gamma \cup \{\mathit{ra'aa}, \mathit{atefl}, \mathit{albab}\} \\ \Gamma &= \{\mathrm{CONJ}, \mathrm{S}, \mathrm{VP}, \mathrm{V}, \mathrm{DT}, \mathrm{NP}, \mathrm{N}\} \end{aligned}$$

and the following weighted rules of R

$$q(\mathrm{x}_1) \xrightarrow{.2} q_{\mathrm{S}}(\mathrm{x}_1) \qquad (\rho_1)$$

$$q(\mathrm{x}_1) \xrightarrow{.8} \mathrm{S}(\mathrm{CONJ}(\mathit{wa}\text{-}), q_{\mathrm{S}}(\mathrm{x}_1)) \qquad (\rho_2)$$

$$q_{\mathrm{S}}(\mathrm{S}(\mathrm{x}_1, \mathrm{VP}(\mathrm{x}_2, \mathrm{x}_3))) \xrightarrow{1} \mathrm{S}'(q_{\mathrm{V}}(\mathrm{x}_2), q_{\mathrm{NP}}(\mathrm{x}_1), q_{\mathrm{NP}}(\mathrm{x}_3)) \qquad (\rho_3)$$

$$q_{\mathrm{V}}(\mathrm{V}(\mathit{saw})) \xrightarrow{.7} \mathrm{V}(\mathit{ra'aa}) \qquad (\rho_4)$$

$$q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\mathit{the}), \mathrm{N}(\mathit{boy}))) \xrightarrow{.6} \mathrm{NP}(\mathrm{N}(\mathit{atefl})) \qquad (\rho_5)$$

$$q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\mathit{the}), \mathrm{N}(\mathit{door}))) \xrightarrow{.5} \mathrm{NP}(\mathrm{N}(\mathit{albab}) \qquad (\rho_6)$$

is linear and nondeleting. In addition, it has 2 input ε-rules, 1 output ε-rule, and 1 pure ε-rule. It is not a top-down tree transducer.

Our reference semantics of the xtt $M = (Q, \Sigma, \Delta, I, R)$ is given by rewriting [13, 22, 41]. Let $\zeta, \xi \in T_\Delta(Q(T_\Sigma))$ and $\rho = (l \to r) \in \mathrm{supp}(R)$. The *leftmost redex* in ζ is the least position $w \in \mathrm{pos}_Q(\zeta)$ with respect to the lexicographic (total) ordering on $\mathbb{N}^*$. In other words, the leftmost redex is the leftmost position in the tree that is labeled with a state. We say that ζ *rewrites to* ξ *using* ρ, denoted by $\zeta \Rightarrow^{\rho}_{M} \xi$, if there exist a minimal position $w \in \mathrm{pos}_Q(\zeta)$ and a substitution $\theta \colon \mathrm{var}(l) \to T_\Sigma$ such that $\zeta|_w = l\theta$ and $\xi = \zeta[r\theta]_w$. Intuitively, we identify a rule, of which the left-hand side matches the subtree at the leftmost redex, and then we replace this subtree by the corresponding instantiated right-hand side. The *weighted relation* τ_M (or weighted tree transformation) *computed by* M is given by

$$\tau_M(t, u) = \sum_{\substack{q \in I, k \in \mathbb{N}, \rho_1, \ldots, \rho_k \in \mathrm{supp}(R) \\ q(t) \Rightarrow^{\rho_1}_{M} \cdots \Rightarrow^{\rho_k}_{M} u}} R(\rho_1) \cdots R(\rho_k) \qquad (2)$$

for every $t \in T_\Sigma$ and $u \in T_\Delta$. Two xtt M and M' are *equivalent* if $\tau_M = \tau_{M'}$. The sum in (2) can be infinite. Let us present a short example before we discuss the finiteness of the sum of (2) in some detail. The semantics shows that variables can be consistently renamed without effect. Consequently, for every xtt $M = (Q, \Sigma, \Delta, I, R)$ there exists an equivalent xtt $M' = (Q, \Sigma, \Delta, I, R')$ such that for every $(l, r) \in \mathrm{supp}(R')$ there exists $k \in \mathbb{N}$ with $\mathrm{var}(l) = \{\mathrm{x}_1, \ldots, \mathrm{x}_k\}$. In the following, we will silently assume this normal form.

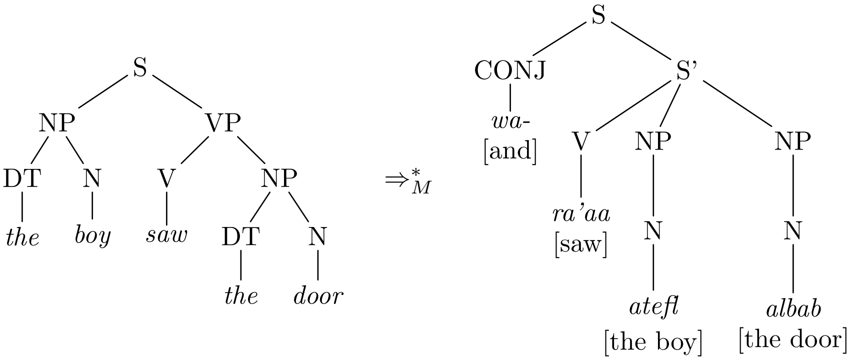

Figure 2: English-to-Arabic translation on syntax trees.

Example 4. Let us reconsider the xtt M of Example 3, and let

$$t = \mathrm{S}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{boy})), \mathrm{VP}(\mathrm{V}(\textit{saw}), \mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{door})))) \enspace ,$$

which is depicted in Figure 2. Then

$$\begin{aligned}
q(t) &\Rightarrow_M^{\rho_2} \mathrm{S}(\mathrm{CONJ}(\textit{wa-}), q_{\mathrm{S}}(t)) \\
&\Rightarrow_M^{\rho_3} \mathrm{S}(\mathrm{CONJ}(\textit{wa-}), \mathrm{S}'(q_{\mathrm{V}}(\mathrm{V}(\textit{saw})), q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{boy}))), \\
&\qquad\qquad q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{door}))))) \\
&\Rightarrow_M^{\rho_4} \mathrm{S}(\mathrm{CONJ}(\textit{wa-}), \mathrm{S}'(\mathrm{V}(\textit{ra'aa}), q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{boy}))), \\
&\qquad\qquad q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{door}))))) \\
&\Rightarrow_M^{\rho_5} \mathrm{S}(\mathrm{CONJ}(\textit{wa-}), \mathrm{S}'(\mathrm{V}(\textit{ra'aa}), \mathrm{NP}(\mathrm{N}(\textit{atefl})), q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{door}))))) \\
&\Rightarrow_M^{\rho_6} \mathrm{S}(\mathrm{CONJ}(\textit{wa-}), \mathrm{S}'(\mathrm{V}(\textit{ra'aa}), \mathrm{NP}(\mathrm{N}(\textit{atefl})), \mathrm{NP}(\mathrm{N}(\textit{albab})))) = u \enspace .
\end{aligned}$$

Since this is the only derivation from $q(t)$ to u, we conclude that

$$\tau_M(t, u) = R(\rho_2)R(\rho_3)R(\rho_4)R(\rho_5)R(\rho_6) = 0.8 \cdot 1 \cdot 0.7 \cdot 0.6 \cdot 0.5 = 0.168 \enspace .$$

For illustration, t and u are displayed in Figure 2.

