Weighted logics for XML

Manfred Droste
Institute of Computer Science
Leipzig University
D-04109 Leipzig, Germany
droste@informatik.uni-leipzig.de

Heiko Vogler
Department of Computer Science
Dresden University of Technology
D-01062 Dresden, Germany
vogler@tcs.inf.tu-dresden.de

February 8, 2007

Abstract: We define a weighted monadic second order logic for XML-documents (weighted uMSO-logic), viewed as unranked trees. Such a logic allows us to pose quantitative queries to XML-databases. Also we define the concept of unranked weighted tree automata and prove that, for every commutative semiring K, they have the same expressive power as (1) restricted weighted uMSO-logic, (2) almost existential weighted uMSO-logic whenever the additive monoid of K is locally finite, (3) (unrestricted) weighted uMSO-logic whenever K is locally finite. Moreover, if K is a computable field, then the equivalence of any two sentences of restricted weighted uMSO-logic is decidable and there is an effective procedure to construct the automaton from the sentence.

1 Introduction

The eXtensible Markup Language XML is a standard for representing and exchanging semistructured data on the web [ABS00, KSS03], this standard has been adopted by the World Wide Web Consortium [BPSM98]. The data of XML have the form of labeled, ordered trees in which the labels do not have a rank or arity, i.e., a label does not determine the number of successors of the node which it labels. Such trees are also called unranked trees.

Whereas the XML-technology and thus, the use of unranked trees as data model is rather new, the investigation of formal languages, automata, and logic on ranked trees started in the 60s of the previous century and is now a well established research area, cf., e.g. [GS84, GS97, CDG+97] for survey books on these topics. A fundamental result in this area is the fact that every recognizable tree language is definable by a formula of monadic second order (MSO) logic and vice versa [TW68, Don70] which generalizes Büchi's theorem [Büc60] for the string case. The corresponding equivalence result for unranked tree languages has been observed by several authors, including [Lib05, Nev02, NS02]. In this logic one can pose qualitative questions like whether, in a given bibliographic database (formalized as an unranked tree), there is an entry which misses optional information of a certain kind, or whether there is a paper with three authors.

A natural problem for XML is to pose *quantitative* queries to documents. For instance, for the scenario of bibliographic databases one might ask how many entries miss optional information. Or, on a more specific level: given the different efforts (measured as natural numbers) for completing a `book`-entry and an `article`-entry, respectively; then one might want to know the weighted sum which shows the whole effort to complete all `book`- and all `article`-entries in the database.

In this paper we present a weighted logic suitable for the formulation of such quantitative queries for unranked trees (weighted uMSO-logic). Depending on the specific situation, the queries (= formulas) are evaluated in different semantic domains, like the natural or real numbers, or any given Boolean algebra. In order to cope with such different situations in a uniform way, weights are often modelled with respect to some semiring K. For instance, graphs with semiring-based weights have been investigated in [Car71] in the area of network routing problems. Starting with [Sch61], weighted automata and the theory of formal power series have been investigated intensively [SS78, KS86, BR88, Kui97]. We will also define our weighted logic for unranked trees with respect to a given semiring.

As an automata-theoretic counterpart of our weighted uMSO-logic, we will introduce weighted bottom-up tree automata over unranked trees (for short: uwta). These generalize the bottom-up tree automata over unranked trees [BMW01, Nev02] which can be obtained from our model by choosing the Boolean semiring for K, and also weighted tree automata on ranked trees [Kui98, Boz99]. Both the behaviour of uwta and the semantics of a sentence of our weighted uMSO-logic are described by unranked tree series, i.e., by functions associating to each unranked tree a value in K.

As the main results in this paper, we will prove the following:

(1) An unranked tree series S over a commutative semiring K is the behaviour of a uwta if and only if S is definable by a sentence from a suitable restriction of our weighted uMSO-logic (cf. Theorem 4.7).

(2) For large classes of semirings including all idempotent semirings (like the tropical and max-plus semirings) and all fields of prime characteristic, we obtain an equivalence between uwta and a purely syntactically defined restriction of our weighted uMSO-logic, called almost existential uMSO-logic (cf. Theorem 8.10). In Section 10 we will show how the above mentioned task (of computing the effort for completing a database of `book`- and `article`-entries) can be specified by a formula of almost existential uMSO-logic.

(3) For the class of locally finite semirings we obtain this equivalence result for the unrestricted weighted uMSO-logic (cf. Theorem 9.2).

(4) Furthermore, if K is any computable field (e.g. the rational numbers) our constructions are effective, and given two sentences from our restricted weighted uMSO-logic, we can decide whether their semantics are equal (cf. Corollary 7.4).

In our proofs we will employ a corresponding equivalence result recently shown in Theorem 5.1 of [DV06] for weighted tree automata and weighted MSO-logics over ranked trees (also cf. [DG05] for the origin of weighted MSO-logics). As in the unweighted case we will encode unranked trees by trees over a binary ranked alphabet (compare [Suc02, Nev02]). Then we will show that this encoding preserves recognizability by weighted tree automata and definability by our weighted MSO-logic in both directions (cf. Figure 1); in each of the underlying Lemmas 6.1, 6.4, 7.2, and 7.5 we will give direct constructions of the tree automata or formulas, and formally prove the correctness of our constructions.

Our proofs involve precise counting arguments for the numbers of successful paths in the weighted automata. We would like to stress that, in contrast to the language case, for weighted

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\begin{array}{c}\text{uwta } M \\ \text{over } \Sigma \text{ and } K\end{array} \arrow[r, shift left, "\text{Lemma 6.1}"] \arrow[d, shift right, "\text{Theorem 6.5}"'] & \begin{array}{c}\text{wta } M' \\ \text{over } enc(\Sigma) \text{ and } K\end{array} \arrow[l, shift left, "\text{Lemma 7.5}"] \arrow[d, shift right, "\begin{array}{c}\text{Theorem 5.11}\\ \text{of [DV06]}\end{array}"'] \\
\begin{array}{c}\text{uRMSO-formula } \varphi \\ \text{over } \Sigma \text{ and } K\end{array} \arrow[r, shift left, "\text{Lemma 7.2}"] \arrow[u, shift right, "\text{Theorem 7.5}"'] & \begin{array}{c}\text{RMSO-formula } \varphi \\ \text{over } enc(\Sigma) \text{ and } K\end{array} \arrow[l, shift left, "\text{Lemma 6.4}"] \arrow[u, shift right, "\begin{array}{c}\text{Theorem 5.6}\\ \text{of [DV06]}\end{array}"']
\end{tikzcd}
$$

Figure 1: An overview of the proof of Theorem 4.7.

automata such precise counting arguments are crucial. Also, often for weighted automata results, additional assumptions on the underlying semiring are necessary, like idempotence. We also encounter additional assumptions here, but we are able to avoid idempotence, and our first main result applies to any commutative semiring.

We note that in [SSM03], an MSO-logic for unranked trees with Presburger constraints on the children of nodes was presented; the satisfiability of this logic was shown to be undecidable. Recently, [SSMH04] presented a modal fixpoint logic with Presburger constraints which becomes decidable. It would be interesting to compare and possibly combine these approaches with the present one formulated for arbitrary semirings of numerical weights.

This paper contains ten sections. After having recalled the basic notions of trees, semirings, formal power series (over words), and weighted finite automata in Section 2, we will define in Sections 3 and 4 the concepts of uwta and of weighted uMSO-logic, respectively. Section 4 also contains the first main result (cf. Theorem 4.7). In Section 5 we formally define the encoding of unranked trees by ranked trees. Section 6 contains the first part of the proof of the first main result, i.e., for every commutative semiring, every recognizable unranked tree series is definable by restricted weighted uMSO-logic, and Section 7 contains the proof of the reverse direction; also in this section we prove the fourth main result (cf. Corollary 7.4). In Sections 8 and 9 we prove the second and the third main result (cf. Theorems 9.2 and 8.10). Finally, Section 10 contains an extended example where we define an XML-database with bibtex entries and show how quantitative queries of the above mentioned form can be formulated.

2 Preliminaries

2.1 Trees

Let $\mathbb{N}$ and $\mathbb{N}_+$ be the sets $\{0, 1, 2, \ldots\}$ and $\{1, 2, \ldots\}$, respectively. Let Σ be an alphabet, i.e., a finite nonempty set. The set of *unranked* Σ*-trees*, denoted by U_Σ, is the smallest subset U of $(\Sigma \cup \{(,)\} \cup \{,\})^*$ such that if $\sigma \in \Sigma$ and $t_1, \ldots, t_k \in U$ with $k \geq 0$, then $\sigma(t_1, \ldots, t_k) \in U$. In case $k = 0$, we identify $\sigma(\,)$ with σ; thus $\Sigma \subseteq U_\Sigma$.

We define the set of *positions in an unranked* Σ*-tree* by means of the mapping $\mathrm{pos} : U_\Sigma \to \mathcal{P}((\mathbb{N}_+ \cup \{\cdot\})^*)$ inductively on the argument $t \in U_\Sigma$ as follows: if $t = \sigma(t_1, \ldots, t_k)$ where $\sigma \in \Sigma$,

$k \geq 0$ and $t_1, \ldots, t_k \in U_\Sigma$, then $\text{pos}(t) = \{\varepsilon\} \cup \{i \cdot v \mid 1 \leq i \leq k, v \in \text{pos}(t_i)\}$. Thus $\text{pos}(t) = \{\varepsilon\}$ for every $t \in \Sigma$. Sometimes we will also drop the dot $\cdot$ and just write iv rather than $i \cdot v$.

For every $t \in U_\Sigma$ and $w \in \text{pos}(t)$, the *label of t at w*, denoted by $t(w) \in \Sigma$, is defined inductively as follows: if $t = \sigma(t_1, \ldots, t_k)$ for some $\sigma \in \Sigma$ with $k \geq 0$ and $t_1, \ldots, t_k \in U_\Sigma$, then $t(\varepsilon) = \sigma$, and if $1 \leq i \leq k$ and $w = iv$, then $t(w) = t_i(v)$.

Given an unranked Σ-tree t, then with every position $w \in \text{pos}(t)$ we associate the rank $rk_t(w)$ at this position, which is the number of direct descendants at w, i.e., $rk_t(w) = k$ if there are $\sigma \in \Sigma$ and $t_1, \ldots, t_k \in U_\Sigma$ such that $t|_w = \sigma(t_1, \ldots, t_k)$. Note that, in an unranked Σ-tree t there can be different positions $w, w' \in \text{pos}(t)$ which are labeled by the same symbol (i.e., $t(w) = t(w')$), but the ranks of w and w' are different (i.e., $rk_t(w) \neq rk_t(w')$).

In this paper Σ will always denote an arbitrary alphabet unless specified otherwise.

The ranked case is a special instance of the unranked case, which is obtained as follows. A *ranked alphabet* is a tuple (Σ, rk) where $rk : \Sigma \to \mathbb{N}$. The set T_Σ of *(ranked) Σ-trees* is the subset of U_Σ defined by $T_\Sigma = \{t \in U_\Sigma \mid \forall w \in \text{pos}(t) : rk_t(w) = rk(t(w))\}$. Clearly, since Σ-trees are particular unranked Σ-trees, the notions of position, subtrees, and label are also defined for Σ-trees. Also, for every $k \geq 0$, we let $\Sigma^{(k)} = rk^{-1}(k)$ and $\max_\Sigma = \max\{k \mid \Sigma^{(k)} \neq \emptyset\}$.

Let Σ be a ranked alphabet, $k \geq 1$, $L_1, \ldots, L_k \subseteq T_\Sigma$ be tree languages, and $\sigma \in \Sigma^{(k)}$. Then we define $\sigma(L_1, \ldots, L_k) = \{\sigma(t_1, \ldots, t_k) \mid t_i \in L_i\}$ and if $\Sigma' \subseteq \Sigma^{(k)}$, then we define $\Sigma'(L_1 \ldots, L_k) = \bigcup_{\sigma \in \Sigma'} \sigma(L_1, \ldots, L_k)$.

Given two sets A and B and an arbitrary element $(a, b) \in A \times B$, we put $(a, b)_1 = a$ and $(a, b)_2 = b$.

2.2 Semirings, formal power series, and weighted finite automata

Here we recall the basic notions on semirings, formal power series, and weighted finite automata on strings. We refer the reader for more information to [Eil74, SS78, KS86].

A semiring is a quintuple $(K, +, \cdot, 0, 1)$ (often abbreviated by K) where $(K, +, 0)$ is a commutative additive monoid, $(K, \cdot, 1)$ is a multiplicative monoid, multiplication distributes over addition from both sides, and 0 is absorbing with respect to multiplication, i.e.. for every $k \in K$ we have that $0 \cdot k = k \cdot 0 = 0$. A semiring is *commutative* if its multiplication is commutative.

In this paper, K will always denote an arbitrary commutative semiring unless specified otherwise.

Important examples of semirings are:

- the *Boolean semiring* $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$ with disjunction $\vee$ and conjunction $\wedge$,
- the *semiring of natural numbers* $(\mathbb{N}, +, \cdot, 0, 1)$, abbreviated by $\mathbb{N}$, with the usual addition and multiplication,
- the *tropical semiring* $Trop = (\mathbb{N} \cup \{\infty\}, min, +, \infty, 0)$ where the sum and the product operations are min and $+$, resp., extended to $\mathbb{N} \cup \{\infty\}$ in the usual way.

Let Z be an arbitrary set. A *(formal) power series* over K and Z is a mapping $S : Z \to K$. The set of all formal power series over K and Z is denoted by $K\langle\langle Z\rangle\rangle$. For $z \in Z$, the value $S(z)$ is here, as usual, denoted as (S, z). We define the operations *sum* and *Hadamard-product* on $K\langle\langle Z\rangle\rangle$ as follows: for $S, S' \in K\langle\langle Z\rangle\rangle$ and $z \in Z$ we let $(S + S', z) = (S, z) + (S', z)$ and $(S \odot S', z) = (S, z) \cdot (S', z)$. Let $k \in K$ and $S \in K\langle\langle Z\rangle\rangle$. Then $k \cdot S \in K\langle\langle Z\rangle\rangle$ is defined by $(k \cdot S, z) = k \cdot (S, z)$ for every $z \in Z$.

Let Σ be an alphabet. Then we call an element $S \in K\langle\langle U_\Sigma\rangle\rangle$ a *unranked tree series*. The set $supp(S) = \{t \in U_\Sigma | (S, t) \neq 0\}$ is called the *support of* S. Let $L \subseteq U_\Sigma$. Then the *characteristic unranked tree series* $\mathbb{1}_L \in K\langle\langle U_\Sigma\rangle\rangle$ *of* L is defined for every $t \in U_\Sigma$ by $(\mathbb{1}_L, t) = 1$ if $t \in L$, and $(\mathbb{1}_L, t) = 0$ otherwise.

Observe that if $K = \mathbb{B}$, the mapping $L \mapsto \mathbb{1}_L$ provides a bijection between unranked tree languages and unranked tree series.

Let $(K, +, \cdot, 0, 1)$ be a semiring and Δ be an alphabet. As usual, a *weighted finite state string automaton (for short: wsa) over* K *and* Δ is a quadruple $A = (P, \lambda, \mu, \nu)$ such that P is a finite set (of *states*), $\mu : \Delta \to K^{P\times P}$ is a mapping (called *transition weight function*), and $\lambda, \nu : P \to K$ are functions (called *initial weight function* and *final weight function*, respectively). A *run (through A)* is a sequence $r = (p_0, a_1, p_1)(p_1, a_2, p_2) \ldots (p_{n-1}, a_n, p_n)$ where $p_i \in P$ and $a_i \in \Delta$ with $0 \leq i \leq n$. We say that r is a *run from* p_0 *to* p_n *with label* $w = a_1 \ldots a_n$; the set of all such runs is denoted by $P_{p_0,p_n}(w)$. The *weight of* r is the product $wt(r) = \prod_{i=1}^{n} \mu(a_i)_{p_{i-1},p_i}$. Note that $r = \varepsilon$ if $n = 0$, and $wt(\varepsilon) = 1$ because, as usual, in K products over empty index sets are defined to be 1. The *behaviour of* A (or: *power series recognized by* A) is the power series $\|A\| \in K\langle\langle \Delta^*\rangle\rangle$ such that for every $w \in \Delta^*$ we have

$$(\|A\|, w) = \sum_{p,p' \in P} \lambda(p) \cdot \mu(w)_{p,p'} \cdot \nu(p')$$

where $\mu(w)_{p,p'} = \sum_{r \in P_{p,p'}(w)} wt(r)$.

A power series $S \in K\langle\langle \Delta^*\rangle\rangle$ is *recognizable over* K *and* Δ if there is a weighted finite automaton A over K and Δ such that $S = \|A\|$.

3 Unranked weighted tree automata

In this section we extend the concept of nondeterministic bottom-up tree automata on unranked trees [BMW01] (also cf. [Nev02]) by weights taken from some semiring $(K, +, \cdot, 0, 1)$. The classical concept of unranked tree automata is obtained by letting $K = \mathbb{B}$.

An *unranked weighted (finite state) bottom-up tree automaton (for short: uwta) over* K is a quadruple $M = (Q, \Sigma, \delta, \gamma)$ where Q is a finite set (of *states*), Σ is an alphabet (of *input symbols*), $\delta : Q \times \Sigma \to K\langle\langle Q^*\rangle\rangle$ is a mapping (*transition function*) such that $\delta(q, \sigma)$ is a recognizable power series over K and Q, and $\gamma : Q \to K$ is a mapping (*final weight function*). A uwta M is *deterministic* if for every $\sigma \in \Sigma$ and $q_1 \ldots q_k \in Q^*$ there is at most one $q \in Q$ such that $(\delta(q, \sigma), q_1 \ldots q_k) \neq 0$.

Now we define the run semantics of a uwta M. The *set of runs of* M is the set $U_{Q\times\Sigma}$ of unranked $(Q \times \Sigma)$-trees. We define the projection mapping $tree : U_{Q\times\Sigma} \to U_\Sigma$ by cancelling the Q-component; that is, if $r = (q, \sigma)(r_1, \ldots, r_k)$ for some $q \in Q$, $\sigma \in \Sigma$, $k \geq 0$, and $r_1, \ldots, r_k \in U_{Q\times\Sigma}$, then $tree(r) = \sigma(tree(r_1), \ldots, tree(r_k))$. If $t = tree(r)$, then we also call r

a *run on t*. The *weight function (of runs)* $wt_M : U_{Q\times\Sigma} \to K$ is defined inductively as follows: if $r = (q,\sigma)(r_1,\dots,r_k)$ for some $q \in Q$, $\sigma \in \Sigma$, $k \geq 0$, and $r_1,\dots,r_k \in U_{Q\times\Sigma}$, then

$$wt_M(r) = (\delta(q,\sigma), q_1 \dots q_k) \cdot wt_M(r_1) \cdot \ldots \cdot wt_M(r_k)$$

where $q_i = r_i(\varepsilon)_1$ for every $1 \leq i \leq k$. Since K is commutative, it is obvious that for every $r \in U_{Q\times\Sigma}$,

$$wt_M(r) = \prod_{w \in \mathrm{pos}(r)} (\delta(q^w, \sigma^w), q_1^w \dots q_{rk_r(w)}^w)$$

where $(q^w, r^w) = r(w) \in Q \times \Sigma$ and $q_i^w = r(wi)_1 \in Q$ for every $1 \leq i \leq rk_r(w)$. Clearly, if M is deterministic, then for every $t \in U_\Sigma$ there is at most one run r on t such that $wt_M(r) \neq 0$.

The *unranked tree series accepted by M over K* is the unranked tree series $S_M \in K\langle\langle U_\Sigma\rangle\rangle$ defined by

$$(S_M, t) = \sum_{r \in tree^{-1}(t)} wt_M(r) \cdot \gamma(r(\varepsilon)_1)$$

for every $t \in U_\Sigma$. An unranked tree series $S \in K\langle\langle U_\Sigma\rangle\rangle$ is *recognizable over K* if there is a uwta M over K such that $S = S_M$. We will denote the class of all recognizable unranked tree series over K and Σ by $K^{\mathrm{rec}}\langle\langle U_\Sigma\rangle\rangle$.

An unranked tree language $L \subseteq U_\Sigma$ is *recognizable* if there is a uwta M over $\mathbb{B}$ such that $L = \mathrm{supp}(S_M)$. It is easy to see that, using the bijection between unranked tree languages and characteristic unranked tree series, this notion coincides with the usual definition of recognizability for unranked tree languages. In fact, nondeterministic (and deterministic) bottom-up tree automaton on unranked trees in the sense of [BMW01, Nev02] are precisely the uwta (and deterministic uwta, respectively) over $\mathbb{B}$. Subsequently, we will often use the fact that the class of recognizable unranked tree languages is closed under intersection and complement, cf. Theorem H of [BMW01]. In Theorem B of [BMW01] it is proved (by using the well known power set construction) that every recognizable unranked tree language is recognizable by a deterministic uwta over $\mathbb{B}$. Using this property, it is easy to see that for every recognizable unranked tree language L, the series $\mathbb{1}_L \in K\langle\langle U_\Sigma\rangle\rangle$ is recognizable.

Also weighted tree automata in the sense of [Boz99, Kui98, DPV05] are particular uwta. More precisely, given a rank function $rk : \Sigma \to \mathbb{N}$, a *weighted tree automaton (over K)* (for short: wta) is a uwta $(Q, \Sigma, \delta, \gamma)$ such that $supp(\delta(q,\sigma)) \subseteq Q^{rk(\sigma)}$ for every $q \in Q$ and $\sigma \in \Sigma$. Since $Q^{rk(\sigma)}$ is a finite set, it follows directly that $\delta(q,\sigma)$ is a recognizable power series over K and Σ. We will denote the class of all recognizable tree series over K and Σ by $K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle$. We will use the notion of recognizability for both, the unranked and the ranked case; there is no danger of confusion, because the respective context will determine the appropriate case. Note that, for every wta M, the support of S_M is a subset of T_Σ. For later use, we will call an alphabet Σ and a wta over Σ *binary* if $\Sigma = \Sigma^{(0)} \cup \Sigma^{(2)}$.

An unranked tree series $S \in K\langle\langle U_\Sigma\rangle\rangle$ is a *recognizable unranked step function* if there are an $n \geq 1$ and recognizable unranked tree languages $L_1,\dots,L_n \subseteq U_\Sigma$ such that $S = \sum_{i=1}^n k_i \cdot \mathbb{1}_{L_i}$. Equivalently, S assumes only finitely many values, and for each $k \in K$ the language $S^{-1}(k) = \{t \in U_\Sigma \mid (S,t) = k\}$ is recognizable.

Lemma 3.1. Let $S \in K\langle\langle U_\Sigma\rangle\rangle$ be a recognizable unranked step function. Then there is a partitioning $L_1,\dots,L_m$ of U_Σ where $L_1,\dots,L_m$ are recognizable unranked tree languages, and there are $l_1,\dots,l_m \in K$ such that $S = \sum_{i=1}^m l_i \cdot \mathbb{1}_{L_i}$.

Proof. Note that the class of recognizable unranked tree languages is closed under complement and intersection to form the partition. □

Next we show closure properties of the classes of recognizable unranked step functions and recognizable unranked tree series.

Proposition 3.2. Let $S, S' \in K\langle\langle U_\Sigma \rangle\rangle$.

(1) If S and S' are recognizable, then $S + S'$, $S \odot S'$, and $k \cdot S$ are recognizable for every $k \in K$.

(2) If S and S' are recognizable unranked step functions, then $S + S'$, $S \odot S'$, and $k \cdot S$ are recognizable unranked step functions for every $k \in K$.

(3) Every recognizable unranked step function is a recognizable unranked tree series.

Proof. (1) We can directly generalize the standard constructions from the ranked case, cf. e.g., Lemma 6.4 of [DPV05] for the sum and Corollary 3.9 of [Bor04] or Proposition 5.1 of [BR82] for the Hadamard-product. From an uwta M we obtain the uwta which recognizes $k \cdot S_M$ by multiplying the results of the final weight function with k.

(2) Let $S = \sum_{i=1}^{n} k_i \cdot \mathbb{1}_{L_i}$ and $S' = \sum_{j=1}^{m} k'_j \cdot \mathbb{1}_{L'_j}$ with every L_i, L'_j being a recognizable unranked tree language. Trivially, $S + S'$ is again of this form. Clearly, $S \odot S' = \sum_{i,j} k_i \cdot k'_j \cdot \mathbb{1}_{L_i \cap L'_j}$ and every $L_i \cap L'_j$ is a recognizable unranked tree language. This implies the claim for $S \odot S'$. Clearly, for every $k \in K$, the tree series $k \cdot S = \sum_{i=1}^{n} (k \cdot k_i) \cdot \mathbb{1}_{L_i}$ is also a recognizable unranked step function.

(3) This claim follows directly from the facts that $\mathbb{1}_L$ is recognizable for every recognizable unranked tree language L and (1) and (2). □

A tree series $S \in K\langle\langle T_\Sigma \rangle\rangle$ is a *recognizable step function* if there are an $n \geq 1$ and recognizable tree languages $L_1, \ldots, L_n \subseteq T_\Sigma$ such that $S = \sum_{i=1}^{n} k_i \cdot \mathbb{1}_{L_i}$. It is easy to see that the analogues of Lemma 3.1 and Proposition 3.2 also hold for recognizable step functions.

Example 3.3. Let $\Sigma = \{\alpha, \beta\}$. We consider the set $L \subseteq U_\Sigma$ of all unranked Σ-trees such that for every $t \in L$ and position $w \in \mathrm{pos}(t)$ which is not a leaf, the sequence of successor symbols of w has the form $\alpha^n \beta^k$ with $n \geq 0$ and $k \geq 1$. More precisely, for every $w \in \mathrm{pos}(t)$ which is not a leaf there is $0 \leq n < rk_t(w)$ such that $t(w \cdot i) = \alpha$ for every $1 \leq i \leq n$, and $t(w \cdot j) = \beta$ if $n + 1 \leq j \leq rk_t(w)$.

For $t \in L$ let $\#_{\alpha\alpha}(t)$ be the number of positions w of t such that the first and the second successor of w are labeled by α. More precisely, $\#_{\alpha\alpha}(t) = |\{w \in \mathrm{pos}(t) \mid t(w \cdot 1) = t(w \cdot 2) = \alpha\}|$. Then we define the unranked tree series $S \in \mathit{Trop}\langle\langle U_\Sigma \rangle\rangle$ such that $(S, t) = \#_{\alpha\alpha}(t)$ if $t \in L$, and $(S, t) = \infty$ if $t \in U_\Sigma \setminus L$.

We show that S is recognizable. We construct the uwta $M = (Q, \Sigma, \delta, \gamma)$ over Trop with $Q = \{q_\alpha, q_\beta\}$ as follows. For every $q_\xi \in Q$ and $\zeta \in \Sigma$ we put

$$\delta(q_\xi, \zeta) = \begin{cases} S' & \text{if } \zeta = \xi \\ \widetilde{\infty} & \text{otherwise} \end{cases}$$

where $S' \in Trop\langle\langle Q^*\rangle\rangle$ and $\widetilde{\infty} \in Trop\langle\langle Q^*\rangle\rangle$ are defined for every $w \in Q^*$ by

$$(S', w) = \begin{cases} 1 & \text{if } w \in \{q_\alpha\}^2\{q_\alpha\}^*\{q_\beta\}^+ \\ 0 & \text{if } w \in \{q_\beta\}^* \cup \{q_\alpha\}\{q_\beta\}^+ \\ \infty & \text{otherwise} \end{cases}$$

and $(\widetilde{\infty}, w) = \infty$. Finally, $\gamma(q_\alpha) = \gamma(q_\beta) = 0$.

Now it is clear that for every $t \in U_\Sigma \setminus L$, every run r on t has the weight $wt_M(r) = \infty$. Also, for every $t \in L$ there is exactly one run $r_t \in U_{Q\times\Sigma}$ such that $wt_M(r) \neq \infty$: for every position w of t we have that $r_t(w) = (q_{t(w)}, t(w))$. Then $wt_M(r_t) = \#_{\alpha\alpha}(t)$, showing that $S_M = S$. E.g., for the unranked tree $t = \alpha(\alpha(\alpha,\beta), \alpha(\alpha,\alpha,\beta), \beta)$ we have

position w in t	ε	1	11	12	2	21	22	23	3
$(\delta(q_{t(w)}, t(w)), q_{t(w1)}, \ldots, q_{t(wrk_t(w))})$	1	0	0	0	1	0	0	0	0

Hence $(S,t) = min_{r\in tree^{-1}(t)}\{wt_M(r) + \gamma(r(\varepsilon)_1)\} = wt_M(r_t) = 2$. □

The following is an easy generalization of the corresponding result of [DV06] for recognizable tree series to the unranked setting. For the string case, this result is well-known (cf. e.g. Lemma III.1.2 of [BR88]).

Lemma 3.4. Let Σ be an alphabet, K and L two commutative semirings, $\varphi : K \to L$ a semiring morphism, and $S \in K^{\mathrm{rec}}\langle\langle U_\Sigma\rangle\rangle$. Then the series $\varphi(S) = \varphi \circ S : U_\Sigma \to L$ with $(\varphi(S), t) = \varphi((S,t))$ for every $t \in U_\Sigma$ is recognizable.

Proof. Let M be a uwta over K and Σ which recognizes S. Applying φ to all the weights occuring in M yields a uwta over L and Σ which recognizes $\varphi(S)$. □

Finally, we show that the class of recognizable unranked tree series is closed under relabelings. This will be used in Section 8, but it is also of independant interest. For this, let Σ and Δ be two alphabets and $\tau : \Sigma \to \mathcal{P}(\Delta)$ be a mapping. This mapping is extended to a mapping $\tau' : U_\Sigma \to \mathcal{P}(U_\Delta)$ by defining inductively $\tau'(\sigma(s_1,\ldots,s_k)) = \{\beta(t_1,\ldots,t_k) \,|\, \beta \in \tau(\sigma), t_1 \in \tau'(s_1), \ldots, t_k \in \tau'(s_k)\}$ for every $\sigma \in \Sigma$, $k \geq 0$, and $s_1,\ldots,s_k \in U_\Sigma$. Note that the set $\{s \mid t \in \tau'(s)\})$ is finite for every $t \in U_\Delta$. We will denote τ' also by τ. Next we extend τ to a mapping $\tau : K\langle\langle U_\Sigma\rangle\rangle \to K\langle\langle U_\Delta\rangle\rangle$ by defining $(\tau(S), t) = \sum_{s\in U_\Sigma, t\in\tau(s)}(S,s)$ for every $S \in K\langle\langle U_\Sigma\rangle\rangle$ and $t \in U_\Delta$. In the sequel we call mappings like τ *relabelings.*

Lemma 3.5. Let $S \in K\langle\langle U_\Sigma\rangle\rangle$ and $\tau : K\langle\langle U_\Sigma\rangle\rangle \to K\langle\langle U_\Delta\rangle\rangle$ be a relabeling. If S is recognizable, then $\tau(S)$ is recognizable.

Proof. Let $M = (Q,\Sigma,\delta,\gamma)$ be a uwta over K and $\tau : \Sigma \to \mathcal{P}(\Delta)$. Construct the uwta $M' = (Q,\Delta,\delta',\gamma)$ over K where $(\delta'(q,\beta), q_1\ldots q_k) = \sum_{\sigma\in\Sigma,\beta\in\tau(\sigma)}(\delta(q,\sigma), q_1\ldots q_k)$ for every $\beta \in \Delta$, $k \geq 0$, and $q_1,\ldots,q_k, q \in Q$.

First we note the following auxiliary statement: for every $t \in U_\Delta$ and $r' \in tree^{-1}_{M'}(t)$ the equation

$$\mathrm{wt}_{M'}(r') = \sum_{r\in U_{Q\times\Sigma}, r'\in\tau(r)} \mathrm{wt}_M(r) \tag{1}$$

holds, where τ is extended canonically to a mapping of type $U_{Q\times\Sigma} \to \mathcal{P}(U_{Q\times\Delta})$ by keeping the q-component at every node and relabeling the Σ-component. The proof of Equation 1 is straightforward and thus left to the reader.

Then we can compute as follows for every $t \in U_\Delta$:

$$\begin{aligned}
& (S_{M'}, t) \\
= \;& \textstyle\sum_{r'\in tree^{-1}_{M'}(t)} \mathrm{wt}_{M'}(r') \cdot \gamma(r'(\varepsilon)_1) \\
= \;& \textstyle\sum_{r'\in tree^{-1}_{M'}(t)} \left(\sum_{r\in U_{Q\times\Sigma}, r'\in\tau(r)} \mathrm{wt}_M(r)\right) \cdot \gamma(r'(\varepsilon)_1) \\
= \;& \textstyle\sum_{r'\in tree^{-1}_{M'}(t)} \sum_{r\in U_{Q\times\Sigma}, r'\in\tau(r)} \mathrm{wt}_M(r) \cdot \gamma(r(\varepsilon)_1) \\
= \;& \textstyle\sum_{s\in U_\Sigma, t\in\tau(s)} \sum_{r\in tree^{-1}_{M}(s)} \mathrm{wt}_M(r) \cdot \gamma(r(\varepsilon)_1) \\
= \;& \textstyle\sum_{s\in U_\Sigma, t\in\tau(s)} (S_M, s) \\
= \;& (\tau(S_M), t). \qquad \square
\end{aligned}$$

4 Unranked weighted MSO

In this section, we will introduce our weighted MSO-logic for unranked trees, and we present our main result. For later use and the convenience of the reader, we also recall from [DV06] our weighted MSO-logic for ranked trees and the corresponding equivalence result.

Definition 4.1. The *set* uMSO(K, Σ) *of all formulas of weighted uMSO-logic over* K *and* Σ *on unranked trees* (for short: weighted uMSO-logic) is defined to be the smallest set F such that:

1. F contains all the *atomic formulas* k, $\mathrm{label}_\sigma(x)$, $\mathrm{desc}(x, y)$, $(x \leq y)$, and $x \in X$ and the negations $\neg\mathrm{label}_\sigma(x)$, $\neg\mathrm{desc}(x, y)$, $\neg(x \leq y)$, and $\neg(x \in X)$, and
2. if φ and ψ are in F, then also $\varphi \vee \psi$, $\varphi \wedge \psi$, $\exists x.\varphi$, $\exists X.\varphi$, $\forall x.\varphi$, $\forall X.\varphi$ are in F,

where $k \in K$, $\sigma \in \Sigma$, x, y are first order variables, and X is a second order variable.

We denote by uMSO$^-(\Sigma)$ the fragment of uMSO(K, Σ) obtained by dropping formulas of the form k with $k \in K$; the quantifier-free fragment qf-uMSO$^-(\Sigma)$ is obtained from uMSO$^-(\Sigma)$ by forbidding first and second order quantification.

In order to have appropriate structures available on which formulas with free variables can be interpreted, we extend the alphabet Σ in the usual way. Formally, let $\mathcal{V}$ be a finite set of first and second order variables, then we denote the alphabet $\Sigma \times \{0,1\}^{\mathcal{V}}$ by $\Sigma_{\mathcal{V}}$; thus, an element of $\Sigma \times \{0,1\}^{\mathcal{V}}$ has the form (σ, f) with $\sigma \in \Sigma$ and $f \in \{0,1\}^{\mathcal{V}}$.

An unranked $\Sigma_{\mathcal{V}}$-tree s is *valid* if for every first order variable $x \in \mathcal{V}$, there is exactly one $w \in \mathrm{pos}(s)$ such that $(s(w)_2)(x) = 1$. The subset of $U_{\Sigma_{\mathcal{V}}}$ containing all valid trees is denoted by $U^v_{\Sigma_{\mathcal{V}}}$. We put $\Sigma_\varphi = \Sigma_{\mathrm{Free}(\varphi)}$.

Every valid unranked $\Sigma_{\mathcal{V}}$-tree s corresponds to a pair (t, ρ) where $t \in U_\Sigma$ and ρ is a $(\mathcal{V}, t)$-*assignment*; such an assignment is a function which maps first order variables in $\mathcal{V}$ to elements of $\mathrm{pos}(t)$ and second order variables in $\mathcal{V}$ to subsets of $\mathrm{pos}(t)$. More precisely, we say that s *and* (t, ρ) *correspond to each other* if $\mathrm{pos}(t) = \mathrm{pos}(s)$, t is obtained from s by replacing $s(w)$ by $s(w)_1$ for every $w \in \mathrm{pos}(t)$, and for every first order variable x, second order variable X, and $w \in \mathrm{pos}(s)$, we have that $(s(w)_2)(x) = 1$ iff $\rho(x) = w$, and $(s(w)_2)(X) = 1$ iff $w \in \rho(X)$. In the sequel we will identify a valid $\Sigma_{\mathcal{V}}$-tree with the corresponding pair (t, ρ).

Let s be an arbitrary unranked $\Sigma_{\mathcal{V}}$-tree, x be a first order variable, and $w \in \mathrm{pos}(s)$. Then $s[x \to w]$ is the $\Sigma_{\mathcal{V}\cup\{x\}}$-labeled tree obtained from s by putting $(s[x \to w](v)_2)(x) = 1$ iff $v = w$, for every $v \in \mathrm{pos}(s)$. Similarly, if X is a second order variable and $I \subseteq \mathrm{pos}(s)$, then $s[X \to I]$ is the unranked $\Sigma_{\mathcal{V}\cup\{X\}}$-tree obtained from s by putting $(s[X \to I](v)_2)(X) = 1$ iff $v \in I$, for every $v \in \mathrm{pos}(s)$. If here $s = (t, \rho)$, we also write $s[x \to w] = (t, \rho[x \to w])$ and $s[X \to I] = (t, \rho[X \to I])$.

Definition 4.2. Let $\varphi \in \mathrm{uMSO}(K, \Sigma)$ and $\mathcal{V}$ be a finite set of variables containing $\mathrm{Free}(\varphi)$. The *semantics* of φ is the formal tree series $[\![\varphi]\!]_{\mathcal{V}} \in K\langle\langle U_{\Sigma_{\mathcal{V}}}\rangle\rangle$ defined as follows. If $s \in U_{\Sigma_{\mathcal{V}}}$ is not valid, then we put $([\![\varphi]\!]_{\mathcal{V}}, s) = 0$. Otherwise, we define $([\![\varphi]\!]_{\mathcal{V}}, s) \in K$ inductively as follows where (t, ρ) corresponds to s.

$$
\begin{aligned}
([\![k]\!]_{\mathcal{V}}, s) &= k \\
([\![\mathrm{label}_\sigma(x)]\!]_{\mathcal{V}}, s) &= \begin{cases} 1 & \text{if } t(\rho(x)) = \sigma \\ 0 & \text{otherwise} \end{cases} \\
([\![\mathrm{desc}(x,y)]\!]_{\mathcal{V}}, s) &= \begin{cases} 1 & \text{if there is an } i \text{ such that } \rho(y) = \rho(x)\cdot i \\ 0 & \text{otherwise} \end{cases} \\
([\![x \le y]\!]_{\mathcal{V}}, s) &= \begin{cases} 1 & \text{if } \rho(x) = \rho(y) = \varepsilon \text{ or if there are } w \in \mathrm{pos}(s) \text{ and } i, j \ge 1 \text{ such that} \\ & \rho(x) = w\cdot i, \rho(y) = w \cdot j, \text{ and } i \le j \\ 0 & \text{otherwise} \end{cases} \\
([\![x \in X]\!]_{\mathcal{V}}, s) &= \begin{cases} 1 & \text{if } \rho(x) \in \rho(X) \\ 0 & \text{otherwise} \end{cases} \\
([\![\neg\varphi]\!]_{\mathcal{V}}, s) &= \begin{cases} 1 & \text{if } ([\![\varphi]\!]_{\mathcal{V}}, s) = 0 \\ 0 & \text{if } ([\![\varphi]\!]_{\mathcal{V}}, s) = 1 \end{cases} \\
& \quad \text{if } \varphi \text{ is of the form } \mathrm{label}_\sigma(x),\ \mathrm{desc}(x,y),\ x \le y, \text{ or } x \in X \\
([\![\varphi \vee \psi]\!]_{\mathcal{V}}, s) &= ([\![\varphi]\!]_{\mathcal{V}}, s) + ([\![\psi]\!]_{\mathcal{V}}, s) \\
([\![\varphi \wedge \psi]\!]_{\mathcal{V}}, s) &= ([\![\varphi]\!]_{\mathcal{V}}, s) \cdot ([\![\psi]\!]_{\mathcal{V}}, s) \\
([\![\exists x.\varphi]\!]_{\mathcal{V}}, s) &= \textstyle\sum_{w \in \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, s[x \to w]) \\
([\![\exists X.\varphi]\!]_{\mathcal{V}}, s) &= \textstyle\sum_{I \subseteq \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}, s[X \to I]) \\
([\![\forall x.\varphi]\!]_{\mathcal{V}}, s) &= \textstyle\prod_{w \in \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, s[x \to w]) \\
([\![\forall X.\varphi]\!]_{\mathcal{V}}, s) &= \textstyle\prod_{I \subseteq \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}, s[X \to I])
\end{aligned}
$$

□

We write $[\![\varphi]\!]$ rather than $[\![\varphi]\!]_{\mathrm{Free}(\varphi)}$. An unranked tree series $S \in K\langle\langle U_\Sigma\rangle\rangle$ is called $\mathrm{uMSO}(K, \Sigma)$-*definable* if there is a sentence $\varphi \in \mathrm{uMSO}(K, \Sigma)$ such that $S = [\![\varphi]\!]$. We denote the class of all $\mathrm{uMSO}(K, \Sigma)$-definable unranked tree series by $K^{\mathrm{mso}}\langle\langle U_\Sigma\rangle\rangle$.

As in the string case (Proposition 3.3 of [DG05]) and the (ranked) tree case (Lemma 4.7 of [DV06]) we note that the semantics' $[\![\varphi]\!]_{\mathcal{V}}$ for every $\mathcal{V}$ containing $\mathrm{Free}(\varphi)$ are consistent, which follows by a standard induction on the structure of φ.

Lemma 4.3. Let $\varphi \in \mathrm{uMSO}(K,\Sigma)$ and $\mathcal{V}$ a finite set of variables containing $\mathrm{Free}(\varphi)$. Then, for every $(t,\rho) \in U^v_{\Sigma_\mathcal{V}}$, we have that $(\llbracket\varphi\rrbracket_\mathcal{V},(t,\rho)) = (\llbracket\varphi\rrbracket,(t,\rho|_{\mathrm{Free}(\varphi)}))$.

If $\varphi \in \text{qf-uMSO}^-(\Sigma)$ and $\mathcal{V} \supseteq \mathrm{Free}(\varphi)$, then φ can also be considered as an unranked (unweighted) MSO-formula over Σ with accepted language $L_\mathcal{V}(\varphi)$. However, since φ can contain disjunctions $\psi \vee \psi'$ as subformulas with $(\llbracket\psi\rrbracket_\mathcal{V},s) \neq 0 \neq (\llbracket\psi'\rrbracket_\mathcal{V},s)$ for some $s \in U_{\Sigma_\mathcal{V}}$, possibly $\llbracket\varphi\rrbracket_\mathcal{V} \neq \mathbb{1}_{L_\mathcal{V}(\varphi)}$. Therefore we recall from [DG05, DV06] the two transformations $(.)^+$ and $(.)^-$ on $\text{qf-uMSO}^-(\Sigma)$ in order to disambiguate formulas. For every $\varphi,\psi \in \text{qf-uMSO}^-(\Sigma)$:

1. if φ is atomic or the negation of an atomic formula, then we define $\varphi^+ = \varphi$ and $\varphi^- = \neg\varphi$ with the convention that $\neg\neg\psi$ is ψ,
2. $(\varphi \vee \psi)^+ = \varphi^+ \vee (\varphi^- \wedge \psi^+)$ and $(\varphi \vee \psi)^- = \varphi^- \wedge \psi^-$, and
3. $(\varphi \wedge \psi)^- = \varphi^- \vee (\varphi^+ \wedge \psi^-)$ and $(\varphi \wedge \psi)^+ = \varphi^+ \wedge \psi^+$.

Then, for every $\varphi \in \text{qf-uMSO}^-(\Sigma)$ and $\psi \in \mathrm{uMSO}(K,\Sigma)$, we define $\varphi \to \psi = \varphi^- \vee (\varphi^+ \wedge \psi)$.

Observation 4.4. Let $\varphi \in \text{qf-uMSO}^-(\Sigma)$ and $\mathcal{V}$ be a finite set of variables containing $\mathrm{Free}(\varphi)$. Then $\llbracket\varphi^+\rrbracket_\mathcal{V} = \mathbb{1}_{\mathcal{L}_\mathcal{V}(\varphi)}$ and $\llbracket\varphi^-\rrbracket_\mathcal{V} = \mathbb{1}_{\mathcal{L}_\mathcal{V}(\neg\varphi)}$.