Let us return to the sum in (2). Clearly, the length k of the derivation is at most $|t| + |u|$ if no pure ε-rule is used (i.e., if $\rho_i \notin R^\varepsilon$ for all $i \in [k]$). In this case, the sum in (2) is finite, which yields that the semantics of any top-down tree transducer is well-defined. In the presence of pure ε-rules, we assume that $(A, +, \cdot, 0, 1)$ is countably complete with respect to $\sum$. Consequently, the sum is well-defined. We will get rid of this case distinction after an important characterization result that relates linear nondeleting xtt to another device that computes weighted tree transformations: the weighted linear complete bimorphism.

A *linear and complete tree homomorphism* $f \colon T_\Gamma \to T_\Sigma$ is such that for every $k \in \mathbb{N}$ and $\gamma \in \Gamma$ there exists $f_k(\gamma) \in C_\Sigma(\mathrm{X}_k)$ such that

$$f(\gamma(s_1, \dots, s_k)) = f_k(\gamma)[f(s_1), \dots, f(s_k)] \enspace .$$

If additionally $f_k(\gamma) \neq \mathrm{x}_1$ for every $k \in \mathbb{N}$ and $\gamma \in \Gamma$, then f is *ε-free.*

Since wta are rather unsuitable for the next result, we use another model: the *weighted regular tree grammar* [1]. Such a grammar is tuple $G = (N, \Sigma, S, P)$ where N is a finite set of nonterminals, Σ is an alphabet of input symbols, $S \subseteq N$ is a set of start symbols, and $P\colon N \times T_\Sigma(N) \to A$ assigns weights to productions such that $\mathrm{supp}(P)$ is finite. It computes in a step-wise fashion. Given $\xi \in T_\Sigma(N)$, let $w \in \mathrm{pos}_N(\xi)$ be the leftmost position (i.e., the smallest of $\mathrm{pos}_N(\xi)$ with respect to the lexicographic ordering on $\mathbb{N}^*$). If there exists a production $\rho = (n, s) \in \mathrm{supp}(P)$ such that $n = \xi(w)$, then we write $\xi \Rightarrow_G^\rho \zeta$ where $\zeta = \xi[s]_w$. In other words, the nonterminal is replaced by the corresponding right-hand side of the production. The semantics of G is then given by

$$\varphi_G(t) = \sum_{\substack{s \in S, k \in \mathbb{N}, \rho_1, \dots, \rho_k \in \mathrm{supp}(P) \\ s \Rightarrow_G^{\rho_1} \cdots \Rightarrow_G^{\rho_k} t}} P(\rho_1) \cdots P(\rho_k)$$

for every $t \in T_\Sigma$. To avoid a discussion of infinite summations here, we assume that $s \notin N$ for every $(n, s) \in \mathrm{supp}(P)$, which guarantees that the above sum is finite. It is known that such weighted regular tree grammars also compute the recognizable weighted tree languages [1, Proposition 3.1] (also see [23] for a detailed account). The following important result is well-known from the literature [35, 17].

Theorem 2 (see [35] and [17, Corollary 6.10])**.** *For every linear, complete, and ε-free tree homomorphism $f\colon T_\Gamma \to T_\Sigma$ and every recognizable weighted tree language $\psi\colon T_\Gamma \to A$, the weighted tree language $\varphi\colon T_\Sigma \to A$, which is given by $\varphi(t) = \sum_{s \in T_\Gamma, f(s) = t} \psi(s)$, is again recognizable.*

Proof. Intuitively, we translate a symbol $\gamma \in \Gamma$ with the help of f into a context, which we then process in a single step charging the original weight. Now let us construct a weighted regular tree grammar for φ. Formally, let $M = (Q, \Gamma, \delta, F)$ be a wta recognizing ψ. The weighted regular tree grammar $G = (Q, \Sigma, F, P)$ is such that

$$P(q, s) = \sum_{\substack{k \in \mathbb{N}, q_1, \dots, q_k \in Q \\ s = f_k(\gamma)[q_1, \dots, q_k]}} \delta(q_1 \cdots q_k, \gamma, q)$$

for every $q \in Q$ and $s \in T_\Sigma(Q)$. Clearly, $s \notin Q$ for every $(q, s) \in \mathrm{supp}(P)$. Moreover, it should be clear that $\varphi_G = \varphi$, which proves that φ is recognizable. □

A *weighted linear and complete bimorphism* [3, 21] is a tuple $B = (f, \varphi, g)$ such that $f\colon T_\Gamma \to T_\Sigma$ and $g\colon T_\Gamma \to T_\Delta$ are linear and complete tree homomorphisms and $\varphi\colon T_\Gamma \to A$ is recognizable. The weighted tree transformation *computed by B* is

$$\tau_B(t, u) = \sum_{\substack{s \in T_\Gamma \\ f(s) = t, g(s) = u}} \varphi(s)$$

for every $t \in T_\Sigma$ and $u \in T_\Delta$.