Proof. This claims follows by a straightforward induction on the structure of φ. ☐

Lemma 4.5. Let $\varphi \in \text{qf-uMSO}^-(\Sigma)$, let $\psi \in \mathrm{MSO}(K,\Sigma)$, and let $\mathcal{V}$ be a finite set of variables containing $\mathrm{Free}(\varphi) \cup \mathrm{Free}(\psi)$. Then $\llbracket\varphi \to \psi\rrbracket_\mathcal{V} = \llbracket\varphi^-\rrbracket_\mathcal{V} + \llbracket\varphi^+\rrbracket_\mathcal{V} \odot \llbracket\psi\rrbracket_\mathcal{V}$. Moreover, if $\llbracket\psi\rrbracket$ is a recognizable step function, then so is $\llbracket\varphi \to \psi\rrbracket$.

Proof. The equation claimed is immediate from the definition of the macro $\varphi \to \psi$. The final statement follows from this equation, Observation 4.4 and Proposition 3.2. ☐

Next we define some useful macros and show an example.

x is a leaf	$\mathrm{leaf}(x)$	$=$	$\forall y.\neg\mathrm{desc}(x,y)$
x and y are the same nodes:	$(x = y)$	$=$	$(x \leq y) \wedge (y \leq x)$
$x \leq y$ and x and y are different:	$(x < y)$	$=$	$(x \leq y) \wedge \neg(y \leq x)$
y is oldest child of x:	$\mathrm{oldest.child}(x,y)$	$=$	$\mathrm{desc}(x,y) \wedge \forall z.\mathrm{desc}(x,z) \to (y \leq z)$
y is youngest child of x:	$\mathrm{youngest.child}(x,y)$	$=$	$\mathrm{desc}(x,y) \wedge \forall z.\mathrm{desc}(x,z) \to (z \leq y)$
x is the next older sibling of y:	$\mathrm{next}(x,y)$	$=$	$(x < y) \wedge \forall u.((x < u) \to (y \leq u))$

Example 4.6. Reconsider the set $L \subseteq U_\Sigma$ and the tree series S of Example 3.3. Here we will specify S by means of a formula φ of weighted uMSO-logic over $Trop$ and Σ. Let

$$\varphi = \mathrm{member}_L \wedge \mathrm{count}$$

where

$$\mathrm{member}_L = \forall x.\Big(\mathrm{leaf}(x) \vee \exists y.\big(\mathrm{desc}(x,y) \wedge \text{alpha–beta}(y)\big)\Big),$$

$$\text{alpha–beta}(y) = \forall u.\Big(\big((u < y) \to \mathrm{label}_\alpha(u)\big) \wedge \big((y \leq u) \to \mathrm{label}_\beta(u)\big)\Big)$$

and

$$\begin{aligned}\mathrm{count} \;=\; \forall x.\; &\Big(\mathrm{leaf}(x) \wedge 0\Big)\\ &\vee\Big(\exists y.\mathrm{oldest.child}(x,y) \wedge \mathrm{youngest.child}(x,y) \wedge 0\Big)\\ &\vee\Big(\exists y_1.\exists y_2.\big(\mathrm{oldest.child}(x,y_1) \wedge \mathrm{label}_\beta(y_1) \wedge 0\big)\\ &\qquad\vee\big(\mathrm{oldest.child}(x,y_1) \wedge \mathrm{label}_\alpha(y_1) \wedge \mathrm{next}(y_1,y_2) \wedge \mathrm{label}_\beta(y_2) \wedge 0\big)\\ &\qquad\vee\big(\mathrm{oldest.child}(x,y_1) \wedge \mathrm{label}_\alpha(y_1) \wedge \mathrm{next}(y_1,y_2) \wedge \mathrm{label}_\alpha(y_2) \wedge 1\big)\Big)\end{aligned}$$

Now it is easy to see that $([\![\text{member}_L]\!], t) = 0$ if $t \in L$ and $([\![\text{member}_L]\!], t) = \infty$ if $t \notin L$. Moreover, for every $t \in L$ we have $([\![\text{count}]\!], t) = \#_{\alpha\alpha}(t)$. Hence $S = [\![\varphi]\!]$. □

A formula $\varphi \in \text{uMSO}(K, \Sigma)$ is *restricted*, if it does not contain a universal set quantification of the form $\forall X.\psi$, and whenever φ contains a universal first order quantification of the form $\forall x.\psi$, then $[\![\psi]\!]$ is a recognizable unranked step function. The *set of all restricted formulas of* $\text{uMSO}(K, \Sigma)$ is denoted by $\text{uRMSO}(K, \Sigma)$. The *set of all restricted existential formulas of* $\text{uMSO}(K, \Sigma)$, denoted by $\text{uREMSO}(K, \Sigma)$, is the set of all restricted formulas of $\text{uMSO}(K, \Sigma)$ of the form $\varphi = \exists X_1, \ldots, X_n.\psi$ with ψ containing no set quantification. The set of all unranked tree series $S \in K\langle\langle U_\Sigma \rangle\rangle$ which are definable by some sentence in $\text{uRMSO}(K, \Sigma)$ (resp. in $\text{uREMSO}(K, \Sigma)$) is denoted by $K^{\text{rmso}}\langle\langle U_\Sigma \rangle\rangle$ (resp. $K^{\text{remso}}\langle\langle U_\Sigma \rangle\rangle$).

Now we can formulate our first main theorem of this paper. It will follow directly from Theorems 6.6 and 7.6.

Theorem 4.7. *Let K be any commutative semiring and Σ an alphabet. Then $K^{\text{rec}}\langle\langle U_\Sigma \rangle\rangle = K^{\text{rmso}}\langle\langle U_\Sigma \rangle\rangle = K^{\text{remso}}\langle\langle U_\Sigma \rangle\rangle$.*

Finally, we indicate that the weighted MSO-logic over a ranked alphabet Σ and a commutative semiring K on trees [DV06] can be obtained as a special case of the unranked weighted MSO-logic.

Definition 4.8. The *set* $\text{MSO}(K, \Sigma)$ *of all formulas of weighted MSO-logic over K and Σ on trees* (for short: weighted MSO-logic) is defined to be the smallest set F such that:

1. F contains all the *atomic formulas* k, $\text{label}_\sigma(x)$, $\text{edge}_i(x, y)$, and $x \in X$ and the negations $\neg\text{label}_\sigma(x)$, $\neg\text{edge}_i(x, y)$, and $\neg(x \in X)$, and
2. if φ, ψ are in F, then also $\varphi \vee \psi$, $\varphi \wedge \psi$, $\exists x.\varphi$, $\exists X.\varphi$, $\forall x.\varphi$, and $\forall X.\varphi$ are in F,

where $k \in K$, $\sigma \in \Sigma$, x, y are first order variables, $1 \leq i \leq \max_\Sigma$, and X is a second order variable.

Obviously, apart from the atomic formula $\text{edge}_i(x, y)$ and its negation, the set $\text{MSO}(K, \Sigma)$ is a subset of $\text{uMSO}(K, \Sigma)$. However, for each $1 \leq i \leq \max_\Sigma$ we can express $\text{edge}_i(x, y)$ in $\text{uMSO}(K, \Sigma)$ as follows:

$$\exists y_1. \ldots \exists y_{i-1}.\text{oldest.child}(x, y_1) \wedge \text{next}(y_1, y_2) \wedge \ldots \wedge \text{next}(y_{i-2}, y_{i-1}) \wedge \text{next}(y_{i-1}, y).$$

The negation of $\text{edge}_i(x, y)$ can be expressed as follows:

$$\neg\text{desc}(x, y) \vee \bigvee_{j=1}^{i-1} \text{edge}_j(x, y) \vee \Big(\exists z.\text{edge}_{i+1}(x, z) \wedge (z \leq y)\Big).$$

These formulas of $\text{uMSO}(K, \Sigma)$ take on only 0 or 1 as values. Using them as macros, we can view the complete set $\text{MSO}(K, \Sigma)$ as a subset of $\text{uMSO}(K, \Sigma)$. Then, clearly, the semantics as defined in Definition 4.2 also applies to the set $\text{MSO}(K, \Sigma)$ and, in fact, it coincides with the semantics given in Definition 4.2 of [DV06].

Conversely, given a ranked alphabet (Σ, rk), we can also express the unranked formulas $\mathrm{desc}(x, y)$, $(x \le y)$, and their negations in $\mathrm{MSO}(K, \Sigma)$ as follows:

$$
\begin{aligned}
\mathrm{desc}(x,y): &\quad \bigvee_{1\le i\le \max_\Sigma} \mathrm{edge}_i(x,y)\\
\neg\mathrm{desc}(x,y): &\quad \bigwedge_{1\le i\le \max_\Sigma} \neg\mathrm{edge}_i(x,y)\\
(x\le y): &\quad (\mathrm{root}(x)\wedge\mathrm{root}(y)) \vee \exists z. \bigvee_{1\le i\le j\le \max_\Sigma} (\mathrm{edge}_i(z,x)\wedge \mathrm{edge}_j(z,y))\\
\neg(x\le y): &\quad \forall z. \left(\bigvee_{1\le i,j\le \max_\Sigma} \mathrm{edge}_i(z,x)\wedge \mathrm{edge}_j(z,y)\right) \to \bigvee_{1\le j< i\le \max_\Sigma} \mathrm{edge}_i(z,x)\wedge \mathrm{edge}_j(z,y)
\end{aligned}
$$

using the abbreviation $\mathrm{root}(x) = \forall z. \bigwedge_{1\le i\le \max_\Sigma} \neg\mathrm{edge}_i(z, x)$. Again, on ranked trees, the semantics of these formulas coincide.

Analogously to unranked tree series, a tree series $S \in K\langle\langle T_\Sigma\rangle\rangle$ is called $\mathrm{MSO}(K, \Sigma)$-definable, if $S = [\![\varphi]\!]$ for some sentence $\varphi \in \mathrm{MSO}(K, \Sigma)$. By referring to recognizable step functions, we define *restricted* and *restricted existential* formulas of $\mathrm{MSO}(K, \Sigma)$, and correspondingly we obtain the classes $K^{\mathrm{mso}}\langle\langle T_\Sigma\rangle\rangle$, $K^{\mathrm{rmso}}\langle\langle T_\Sigma\rangle\rangle$ and $K^{\mathrm{remso}}\langle\langle T_\Sigma\rangle\rangle$. We cite the main result of [DV06].

Theorem 4.9. *(Theorem 5.1 of [DV06]). Let Σ be a ranked alphabet and K any commutative semiring. Then $K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle = K^{\mathrm{rmso}}\langle\langle T_\Sigma\rangle\rangle = K^{\mathrm{remso}}\langle\langle T_\Sigma\rangle\rangle$.*

By the discussion above, it is clear that Theorem 4.9 is a special case of Theorem 4.7. However, here we will deduce Theorem 4.7 as a consequence of Theorem 4.9. The essential technique will be to encode unranked trees as ranked trees, see Section 5 and Figure 1.

Finally, we recall a corollary from [DV06] which we will use in Section 8 and which shows that the formulas for the simulation of a recognizable tree series can be chosen to have a particular form.

Corollary 4.10. *(cf. Corollary 5.12 of [DV06]) Let $S \in K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle$. Then $S = [\![\xi]\!]$ for a formula $\xi \in \mathrm{REMSO}(K, \Sigma)$ of the form $\xi = \exists X_1 \ldots \exists X_n.\forall x.\theta$ where θ contains the formula* $\mathrm{root}(z)$ *and only first order existential quantifications, and $[\![\theta]\!]$ is a recognizable step function.*

5 Encoding

According to Section 4.2 of [Nev02], we encode an unranked tree as a ranked tree. The intuitive idea can be grasped from Figure 2. Given a graph s which is an unranked tree with additional arrows from every node to its first successor node and from every successor node to its next younger sibling node; moreover, arrows to #-labeled nodes are added at the end of the sequence of descendants of a node and at nodes without descendants. Then, the unranked tree t is reobtained from s by dropping the arrows and the #-labeled nodes. The encoding $enc(t)$ is obtained from s by dropping the (ordinary) edges from s, replacing the arrows by edges, and marking the root symbol with a prime.

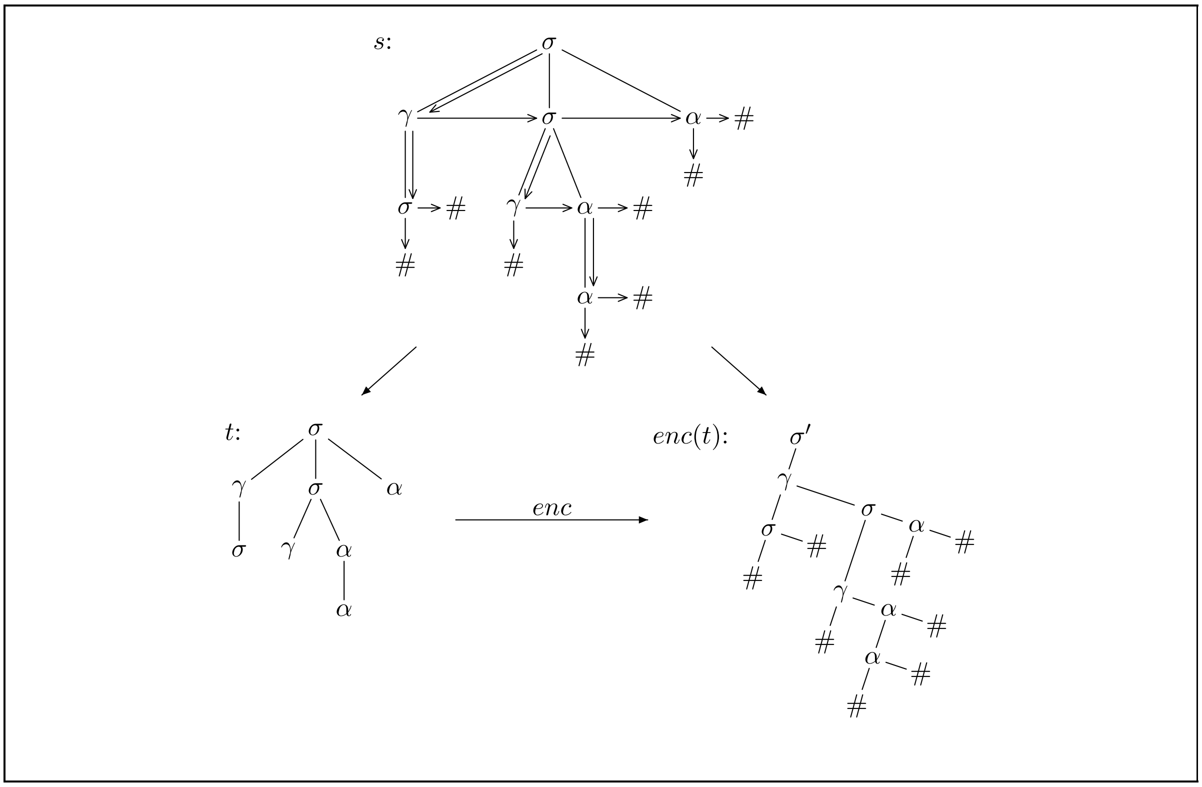

Figure 2: An unranked tree t and its encoding $enc(t)$.

The formal definition of the encoding requires the concept of hedge. A *hedge of unranked* Σ-*trees* is a sequence $(t_1, \ldots, t_n)$ of unranked Σ-trees $t_1, \ldots, t_n \in U_\Sigma$ for some $n \geq 0$ (cf. [BW98, BMW01]). We denote the set of all hedges of unranked Σ-trees by H_Σ. Note that for every $\sigma \in \Sigma$ and $(t_1, \ldots, t_n) \in H_\Sigma$, the expression $\sigma(t_1, \ldots, t_n) \in U_\Sigma$, and vice versa. We identify a hedge $(t) \in H_\Sigma$ with the unranked tree $t \in U_\Sigma$. Thus we have that $U_\Sigma \subseteq H_\Sigma$.

Let Σ be an alphabet. We define the ranked alphabet $(enc(\Sigma), rk)$ by $enc(\Sigma) = \Sigma \cup \Sigma' \cup \{\#\}$ with $\Sigma' = \{\sigma' \mid \sigma \in \Sigma\}$, $rk(\sigma) = 2$ and $rk(\sigma') = 1$ for every $\sigma \in \Sigma$, and $rk(\#) = 0$. Next we map every unranked tree $t \in U_\Sigma$ to a tree $enc(t) \in T_{enc(\Sigma)}$ of which the root is labeled by some element in Σ' and every other node is labeled by some element in $\Sigma \cup \{\#\}$.

The *binary* Σ-*encoding* is the mapping $enc : U_\Sigma \to \Sigma'(T_{\Sigma \cup \{\#\}})$ defined for every $\sigma(t_1, \ldots, t_n) \in U_\Sigma$ with $\sigma \in \Sigma$, $n \geq 0$, and $t_1, \ldots, t_n \in U_\Sigma$ as follows:

$$enc(\sigma(t_1, \ldots, t_n)) = \sigma'(henc((t_1, \ldots, t_n)))$$

and the mapping $henc : H_\Sigma \to T_{enc(\Sigma)}$ is defined for every $n \geq 0$ and $t_1, \ldots, t_n \in U_\Sigma$ as follows:

- if $n \geq 1$ and $t_1 = \sigma(s_1, \ldots, s_k)$ for some $\sigma \in \Sigma$, $k \geq 0$, and $s_1, \ldots, s_k \in U_\Sigma$, then

$$henc(t_1, t_2, \ldots, t_n)) = \sigma(henc((s_1, \ldots, s_k)), henc((t_2, \ldots, t_n)));$$

- if $n = 0$, then $henc(()) = \#$.

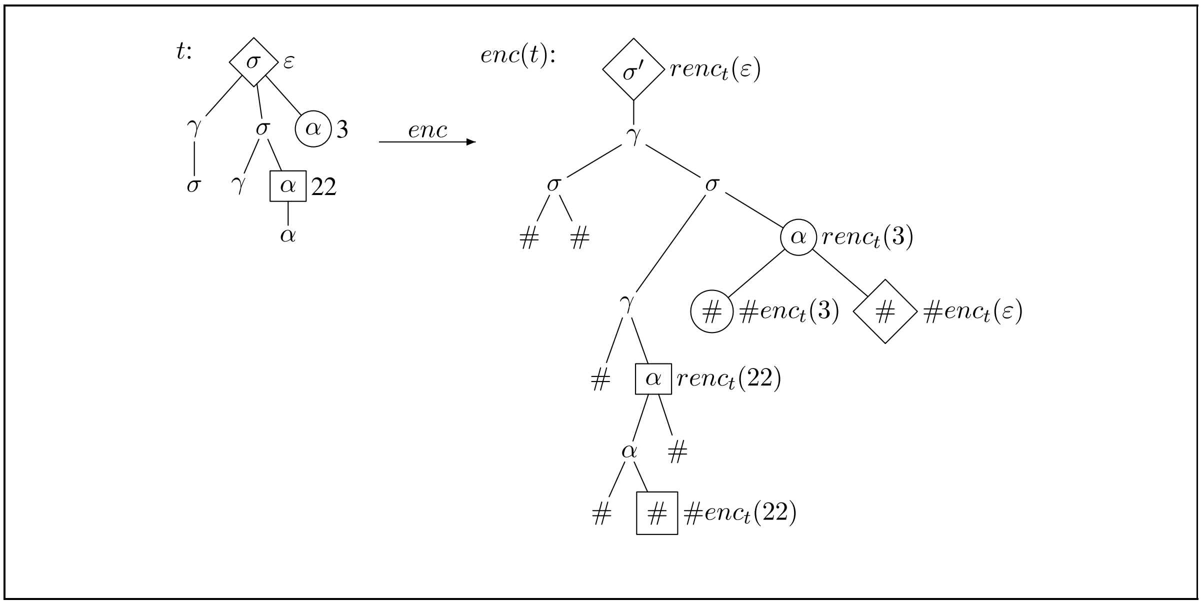

Figure 3: An unranked tree t with the three designated positions ε, 22, and 3; and the encoding $enc(t)$ with the encodings of these designated positions as regular positions (i.e., ε, 1212, and 122, respectively) and as #-positions (i.e., 1222, 121212, and 1221, respectively).

For example, $enc(\sigma(\gamma(\alpha), \beta)) = \sigma'(\gamma(\alpha(\#, \#), \beta(\#, \#)))$. In fact, symbols from Σ' only occur at the root of elements in $range(enc)$.

It is clear that enc is bijective. Even more: for every position of $t \in U_\Sigma$ we can associate in a schematic way exactly two positions of $enc(t)$ as follows (cf. Figure 3). For this we define the *encoding of positions of* t *as regular positions* as the mapping $renc_t : \mathrm{pos}(t) \to \mathrm{pos}(enc(t))$ and the *encoding of positions of* t *as #-positions* as the mapping $\#enc_t : \mathrm{pos}(t) \to \mathrm{pos}(enc(t))$ such that for every position $i_1 \cdot \ldots \cdot i_n \in \mathrm{pos}(t)$ [Küh06]:

$$renc_t(i_1 \cdot \ldots \cdot i_n) = 1 \cdot \underbrace{2 \cdot \ldots \cdot 2}_{i_1 - 1} \cdot 1 \cdot \underbrace{2 \cdot \ldots \cdot 2}_{i_2 - 1} \cdot \ldots \cdot 1 \cdot \underbrace{2 \cdot \ldots \cdot 2}_{i_n - 1}$$

and

$$\#enc_t(i_1 \cdot \ldots \cdot i_n) = renc_t(i_1 \cdot \ldots \cdot i_n) \cdot 1 \cdot \underbrace{2 \cdot \ldots \cdot 2}_{rk_t(i_1 \ldots i_n)}.$$

Next we list some useful properties of the mappings $renc_t$ and $\#enc_t$; they follow directly from the definition of these functions.

Observation 5.1. Let $t \in U_\Sigma$.

1. $renc_t$ and $\#enc_t$ are injective.
2. $range(renc_t) \cup range(\#enc_t) = \mathrm{pos}(enc(t))$ and $range(renc_t) \cap range(\#enc_t) = \emptyset$.
3. For every $w \in \mathrm{pos}(t)$, we have $enc(t)(\#enc_t(w)) = \#$ and if $w \neq \varepsilon$, then $enc(t)(renc_t(w)) = t(w)$; moreover, $enc(t)(renc_t(\varepsilon)) = t(\varepsilon)'$.

4. For every $w, u \in \mathrm{pos}(t)$ such that $l = rk(u) \geq 1$, we have: $w = u \cdot l$ iff $renc_t(w) = renc_t(u) \cdot 1 \cdot \underbrace{2 \cdot \ldots \cdot 2}_{l-1}$.

5. For every $w, u \in \mathrm{pos}(t)$, we have: [there are $v \in \mathrm{pos}(t)$ and $i \geq 1$ such that $w = v \cdot i$ and $u = v \cdot (i+1)$] iff $renc_t(u) = renc_t(w) \cdot 2$.

The fact that the mappings *enc* and enc^{-1} preserve recognizability of unranked tree languages is a folklore result. In [Suc02] it appears in Theorem 2.3 (with a slightly different definition of *enc*). In [Nev02] it is displayed in Proposition 1 (with the same encoding as we use it). A formal proof can be obtained from Lemmas 6.1 and 7.5 of the present paper using the Boolean semiring as K. Also, clearly $\Sigma'(T_{\Sigma \cup \{\#\}}) \subseteq T_{enc(\Sigma)}$ is recognizable.

Proposition 5.2. Let $L \subseteq U_\Sigma$. Then L is recognizable iff $enc(L) \subseteq T_{enc(\Sigma)}$ is recognizable. Furthermore, $L' \subseteq T_{enc(\Sigma)}$ is recognizable iff $enc^{-1}(L') \subseteq U_\Sigma$ is recognizable.

6 Recognizable unranked tree series are uRMSO-definable

In this section we want to prove that every recognizable unranked tree series $S \in K\langle\langle U_\Sigma \rangle\rangle$ is also uREMSO(K, Σ)-definable (cf. Theorem 6.6 in Section 6.3). The proof of this theorem is based on Lemmas 6.1 and 6.5 which, in their turn, will be proved in Sections 6.1 and 6.2, respectively (cf. again Figure 1).

6.1 From unranked weighted tree automata to weighted tree automata

Let $M = (Q, \Sigma, \delta, \gamma)$ be a uwta over K. In the next lemma we simulate such an M by a binary wta M' over $enc(\Sigma)$ and K. Roughly speaking, the idea is that M' takes over the state set Q of M and, being in state $q \in Q$ at a σ-labeled position w of some given input tree $enc(t)$, additionally simulates the behaviour of the weighted finite automaton $A_{q,\sigma}$ (which accepts the power series $\delta(q, \sigma) \in K\langle\langle Q^* \rangle\rangle$) on the right growing comb in the first successor of w. The right part of Figure 4 shows the relationship between a run r of M on some unranked tree t and the run $enc^r(r)$ of M' on $enc(t)$ which corresponds to the run $(p'_1, q_1, p'_2)(p'_2, q_2, p'_3) \ldots (p'_k, q_k, p'_{k+1})$ of $A_{q,\sigma}$ on $q_1 q_2 \ldots q_k$.

Lemma 6.1. For every uwta M over Σ and K there is a binary wta M' over $enc(\Sigma)$ and K such that $(S_M, t) = (S_{M'}, enc(t))$ for every $t \in U_\Sigma$.

Proof. Let $M = (Q, \Sigma, \delta, \gamma)$ be a uwta over K. For every $q \in Q$ and $\sigma \in \Sigma$, let $A_{q,\sigma} = (P_{q,\sigma}, \lambda_{q,\sigma}, \mu_{q,\sigma}, \nu_{q,\sigma})$ be the wsa over K and Q such that $\|A_{q,\sigma}\| = \delta(q, \sigma)$. We may assume that the state sets of the wsa's are mutually disjoint.

We construct the binary wta $M' = (Q', enc(\Sigma), \delta', \gamma')$ over K as follows:

- $Q' = Q \cup \left(Q \times \bigcup_{q \in Q, \sigma \in \Sigma} P_{q,\sigma}\right) \cup \left(\bigcup_{q \in Q, \sigma \in \Sigma} P_{q,\sigma}\right)$
(intuitively, the states from Q are only used by M' at the root of $enc(\Sigma)$-trees (note that for every unranked Σ-tree $\sigma(s_1, \ldots, s_k)$ we have that $enc(\sigma(s_1, \ldots, s_k)) = \sigma'(henc((s_1, \ldots, s_k))))$; if an interior position w of an $enc(\Sigma)$-tree is labeled by a state

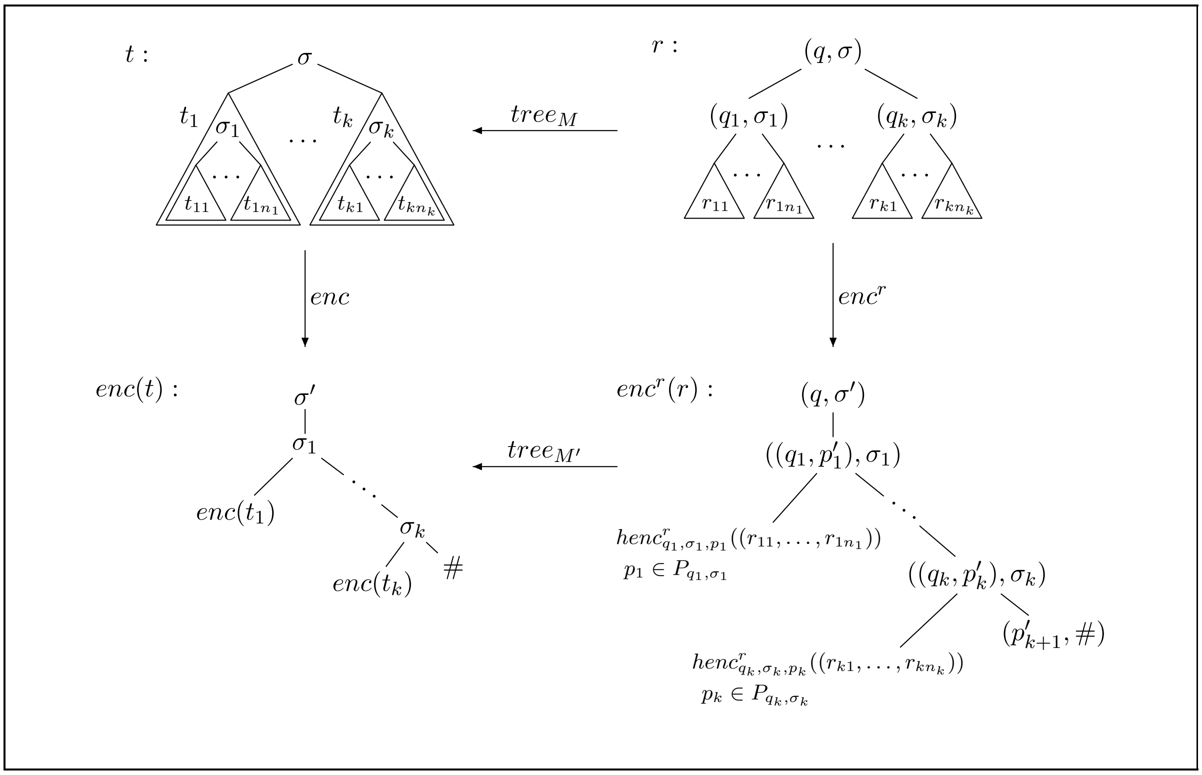

Figure 4: Relationship between runs of M and M'.

of the form (q,p) (with $q \in Q$ and $p \in P_{q_0,\sigma_0}$ for some $q_0 \in Q$ and $\sigma_0 \in \Sigma$), then the corresponding position in the unranked Σ-tree is in state q, and q is the next input symbol which has to be processed by the wsa A_{q_0,σ_0} in state p; finally, states of the form $p \in P_{q,\sigma}$ are used to terminate the computations of $A_{q,\sigma}$ at $\#$),

- δ' is defined for every $\xi \in enc(\Sigma)$ and $w \in (Q')^*$ as follows (according to the three different subsets of Q'):

  1. for every $q \in Q$ we define (cf. Figure 5)

$$\big(\delta'(q,\xi),w\big) = \begin{cases} \lambda_{q,\sigma}(p) & \text{if } \exists\sigma \in \Sigma : \xi = \sigma' \text{ and } \exists q_1 \in Q, p \in P_{q,\sigma} : \text{(a) } w = (q_1,p) \text{ or (b) } w = p \\ 0 & \text{otherwise} \end{cases}$$

  2. for every $q, q_0 \in Q$, $\sigma_0 \in \Sigma$, and $p \in P_{q_0,\sigma_0}$, we define (cf. Figure 6)

$$\big(\delta'((q,p),\xi),w\big) = \begin{cases} \lambda_{q,\xi}(p_1)\cdot\mu_{q_0,\sigma_0}(q)_{p,p'} & \text{if } \xi \in \Sigma \text{ and } \exists q_1,q_2 \in Q, p_1 \in P_{q,\xi}, p' \in P_{q_0,\sigma_0} : \\ & \text{(a) } w = (q_1,p_1)(q_2,p') \text{ or} \\ & \text{(b) } w = (q_1,p_1)p' \text{ or} \\ & \text{(c) } w = p_1(q_2,p') \text{ or} \\ & \text{(d) } w = p_1p' \\ 0 & \text{otherwise} \end{cases}$$

3. for every $q_0 \in Q$, $\sigma_0 \in \Sigma$, and $p \in P_{q_0,\sigma_0}$, we define

$$\big(\delta'(p,\xi),w\big) = \begin{cases} (\nu_{q_0,\sigma_0})(p) & \text{if } \xi = \# \text{ and } w = \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

- the final weight function $\gamma' : Q' \to K$ is defined for every $q' \in Q'$ by

$$\gamma'(q') = \begin{cases} \gamma(q') & \text{if } q' \in Q \\ 0 & \text{otherwise.} \end{cases}$$

In order to prove that $(S_M, t) = (S_{M'}, enc(t))$, we encode runs of M as follows (cf. Figure 4). We define the mapping $enc^r : U_{Q\times\Sigma} \to \mathcal{P}(T_{Q'\times enc(\Sigma)})$ for every $(q,\sigma) \in Q \times \Sigma$ and $r_1, \ldots, r_k \in U_{Q\times\Sigma}$ by

$$enc^r((q,\sigma)(r_1,\ldots,r_k)) = \bigcup_{p \in P_{q,\sigma}} (q,\sigma')(henc^r_{q,\sigma,p}((r_1,\ldots,r_k)))$$

(cf. the definition of $\xi(L_1,\ldots,L_k)$ in Section 2.1). For every $q_0 \in Q$, $\sigma_0 \in \Sigma$, and $p \in P_{q_0,\sigma_0}$ we define the mapping $henc^r_{q_0,\sigma_0,p} : H_{Q\times\Sigma} \to \mathcal{P}(T_{Q'\times enc(\Sigma)})$ for every $n \geq 0$ and $t_1,\ldots,t_n \in U_{Q\times\Sigma}$ as follows:

– if $n \geq 1$ and $t_1 = (q,\sigma)(r_1,\ldots,r_k)$ for some $(q,\sigma) \in Q \times \Sigma$, $k \geq 0$, and $r_1,\ldots,r_k \in U_{Q\times\Sigma}$, then

$$henc^r_{q_0,\sigma_0,p}((q,\sigma)(r_1,\ldots,r_k),t_2,\ldots,t_n) =$$
$$\bigcup_{p_1 \in P_{q,\sigma}, p' \in P_{q_0,\sigma_0}} ((q,p),\sigma)(henc^r_{q,\sigma,p_1}((r_1,\ldots,r_k)), henc^r_{q_0,\sigma_0,p'}((t_2,\ldots,t_n)))$$

– if $n = 0$, then

$$henc^r_{q_0,\sigma_0,p}((t_1,\ldots,t_n)) = \{(p,\#)\}.$$

Obviously, for every $r \in U_{Q\times\Sigma}$ we have that $enc(tree_M(r)) = tree_{M'}(enc^r(r))$, and also for every $t \in U_\Sigma$ we have that $enc^r(tree_M^{-1}(t)) \subseteq tree_{M'}^{-1}(enc(t))$. By definition of δ' it follows that (†) $wt_{M'}(r) = 0$ for every $r \in tree_{M'}^{-1}(enc(t)) \setminus enc^r(tree_M^{-1}(t))$.

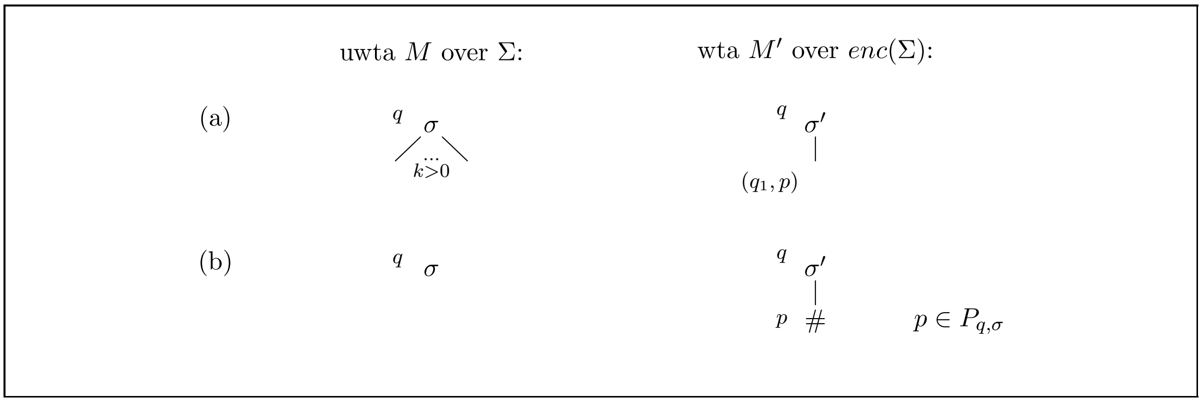

Figure 5: Construction of rules (part 1).

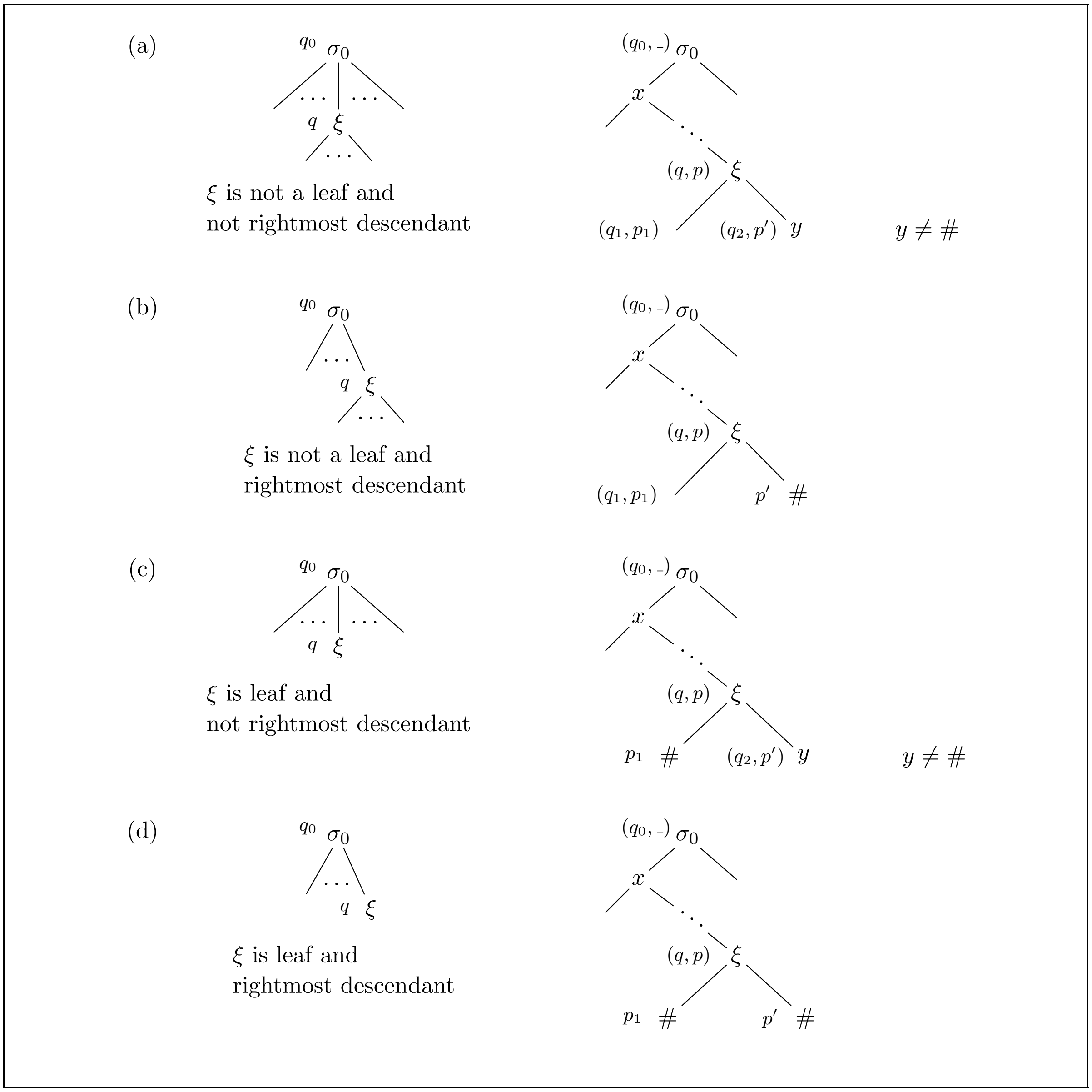

Figure 6: Construction of rules (part 2).

Next we prove that for every $t \in U_\Sigma$ the equation $(S_M, t) = (S_{M'}, enc(t))$ holds; the corresponding calculation needs an auxiliary statement (††) which we will prove afterwards. Now let $t = \sigma(t_1, \ldots, t_k)$ for some $k \geq 0$, $\sigma \in \Sigma$, and $t_1, \ldots, t_k \in U_\Sigma$. Then:

$$
\begin{aligned}
& (S_M, \sigma(t_1, \ldots, t_k)) \\
= \;& \textstyle\sum_{r \in tree_M^{-1}(t)} wt_M(r) \cdot \gamma(r(\varepsilon)_1) \\
= \;& \textstyle\sum_{q \in Q} \sum_{r_i \in tree_M^{-1}(t_i)} wt_M((q,\sigma)(r_1, \ldots, r_k)) \cdot \gamma(q) \\
=^{\dagger\dagger} \;& \textstyle\sum_{q \in Q} \sum_{r_i \in tree_M^{-1}(t_i)} \big(\sum_{p \in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \sum_{r' \in henc^r_{q,\sigma,p}((r_1,\ldots,r_k))} wt_{M'}(r') \big) \cdot \gamma(q) \\
= \;& \textstyle\sum_{q \in Q} \sum_{r_i \in tree_M^{-1}(t_i)} \big(\sum_{p \in P_{q,\sigma}} \sum_{r' \in henc^r_{q,\sigma,p}((r_1,\ldots,r_k))} \lambda_{q,\sigma}(p) \cdot wt_{M'}(r') \big) \cdot \gamma(q) \\
= \;& \textstyle\sum_{q \in Q} \sum_{r_i \in tree_M^{-1}(t_i)} \big(\sum_{p \in P_{q,\sigma}} \sum_{r' \in henc^r_{q,\sigma,p}((r_1,\ldots,r_k))} (\delta'(q,\sigma'), w) \cdot wt_{M'}(r') \big) \cdot \gamma(q) \\
& \text{and there is a } q_1 \in Q \text{ such that } w = (q_1, p) \text{ or } w = p \\
= \;& \textstyle\sum_{q \in Q} \sum_{r_i \in tree_M^{-1}(t_i)} \big(\sum_{p \in P_{q,\sigma}} \sum_{r' \in henc^r_{q,\sigma,p}((r_1,\ldots,r_k))} wt_{M'}((q,\sigma')(r')) \big) \cdot \gamma(q) \\
= \;& \textstyle\sum_{q \in Q} \sum_{r_i \in tree_M^{-1}(t_i)} \big(\sum_{r \in enc^r((q,\sigma)(r_1,\ldots,r_k))} wt_{M'}(r) \big) \cdot \gamma(q) \\
= \;& \textstyle\sum_{r \in enc^r(tree_M^{-1}(t))} wt_{M'}(r) \cdot \gamma(r(\varepsilon)_1) \\
= \;& \textstyle\sum_{r \in tree_{M'}^{-1}(enc(t))} wt_{M'}(r) \cdot \gamma(r(\varepsilon)_1) \\
& \text{because } enc^r(tree_M^{-1}(t)) \subseteq tree_{M'}^{-1}(enc(t)) \text{ and by } (\dagger) \\
= \;& (S_{M'}, enc(t))
\end{aligned}
$$

Now let us prove equation (††): for every $r \in U_{Q\times\Sigma}$ if there are $k \geq 0$, $r_1 \ldots, r_k \in U_{Q\times\Sigma}$, $q \in Q$, and $\sigma \in \Sigma$ such that $r = (q,\sigma)(r_1, \ldots, r_k)$, then

$$wt_M((q,\sigma)(r_1, \ldots, r_k)) = \sum_{p \in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \sum_{r' \in henc^r_{q,\sigma,p}((r_1,\ldots,r_k))} wt_{M'}(r').$$

We prove (††) by induction on r.

$\underline{r = (q,\sigma)}$: $wt_M((q,\sigma)) = (\delta(q,\sigma), \varepsilon) = \sum_{p \in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \nu_{q,\sigma}(p) = \sum_{p \in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot wt_{M'}((p, \#))$ $= \sum_{p \in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \sum_{r' \in henc^r_{q,\sigma,p}(())} wt_{M'}(r')$.