Theorem 3 (see [38, Theorem 4] and [21])**.** *For every linear and nondeleting xtt there exists an equivalent weighted linear and complete bimorphism and vice versa.*

Proof. We have to prove both directions. Let $B = (f, \varphi, g)$ be a weighted linear and complete bimorphism with $f \colon T_\Gamma \to T_\Sigma$ and $g \colon T_\Gamma \to T_\Delta$. Moreover, let $N = (Q, \Gamma, \delta, F)$ be a wta such that $\varphi_N = \varphi$. Roughly speaking, we use the control structure of N as control structure of the xtt M that we construct and use f and g to determine the left- and right-hand sides of the rules, respectively. Formally, we construct the linear nondeleting xtt $M = (Q, \Sigma, \Delta, F, R)$ as follows. For every $l \in Q(T_\Sigma)$ and $r \in T_\Delta(Q(\mathrm{X}))$, let

$$R(l, r) = \sum_{\substack{(q_1 \cdots q_k, \gamma, q) \in \mathrm{supp}(\delta) \\ l = q(f_k(\gamma)), r = g_k(\gamma)[q_1(\mathrm{x}_1), \dots, q_k(\mathrm{x}_k)]}} \delta(q_1 \cdots q_k, \gamma, q) \enspace .$$

Note that M is linear and nondeleting.

For the converse, let a linear and nondeleting xtt $M = (Q, \Sigma, \Delta, I, R)$ be given. Without loss of generality, we suppose that for every $l \to r \in \mathrm{supp}(R)$ there exists $k \in \mathbb{N}$ with $\mathrm{var}(l) = \{\mathrm{x}_1, \dots, \mathrm{x}_k\}$. We construct f, g, and a wta $N = (Q, \Gamma, \delta, I)$ with $\Gamma = \mathrm{supp}(R)$ as follows. For every $\rho \in \mathrm{supp}(R)$, we have $\rho = q(l) \to r\theta$ with $l \in C_\Sigma(\mathrm{X}_k)$, $r \in C_\Delta(\mathrm{X}_k)$, and $q, q_1, \dots, q_k \in Q$ where θ is the substitution such that $\mathrm{x}_i\theta = q_i(\mathrm{x}_i)$ for every $i \in [k]$. For this rule ρ, let $f_k(\rho) = l$, $g_k(\rho) = r$, and $\delta(q_1 \cdots q_k, \rho, q) = R(\rho)$. The remaining values of $f_k(\gamma)$ and $g_k(\gamma)$ are irrelevant and all unmentioned values of δ are 0.

For both directions it remains to prove that $\tau_M = \tau_B$. To this end, it can be shown for every $q \in Q$, $t \in T_\Sigma$, and $u \in T_\Delta$ that

$$\sum_{\substack{n \in \mathbb{N}, \rho_1, \dots, \rho_n \in \mathrm{supp}(R) \\ q(t) \Rightarrow_M^{\rho_1} \cdots \Rightarrow_M^{\rho_n} u}} R(\rho_1) \cdots R(\rho_n) = \sum_{\substack{s \in T_\Gamma \\ f(s) = t, g(s) = u}} \delta(s, q) \enspace .$$

The proof of that statement is omitted here. The unweighted case is proved in [38, Theorem 4] and the weighted case is discussed in [20, 21]. □

It follows immediately from Theorem 3 that linear and nondeleting xtt are symmetric; i.e., for every linear and nondeleting xtt $M = (Q, \Sigma, \Delta, I, R)$ there exists a linear and nondeleting xtt M' such that $\tau_{M'}(u, t) = \tau_M(t, u)$ for every $t \in T_\Sigma$ and $u \in T_\Delta$. This property is not quite obvious from the definition of such xtt, but can trivially be observed on the bimorphism representation.

Now, we will eliminate pure ε-rules in the standard manner in order to avoid infinite sums, which only occur in the semantics of xtt with pure ε-rules. To this end, let $(A, +, \cdot, 0, 1)$ be countably complete with respect to $\sum$, and let

$$E_{p,q} = \sum_{\substack{l \to r \in R^\varepsilon \\ l(\varepsilon) = p, r(\varepsilon) = q}} R(l, r)$$

for every $p, q \in Q$. Using the matrix E^*, which is well-defined due to the countable completeness, we can construct the equivalent xtt $M' = (Q, \Sigma, \Delta, I, R')$ such that

$$\begin{aligned} R'(l, r) &= 0 && \text{for all } l, r \in Q(\mathrm{X}) \\ R'(p(\ell), r) &= \sum_{q \in Q} E^*_{p,q} R(q(\ell), r) && \text{for all } p, q \in Q,\ \ell \in T_\Sigma(\mathrm{X}), \text{ and } r \in T_\Delta(Q(\mathrm{X}))\ . \end{aligned}$$

In fact, the countable completeness is only required if there are cyclic pure ε-rules. We omit the proof that M and M' are equivalent. Clearly, the xtt M' has no pure ε-rules.

Theorem 4. *If $(A, +, \cdot, 0, 1)$ is countably complete with respect to $\sum$, then for every xtt we can construct an equivalent xtt without pure ε-rules.*

Example 5. Recall the xtt $(Q, \Sigma, \Delta, \{q\}, R)$ of Example 3, which has the pure ε-rule ρ_1. Pure ε-rule elimination as outlined above yields the xtt with the rules (ρ_2)–(ρ_6) with their original weight and the new rule

$$q(\mathrm{S}(\mathrm{x}_1, \mathrm{VP}(\mathrm{x}_2, \mathrm{x}_3))) \xrightarrow{.2} \mathrm{S}'(q_{\mathrm{V}}(\mathrm{x}_2), q_{\mathrm{NP}}(\mathrm{x}_1), q_{\mathrm{NP}}(\mathrm{x}_3))\ . \qquad (\rho'_3)$$

For the rest of the section, we will assume that all used xtt do not have pure ε-rules. This assumption is often made immediately in the literature [40, 20] to ensure that the sum in (2) is always well-defined. The countable completeness of the semiring is thus only needed in the elimination of the pure ε-rules. The class of weighted tree transformations computed by xtt is denoted by XTOP. The subclasses computed by linear and linear nondeleting xtt are denoted by l-XTOP and ln-XTOP, respectively. The corresponding classes of weighted tree transformations computed by top-down tree transducers are TOP, l-TOP, and ln-TOP.

The rewrite semantics is very illustrative, but difficult to handle in proofs due to its essentially non-recursive specification. Next, we are going to present an alternative way to recursively define the semantics and then show that for every xtt both semantics indeed define the same weighted tree transformation. We need one additional notion. Let Σ be an alphabet and $t \in T_\Sigma$. Then

$$\mathrm{match}(t) = \{(c, \theta) \mid k \in \mathbb{N}, c \in C_\Sigma(\mathrm{X}_k), \theta \colon \mathrm{X}_k \to T_\Sigma \text{ with } t = c\theta\}\ .$$

Note that match(t) is finite. Recall that a normalized tree is linear and nondeleting in X_k for some $k \in \mathbb{N}$ and its variables $\{\mathrm{x}_1, \ldots, \mathrm{x}_k\}$ occur in order.

Definition 3. *Let $M = (Q, \Sigma, \Delta, I, R)$ be an xtt (without pure ε-rules). We define the mapping $h_R \colon Q(T_\Sigma) \times T_\Delta \to A$ for every $\xi \in Q(T_\Sigma)$ and $u \in T_\Delta$ by*

$$h_R(\xi, u) = \sum_{\substack{l \to r \in \mathrm{supp}(R) \\ (l, \theta') \in \mathrm{match}(\xi) \\ (s, \theta'') \in \mathrm{match}(u) \\ s \text{ normalized} \\ \theta \colon \mathrm{var}(s) \to Q(\mathrm{var}(l)) \\ r = s\theta}} R(l, r) \cdot \prod_{x \in \mathrm{var}(s)} h_R(x\theta\theta', x\theta'')\ .$$

Note that this recursion is well-defined because $|x\theta\theta'| \leq |t|$ and $|x\theta''| \leq |u|$ and one of the inequalities is strict due to the fact that $\{l, r\} \not\subseteq Q(\mathrm{X})$ for every $l \to r \in \mathrm{supp}(R)$. Consequently, we can define the weighted tree transformation $\tau'_M \colon T_\Sigma \times T_\Delta \to A$ by

$$\tau'_M(t, u) = \sum_{q \in I} h_R(q(t), u)$$

for every $t \in T_\Sigma$ and $u \in T_\Delta$.

Theorem 5. *For every xtt M (without pure ε-rules) we have $\tau_M = \tau'_M$.*

Proof. The statement follows immediately from the following statement, which we prove by induction on $|\xi| + |u|$.

$$h_R(\xi, u) = \sum_{\substack{k \in \mathbb{N}, \rho_1, \dots, \rho_k \in \mathrm{supp}(R) \\ \xi \Rightarrow^{\rho_1}_M \cdots \Rightarrow^{\rho_k}_M u}} R(\rho_1) \cdots R(\rho_k)$$

for every $\xi \in Q(T_\Sigma)$, and $u \in T_\Delta$.

$$\begin{aligned}
&\sum_{\substack{k \in \mathbb{N}, \rho_1, \dots, \rho_k \in \mathrm{supp}(R) \\ \xi \Rightarrow^{\rho_1}_M \cdots \Rightarrow^{\rho_k}_M u}} R(\rho_1) \cdots R(\rho_k) \\
= &\sum_{\substack{k \in \mathbb{N}, \rho_1, \dots, \rho_k \in \mathrm{supp}(R) \\ \rho_1 = l \to r, \theta \colon \mathrm{var}(l) \to T_\Sigma, \xi = l\theta \\ \xi \Rightarrow^{\rho_1}_M r\theta \Rightarrow^{\rho_2}_M \cdots \Rightarrow^{\rho_k}_M u}} R(\rho_1) \cdots R(\rho_k) \\
= &\sum_{\substack{l \to r \in \mathrm{supp}(R) \\ (l, \theta) \in \mathrm{match}(\xi)}} R(l, r) \cdot \Bigg(\sum_{\substack{k \in \mathbb{N}, \rho_2, \dots, \rho_k \in \mathrm{supp}(R) \\ r\theta \Rightarrow^{\rho_2}_M \cdots \Rightarrow^{\rho_k}_M u}} R(\rho_2) \cdots R(\rho_k) \Bigg) \\
= &\sum_{\substack{l \to r \in \mathrm{supp}(R) \\ (l, \theta') \in \mathrm{match}(\xi) \\ (s, \theta'') \in \mathrm{match}(u) \\ s \text{ normalized} \\ \theta \colon \mathrm{var}(s) \to Q(\mathrm{X}) \\ r = s\theta}} R(l, r) \cdot \prod_{w \in \mathrm{pos}_Q(r)} \Bigg(\sum_{\substack{n \in \mathbb{N}, \rho'_1, \dots, \rho'_n \in \mathrm{supp}(R) \\ r\theta'|_w \Rightarrow^{\rho'_1}_M \cdots \Rightarrow^{\rho'_n}_M u|_w}} R(\rho'_1) \cdots R(\rho'_n) \Bigg) \\
= &\sum_{\substack{l \to r \in \mathrm{supp}(R) \\ (l, \theta') \in \mathrm{match}(\xi) \\ (s, \theta'') \in \mathrm{match}(u) \\ s \text{ normalized} \\ \theta \colon \mathrm{var}(s) \to Q(\mathrm{X}) \\ r = s\theta}} R(l, r) \cdot \prod_{x \in \mathrm{var}(s)} \Bigg(\sum_{\substack{n \in \mathbb{N}, \rho'_1, \dots, \rho'_n \in \mathrm{supp}(R) \\ x\theta\theta' \Rightarrow^{\rho'_1}_M \cdots \Rightarrow^{\rho'_n}_M x\theta''}} R(\rho'_1) \cdots R(\rho'_n) \Bigg) \\
\overset{\text{I.H.}}{=} &\sum_{\substack{l \to r \in \mathrm{supp}(R) \\ (l, \theta') \in \mathrm{match}(\xi) \\ (s, \theta'') \in \mathrm{match}(u) \\ s \text{ normalized} \\ \theta \colon \mathrm{var}(s) \to Q(\mathrm{X}) \\ r = s\theta}} R(l, r) \cdot \prod_{x \in \mathrm{var}(s)} h_R(x\theta\theta', x\theta'')
\end{aligned}$$

$$= h_R(\xi, u) \ ,$$

where we isolated the first derivation step in the first 2 steps, split the subderivations in the third step, and simplified the obtained expression in the fourth step before we used the induction hypothesis in the fifth step. □

In the following, we will often use this alternative semantics of xtt to prove the correctness of constructions. We will not explicitly recall that it yields the same results as our reference semantics based on rewriting.