$\underline{r = (q,\sigma)(r_1, \ldots, r_k) \text{ with } k \geq 1}$: Then, for every $1 \leq i \leq k$, let r_i have the form $r_i = (q_i, \sigma_i)(r_{i1}, \ldots, r_{ik_i})$ for some $k_i \geq 0$, $q_i \in Q$, $\sigma_i \in \Sigma$, and $r_{i1}, \ldots, r_{ik_i} \in U_{Q\times\Sigma}$. Then

$$
\begin{aligned}
& wt_M((q,\sigma)(r_1, \ldots, r_k)) \\
= \;& (\delta(q,\sigma), q_1 \ldots q_k) \cdot \textstyle\prod_{1 \leq i \leq k} wt_M(r_i) \\
= \;& (\|A_{q,\sigma}\|, q_1 \ldots q_k) \cdot \textstyle\prod_M \\
& \text{where } \textstyle\prod_M \text{ abbreviates } \prod_{1 \leq i \leq k} wt_M(r_i) \\
= \;& \Big(\textstyle\sum_{p,p' \in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \mu_{q,\sigma}(q_1 \ldots q_k)_{p,p'} \cdot \nu_{q,\sigma}(p') \Big) \cdot \textstyle\prod_M \\
= \;& \Big(\textstyle\sum_{p \in P_{q,\sigma}} \sum_{p_1,\ldots,p_k \in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \mu_{q,\sigma}(q_1)_{p,p_1} \cdot \mu_{q,\sigma}(q_2)_{p_1,p_2} \cdot \ldots \cdot \mu_{q,\sigma}(q_k)_{p_{k-1},p_k} \cdot \nu_{q,\sigma}(p_k) \Big) \cdot \textstyle\prod_M \\
= \;& \Big(\textstyle\sum_{p \in P_{q,\sigma}} \sum_{p_1,\ldots,p_k \in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \mu_{q,\sigma}(q_1)_{p,p_1} \cdot \mu_{q,\sigma}(q_2)_{p_1,p_2} \cdot \ldots \cdot \mu_{q,\sigma}(q_k)_{p_{k-1},p_k} \cdot \nu_{q,\sigma}(p_k) \Big) \cdot \\
& \textstyle\prod_{1 \leq i \leq k} \Big(\sum_{\overline{p_i} \in P_{q_i,\sigma_i}} \lambda_{q_i,\sigma_i}(\overline{p_i}) \cdot \sum_{r'_i \in henc^r_{q_i,\sigma_i,\overline{p_i}}((r_{i1},\ldots,r_{ik_i}))} wt_{M'}(r'_i) \Big) \\
& \text{(by induction hypothesis)}
\end{aligned}
$$

$$= \sum_{p\in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \sum_{p_1,\ldots,p_k\in P_{q,\sigma}} \sum_{\substack{\forall i:1\le i\le k:\ \overline{p_i}\in P_{q_i,\sigma_i} \text{ and}\\ r'_i\in henc^r_{q_i,\sigma_i,\overline{p_i}}((r_{i1},\ldots,r_{ik_i}))}}$$

$$\lambda_{q_1,\sigma_1}(\overline{p_1}) \cdot \mu_{q,\sigma}(q_1)_{p,p_1} \cdot \prod_{2\le i\le k} \lambda_{q_i,\sigma_i}(\overline{p_i}) \cdot \mu_{q,\sigma}(q_i)_{p_{i-1},p_i} \cdot \nu_{q,\sigma}(p_k) \cdot \prod_{1\le i\le k} wt_{M'}(r'_i)$$

$$= \sum_{p\in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \sum_{p_1,\ldots,p_k\in P_{q,\sigma}} \sum_{\substack{\forall i:1\le i\le k:\overline{p_i}\in P_{q_i,\sigma_i} \text{ and}\\ r'_i\in henc^r_{q_i,\sigma_i,\overline{p_i}}((r_{i1},\ldots,r_{ik_i}))}}$$

$$wt_{M'}(((q_1,p),\sigma_1)(r'_1,((q_2,p_1),\sigma_2)(r'_2,\ldots((q_k,p_{k-1}),\sigma_k)(r'_k,(p_k,\#))\ldots)))$$

$$= \sum_{p\in P_{q,\sigma}} \lambda_{q,\sigma}(p) \cdot \sum_{r'\in henc^r_{q,\sigma,p}((r_1,\ldots,r_k))} wt_{M'}(r')$$

□

As a consequence of this translation, we obtain that the equivalence problem for uwta's is decidable, provided that K is a computable field.

Corollary 6.2. Let K be a computable field. It is decidable for two effectively given unranked weighted tree automata M and N, whether $S_M = S_N$ or not.

Proof. Given two uwta's M and N, construct the corresponding wta's M' and N' as in Lemma 6.1. The tree language $L = enc(U_\Sigma) = \Sigma'(T_{\Sigma\cup\{\#\}})$ is clearly recognizable, hence $\mathbb{1}_L$ is a recognizable tree series. Since the class of recognizable tree series is closed under the Hadamard product (cf. Corollary 3.9 of [Bor04]), it follows that the series $S_{M'} \odot \mathbb{1}_L$ and $S_{N'} \odot \mathbb{1}_L$ are recognizable. By Lemma 6.1, we have $S_M = S_N$ iff $S_{M'} \odot \mathbb{1}_L = S_{N'} \odot \mathbb{1}_L$. Using the assumption that K is a computable field, by Theorem 4.2 of [Sei90], also cf. [Boz91], the latter equality is decidable. □

6.2 From weighted MSO to weighted uMSO

Here we wish to construct for every sentence $\psi \in \mathrm{MSO}(K, enc(\Sigma))$ a sentence $\psi' \in \mathrm{uMSO}(K,\Sigma)$ such that $([\![\psi]\!], enc(t)) = ([\![\psi']\!], t)$ for every $t \in U_\Sigma$. Note that the #-positions of $enc(t)$ have to be considered in the definition of ψ'. We proceed inductively. Consider a formula $\psi \in \mathrm{MSO}(K, enc(\Sigma))$ and two sets V and $\mathcal{V}$ such that $V \cup \mathrm{Free}(\psi) \subseteq \mathcal{V}$ and no variable of V occurs bound in ψ. We will define a formula $\psi_V \in \mathrm{uMSO}(K,\Sigma)$ and an assignment ρ_V such that, for every $(t,\rho) \in U^v_{\Sigma_\mathcal{V}}$, the semantics of ψ on $(enc(t), \rho_V)$ is the same as the semantics of ψ_V on (t,ρ). The idea for obtaining the assignment ρ_V from ρ is easy (cf. Figure 7): if a variable $\xi \in \mathcal{V}$ is not in V (respectively, is in V), then ρ_V maps it to the corresponding regular position (respectively, #-position). Formally, we define the $(\Sigma, V, \mathcal{V})$-*encoding* to be the mapping $enc_V : U^v_{\Sigma_\mathcal{V}} \to T^v_{enc(\Sigma_\mathcal{V})}$ defined for every $s = (t,\rho) \in U^v_{\Sigma_\mathcal{V}}$ by

$$enc_V(s) = (enc(t), \rho_V)$$

where $\rho_V : \mathcal{V} \to \mathrm{pos}(enc(t))$ is defined by

$$\rho_V(\xi) = \begin{cases} renc_t(\rho(\xi)) & \text{if } \xi \in \mathcal{V} \setminus V \\ \#enc_t(\rho(\xi)) & \text{if } \xi \in V. \end{cases}$$

We may assume that for every formula $\varphi \in \mathrm{uMSO}(K,\Sigma)$ and every (first-order or second-order) variable ξ there is at most one quantification over ξ in φ and no variable occurs both free and

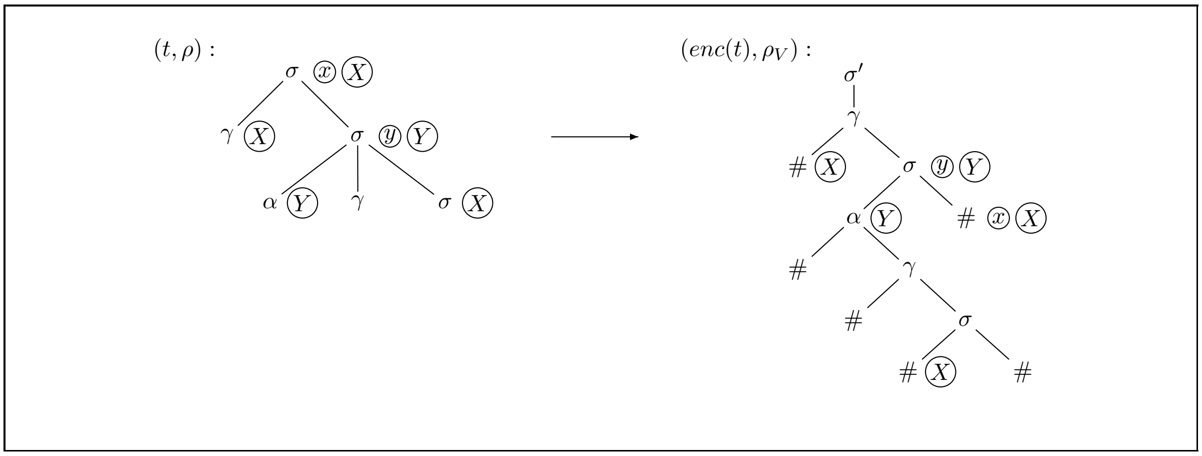

Figure 7: A tree (t, ρ) and the tree $(enc(t), \rho_V)$ for $V = \{x, X\}$ and $\mathcal{V} = \{x, y, X, Y\}$.

bound in φ. Clearly, this syntactic form can always be achieved by appropriate renaming of variables; and this does not change the semantics of sentences.

Definition 6.3. Let $\psi \in \mathrm{MSO}(K, enc(\Sigma))$ and $V \cup \mathrm{Free}(\psi) \subseteq \mathcal{V}$ such that no variable of V occurs bound in ψ. We define the formula $\psi_V \in \mathrm{uMSO}(K, \Sigma)$ by induction on the structure of ψ as follows (where the necessity of V becomes clear from cases 7-10):

<u>Case 1.</u> $\psi = k$ with $k \in K$: $\quad \psi_V = k$

<u>Case 2.</u> $\psi = \mathrm{label}_\sigma(x)$ and $\psi' = \neg\mathrm{label}_\sigma(x)$:

if $\sigma = \#$ and $x \in V$, then:	$\psi_V = 1$	$\psi'_V = 0$
if $\sigma = \#$ and $x \notin V$, then:	$\psi_V = 0$	$\psi'_V = 1$
if $\sigma \neq \#$ and $x \in V$, then:	$\psi_V = 0$	$\psi'_V = 1$
if $\sigma \neq \#$ and $x \notin V$, then:	$\psi_V = \mathrm{label}_\sigma(x)$	$\psi'_V = \neg\mathrm{label}_\sigma(x)$

<u>Case 3.</u> $\psi = \mathrm{edge}_i(x, y)$ and $\psi' = \neg\mathrm{edge}_i(x, y)$ (for some cases cf. Figures 8 and 9):

if $x \in V$, then:		$\psi_V = 0$ and $\psi'_V = 1$
if $x \notin V$, $y \in V$, then	if $i = 1$, then	$\psi_V = (x = y) \wedge \mathrm{leaf}(y)$
	and	$\psi'_V = (x = y) \rightarrow \exists u.(\mathrm{oldest.child}(y, u))$
	if $i = 2$, then	$\psi_V = \mathrm{youngest.child}(y, x)$
	and	$\psi'_V = \mathrm{desc}(y, x) \rightarrow \exists z.\mathrm{youngest.child}(y, z) \wedge \neg(z \leq x)$
if $x, y \notin V$, then	if $i = 1$, then	$\psi_V = \mathrm{oldest.child}(x, y)$
	and	$\psi'_V = \mathrm{desc}(x, y) \rightarrow \exists z.\mathrm{oldest.child}(x, z) \wedge \neg(y \leq z)$
	if $i = 2$, then	$\psi_V = \mathrm{next}(x, y)$
	and	$\psi'_V = \exists z.\mathrm{next}(x, z) \wedge ((y \leq z) \rightarrow \neg(z \leq y))$

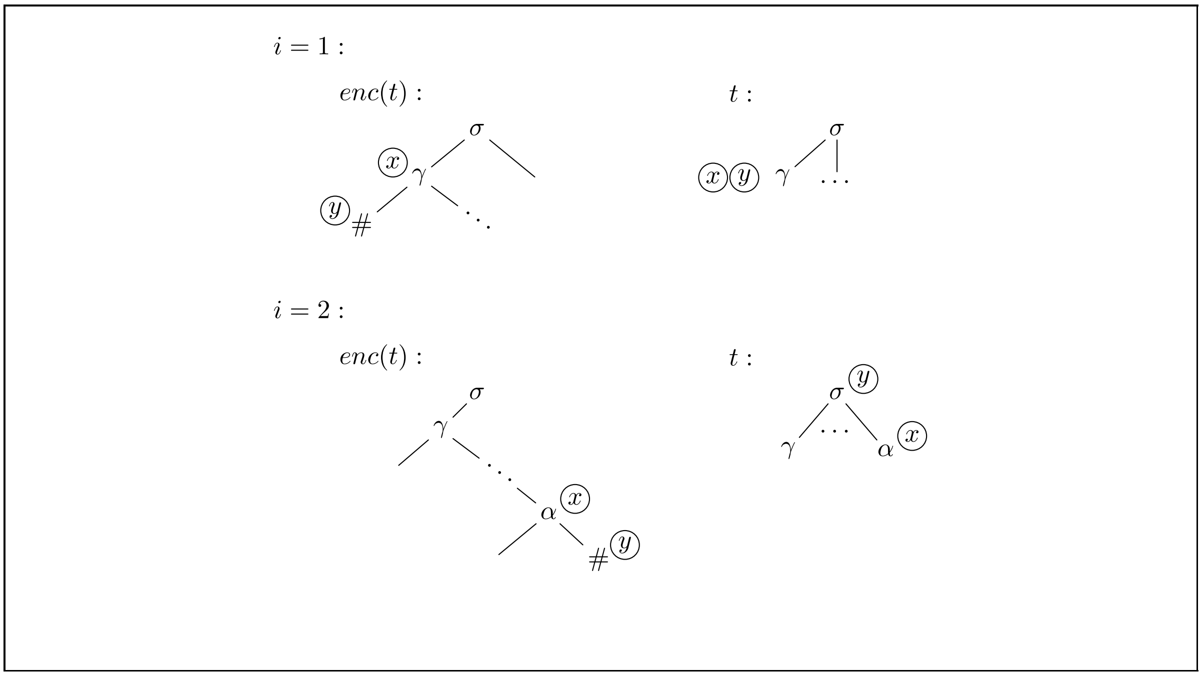

Figure 8: Encoding of $\mathrm{edge}_i(x,y)$ for $x \notin V$ and $y \in V$.

Case 4. $\psi = (x \in X)$ and $\psi' = \neg(x \in X)$:

if $x \in V$ and $X \in V$, then	$\psi_V = (x \in X)$	$\psi'_V = \neg(x \in X)$
if $x \in V$ and $X \notin V$, then	$\psi_V = 0$	$\psi'_V = 1$
if $x \notin V$ and $X \in V$, then	$\psi_V = 0$	$\psi'_V = 1$
if $x \notin V$ and $X \notin V$, then	$\psi_V = (x \in X)$	$\psi'_V = \neg(x \in X)$

Case 5. $\psi = \psi_1 \vee \psi_2$: $\psi_V = (\psi_1)_V \vee (\psi_2)_V$

Case 6. $\psi = \psi_1 \wedge \psi_2$: $\psi_V = (\psi_1)_V \wedge (\psi_2)_V$

Case 7. $\psi = \exists x.\varphi$: $\psi_V = \exists x.(\varphi_V \vee \varphi_{V\cup\{x\}})$

Case 8. $\psi = \forall x.\varphi$: $\psi_V = \forall x.(\varphi_V \wedge \varphi_{V\cup\{x\}})$ (note that K is commutative)

Case 9. $\psi = \exists X.\varphi$: $\psi_V = \exists Xr.\exists X\#.(\varphi')_{V\cup\{X\#\}}$ where φ' is obtained from φ by replacing every occurrence of $(x \in X)$ by $(x \in Xr) \vee (x \in X\#)$.

Case 10. $\psi = \forall X.\varphi$: $\psi_V = \forall Xr.\forall X\#.(\varphi')_{V\cup\{X\#\}}$ where φ' is obtained from φ by replacing every occurrence of $(x \in X)$ by $(x \in Xr) \vee (x \in X\#)$.

In the next lemma we will use some abbreviations. For this, we assume that $n \geq 0$. Also let $\psi \in \mathrm{MSO}(K, enc(\Sigma))$ and $V, \mathcal{V}$ be two sets of variables such that $V \cup \mathrm{Free}(\psi) \subseteq \mathcal{V}$ and no variable of V occurs bound in ψ. Moreover, let $s = (t, \rho) \in U^{v}_{\Sigma_{\mathcal{V}}}$. Finally, let $\mathcal{I} = \{I_i \mid i \in [n]\}$ and $\mathcal{J} = \{J_i \mid i \in [n]\}$ be two families of subsets of $\mathrm{pos}(t)$, and let $\mathcal{I}' = \{I'_i \mid i \in [n]\}$ and $\mathcal{J}' = \{J'_i \mid i \in [n]\}$ be two families of subsets of $\mathrm{pos}(enc(t))$.

Then consider the families of set variables $\mathcal{Y} = \{Y_i \mid i \in [n]\}$, $\mathcal{Y}r = \{Y_i r \mid i \in [n]\}$, and $\mathcal{Y}\# = \{Y_i\# \mid i \in [n]\}$.

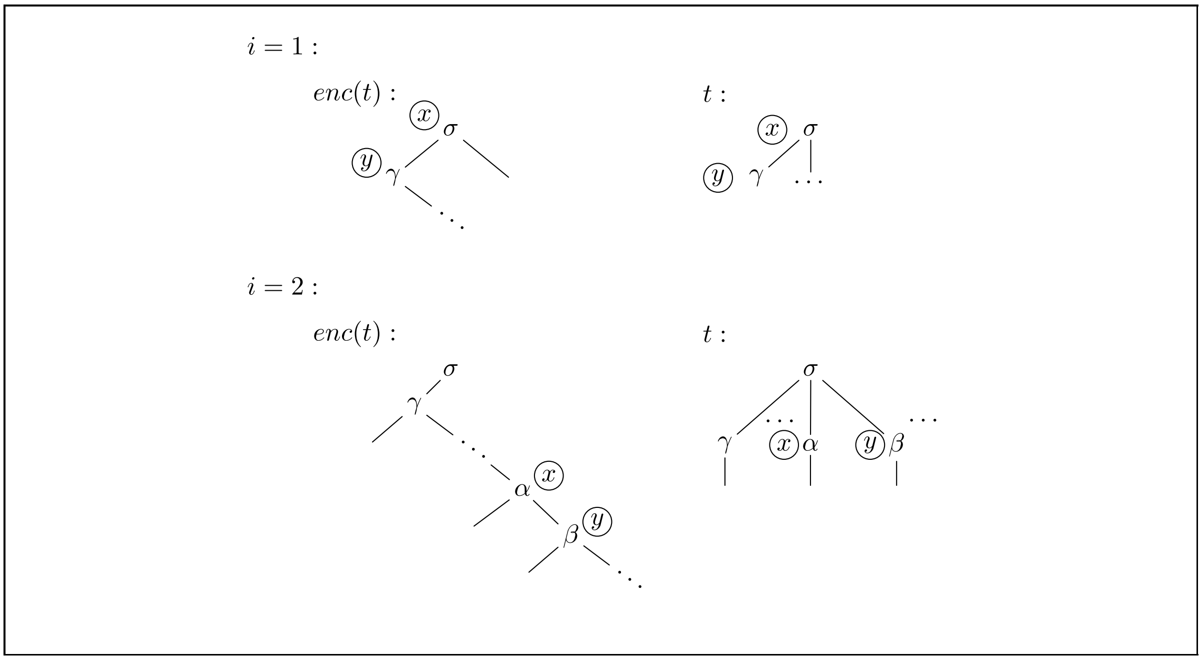

Figure 9: Encoding of $\text{edge}_i(x, y)$ for $x, y \notin V$.

We denote by $\psi^{\mathcal{Y}}$ the formula which is obtained from ψ by replacing every occurrence of $(x \in Y_i)$ with Y_i free by $(x \in Y_i r) \vee (x \in Y_i \#)$, and every occurrence of $\neg(x \in Y_i)$ by $\neg(x \in Y_i r) \wedge \neg(x \in Y_i \#)$.

We denote by $\rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}]$ the assignment which is obtained from ρ by updating $Y_i r$ by I_i and $Y_i\#$ by J_i for every $i \in [n]$. Also we denote by $\rho_V[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}']$ the assignment which is obtained from ρ_V by updating Y_i by $I'_i \cup J'_i$ for every $i \in [n]$.

Lemma 6.4. Let K be commutative. Let $\psi \in \text{MSO}(K, enc(\Sigma))$ and $V, \mathcal{V}$ be two sets of variables such that $V \cup \text{Free}(\psi) \subseteq \mathcal{V}$ and no variable of V occurs bound in ψ. Moreover, let $s = (t, \rho) \in U^v_{\Sigma_{\mathcal{V}}}$, $n \geq 0$, and $\mathcal{Y} = \{Y_i \mid i \in [n]\}$. Finally, let $\mathcal{I} = \{I_i \mid i \in [n]\}$ and $\mathcal{J} = \{J_i \mid i \in [n]\}$ be two families of subsets of $\text{pos}(t)$. We put $I'_i = renc_t(I_i)$ and $J'_i = \#enc_t(J_i)$ for every $1 \leq i \leq n$ and let $\mathcal{I}' = \{I'_i \mid i \in [n]\}$ and $\mathcal{J}' = \{J'_i \mid i \in [n]\}$.

Then $([\![\psi]\!]_{\mathcal{V}\cup\mathcal{Y}}, (enc(t), \rho_V[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}'])) = ([\![(\psi^{\mathcal{Y}})_{V\cup\mathcal{Y}\#}]\!]_{\mathcal{V}\cup\mathcal{Y}r\cup\mathcal{Y}\#}, (t, \rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}])$ where $(\psi^{\mathcal{Y}})_{V\cup\mathcal{Y}\#}$ is defined in Definition 6.3.

Proof. We first note that, if no variable Y_i occurs in ψ, then the claim of the lemma reduces to the equality $([\![\psi]\!]_{\mathcal{V}}, (enc(t), \rho_V)) = ([\![\psi_V]\!]_{\mathcal{V}}, (t, \rho))$. Moreover, we may assume that no variable of $\mathcal{Y}$ occurs as a bound variable in ψ. We now show the claim of the lemma by induction on ψ.

<u>Case 1.</u> $\psi = k \in K$: $([\![k]\!]_{\mathcal{V}}, (enc(t), \rho_V)) = k = ([\![k_V]\!]_{\mathcal{V}}, (t, \rho))$.

<u>Case 2.</u> $\psi = \text{label}_\sigma(x)$ for some $\sigma \in enc(\Sigma)$: We distinguish four cases. (i) $\sigma = \#$ and $x \in V$: We have $enc(t)(\rho_V(x)) = enc(t)(\#enc_t(\rho(x))) = \#$, so $([\![\text{label}_\#(x)]\!]_{\mathcal{V}}, (enc(t), \rho_V)) = 1$ $= ([\![1]\!]_{\mathcal{V}}, (t, \rho)) = ([\![\text{label}_\#(x)_V]\!]_{\mathcal{V}}, (t, \rho))$.

(ii)$\sigma = \#$ and $x \notin V$: Here $enc(t)(\rho_V(x)) = enc(t)(renc_t(\rho(x))) \neq \#$, hence $([\![\text{label}_\#(x)]\!]_{\mathcal{V}}, (enc(t), \rho_V)) = 0 = ([\![0]\!]_{\mathcal{V}}, (t, \rho)) = ([\![\text{label}_\#(x)_V]\!]_{\mathcal{V}}, (t, \rho))$.