Example 6. Let us reconsider the xtt M of Example 5, and let

$$\begin{aligned} t &= \mathrm{S}(\mathrm{NP}(\mathrm{DT}(\mathit{the}), \mathrm{N}(\mathit{boy})), \mathrm{VP}(\mathrm{V}(\mathit{saw}), \mathrm{NP}(\mathrm{DT}(\mathit{the}), \mathrm{N}(\mathit{door})))) \\ u &= \mathrm{S}(\mathrm{CONJ}(\mathit{wa\text{-}}), \mathrm{S}'(\mathrm{V}(\mathit{ra'aa}), \mathrm{NP}(\mathrm{N}(\mathit{atefl})), \mathrm{NP}(\mathrm{N}(\mathit{albab})))) \\ u' &= \mathrm{S}'(\mathrm{V}(\mathit{ra'aa}), \mathrm{NP}(\mathrm{N}(\mathit{atefl})), \mathrm{NP}(\mathrm{N}(\mathit{albab}))) \end{aligned}$$

as in Example 4. Then

$$\begin{aligned} h_R(q(t), u) &= \sum_{\substack{l \to r \in \mathrm{supp}(R) \\ (l, \theta') \in \mathrm{match}(\xi) \\ (s, \theta'') \in \mathrm{match}(u) \\ s \text{ normalized} \\ \theta \colon \mathrm{var}(s) \to Q(\mathrm{var}(l)) \\ r = s\theta}} R(l, r) \cdot \prod_{x \in \mathrm{var}(s)} h_R(x\theta\theta', x\theta'') \\ &= R(\rho_2) \cdot h_R(q_{\mathrm{S}}(t), u') \\ &= R(\rho_2) \cdot h_R(q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\mathit{the}), \mathrm{N}(\mathit{boy}))), \mathrm{NP}(\mathrm{N}(\mathit{atefl}))) \\ &\quad \cdot h_R(q_{\mathrm{V}}(\mathrm{V}(\mathit{saw})), \mathrm{V}(\mathit{ra'aa})) \\ &\quad \cdot h_R(q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\mathit{the}), \mathrm{N}(\mathit{door}))), \mathrm{NP}(\mathrm{N}(\mathit{albab}))) \end{aligned}$$

where we see that the subtrees are evaluated independently and in parallel, whereas the derivation processed the leftmost subtree first. In addition, nondeterminism inside a particular subtree translation is handled locally, whereas nondeterminism is always handled globally in the rewrite semantics.

4 Expressive Power

In this section, we explore the expressive power of xtt and compare the introduced classes of weighted tree transformations. The number of classes was intentionally kept low in order to illustrate a particular approach. A more complete picture is shown in [21], but can also easily be obtained using the techniques recalled here.

Let us first recall the HASSE diagram for the unweighted case of [41, Figure 4.5]. Figure 3 shows the relevant subpart that we are interested in. The contribution [41] contains a much more refined HASSE diagram that relates many more classes. The interested reader might consult [41, Figure 4.5] and translate those additional results to the weighted setting using the approach demonstrated here.

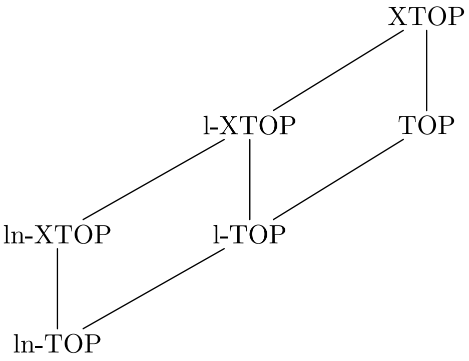

Figure 3: Hasse diagram of the classes of weighted tree transformations computed by xtt.

Theorem 6 (see [41, Theorem 4.11])**.** *Figure 3 is a* Hasse *diagram if* $(A, +, \cdot, 0, 1)$ *is the* Boolean *semiring.*

The approach that we want to demonstrate only concerns the strictness of the inclusions or the incomparability of classes. Variations of the approach are (implicitly and explicitly) used, for example, in [19, 37, 23]. Since the approach only covers inequalities, the inclusions have to be shown in the standard way. We choose the set of classes of weighted tree transformations such that all inclusions of Figure 3 trivially hold in every semiring. Now let us show how to lift a statement of the form $\mathcal{C} \nsubseteq \mathcal{C}'$ from the Boolean semiring to proper semirings. Recall that a nontrivial semiring is proper if it is not a ring and that every countably complete semiring is proper by Proposition 2.

First we lift the application of a semiring homomorphism $h\colon A \to B$ from semiring elements to weighted tree transformations and to xtt. Given a weighted tree transformation $\tau\colon T_\Sigma \times T_\Delta \to A$, we write $h(\tau)$ for the weighted tree transformation $h(\tau)\colon T_\Sigma \times T_\Delta \to B$ such that $h(\tau)(t, u) = h(\tau(t, u))$ for every $t \in T_\Sigma$ and $u \in T_\Delta$. Moreover, given an xtt $M = (Q, \Sigma, \Delta, I, R)$ we write $h(M)$ for the xtt $h(M) = (Q, \Sigma, \Delta, I, h(R))$ where $h(R)$ is such that $\mathrm{supp}(h(R)) \subseteq \mathrm{supp}(R)$ and $h(R)(\rho) = h(R(\rho))$ for every $\rho \in \mathrm{supp}(R)$.

The next theorem shows that applying an essentially complete semiring homomorphism h to an xtt M yields an xtt $h(M)$ that computes the weighted tree transformation $h(\tau_M)$. In other words, such a homomorphism is also compatible with xtt and its computed weighted tree transformations.

Theorem 7. *Let* $h\colon A \to B$ *be an essentially complete semiring homomorphism. Then* $\tau_{h(M)} = h(\tau_M)$ *for every xtt* M*.*

Proof. Let $M = (Q, \Sigma, \Delta, I, R)$, $t \in T_\Sigma$, and $u \in T_\Delta$. Then

$$h(\tau_M)(t, u) = h\Big(\sum_{\substack{q \in I, k \in \mathbb{N}, \rho_1, \ldots, \rho_k \in \mathrm{supp}(R) \\ q(t) \Rightarrow_M^{\rho_1} \cdots \Rightarrow_M^{\rho_k} u}} R(\rho_1) \cdots R(\rho_k)\Big)$$

$$
\begin{aligned}
&\overset{\dagger}{=} \sum_{\substack{q\in I, k\in\mathbb{N}, \rho_1,\dots,\rho_k\in\mathrm{supp}(R)\\ q(t)\Rightarrow_M^{\rho_1}\cdots\Rightarrow_M^{\rho_k} u}} h\big(R(\rho_1)\cdots R(\rho_k)\big)\\
&= \sum_{\substack{q\in I, k\in\mathbb{N}, \rho_1,\dots,\rho_k\in\mathrm{supp}(R)\\ q(t)\Rightarrow_M^{\rho_1}\cdots\Rightarrow_M^{\rho_k} u}} h(R(\rho_1))\cdots h(R(\rho_k))\\
&= \sum_{\substack{q\in I, k\in\mathbb{N}, \rho_1,\dots,\rho_k\in\mathrm{supp}(h(R))\\ q(t)\Rightarrow_M^{\rho_1}\cdots\Rightarrow_M^{\rho_k} u}} h(R)(\rho_1)\cdots h(R)(\rho_k)\\
&= \tau_{h(M)}(t,u) \enspace ,
\end{aligned}
$$

where we used the essential completeness of h in the step marked †. The summands $R(\rho_1)\cdots R(\rho_k)$ in the previous step clearly are in the finitely generated subsemiring $\langle C\rangle$ where $C = \{R(\rho) \mid \rho \in \mathrm{supp}(R)\}$, which is a finite set. □