(iii) $\sigma \neq \#$ and $x \in V$: Now $enc(t)(\rho_V(x)) = enc(t)(\#enc_t(\rho(x))) = \# \neq \sigma$, thus $([\![\text{label}_\sigma(x)]\!]_\mathcal{V}, (enc(t), \rho_V)) = 0 = ([\![0]\!]_\mathcal{V}, (t,\rho))= ([\![\text{label}_\sigma(x)_V]\!]_\mathcal{V}, (t,\rho))$.

(iv) $\sigma \neq \#$ and $x \notin V$: Observe $enc(t)(\rho_V(x)) = enc(t)(renc_t(\rho(x))) = t(\rho(x))$. Hence $([\![\text{label}_\sigma(x)]\!]_\mathcal{V}, (enc(t), \rho_V)) = \left\{ \begin{array}{ll} 1 & \text{if } t(\rho(x)) = \sigma \\ 0 & \text{otherwise} \end{array} \right\} = ([\![\text{label}_\sigma(x)_V]\!]_\mathcal{V}, (t,\rho))$.

Case 3. $\psi = \text{edge}_i(x,y)$: Observe that $[\![\psi_V]\!]$ only takes $0,1$ as values, as does $[\![\psi]\!]$. We have $([\![\text{edge}_i(x,y)]\!]_\mathcal{V}, (enc(t), \rho_V)) = \left\{ \begin{array}{ll} 1 & \text{if } \rho_V(y) = \rho_V(x) \cdot i \\ 0 & \text{otherwise} \end{array} \right.$.

Now we distinguish three cases.

(i) $x \in V$: Then $\rho_V(x) = \#enc_t(\rho(x))$; since this is a leaf in $enc(t)$, there is no i such that $\#enc_t(\rho(x)) \cdot i \in \text{pos}(enc(t))$. Hence $([\![\text{edge}_i(x,y)]\!]_\mathcal{V}, (enc(t), \rho_V)) = 0 = ([\![0]\!]_\mathcal{V}, (t,\rho)) = ([\![\text{edge}_i(x,y)_V]\!]_\mathcal{V}, (t,\rho))$.

(ii) $x \notin V$, $y \in V$: Here we have $\rho_V(y) = \#enc_t(\rho(y)) = renc(\rho(y)) \cdot 1 \cdot 2^{\text{rk}(\rho(y))}$ and $\rho_V(x) = renc_t(\rho(x))$. Now let $i = 1$. Then $\rho_V(y) = \rho(x) \cdot 1$ iff $\text{rk}(\rho(y)) = 0$ and $renc_t(\rho(y)) = renc_t(\rho(x))$ iff $\text{rk}(\rho(y)) = 0$ and $\rho(y) = \rho(x)$. Hence $([\![\text{edge}_i(x,y)]\!]_\mathcal{V}, (enc(t), \rho_V)) = ([\![(x = y) \wedge \text{leaf}(y)]\!]_\mathcal{V}, (t,\rho)) = ([\![\text{edge}_i(x,y)_V]\!]_\mathcal{V}, (t,\rho))$. Now let $i = 2$. Here $\rho_V(y) = \rho(x) \cdot 2$ iff $\rho(x) = \rho(y) \cdot \text{rk}(\rho(y))$ iff $\rho(x)$ is the youngest child of $\rho(y)$ in t. Hence $([\![\text{edge}_i(x,y)]\!]_\mathcal{V}, (enc(t), \rho_V)) = ([\![\text{youngest.child}(y,x)]\!]_\mathcal{V}, (t,\rho)) = ([\![\text{edge}_i(x,y)_V]\!]_\mathcal{V}, (t,\rho))$.

(iii) $x, y \notin V$: Now $\rho_V(y) = renc_t(\rho(y))$ and $\rho_V(x) = renc_t(\rho(x))$. First let $i = 1$. Then $\rho_V(y) = \rho_V(x) \cdot 1$ iff $\rho(y) = \rho(x) \cdot 1$ iff $\rho(y)$ is the oldest child of $\rho(x)$ in t. Hence $([\![\text{edge}_i(x,y)]\!]_\mathcal{V}, (enc(t), \rho_V)) = ([\![\text{oldest.child}(x,y)]\!]_\mathcal{V}, (t,\rho)) = ([\![\text{edge}_i(x,y)_V]\!]_\mathcal{V}, (t,\rho))$. Now let $i = 2$. Then $\rho_V(y) = \rho_V(x) \cdot 2$ iff $\rho(x), \rho(y)$ are children of the same node in t with $\rho(y)$ being the direct sucessor of $\rho(x)$. Hence $([\![\text{edge}_i(x,y)]\!]_\mathcal{V}, (enc(t), \rho_V)) = ([\![\text{next}(x,y)]\!]_\mathcal{V}, (t,\rho)) = ([\![\text{edge}_i(x,y)_V]\!]_\mathcal{V}, (t,\rho))$.

Case 4. $(x \in X)$ with $X \notin \mathcal{Y}$: We distinguish four cases.

(i) $x \in V$ and $X \in V$: Clearly, $\rho_V(x) \in \rho_V(X)$ iff $\#enc_t(\rho(x)) \in \#enc_t(\rho(X))$ iff $\rho(x) \in \rho(X)$. Hence $([\![(x \in X)]\!]_\mathcal{V}, (enc(t), \rho_V)) = ([\![(x \in X)_V]\!], (t,\rho))$.

(ii) $x \in V$ and $X \notin V$: First note that $\#enc_t(\rho(x)) \notin renc_t(\rho(X))$. Thus $([\![(x \in X)]\!]_\mathcal{V}, (enc(t), \rho_V)) = 0$. On the other hand $(x \in X)_V = 0$.

(iii) $x \notin V$ and $X \in V$: Now note that $renc_t(\rho(x)) \notin \#enc_t(\rho(X))$ and then argue as in the previous case.

(iv) $x \notin V$ and $X \notin V$: This case is proved in the same way as case (i) except that $\#enc_t$ has to be replaced by $renc_t$.

Case 5. $\varphi = \neg\psi$ where $\psi = \text{label}_\sigma(x)$ or $\psi = \text{edge}_i(x,y)$ or $\psi = (x \in X)$: By our definitions, $[\![\varphi]\!]_\mathcal{V}$ and $[\![\varphi_V]\!]_\mathcal{V}$ take values only in $\{0,1\}$. As before, we have to distinguish whether $\sigma = \#$ or $\sigma \neq \#$ and whether the variables x, y resp. X belong to $\mathcal{V}$ or not. In each case we have defined φ_V as the logical negation of ψ_V. This implies the result.

Case 6. $\varphi = (x \in X)$ with $X = Y_j$ for some $j \in [n]$: On the left hand side of the claim we have

$$([\![(x \in Y_j)]\!]_{\mathcal{V}\cup\mathcal{Y}}, (enc(t), \rho_V[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}'])) = \left\{ \begin{array}{ll} 1 & \text{if } \rho_V(x) \in I'_j \cup J'_j \\ 0 & \text{otherwise} \end{array} \right.$$

On the right hand side of the claim we can calculate as follows:

$$
\begin{array}{rl}
& ([\![((x \in Y_j)^{\mathcal{Y}})_{V \cup \mathcal{Y}\#}]\!]_{\mathcal{V} \cup \mathcal{Y}r \cup \mathcal{Y}\#}, (t, \rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}])) \\
= & ([\![((x \in Y_j r) \vee (x \in Y_j\#))_{V \cup \{Y_j\#\}}]\!]_{\mathcal{V} \cup \{Y_j r, Y_j\#\}}, (t, \rho[Y_j r \to I_j, Y_j\# \to J_j])) \\
& \text{(using the consistency lemma)} \\
= & ([\![(x \in Y_j r)_{V \cup \{Y_j\#\}} \vee (x \in Y_j\#)_{V \cup \{Y_j\#\}}]\!]_{\mathcal{V} \cup \{Y_j r, Y_j\#\}}, (t, \rho[Y_j r \to I_j, Y_j\# \to J_j])). \\
= & a + b \\
& \text{where } a = ([\![(x \in Y_j r)_{V \cup \{Y_j\#\}}]\!]_{\mathcal{V} \cup \{Y_j r, Y_j\#\}}, (t, \rho[Y_j r \to I_j, Y_j\# \to J_j])) \\
& \text{and } b = [\![(x \in Y_j\#)_{V \cup \{Y_j\#\}}]\!]_{\mathcal{V} \cup \{Y_j r, Y_j\#\}}, (t, \rho[Y_j r \to I_j, Y_j\# \to J_j])).
\end{array}
$$

Clearly, $a = \left\{ \begin{array}{ll} 1 & \text{if } x \notin V \text{ and } \rho(x) \in I_j \\ 0 & \text{otherwise} \end{array} \right\}$ and $b = \left\{ \begin{array}{ll} 1 & \text{if } x \in V \text{ and } \rho(x) \in J_j \\ 0 & \text{otherwise} \end{array} \right\}$ and hence

$$
a + b = \left\{ \begin{array}{ll} 1 & \text{if } (x \notin V \text{ and } \rho(x) \in I_j) \text{ or } (x \in V \text{ and } \rho(x) \in J_j) \\ 0 & \text{otherwise} \end{array} \right.
$$

Since it is easy to see that the two conditions $[\rho_V(x) \in I'_j \cup J'_j]$ and $[(x \notin V$ and $\rho(x) \in I_j)$ or $(x \in V$ and $\rho(x) \in J_j)]$ are equivalent, the claim of the lemma follows for $\varphi = (x \in X)$.

Case 7. $\varphi = \neg(x \in X)$ with $X = Y_j$ for some $j \in [n]$: This case is handled in exactly the same way as the previous one except that we have to deal with $(\neg(x \in Y_j))^{\mathcal{Y}} = \neg(x \in Y_j r) \wedge \neg(x \in Y_j\#)$, and we have to exchange the roles of 0 and 1 and replace addition by multiplication.

Case 8. $\varphi = \varphi_1 \vee \varphi_2$ or $\varphi = \varphi_1 \wedge \varphi_2$: The case of disjunction follows directly from the definitions of the expressions $([\![\varphi_1 \vee \varphi_2]\!]_{\mathcal{V} \cup \mathcal{Y}}, (enc(t), \rho_V[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}'])$ and $([\![(\varphi_1^{\mathcal{Y}} \vee \varphi_2^{\mathcal{Y}})_{V \cup \mathcal{Y}\#}]\!]_{\mathcal{V} \cup \mathcal{Y}r \cup \mathcal{Y}\#}, (t, \rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}])$ by using the induction hypothesis, and correspondingly for the conjunction.

Case 9. $\varphi = \exists y.\psi$: By our general assumption, $y \notin \text{Free}(\varphi)$ and so $y \notin V$, because $V \subseteq \text{Free}(\varphi)$. Then we can compute as follows:

$$
\begin{array}{rl}
& ([\![\exists y.\psi]\!]_{\mathcal{V} \cup \mathcal{Y}}, (enc(t), \rho_V[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}'])) \\
= & \sum_{w' \in \text{pos}(enc(t))} ([\![\psi]\!]_{\mathcal{V} \cup \mathcal{Y} \cup \{y\}}, (enc(t), \rho_V[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}', y \to w'])) \\
= & \sum_{w' \in range(renc_t)} ([\![\psi]\!]_{\mathcal{V} \cup \mathcal{Y} \cup \{y\}}, (enc(t), (\rho_V[y \to w'])[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}'])) + \\
& \sum_{w' \in range(\#enc_t)} ([\![\psi]\!]_{\mathcal{V} \cup \mathcal{Y} \cup \{y\}}, (enc(t), (\rho_V[y \to w'])[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}'])) \\
= & \sum_{w' \in range(renc_t)} ([\![\psi]\!]_{\mathcal{V} \cup \mathcal{Y} \cup \{y\}}, (enc(t), (\underbrace{\rho[y \to renc_t^{-1}(w')]}_{\rho'})_V[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}'])) + \\
& \sum_{w' \in range(\#enc_t)} ([\![\psi]\!]_{\mathcal{V} \cup \mathcal{Y} \cup \{y\}}, (enc(t), (\underbrace{\rho[y \to \#enc_t^{-1}(w')]}_{\rho''})_{V \cup \{y\}}[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}'])) \\
= & \sum_{w' \in range(renc_t)} ([\![(\psi^{\mathcal{Y}})_{V \cup \mathcal{Y}\#}]\!]_{\mathcal{V} \cup \{y\} \cup \mathcal{Y}r \cup \mathcal{Y}\#}, (t, \rho'[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}])) + \\
& \sum_{w' \in range(\#enc_t)} ([\![(\psi^{\mathcal{Y}})_{V \cup \mathcal{Y}\# \cup \{y\}}]\!]_{\mathcal{V} \cup \{y\} \cup \mathcal{Y}r \cup \mathcal{Y}\#}, (t, \rho''[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}])) \\
& \text{(by induction hypothesis)} \\
= & \sum_{w \in \text{pos}(t)} ([\![((\psi^{\mathcal{Y}})_{V \cup \mathcal{Y}\#}]\!]_{\mathcal{V} \cup \mathcal{Y}r \cup \mathcal{Y}\# \cup \{y\}}, (t, \rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}, y \to w])) + \\
& \sum_{w \in \text{pos}(t)} ([\![(\psi^{\mathcal{Y}})_{V \cup \mathcal{Y}\# \cup \{y\}}]\!]_{\mathcal{V} \cup \mathcal{Y}r \cup \mathcal{Y}\# \cup \{y\}}, (t, \rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}, y \to w])) \\
= & \sum_{w \in \text{pos}(t)} ([\![(\psi^{\mathcal{Y}})_{V \cup \mathcal{Y}\#} \vee (\psi^{\mathcal{Y}})_{V \cup \mathcal{Y}\# \cup \{y\}}]\!]_{\mathcal{V} \cup \mathcal{Y}r \cup \mathcal{Y}\# \cup \{y\}}, (t, \rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}, y \to w])) \\
= & ([\![\exists y.((\psi^{\mathcal{Y}})_{V \cup \mathcal{Y}\#} \vee (\psi^{\mathcal{Y}})_{V \cup \mathcal{Y}\# \cup \{y\}})]\!]_{\mathcal{V} \cup \mathcal{Y}r \cup \mathcal{Y}\#}, (t, \rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}]))
\end{array}
$$

$$= ([\![(\exists y.\psi^{\mathcal{Y}})_{V\cup\mathcal{Y}\#}]\!]_{\mathcal{V}\cup\mathcal{Y}r\cup\mathcal{Y}\#}, (t,\rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}]))$$
$$= ([\![((\exists y.\psi)^{\mathcal{Y}})_{V\cup\mathcal{Y}\#}]\!]_{\mathcal{V}\cup\mathcal{Y}r\cup\mathcal{Y}\#}, (t,\rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}])).$$

Case 10. $\varphi = \forall y.\psi$: This case can be proved literally in the same way as the case in which $\varphi = \exists y.\psi$ except that one has to replace the addition of the semiring by its multiplication. Note that we assume that K is commutative.

Case 11. $\varphi = \exists Z.\psi$: Since Z is a bound variable in φ, we have that $Z \notin \mathcal{Y}$ by our assumption. Then we can compute as follows:

$$([\![\exists Z.\psi]\!]_{\mathcal{V}\cup\mathcal{Y}}, (enc(t), \rho_V[\mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}']))$$
$$= \textstyle\sum_{L'\subseteq \mathrm{pos}(enc(t))}([\![\psi]\!]_{\mathcal{V}\cup\mathcal{Y}\cup\{Z\}}, (enc(t), \rho_V[Z \to L', \mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}']))$$
$$= \textstyle\sum_{L_1'\subseteq \mathrm{range}(renc_t)} \sum_{L_2'\subseteq \mathrm{range}(\#enc_t)}([\![\psi]\!]_{\mathcal{V}\cup\mathcal{Y}\cup\{Z\}}, (enc(t), \rho_V[Z \to L_1' \cup L_2', \mathcal{Y} \to \mathcal{I}' \cup \mathcal{J}']))$$
$$= \textstyle\sum_{L_1\subseteq \mathrm{pos}(t)} \sum_{L_2\subseteq \mathrm{pos}(t)} ([\![(\psi^{\mathcal{Y}\cup\{Z\}})_{V\cup\mathcal{Y}\#\cup\{Z\#\}}]\!]_{\mathcal{V}\cup\mathcal{Y}r\cup\mathcal{Y}\#\cup\{Zr,Z\#\}}, (t,\rho[Zr \to L_1, Z\# \to L_2, \mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}]))$$
(by induction hypothesis)
$$= ([\![\exists Zr.\exists Z\#.(\psi^{\mathcal{Y}\cup\{Z\}})_{V\cup\mathcal{Y}\#\cup\{Z\#\}}]\!]_{\mathcal{V}\cup\mathcal{Y}r\cup\mathcal{Y}\#\cup\{Zr,Z\#\}}, (t,\rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}]))$$
$$= ([\![(\exists Z.(\psi^{\mathcal{Y}})_{V\cup\mathcal{Y}\#}]\!]_{\mathcal{V}\cup\mathcal{Y}r\cup\mathcal{Y}\#\cup\{Zr,Z\#\}}, (t,\rho[\mathcal{Y}r \to \mathcal{I}, \mathcal{Y}\# \to \mathcal{J}])).$$

Case 12. $\varphi = \forall Y.\psi$: This case can be proved literally in the same way as the case in which $\varphi = \forall Y.\psi$ except that one has to replace the addition of the semiring by its multiplication. Note again that we assume that K is commutative. ☐

As a special case of Lemma 6.4 in which no variable Y_i occurs in φ, we obtain the following corollary.

Corollary 6.5. Let K be commutative. Let $\psi \in \mathrm{MSO}(K, enc(\Sigma))$ and $V, \mathcal{V}$ be two sets of variables such that $V \cup \mathrm{Free}(\psi) \subseteq \mathcal{V}$ and no variable of V occurs bound in ψ. Moreover, let $s = (t,\rho) \in U^v_{\Sigma_\mathcal{V}}$. Then $([\![\psi]\!]_\mathcal{V}, (enc(t), \rho_V)) = ([\![\psi_V]\!]_\mathcal{V}, (t,\rho))$. Also, if $\psi \in \mathrm{RMSO}(K, enc(\Sigma))$, then $\psi_V \in \mathrm{uRMSO}(K,\Sigma)$.

Proof. The first claim follows from Lemmas 6.4 and 4.3.

Next assume that $[\![\psi]\!]_\mathcal{V}$ is a recognizable step function of the form $[\![\psi]\!]_\mathcal{V} = \sum_{i=1}^n k_i \cdot \mathbb{1}_{L_i}$. We show that $[\![\psi_V]\!]_\mathcal{V}$ is a recognizable unranked step function. For every $1 \leq i \leq n$, let $L_i' = \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid (enc(t), \rho_V) \in L_i\}$. Since $enc_V(L_i') = L_i \cap \mathrm{enc}_V(U^v_{\Sigma_\mathcal{V}})$, it follows from Proposition 5.2 that L_i' is a recognizable unranked tree language. By the first claim, we have $[\![\psi_V]\!]_\mathcal{V} = \sum_{i=1}^n k_i \cdot \mathbb{1}_{L_i'}$.

Finally, let ψ be restricted. Since ψ does not contain a universal set quantifier, neither does ψ_V. Now let ψ_V contain a subformula $\gamma = \forall x.\delta$, and let $\mathcal{W}$ be a finite set of variables such that $\mathrm{Free}(\delta) \subseteq \mathcal{W}$. We will show that $[\![\delta]\!]_\mathcal{W}$ is a recognizable step function which implies that ψ_V is restricted. By construction of ψ_V, ψ contains a formula $\alpha = \forall x.\beta$ such that $\gamma = \alpha_V$ and $\delta = \beta_V \wedge \beta_{V\cup\{x\}}$. Since ψ is restricted, $[\![\beta]\!]_\mathcal{W}$ is a recognizable step function. As shown above, then $[\![\beta_V]\!]_\mathcal{W}$ and $[\![\beta_{V\cup\{x\}}]\!]_\mathcal{W}$ are recognizable unranked step functions. By Proposition 3.2, $[\![\delta]\!]_\mathcal{W} = [\![\beta_V]\!]_\mathcal{W} \odot [\![\beta_{V\cup\{x\}}]\!]_\mathcal{W}$ is also a recognizable unranked step function. ☐

6.3 First inclusion result and its proof

Theorem 6.6. *Let K be a commutative semiring, Σ be an alphabet, and $S \in K\langle\langle U_\Sigma\rangle\rangle$ be recognizable. Then S is* uREMSO(K,Σ)*-definable.*

Proof. Let $M = (Q, \Sigma, \delta, \gamma)$ be an uwta such that $S = S_M$. Then, by Lemma 6.1 there is a binary wta M' over K and $enc(\Sigma)$ such that for every $s \in U_\Sigma$ we have that $(S_M, s) = (S_{M'}, enc(s))$. By Theorem 4.9, there is a sentence $\varphi \in$ REMSO$(K, enc(\Sigma))$ such that $S_{M'} = [\![\varphi]\!]$. Note that the $(\Sigma, \emptyset, \emptyset)$-encoding $enc_\emptyset(s)$ and $enc(s)$ are equal. Thus, $([\![\varphi]\!], enc(s)) = ([\![\varphi]\!], enc_\emptyset(s)) = ([\![\varphi]\!]_\emptyset, enc_\emptyset(s))$ where the latter equation holds because φ does not contain free variables. Then, by Corollary 6.5, we have that $([\![\varphi]\!]_\emptyset, enc_\emptyset(s)) = ([\![\varphi_\emptyset]\!], s)$ where $\varphi_\emptyset \in$ uREMSO(K, Σ). Thus $S = [\![\varphi_\emptyset]\!]$ is uREMSO(K,Σ)-definable. □

7 uRMSO-definable unranked tree series are recognizable

In this section we want to prove that every uRMSO(K,Σ)-definable unranked tree series $S \in K\langle\langle U_\Sigma\rangle\rangle$ is also recognizable (cf. Theorem 7.6 in Section 7.3). The proof of this theorem is based on Lemmas 7.2 and 7.5 which, in their turn, will be proved in Sections 7.1 and 7.2, respectively. We will also show that the equivalence problem for uRMSO(K,Σ)-sentences is decidable if K is a computable field.