With the help of the previous theorem we can now prove that if an inclusion is valid in the proper semiring $(A, +, \cdot, 0, 1)$, then it must also be valid in the Boolean semiring. Intuitively, this is achieved by just applying the essentially complete semiring homomorphism h of Section 2. We will typically use this statement as contraposition, if two classes are not contained in the Boolean semiring, then they also are not contained in the proper semiring $(A, +, \cdot, 0, 1)$, which is the desired lift result.

Theorem 8. *Let* $\mathcal{C}, \mathcal{C}' \in \{\text{ln-TOP}, \text{l-TOP}, \text{TOP}, \text{ln-XTOP}, \text{l-XTOP}, \text{XTOP}\}$*. If* $\mathcal{C} \subseteq \mathcal{C}'$ *holds in the proper commutative semiring* $(A, +, \cdot, 0, 1)$*, then it also holds in the* Boolean *semiring.*

Proof. Let h be the essentially complete semiring homomorphism discussed in Section 2. For every xtt N with the properties required by $\mathcal{C}$ over the Boolean semiring, we can easily construct an xtt M (with the same properties) over the proper semiring $(A, +, \cdot, 0, 1)$ such that $h(M) = N$. This can be achieved by reinterpreting N (up to the identity of the unit elements 0 and 1) as an xtt over the semiring $(A, +, \cdot, 0, 1)$. By Theorem 7, we have $h(\tau_M) = \tau_{h(M)} = \tau_N$. Since $\mathcal{C} \subseteq \mathcal{C}'$ is true in the proper semiring $(A, +, \cdot, 0, 1)$, there exists an xtt M' with the properties required by $\mathcal{C}'$ such that $\tau_{M'} = \tau_M$. Note that both M and M' compute over $(A, +, \cdot, 0, 1)$. Again we use Theorem 7 to conclude that the xtt $h(M')$ computes $\tau_{h(M')} = h(\tau_{M'}) = h(\tau_M) = \tau_N$ over the Boolean semiring. It is an easy exercise to verify that $h(M')$ has the same properties (linear, nondeleting, top-down tree transducer) as M'. Thus, we proved that for every xtt over the Boolean semiring with the properties required by $\mathcal{C}$ we can construct an equivalent xtt also over the Boolean semiring with the properties required by $\mathcal{C}'$, which proves the statement. □

Since the inclusions of Figure 3 are trivial and the inequalities can be lifted from the unweighted case using Theorem 8, we can immediately conclude the following theorem.

Theorem 9. *Figure 3 is a* Hasse *diagram for every proper commutative semiring* $(A, +, \cdot, 0, 1)$.

As already indicated the presented method applies just as well to other weighted devices such as weighted string transducers, weighted tree-walking automata, etc. In fact, it would be relatively easy to lift even the full Hasse diagram of [41, Figure 4.5] to the weighted case but since that involves a number of additional notions such as look-ahead, we leave this exercise to the reader.

We end this section with a demonstration of the usefulness of the different semantics and presentations of xtt. Again we do not strive to obtain the most general results, but rather we want to illustrate the principles. We start with *domain* and *range*. Let $\tau \colon T_\Sigma \times T_\Delta \to A$ be a weighted tree transformation. Then the *domain of* τ is the weighted tree language $\varphi \colon T_\Sigma \to A$ such that

$$\varphi(t) = \sum_{u \in T_\Delta} \tau(t, u)$$

for every $t \in T_\Sigma$. Mind that the sum might be infinite. It is finite if for every $t \in T_\Sigma$ there exist only finitely many $u \in T_\Delta$ such that $(t, u) \in \mathrm{supp}(\tau)$. For example, if τ is computed by an xtt without input ε-rules, then this property holds and the sum is finite. Intuitively, if an xtt does not have input ε-rules, then each derivation step consumes at least one input symbol. Thus, the number of derivation steps is limited by $|t|$, which can be used to derive an upper bound for the size of any output tree. If the sum is infinite, then we assume that the semiring $(A, +, \cdot, 0, 1)$ is countably complete with respect to $\sum$ as usual. Dually, the *range of* τ is the weighted tree language $\psi \colon T_\Delta \to A$ such that

$$\psi(u) = \sum_{t \in T_\Sigma} \tau(t, u)$$

for every $u \in T_\Delta$.

To keep the presentation simple, let $\tau = \tau_M$ for some linear and nondeleting xtt without input ε-rules. As already remarked the absence of input ε-rules guarantees that the sum in the definition of the domain is finite. Using our approach the result for arbitrary linear and nondeleting xtt over countably complete semirings can be derived using a result of [35] (see [17, Corollary 6.10]). Here we focus on the domain φ of τ_M. Since M is linear and nondeleting we can use Theorem 3 to obtain an equivalent weighted linear and complete bimorphism $B = (f, \psi, g)$ with $\psi \colon T_\Gamma \to A$. Since $\tau_B = \tau_M$, we can equivalently consider the domain of τ_B. By definition

$$\varphi(t) = \sum_{u \in T_\Delta} \tau_B(t, u) = \sum_{u \in T_\Delta} \Big(\sum_{\substack{s \in T_\Gamma \\ f(s) = t, g(s) = u}} \psi(s) \Big) = \sum_{\substack{s \in T_\Gamma, u \in T_\Delta \\ f(s) = t, g(s) = u}} \psi(s)$$

$$= \sum_{s \in T_\Gamma, f(s) = t} \psi(s) \enspace ,$$

which can be rewritten as $\varphi = \sum_{s \in T_\Gamma} \psi(s).f(s)$ where

- $\psi(s).f(s)$ is the weighted tree language that is 0 everywhere besides $f(s)$ where the weight is $\psi(s)$, and
- weighted tree languages are added componentwise.

This last presentation shows that the domain is just the application of the linear, complete, and ε-free tree homomorphism f to the recognizable weighted tree language ψ, where ε-free means that $f_k(\gamma) \neq \mathrm{x}_1$ for every $k \in \mathbb{N}$ and $\gamma \in \Gamma$ and it follows from the fact that M has no input ε-rule.

Theorem 10 (see [20, Corollary 8])**.** *The domain of a linear and nondeleting xtt without input ε-rules is a recognizable weighted tree language.*

Proof. Using the steps presented above and Theorem 2 we immediately obtain the statement. □

Clearly, the range of a linear and nondeleting xtt without output ε-rules is recognizable due to symmetry.

Finally, let us consider the input and the output product, which together with domain and range can be used to prove preservation of recognizability [20, 21]. But let us first define the mentioned input and output product. Given a weighted tree transformation $\tau\colon T_\Sigma \times T_\Delta \to A$ and weighted tree languages $\varphi\colon T_\Sigma \to A$ and $\psi\colon T_\Delta \to A$, the *input product* $\varphi \lhd \tau$ of τ by φ and the *output product* $\tau \rhd \psi$ of τ by ψ are defined by

$$(\varphi \lhd \tau)(t, u) = \varphi(t) \cdot \tau(t, u) \qquad \text{and} \qquad (\tau \rhd \psi)(t, u) = \tau(t, u) \cdot \psi(u) \enspace ,$$

respectively, for every $t \in T_\Sigma$ and $u \in T_\Delta$.

Often input and output products are handled by specialized Bar-Hillel constructions [5, 45] or compositions [4, 12]. We will discuss the composition approach in the third part of this survey, but let us present an input product construction for weighted tree transformations computed by linear and nondeleting weighted top-down tree transducers and recognizable weighted tree languages. We will show that every such input product can again be computed by a linear and nondeleting weighted top-down tree transducer. A more detailed overview on input and output products can be found in [39].

From now on, let $M = (Q, \Sigma, \Delta, I, R)$ be a linear and nondeleting weighted top-down tree transducer and $N = (P, \Sigma, \delta, F)$ be a wta. We want to construct a linear and nondeleting weighted top-down tree transducer M' such that $\tau_{M'} = \varphi_N \lhd \tau_M$. Since M is linear and nondeleting, it visits each input subtree exactly once.

Definition 4. *The input product $N \lhd M$ is the weighted top-down tree transducer $(Q', \Sigma, \Delta, I', R')$ where*

- *$Q' = Q \times P$,*
- *$I' = \{(q, p) \mid q \in I, p \in F\}$, and*
- *$R'(l, r) = \delta(p_1 \cdots p_k, \sigma, p) \cdot R(l', r')$ for all $l \in Q'(T_\Sigma(\mathrm{X}))$ and $r \in T_\Delta(Q'(\mathrm{X}))$ where*
  - *l' and r' are obtained from l and r, respectively, by dropping the second component in the states that occur in l' and r',*

- *$p \in P$, $k \in \mathbb{N}$, and $\sigma \in \Sigma$ are such that $l = (q, p)(\sigma(\mathrm{x}_1, \dots, \mathrm{x}_k))$ for some $q \in Q$, and*
- *for every $i \in [k]$, the state p_i is such that there is a position $w_i \in \mathrm{pos}_{Q'}(r)$ in the right-hand side with $r|_{w_i} = (q_i, p_i)(\mathrm{x}_i)$ for some $q_i \in Q$.*

To illustrate the use of the alternative semantics, we will prove that the input product transducer indeed computes the input product as desired.

Theorem 11 (see [39, Theorem 2])**.** *$\tau_{(N \triangleleft M)} = \varphi_N \triangleleft \tau_M$.*

Proof. Let $N \triangleleft M = (Q', \Sigma, \Delta, I', R')$ as in Definition 4. We prove that

$$h_{R'}((q, p)(t), u) = \delta(t, p) \cdot h_R(q(t), u)$$

for every $t \in T_\Sigma$, $u \in T_\Delta$, $q \in Q$, and $p \in P$. Let $\xi = (q, p)(t)$.

$$\begin{aligned}
&h_{R'}(\xi, u) \\
={}& \sum_{\substack{l \to r \in \mathrm{supp}(R') \\ (l, \theta') \in \mathrm{match}(\xi) \\ (s, \theta'') \in \mathrm{match}(u) \\ s \text{ normalized} \\ \theta\colon \mathrm{var}(s) \to Q'(\mathrm{var}(l)) \\ r = s\theta}} R'(l, r) \cdot \prod_{x \in \mathrm{var}(s)} h_{R'}(x\theta\theta', x\theta'') \\
={}& \sum_{\substack{l \to r \in \mathrm{supp}(R) \\ (l, \theta') \in \mathrm{match}(q(t)) \\ (s, \theta'') \in \mathrm{match}(u) \\ s \text{ normalized} \\ \theta\colon \mathrm{var}(s) \to Q(\mathrm{var}(l)) \\ k = |\mathrm{var}(s)|, p_1, \dots, p_k \in P \\ r = s\theta}} \delta(p_1 \cdots p_k, l(1), p) \cdot R(l, r) \cdot \prod_{i=1}^{k} h_{R'}(((\mathrm{x}_i\theta)(\varepsilon), p_i)(\mathrm{x}_i\theta'), \mathrm{x}_i\theta'') \\
\overset{\dagger}{=}{}& \sum_{\substack{l \to r \in \mathrm{supp}(R) \\ (l, \theta') \in \mathrm{match}(q(t)) \\ (s, \theta'') \in \mathrm{match}(u) \\ s \text{ normalized} \\ \theta\colon \mathrm{var}(s) \to Q(\mathrm{var}(l)) \\ k = |\mathrm{var}(s)|, p_1, \dots, p_k \in P \\ r = s\theta}} \delta(p_1 \cdots p_k, l(1), p) \cdot R(l, r) \cdot \prod_{i=1}^{k} \delta(\mathrm{x}_i\theta', p_i) \cdot h_R(\mathrm{x}_i\theta\theta', \mathrm{x}_i\theta'') \\
\overset{\dagger}{=}{}& \delta(t, p) \cdot h_R(q(t), u) \enspace ,
\end{aligned}$$

where we used the induction hypothesis in the step marked †. With the auxiliary statement established, the proof of the main statement is now easy. Let $t \in T_\Sigma$ and $u \in T_\Delta$. Then

$$\begin{aligned}
\tau_{(N \triangleleft M)}(t, u) &= \sum_{q' \in I'} h_{R'}(q'(t), u) = \sum_{q \in I, p \in F} h_{R'}((q, p)(t), u) \\
&= \sum_{q \in I, p \in F} \delta(t, p) \cdot h_R(q(t), u) = \varphi_N(t) \cdot \tau_M(t, u) = (\varphi_N \triangleleft \tau_M)(t, u) \enspace .
\end{aligned}$$

□

Acknowledgments

The author would like to express his gratitude to the reviewers. Their insight and remarks improved the article. In addition, the author would like to acknowledge the financial support of the *Ministerio de Educación y Ciencia* (MEC) grant JDCI-2007-760.