7.1 From weighted uMSO to weighted MSO

In this section, for every formula $\psi \in$ uMSO(K,Σ) we will define a formula $\psi' \in$ MSO$(K, enc(\Sigma))$ such that for every finite set $\mathcal{V}$ of variables with Free$(\psi) \subseteq \mathcal{V}$ and for every $s \in U_{\Sigma_\mathcal{V}}$ we have $([\![\psi]\!]_\mathcal{V}, s) = ([\![\psi']\!]_\mathcal{V}, enc(s))$. Intuitively, we will interpret the formula ψ for each $s \in U_{\Sigma_\mathcal{V}}$ by ψ' on the regular positions of $enc(s)$. For this purpose, we first introduce the following macros in MSO$(K, enc(\Sigma))$ where y, y', z are first order and X, Y, Z second order variables.

0. $\mathrm{root}(x) = \forall y.\neg\mathrm{edge}_1(y,x) \wedge \neg\mathrm{edge}_2(y,x)$
   $(x = y) = \forall z. \bigwedge_{i=1,2}(\mathrm{edge}_i(z,x) \leftrightarrow \mathrm{edge}_i(z,y))$
   $(x \neq y) = (\mathrm{root}(x) \wedge \exists z. \bigvee_{i=1,2} \mathrm{edge}_i(z,y)) \vee \exists z. \bigvee_{i=1,2}(\mathrm{edge}_i(z,x) \vee \neg\mathrm{edge}_i(z,y))$

1. $\mathrm{down}_2\text{–path}(Z,z) = (z \in Z) \wedge \forall v.((v \in Z) \to (v = z \vee (v \neq z \wedge \exists u.\mathrm{edge}_2(u,v) \wedge (u \in Z))))$
   $\wedge \forall v.\forall w.((v \in Z) \to (\mathrm{edge}_2(v,w) \to (w \in Z)))$

Clearly, for any $s = (t, \rho) \in T_{enc(\Sigma)_{\{Z,z\}}}$ with $t \in \mathrm{enc}(U_\Sigma)$ the root of t has a single successor. It follows that $([\![\mathrm{down}_2\text{–path}(Z,z)]\!], s) = 1$ iff $\rho(Z)$ is the set of nodes of a path in t from $\rho(z)$ to the leaves of t using only the edge$_2$-relation between successive nodes. Note that this path is determined uniquely in t by $\rho(z)$. Otherwise, $([\![\mathrm{down}_2\text{–path}(Z,z)]\!], s) = 0$.

2. $\mathrm{up}\text{–path}(Y,y) =$
   $(y \in Y) \wedge \exists x.(\mathrm{root}(x) \wedge x \in Y) \wedge \forall u.\forall v.\forall w.(((v \in Y) \wedge ((\mathrm{edge}_1(u,v) \wedge \mathrm{edge}_2(u,w))$
   $\vee(\mathrm{edge}_2(u,v) \wedge \mathrm{edge}_1(u,w)))) \to ((u \in Y) \wedge \neg(w \in Y)))$
   $\wedge \forall v.(\mathrm{edge}_1(y,v) \vee \mathrm{edge}_2(y,v)) \to \neg(v \in Y)$

Then for any $s = (t, \rho) \in T_{enc(\Sigma)_{\{Y,y\}}}$ with $t \in \mathrm{enc}(U_\Sigma)$ we have $([\![\text{up–path}(Y, y)]\!], s) = 1$ iff $\rho(Y)$ is the set of nodes of a path in t from the root of t to $\rho(y)$. Note that this path is again determined uniquely in t by $\rho(y)$. Otherwise, $([\![\text{up–path}(Y, y)]\!], s) = 0$.

3. $\mathrm{lex}(y, y') = \exists Y.\exists Y'.\exists x \exists z.\exists z'.\text{up–path}(Y, y) \wedge \text{up–path}(Y', y') \wedge \mathrm{edge}_1(x, z) \wedge \mathrm{edge}_2(x, z')$
$\wedge (z \in Y) \wedge (z' \in Y')$

Then for any $s = (t, \rho) \in T_{enc(\Sigma)_{\{y,y'\}}}$ with $t \in \mathrm{enc}(U_\Sigma)$ we have $([\![\mathrm{lex}(y, y')]\!], s) = 1$ iff the position $\rho(y)$ is not a prefix of $\rho(y')$, but $\rho(y)$ precedes $\rho(y')$ in the natural lexicographic ordering of $\{1, 2\}^*$. Otherwise, $([\![\mathrm{lex}(y, y')]\!], s) = 0$.

4. $\text{non–empty}_\#(X) = \exists x.(x \in X) \wedge \mathrm{label}_\#(x) \wedge \forall y.(y \in X \wedge \mathrm{label}_\#(y)) \rightarrow (x = y \vee \mathrm{lex}(x, y)))$
Then for any $s = (t, \rho) \in T_{enc(\Sigma)_{\{X\}}}$ with $t \in \mathrm{enc}(U_\Sigma)$ we have $([\![\text{non–empty}_\#(X)]\!], s) = 1$ iff there exists $w \in \rho(X)$ with $t(w) = \#$. The reason for this value then to be 1 is that the formula involving $\mathrm{lex}(x, y)$ makes this position w with $\rho(x) = w$ the lexicographic smallest one and hence unique. On the other hand, if $t(w) \neq \#$ for each $w \in \rho(X)$, then $([\![\text{non–empty}_\#(X)]\!], s) = 0$.

5. $\mathrm{Enc} = \forall x.(\mathrm{root}(x) \leftrightarrow \bigvee_{\sigma \in \Sigma} \mathrm{label}_{\sigma'}(x))$

Then for any $t' \in T_{\mathrm{enc}(\Sigma)}$ we have $([\![\mathrm{Enc}]\!], t') = 1$ iff the root of t' is the only node labelled with some σ' ($\sigma \in \Sigma$) iff $t' = \mathrm{enc}(t)$ for some $t \in U_\Sigma$. Otherwise, $([\![\mathrm{Enc}]\!], t') = 0$. That is, the sentence Enc defines the language $\mathrm{enc}(U_\Sigma)$ in $T_{\mathrm{enc}(\Sigma)}$.

Now let $\mathcal{V}$ be a finite set of variables and consider the alphabet $\Sigma_\mathcal{V}$. Its encoding mapping, also denoted enc, equals the $(\Sigma, \emptyset, \mathcal{V})$-encoding $\mathrm{enc}_\emptyset$ (cf. section 6.2). So, $\mathrm{enc}(U^v_{\Sigma_\mathcal{V}}) = \{(\mathrm{enc}(t), \mathrm{renc}_t \circ \rho) \mid (t, \rho) \in U^v_{\Sigma_\mathcal{V}}\}$; note that the assignment $\mathrm{renc}_t \circ \rho$ assigns to variables only regular positions resp. sets of regular positions. Let $\mathcal{V}_1$ (resp., $\mathcal{V}_2$) comprise all first order (resp., second order) variables of $\mathcal{V}$. By the above remarks, the language $\mathrm{enc}(U^v_{\Sigma_\mathcal{V}})$ in $T_{\mathrm{enc}(\Sigma)_\mathcal{V}}$ is defined by the formula

$$\mathrm{Enc}_\mathcal{V} = \mathrm{Enc} \wedge \bigwedge_{x \in \mathcal{V}_1} \neg \mathrm{label}_\#(x) \wedge \bigwedge_{X \in \mathcal{V}_2} \forall z.((z \in X) \rightarrow \neg \mathrm{label}_\#(z)).$$

In particular, $([\![\mathrm{Enc}_\mathcal{V}]\!], s') = 1$ if $s' = \mathrm{enc}(s)$ for some $s \in U^v_{\Sigma_\mathcal{V}}$ and $([\![\mathrm{Enc}_\mathcal{V}]\!], s') = 0$ otherwise. For each formula $\psi \in \mathrm{MSO}(K, \Sigma)$, we abbreviate $\mathrm{Enc}_\psi = \mathrm{Enc}_{\mathrm{Free}(\psi)}$.

Definition 7.1. For every formula $\psi \in \mathrm{uMSO}(K, \Sigma)$ we define a formula $\psi' \in \mathrm{MSO}(K, enc(\Sigma))$ inductively as follows

$\psi = k$: $\quad\psi' = k$
ψ is $\mathrm{label}_\sigma(x)$ or $(x \in X)$ or their negations: $\psi' = \psi$.
$\psi = \mathrm{desc}(x, y)$: $\quad\psi' = \exists Z.\exists z.\mathrm{edge}_1(x, z) \wedge \text{down}_2\text{–path}(Z, z) \wedge (y \in Z)$
$\psi = \neg\mathrm{desc}(x, y)$: $\quad\psi' = \exists Z.\exists z.\mathrm{edge}_1(x, z) \wedge \text{down}_2\text{–path}(Z, z) \wedge \neg(y \in Z)$
$\psi = (x \leq y)$: $\quad\psi' = \exists Z.\text{down}_2\text{–path}(Z, x) \wedge (y \in Z)$
$\psi = \neg(x \leq y)$: $\quad\psi' = \exists Z.\text{down}_2\text{–path}(Z, x) \wedge \neg(y \in Z)$
$\psi = \psi_1 \wedge \psi_2$: $\quad\psi' = \psi'_1 \wedge \psi'_2$
$\psi = \psi_1 \vee \psi_2$: $\quad\psi' = \psi'_1 \vee \psi'_2$
$\psi = \exists x.\varphi$: $\quad\psi' = \exists x.\neg\mathrm{label}_\#(x) \wedge \varphi'$

$\psi = \forall x.\varphi$: $\quad \psi' = \forall x.(\neg \mathrm{label}_{\#}(x) \to (\varphi' \wedge \mathrm{Enc}_{\varphi}))$
$\psi = \exists X.\varphi$: $\quad \psi' = \exists X.(\forall x.(x \in X) \to \neg \mathrm{label}_{\#}(x)) \wedge \varphi'$
$\psi = \forall X.\varphi$: $\quad \psi' = \forall X.\text{non–empty}_{\#}(X) \vee ((\forall x.(x \in X) \to \neg \mathrm{label}_{\#}(x)) \wedge \varphi')$

Lemma 7.2. Let K be commutative. Let $\psi \in \mathrm{uMSO}(K, \Sigma)$, $\mathcal{V}$ a set of variables such that $\mathrm{Free}(\psi) \subseteq \mathcal{V}$, and let $\psi' \in \mathrm{MSO}(K, enc(\Sigma))$ be defined as in Definition 7.1. Moreover, let $s = (t, \rho) \in U^{v}_{\Sigma_{\mathcal{V}}}$. Then $([\![\psi]\!]_{\mathcal{V}}, s) = ([\![\psi']\!]_{\mathcal{V}}, enc(s))$.

Proof. We proceed by induction on the structure of ψ. We put $t' = enc(t)$ and $\rho' = renc_t \circ \rho$ (where $\circ$ denotes the usual functional composition 'first ρ then $renc_t$'), hence $enc(s) = (t', \rho')$.

Case 1. $\psi = k$: Clearly $([\![\psi]\!]_{\mathcal{V}}, s) = k = ([\![\psi']\!]_{\mathcal{V}}, enc(s))$.

Case 2. $\psi = \mathrm{label}_{\sigma}(x)$, $\psi = (x \in X)$, $\psi = \neg \mathrm{label}_{\sigma}(x)$ or $\psi = \neg(x \in X)$: These cases are easy. We only consider the first case, $\psi = \mathrm{label}_{\sigma}(x)$. We have $([\![\psi]\!]_{\mathcal{V}}, s) = 1$ iff $t(\rho(x)) = \sigma$ iff $enc(t)(renc_t(\rho(x))) = \sigma$ iff $([\![\psi']\!]_{\mathcal{V}}, enc(s)) = 1$.

Case 3. $\psi = \mathrm{desc}(x, y)$: We have

$([\![\psi]\!]_{\mathcal{V}}, s) = 1$ iff $\rho(y) = \rho(x) \cdot i$ for some $i \geq 1$
iff $renc_t(\rho(y)) = renc_t(\rho(x)) \cdot 1 \cdot \underbrace{2 \cdot \ldots \cdot 2}_{i-1}$ for some $i \geq 1$
iff there is a path in t' from $z = \rho'(x) \cdot 1$ to the leaves of t' containing the node $\rho'(y)$ and only the edge$_2$-relation between successive nodes
iff $([\![\psi']\!]_{\mathcal{V}}, (t', \rho')) = 1$

Case 4. $\psi = \neg \mathrm{desc}(x, y)$: We note that here the formula ψ' is not the logical negation of $\mathrm{desc}(x, y))'$. However, we will exploit the uniqueness properties of down$_2$–paths mentioned before. We have

$([\![\psi]\!]_{\mathcal{V}}, s) = 1$ iff not $\rho(y) = \rho(x) \cdot i$ for some $i \geq 1$
iff $renc_t(\rho(y))) \neq renc_t(\rho(x)) \cdot 1 \cdot 2 \cdot \ldots \cdot 2$ for any sequence of 2's
iff the unique path in t' from $z = \rho'(x) \cdot 1$ to the leaves of t' containing only the edge$_2$-relation between successive nodes does not pass through $\rho'(y)$
iff $([\![\psi']\!]_{\mathcal{V}}, (t', \rho')) = 1$

Case 5. $\psi = (x \leq y)$: Here

$([\![\psi]\!]_{\mathcal{V}}, s) = 1$ iff $\rho(x) = \rho(y)$ or $\rho(y)$ is an older sibling of $\rho(x)$
iff $\rho'(y) = \rho'(x) \cdot 2 \cdot \ldots \cdot 2$ for some sequence of 2's
iff the uniquely determined down$_2$–path in t' from $\rho'(x)$ to the leaves of t' contains $\rho'(y)$
iff $([\![\psi']\!]_{\mathcal{V}}, (t', \rho')) = 1$

Case 6. $\psi = \neg(x \leq y)$: Again, note that ψ' is not the logical negation of $(x \leq y)'$. Similarly as before, we obtain

$$\begin{aligned}
(\llbracket\psi\rrbracket_{\mathcal{V}}, s) = 1 \;\text{ iff }\; & \rho'(y) \neq \rho'(x) \cdot 2 \cdot \ldots \cdot 2 \text{ for any sequence of 2's} \\
\text{iff }\; & \text{the uniquely determined down}_2\text{–path in } t' \text{ from } \rho'(x) \text{ to the leaves of } t' \\
& \quad \text{does not contain } \rho'(x) \\
\text{iff }\; & (\llbracket\psi'\rrbracket_{\mathcal{V}}, (t', \rho')) = 1
\end{aligned}$$

Case 7. $\psi = \psi_1 \wedge \psi_2$ or $\psi = \psi_1 \vee \psi_2$: Straightforward using the induction hypothesis for ψ_1 and ψ_2.

Case 8. $\psi = \exists x.\varphi$: Here

$$\begin{aligned}
(\llbracket\psi\rrbracket_{\mathcal{V}}, s) &= \sum_{i \in \mathrm{pos}(s)} (\llbracket\varphi\rrbracket_{\mathcal{V}\cup\{x\}}, s[x \to i]) \\
&= \sum_{i \in \mathrm{pos}(s)} (\llbracket\varphi'\rrbracket_{\mathcal{V}\cup\{x\}}, enc(s[x \to i])) \\
&\quad \text{(by induction hypothesis)} \\
&= \sum_{i \in \mathrm{pos}(s)} (\llbracket\varphi'\rrbracket_{\mathcal{V}\cup\{x\}}, enc(s)[x \to renc_t(i)]) \\
&= \sum_{j \in renc_t(\mathrm{pos}(s))} (\llbracket\varphi'\rrbracket_{\mathcal{V}\cup\{x\}}, enc(s)[x \to j]) \\
&= \sum_{j \in \mathrm{pos}(enc(s))} (\llbracket\varphi'\rrbracket_{\mathcal{V}\cup\{x\}}, enc(s)[x \to j]) \cdot (\llbracket\neg\mathrm{label}_{\#}(x)\rrbracket_{\mathcal{V}\cup\{x\}}, enc(s)[x \to j]) \\
&= (\llbracket\psi'\rrbracket_{\mathcal{V}}, enc(s)).
\end{aligned}$$

Case 9. $\psi = \forall x.\varphi$: The reason for employing also the formula Enc_φ in this translation will become apparent in Corollary 7.3. Arguing as in Case 8 and replacing sums by products, we obtain

$$\begin{aligned}
(\llbracket\psi\rrbracket_{\mathcal{V}}, s) &= \prod_{j \in renc_t(\mathrm{pos}(s))} (\llbracket\varphi'\rrbracket_{\mathcal{V}\cup\{x\}}, enc(s)[x \to j]) \\
&= \prod_{j \in renc_t(\mathrm{pos}(s))} (\llbracket\neg\mathrm{label}_{\#}(x) \wedge \varphi' \wedge \mathrm{Enc}_\varphi\rrbracket_{\mathcal{V}\cup\{x\}}, enc(s)[x \to j]) \\
&\quad \cdot \prod_{j \in \# enc_t(\mathrm{pos}(s))} (\llbracket\mathrm{label}_{\#}(x)\rrbracket_{\mathcal{V}\cup\{x\}}, enc(s)[x \to j]) \\
&= (\llbracket\psi'\rrbracket_{\mathcal{V}\cup\{x\}}, enc(s));
\end{aligned}$$

in the final equation we employ the definition of the implication inside ψ' and commutativity of K.

Case 10. $\psi = \exists X.\varphi$: Again we argue similarly as in Case 8 and obtain

$$
\begin{aligned}
&(\llbracket \psi \rrbracket_{\mathcal{V}}, s) \\
&= \sum_{J \subseteq renc_t(\mathrm{pos}(s))} (\llbracket \varphi' \rrbracket_{\mathcal{V} \cup \{X\}}, enc(s)[X \to J]) \\
&= \sum_{J \subseteq \mathrm{pos}(enc(s))} (\llbracket \varphi' \rrbracket_{\mathcal{V} \cup \{X\}}, enc(s)[X \to J]) \cdot \prod_{j \in J} (\llbracket \neg \mathrm{label}_{\#}(x) \rrbracket_{\mathcal{V} \cup \{X,x\}}, enc(s)[X \to J, x \to j]) \\
&= (\llbracket \psi' \rrbracket_{\mathcal{V}}, enc(s)).
\end{aligned}
$$

Case 11. $\psi = \forall X.\varphi$: Similarly as in Case 10, replacing sums by products, we get

$$
\begin{aligned}
&(\llbracket \psi \rrbracket_{\mathcal{V}}, s) \\
&= \prod_{J \subseteq renc_t(\mathrm{pos}(s))} (\llbracket \varphi' \rrbracket_{\mathcal{V} \cup \{X\}}, enc(s)[X \to J]) \\
&= \prod_{J \subseteq renc_t(\mathrm{pos}(s))} ((\llbracket \varphi' \rrbracket_{\mathcal{V} \cup \{X\}}, enc(s)[X \to J]) \cdot \prod_{j \in J} (\llbracket \neg \mathrm{label}_{\#}(x) \rrbracket_{\mathcal{V} \cup \{X,x\}}, enc(s)[X \to J, x \to j]) \\
&\quad \cdot \prod_{\substack{J \subseteq \mathrm{pos}(enc(s)) \\ J \cap \# enc_t(\mathrm{pos}(s)) \neq \emptyset}} (\llbracket \text{non–empty}_{\#}(X) \rrbracket_{\mathcal{V} \cup \{X\}}, enc(s)[X \to J]) \\
&= (\llbracket \psi' \rrbracket_{\mathcal{V}}, enc(s)).
\end{aligned}
$$

This finishes the induction and proves the result. □

Corollary 7.3. *Let K be commutative. Let $\psi \in \mathrm{uMSO}(K, \Sigma)$ and let $\mathcal{V}$ be a finite set of variables such that* $\mathrm{Free}(\psi) \subseteq \mathcal{V}$*. If $\llbracket \psi \rrbracket_{\mathcal{V}}$ is a recognizable unranked step function, then so is $\llbracket \psi' \wedge \mathrm{Enc}_{\psi} \rrbracket_{\mathcal{V}}$. If $\psi \in \mathrm{uRMSO}(K, \Sigma)$, then $\psi' \in \mathrm{RMSO}(K, enc(\Sigma))$.*

Proof. First let $\llbracket \psi \rrbracket_{\mathcal{V}}$ be a recognizable unranked step function. By Lemma 7.2, for any $s \in U^{v}_{\Sigma_{\mathcal{V}}}$ we have $(\llbracket \psi \rrbracket_{\mathcal{V}}, s) = (\llbracket \psi' \rrbracket_{\mathcal{V}}, enc(s)) = (\llbracket \psi' \wedge \mathrm{Enc}_{\psi} \rrbracket_{\mathcal{V}}, enc(s))$, and $(\llbracket \psi' \wedge \mathrm{Enc}_{\psi} \rrbracket_{\mathcal{V}}, s') = 0$ for any $s' \notin enc(U_{\Sigma_{\mathcal{V}}})$. Hence $\llbracket \psi' \wedge \mathrm{Enc}_{\psi} \rrbracket_{\mathcal{V}}$ assumes only finitely many values. Since for any recognizable unranked tree language $L \subseteq U_{\Sigma_{\mathcal{V}}}$, $enc(L)$ is recognizable in $T_{enc(\Sigma_{\mathcal{V}})}$ (by Proposition 5.2), $\llbracket \psi' \wedge \mathrm{Enc}_{\psi} \rrbracket_{\mathcal{V}}$ is a recognizable step function.

Secondly, let ψ be restricted. Since ψ does not contain a universal set quantifier, neither does ψ'. Now let ψ' contain a formula $\gamma = \forall x.\delta$, and let $\mathcal{W}$ be a finite set of variables such that $\mathrm{Free}(\delta) \subseteq \mathcal{W}$. We will show that $\llbracket \delta \rrbracket_{\mathcal{W}}$ is a recognizable step function which implies that ψ' is restricted. By construction of ψ', ψ contains a formula $\alpha = \forall x.\beta$ such that $\gamma = \alpha'$ and $\delta = (\neg \mathrm{label}_{\#}(x) \to (\beta' \wedge \mathrm{Enc}_{\beta}))$. Since ψ is restricted, $\llbracket \beta \rrbracket_{\mathcal{W}}$ is a recognizable unranked step function. As shown above, then $\llbracket \beta' \wedge \mathrm{Enc}_{\beta} \rrbracket_{\mathcal{W}}$ is a recognizable step function. By the analogue of Proposition 3.2, the class of recognizable step functions is closed under sum and Hadamard product (or compare Lemma 4.9 of [DV06]). Hence $\llbracket \delta \rrbracket_{\mathcal{W}}$ is also a recognizable step function. □

As a consequence of these translations of formulas, we obtain a decidability result.