References

[1] Alexandrakis, Athanasios and Bozapalidis, Symeon. Weighted grammars and Kleene's theorem. *Inf. Process. Lett.*, 24(1):1–4, 1987.

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

[3] Arnold, André and Dauchet, Max. Morphismes et bimorphismes d'arbres. *Theoret. Comput. Sci.*, 20(4):33–93, 1982.

[4] Baker, Brenda S. Composition of top-down and bottom-up tree transductions. *Inform. and Control*, 41(2):186–213, 1979.

[5] Bar-Hillel, Yehoshua, Perles, Micha, , and Shamir, Eliyahu. On formal properties of simple phrase structure grammars. In Bar-Hillel, Yehoshua, editor, *Language and Information: Selected Essays on their Theory and Application*, chapter 9, pages 116–150. Addison Wesley, 1964.

[6] Berstel, Jean and Reutenauer, Christophe. Recognizable formal power series on trees. *Theoret. Comput. Sci.*, 18(2):115–148, 1982.

[7] Courcelle, Bruno and Franchi–Zannettacci, Paul. Attribute grammars and recursive program schemes. *Theor. Comput. Sci.*, 17(1):163–191 & 235–257, 1982.

[8] Dauchet, Max. *Transductions inversibles de forêts.* Thèse 3ème cycle, Université de Lille, 1975.

[9] Droste, Manfred, Kuich, Werner, and Vogler, Heiko, editors. *Handbook of Weighted Automata.* EATCS Monographs on Theoret. Comput. Sci. Springer, 2009.

[10] Droste, Manfred, Stüber, Torsten, and Vogler, Heiko. Weighted finite automata over strong bimonoids. *Inf. Sci.*, 180(1):156–166, 2010.

[11] Eilenberg, Samuel. *Volume A: Automata, Languages, and Machines*, volume 59 of *Pure and Applied Math.* Academic Press, 1974.

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

[13] Engelfriet, Joost. Top-down tree transducers with regular look-ahead. *Math. Systems Theory*, 10(1):289–303, 1976.

[14] Engelfriet, Joost, Fülöp, Zoltán, and Vogler, Heiko. Bottom-up and top-down tree series transformations. *J. Autom. Lang. Combin.*, 7(1):11–70, 2002.

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

[16] Engelfriet, Joost and Vogler, Heiko. Modular tree transducers. *Theor. Comput. Sci.*, 78(2):267–303, 1991.

[17] Ésik, Zoltán and Kuich, Werner. Formal tree series. *J. Autom. Lang. Combin.*, 8(2):219–285, 2003.

[18] Fülöp, Zoltán. On attributed tree transducers. *Acta Cybernet.*, 5(1):261–279, 1981.

[19] Fülöp, Zoltán, Gazdag, Zsolt, and Vogler, Heiko. Hierarchies of tree series transformations. *Theoret. Comput. Sci.*, 314(3):387–429, 2004.

[20] Fülöp, Zoltán, Maletti, Andreas, and Vogler, Heiko. Preservation of recognizability for synchronous tree substitution grammars. In *Proc. 1st Workshop Applications of Tree Automata in Natural Language Processing*, pages 1–9. Association for Computational Linguistics, 2010.

[21] Fülöp, Zoltán, Maletti, Andreas, and Vogler, Heiko. Weighted extended tree transducers. submitted, 2011.

[22] Fülöp, Zoltán and Vogler, Heiko. Weighted tree transducers. *J. Autom. Lang. Combin.*, 9(1):31–54, 2004.

[23] Fülöp, Zoltán and Vogler, Heiko. Weighted tree automata and tree transducers. In Droste et al. [9], chapter 9, pages 313–403.

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

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

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

[27] Graehl, Jonathan and Knight, Kevin. Training tree transducers. In *Proc. 2004 Human Language Technology Conf. NAACL*, pages 105–112. Association for Computational Linguistics, 2004.

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

[29] Hopcroft, John E. and Ullman, Jeffrey D. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[30] Jurafsky, Daniel and Martin, James H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Processing.* Prentice-Hall, 2000.

[31] Karner, Georg. Continuous monoids and semirings. *Theoret. Comput. Sci.*, 318(3):355–372, 2004.

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

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

[34] Kuich, Werner. Formal power series over trees. In *Proc. 3rd Int. Conf. Developments in Language Theory*, pages 61–101. Aristotle University of Thessaloniki, 1998.

[35] Kuich, Werner. Tree transducers and formal tree series. *Acta Cybernet.*, 14(1):135–149, 1999.

[36] Kuich, Werner and Salomaa, Arto. *Semirings, Automata, Languages*, volume 5 of *Monographs in Theoretical Computer Science. An EATCS Series.* Springer, 1986.

[37] Maletti, Andreas. The power of tree series transducers of type I and II. In *Proc. 9th Int. Conf. Developments in Language Theory*, volume 3572 of LNCS, pages 338–349. Springer, 2005.

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

[39] Maletti, Andreas. 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, 2010.

[40] Maletti, Andreas. Why synchronous tree substitution grammars? In *Proc. Human Language Technology Conf. NAACL*, pages 876–884. Association for Computational Linguistics, 2010.

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

[42] Manning, Chris and Schütze, Hinrich. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

[43] Raoult, Jean-Claude. Rational tree relations. *Bull. Belg. Math. Soc.*, 4(1):149–176, 1997.

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

[45] Satta, Giorgio. Translation algorithms by means of language intersection. *Manuscript*, 2011. available at: `http://www.dei.unipd.it/~satta`.

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

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

[48] Thatcher, James W. Tree automata: an informal survey. In Aho, Alfred V., editor, *Currents in the Theory of Computing*, pages 143–172. Prentice Hall, 1973.

[49] Wang, Huaxiong. On characters of semirings. *Houston J. Math.*, 23(3):391–405, 1997.

[50] Wang, Huaxiong. On rational series and rational languages. *Theoret. Comput. Sci.*, 205(1–2):329–336, 1998.

Received April 13, 2011