Corollary 7.4. Let K be a computable field, and let ψ_1 and ψ_2 be two $\mathrm{uRMSO}(K, \Sigma)$-sentences. Then it is decidable whether $\llbracket \psi_1 \rrbracket = \llbracket \psi_2 \rrbracket$.

Proof. By Corollary 7.3 and Proposition 3.2, the sentences $\psi_1' \wedge \text{Enc}$ and $\psi_2' \wedge \text{Enc}$ belong to $\text{RMSO}(K, enc(\Sigma))$. From Lemma 7.2 it follows that $[\![\psi_1]\!] = [\![\psi_2]\!]$ iff $[\![\psi_1' \wedge \text{Enc}]\!] = [\![\psi_2' \wedge \text{Enc}]\!]$. And by [DV06], Corollary 5.9 (using the assumption on K), the latter equality is decidable. $\square$

7.2 From weighted tree automata to unranked weighted tree automata

Here we show that a wta over $enc(\Sigma$ can be simulated by a uwta over Σ.

Lemma 7.5. For every wta M over $enc(\Sigma)$ and K there is a uwta M' over Σ and K such that $(S_M, enc(t)) = (S_{M'}, t)$ for every $t \in U_\Sigma$.

Proof. Let $M = (Q, enc(\Sigma), \delta, \gamma)$ be a wta over K. We construct the uwta $M' = (Q', \Sigma, \delta', \gamma')$ over K as follows.

- $Q' = (Q \times Q) \cup (Q \times \Sigma \times Q \times Q)$
  We note that states from $Q \times Q$ and from $(Q \times \Sigma \times Q \times Q)$ are only relevant at the root and at interior nodes of a given input tree, respectively.
- The transition function $\delta' : Q' \times \Sigma \to K\langle\!\langle (Q')^* \rangle\!\rangle$ is defined as follows:

  – for every $q, q_1 \in Q$ and $\sigma \in \Sigma$ we define $\delta'((q, q_1), \sigma) = S$ where $S : (Q')^* \to K$ and for every $w \in (Q')^*$

  * if w has the form $(q_1, \sigma_1, p_1, q_2)(q_2, \sigma_2, p_2, q_3) \ldots (q_n, \sigma_n, p_n, q_{n+1})$ for some $n \geq 1$, then
  $$(S, w) = (\delta(q, \sigma'), q_1) \cdot \left(\prod_{i=1}^{n} (\delta(q_i, \sigma_i), p_i q_{i+1}) \right) \cdot (\delta(q_{n+1}, \#), \varepsilon)$$
  * if $w = \varepsilon$, then $(S, w) = (\delta(q_1, \#), \varepsilon)$
  * otherwise $(S, w) = 0$.

  It is obvious that S can be recognized by a deterministic wsa with state set Q.

  – for every $(q, \gamma, q_1, p) \in Q \times \Sigma \times Q \times Q$ and $\sigma \in \Sigma$ we define
  $$\delta'((q, \gamma, q_1, p), \sigma) = \begin{cases} S & \text{if } \sigma = \gamma \\ \widetilde{0} & \text{if } \sigma \neq \gamma \end{cases}$$
  where $S : (Q')^* \to K$ and for every $w \in (Q')^*$

  * if w has the form $w = (q_1, \sigma_1, p_1, q_2)(q_2, \sigma_2, p_2, q_3) \ldots (q_n, \sigma_n, p_n, q_{n+1})$ for some $n \geq 1$, then
  $$(S, w) = \left(\prod_{i=1}^{n} (\delta(q_i, \sigma_i), p_i q_{i+1}) \right) \cdot (\delta(q_{n+1}, \#), \varepsilon)$$
  * if $w = \varepsilon$, then $(S, w) = (\delta(q_1, \#), \varepsilon)$
  * otherwise $(S, w) = 0$.

  Again it is clear that S can be recognized by a deterministic wsa with state set Q.

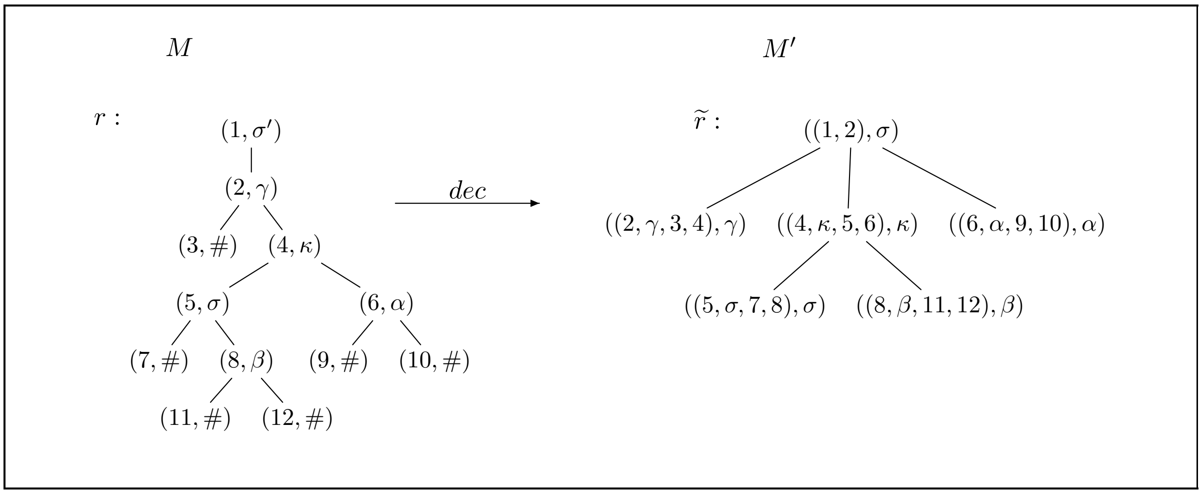

Figure 10: Example of the relationship between a run r of M and the run $\widetilde{r}$ of M'.

- The final weight distribution $\gamma' : Q' \to K$ is defined for every $q' \in Q'$ by

$$\gamma'(q') = \begin{cases} \gamma(q) & \text{if } q' = (q, q_1) \text{ for some } q, q_1 \in Q \\ 0 & \text{if } q' \notin Q \times Q \end{cases}$$

Our goal is to prove that $(S_M, enc(t)) = (S_{M'}, t)$ for every $t \in U_\Sigma$. For this purpose we define a relationship between the runs of M and M' (cf. Figure 10 for an example with states $1, \ldots, 12$). More precisely, we define the decoding mapping $dec : (Q \times \Sigma')(T_{Q\times(\Sigma\cup\{\#\})}) \to C(H_{(Q'\times\Sigma)\setminus C})$ with $C = (Q \times Q) \times \Sigma$ such that for every $(q, \sigma')(r) \in (Q \times \Sigma')(T_{Q\times(\Sigma\cup\{\#\})})$, we have

$$dec((q, \sigma')(r)) = ((q, q_1), \sigma)hdec(r)$$

with $q_1 = r(\varepsilon)_1$, and we define $hdec : T_{Q\times(\Sigma\cup\{\#\})} \to H_{(Q'\times\Sigma)\setminus C}$ such that for every $r = (q_1, \sigma_1)(r_1, (q_2, \sigma_2)(r_2, \ldots (q_n, \sigma_n)(r_n, (q_{n+1}, \#)) \ldots)) \in T_{Q\times(\Sigma\cup\{\#\})}$ with $n \geq 0$ we have

$$hdec(r) = \Big(((q_1, \sigma_1, p_1, q_2), \sigma_1)\, hdec(r_1), \;\; \ldots, \;\; ((q_n, \sigma_n, p_n, q_{n+1}), \sigma_n)\, hdec(r_n) \Big)$$

with $p_i = r_i(\varepsilon)_1$. Obviously, every run $\widetilde{r}$ in the range of dec has a certain consistency property; namely, for every position $w \in pos(\widetilde{r}) \setminus \{\varepsilon\}$, if $\widetilde{r}(w) = ((q, \sigma, p, q'), \gamma)$, then $\sigma = \gamma$. Let us denote the subset of $C(H_{(Q'\times\Sigma)\setminus C})$ which only contains consistent runs, by $C(H_{(Q'\times\Sigma)\setminus C})^{con}$. It is clear that dec is a bijection between $(Q \times \Sigma')(T_{Q\times(\Sigma\cup\{\#\})})$ and $C(H_{(Q'\times\Sigma)\setminus C})^{con}$. Even, for every $t \in U_\Sigma$, (the restriction of) dec is a bijection between $tree_M^{-1}(enc(t))$ and $tree_{M'}^{-1}(t) \cap C(H_{(Q'\times\Sigma)\setminus C})^{con}$.

Now let $r \in (Q \times \Sigma')(T_{Q\times(\Sigma\cup\{\#\})})$ and $\widetilde{r} \in C(H_{(Q'\times\Sigma)\setminus C})^{con}$ such that $dec(r) = \widetilde{r}$. We associate with every $w \in pos(\widetilde{r})$ a set P_w of positions of r in such a way that the transition made by M' at w corresponds to the set of transitions made by M at the positions occurring in P_w. More precisely, for every $w \in pos(\widetilde{r})$ we define the set $P_w \subseteq pos(r)$ as follows:

- if $w = \varepsilon$, then $P_w = \{\varepsilon\} \cup \{12^i \mid 0 \leq i \leq rk_{\widetilde{r}}(\varepsilon)\}$
- if $w = i_1 \ldots i_n$ for some $n \geq 1$, then $P_w = \{p(w)12^i \mid 0 \leq i \leq rk_{\widetilde{r}}(w)\}$ and $p(w) = 12^{i_1-1} \ldots 12^{i_n-1} = renc_{\widetilde{r}}(w)$.

For instance, in Figure 10 we have that $P_\varepsilon = \{\varepsilon, 1, 12, 122, 1222\}$, $P_2 = \{121, 1212, 12122\}$, and $P_{22} = \{12121\}$. It is clear that $(P_w \mid w \in pos(\widetilde{r}))$ is a partitioning of $pos(r)$.

Then we can prove that for every r and $\widetilde{r}$ with $dec(r) = \widetilde{r}$ we have $wt_{M'}(\widetilde{r}) = wt_M(r)$ as follows.

$$
\begin{aligned}
& wt_{M'}(\widetilde{r}) \\
= \; & \prod_{w \in pos(\widetilde{r})} \Big(\delta'(\widetilde{r}(w)),\ \widetilde{r}(w1)_1 \ldots \widetilde{r}(w\, rk_{\widetilde{r}}(w))_1\Big) \\
= \; & \Big(\delta'(\widetilde{r}(\varepsilon)),\ \widetilde{r}(1)_1 \ldots \widetilde{r}(rk_{\widetilde{r}}(\varepsilon))_1\Big) \\
& \cdot \prod_{w \in pos(\widetilde{r}) \setminus \{\varepsilon\}} \Big(\delta'(\widetilde{r}(w),\ \widetilde{r}(w1)_1 \ldots \widetilde{r}(w\, rk_{\widetilde{r}}(w))_1\Big) \\
= \; & \Big(\delta(r(\varepsilon)),\ r(1)_1\Big) \cdot \prod_{i=0}^{rk_{\widetilde{r}}(\varepsilon)-1} \Big(\delta(r(12^i)),\ r(12^i1)_1\, r(12^i2)_1\Big) \\
& \cdot \Big(\delta(r(12^{rk_{\widetilde{r}}(\varepsilon)})),\ \varepsilon\Big) \\
& \cdot \prod_{w \in pos(\widetilde{r}) \setminus \{\varepsilon\}} \Big[\prod_{i=0}^{rk_{\widetilde{r}}(w)-1} \Big(\delta(r(p(w)\, 12^i)),\ r(p(w)\, 12^i1)_1\, r(p(w)\, 12^i2)_1\Big) \\
& \qquad \cdot \Big(\delta(r(p(w)\, 12^{rk_{\widetilde{r}}(w)})),\ \varepsilon\Big)\Big] \\
& \text{(by definition of } \delta'\text{; recall that } r(p(w)\, 12^{rk_{\widetilde{r}}(w)})_2 = \#) \\
= \; & \prod_{u \in P_\varepsilon} \Big(\delta(r(u)),\ r(u1)_1 \ldots r(u\, rk_r(u))_1\Big) \\
& \cdot \prod_{w \in pos(\widetilde{r}) \setminus \{\varepsilon\}} \Big[\prod_{u \in P_w} \Big(\delta(r(u)),\ r(u1)_1 \ldots r(u\, rk_r(u))_1\Big)\Big] \\
= \; & \prod_{u \in pos(r)} \Big(\delta(r(u)),\ r(u1)_1 \ldots r(u\, rk_r(u))_1\Big) \\
& \text{(because } (P_w \mid w \in pos(\widetilde{r})) \text{ is a partitioning of } pos(r)) \\
= \; & wt_M(r).
\end{aligned}
$$

Finally, we prove that $(S_M, enc(t)) = (S_{M'}, t)$ for every $t \in U_\Sigma$.

$$
\begin{aligned}
& (S_M, enc(t)) \\
= \; & \sum_{r \in tree_M^{-1}(enc(t))} wt_M(r) \cdot \gamma(r(\varepsilon)_1) \\
= \; & \sum_{\widetilde{r} \in tree_{M'}^{-1}(t) \cap C(H_{(Q' \times \Sigma) \setminus C})^{con}} wt_{M'}(\widetilde{r}) \cdot \gamma'(\widetilde{r}(\varepsilon)_1) \\
& (dec \text{ is a bijection between } tree_M^{-1}(enc(t)) \text{ and } tree_{M'}^{-1}(t) \cap C(H_{(Q' \times \Sigma) \setminus C})^{con} \text{ and} \\
& wt_M(r) = wt_{M'}(\widetilde{r}) \text{ and } \gamma(r(\varepsilon)_1) = \gamma'(\widetilde{r}(\varepsilon)_1) \text{ for corresponding } r \text{ and } \widetilde{r}) \\
= \; & \sum_{\widetilde{r} \in tree_{M'}^{-1}(t)} wt_{M'}(\widetilde{r}) \cdot \gamma'(\widetilde{r}(\varepsilon)_1) \\
& \text{(by definition of } \delta' \text{ non consistent runs are mapped to 0; } \gamma' \text{ maps every } q' \notin Q \times Q \text{ to 0)} \\
= \; & (S_{M'}, t).
\end{aligned}
$$

□

7.3 Second inclusion result and its proof

Theorem 7.6. *Let K be a commutative semiring, Σ be an alphabet, and $S \in K\langle\langle U_\Sigma \rangle\rangle$ be* uRMSO(K, Σ)*-definable. Then S is recognizable.*

Proof. Let $S = [\![\psi]\!]$ for some sentence $\psi \in$ uRMSO(K, Σ). Then $\psi' \in$ RMSO$(K, enc(\Sigma))$ by Corollary 7.3, and $([\![\psi]\!], t) = ([\![\psi']\!], enc(t))$ for each $t \in U_\Sigma$ by Lemma 7.2. By Theorem 4.9, there is a wta M' over $enc(\Sigma)$ and K such that $[\![\psi']\!] = S_{M'}$. By Lemma 7.5, we obtain a uwta M over Σ and K such that $(S_{M'}, enc(t)) = (S_M, t)$ for each $t \in U_\Sigma$. Hence $S = S_M$ is recognizable. $\square$

Corollary 7.7. *Let K be a computable field. There is an effective procedure which produces, for a given* uRMSO(K,Σ)*-sentence ψ, an unranked weighted tree automaton M such that $[\![\psi]\!] = S_M$.*

Proof. We follow the proof of Theorem 7.6. For the sentence $\psi' \in \mathrm{RMSO}(K, \mathrm{enc}(\Sigma))$, the equivalent wta M' can be constructed effectively by Corollary 5.8 of [DV06] (using the assumption on K). Now construct the uwta M as above to obtain the result. $\square$

8 Almost existential weighted uMSO-logic

Our main result, Theorem 4.7, provides an equivalence between uwta and sentences of our restricted MSO-logic. Recall that the definition of a formula φ to be restricted involves a semantic condition for particular subformulas of φ. In this section, we wish to describe a purely syntactic definition of a class of weighted MSO-formulas and show, for a large class of semirings, again an equivalence result to the class of all uwta.

Let us first describe the class of semirings we will encounter. A monoid M is called *locally finite*, if each finitely generated submonoid of M is finite. Clearly, a commutative monoid M is locally finite iff each cyclic submonoid $\langle a \rangle$ of M is finite. The main result of this section will apply to commutative semirings K for which the commutative monoid $(K, +, 0)$ is locally finite. For instance, such semirings include

- all idempotent commutative semirings K (i.e. $x + x = x$ for each $x \in K$), in particular the tropical semiring $Trop$, the *arctical* or *max-plus semiring* $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$, the semiring $(2^{\Sigma_c^*}, \cup, \cdot, \emptyset, \{\varepsilon\})$ of languages over a free commutative alphabet Σ_c, and the semiring $([0,1], \max, \cdot, 0, 1)$ useful for describing probabilistic settings,
- all fields of characteristic p, for any prime p,
- all locally finite semirings, cf. Section 9,
- all products $K_1 \times \ldots \times K_n$ (with operations defined pointwise) of semirings K_i such that $(K_i, +, 0)$ is locally finite $(i = 1, \ldots, n)$,
- the semiring of polynomials $(K[X], +, \cdot, 0, 1)$ over a variable X and a semiring K such that $(K, +, 0)$ is locally finite.

Next we turn to the definition of our weighted logic.

Definition 8.1. The *set* uMSO$^{\mathrm{ae}}(K,\Sigma)$ *of all formulas of almost existential weighted UMSO-logic over K and Σ on unranked trees* (for short: weighted uMSO^{ae}-logic) is defined to be the smallest set F such that:

1. F contains all the *atomic formulas* k, $\mathrm{label}_\sigma(x)$, $\mathrm{lf}(x)$, $\mathrm{rt}(x)$, $\mathrm{desc}(x,y)$, $(x \leq y)$, $\mathrm{oldest.ch}(x,y)$, $\mathrm{youngest.ch}(x,y)$, $\mathrm{ne}(x,y)$, and $x \in X$ and their negations (except the negation of k), and

2. if φ and ψ are in F, then also $\varphi \vee \psi$, $\varphi \wedge \psi$, $\exists x.\varphi$, $\exists X.\varphi$ are in F, and if φ contains no universal quantifier, then also $\forall x.\varphi$ is in F

where $k \in K$, $\sigma \in \Sigma$, x, y are first order variables, and X is a second order variable.

Recall that in Section 4 we have introduced the macros which we now use to define the semantics of our new logic.

Definition 8.2. Let $\varphi \in \mathrm{uMSO}^{\mathrm{ae}}(K, \Sigma)$ and $\mathcal{V}$ be a finite set of variables containing $\mathrm{Free}(\varphi)$. We define the *semantics* of φ as the formal tree series $[\![\varphi]\!]_{\mathcal{V}} \in K\langle\!\langle U_{\Sigma_{\mathcal{V}}}\rangle\!\rangle$ exactly as in Definition 4.2 with the following additions for the new atomic formulas.

$$\begin{aligned}
[\![\mathrm{lf}(x)]\!]_{\mathcal{V}} &= [\![\mathrm{leaf}(x)]\!]_{\mathcal{V}} \\
[\![\mathrm{oldest.ch}(x,y)]\!]_{\mathcal{V}} &= [\![\mathrm{oldest.child}(x,y)]\!]_{\mathcal{V}} \\
[\![\mathrm{youngest.ch}(x,y)]\!]_{\mathcal{V}} &= [\![\mathrm{youngest.child}(x,y)]\!]_{\mathcal{V}} \\
[\![\mathrm{ne}(x,y)]\!]_{\mathcal{V}} &= [\![\mathrm{next}(x,y)]\!]_{\mathcal{V}}
\end{aligned}$$

and for every $(t, \rho) \in U_{\Sigma_{\mathcal{V}}}$, let

$$\big([\![\mathrm{rt}(z)]\!], (t,\rho)\big) = \begin{cases} 1 & \text{if } (t,\rho) \text{ is valid and } \rho(z) = \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, if φ is of the form $\mathrm{lf}(x)$, $\mathrm{rt}(x)$, $\mathrm{oldest.ch}(x,y)$, $\mathrm{youngest.ch}(x,y)$, or $\mathrm{ne}(x,y)$ and $s \in U_{\Sigma_{\mathcal{V}}}$ let

$$([\![\neg\varphi]\!]_{\mathcal{V}}, s) = \begin{cases} 1 & \text{if } s \text{ is valid and } ([\![\varphi]\!]_{\mathcal{V}}, s) = 0 \\ 0 & \text{otherwise.} \end{cases}$$

First we prove a consistency result for the semantics of our new logic which is analogous to Lemma 4.3.

Lemma 8.3. Let $\varphi \in \mathrm{uMSO}(K, \Sigma)$ and $\mathcal{V}$ a finite set of variables containing $\mathrm{Free}(\varphi)$. Then, for every $(t, \rho) \in U^v_{\Sigma_{\mathcal{V}}}$, we have $([\![\varphi]\!]_{\mathcal{V}}, (t,\rho)) = ([\![\varphi]\!], (t, \rho|_{\mathrm{Free}(\varphi)}))$. Moreover, $[\![\varphi]\!]$ is a recognizable step function iff $[\![\varphi]\!]_{\mathcal{V}}$ is a recognizable step function.

Proof. The first statement can be proved by induction on the structure of φ. For the final claim consider the mapping $\pi : \Sigma_{\mathcal{V}} \to \Sigma_\varphi$ such that $\pi(s) = (t, \rho|_{\mathrm{Free}(\varphi)})$ for every $s = (t, \rho) \in U_{\Sigma_{\mathcal{V}}}$. The first assertion implies that $[\![\varphi]\!]_{\mathcal{V}}$ assumes the same values on $U^v_{\Sigma_{\mathcal{V}}}$ as $[\![\varphi]\!]$ does on $U^v_{\Sigma_\varphi}$; also $([\![\varphi]\!]_{\mathcal{V}}, s) = 0$ for any $s \in U_{\Sigma_\varphi} \setminus U^v_{\Sigma_\varphi}$. Hence it suffices to show for each $k \in K$ that $[\![\varphi]\!]^{-1}_{\mathcal{V}}(k)$ is recognizable iff $[\![\varphi]\!]^{-1}(k)$ is recognizable. Indeed, if $k \neq 0$, we have $[\![\varphi]\!]^{-1}_{\mathcal{V}}(k) = \pi^{-1}([\![\varphi]\!]^{-1}(k)) \cap U^v_{\Sigma_{\mathcal{V}}}$, and if $k = 0$, then $[\![\varphi]\!]^{-1}_{\mathcal{V}}(0) = \pi^{-1}([\![\varphi]\!]^{-1}(0))$. Clearly, $U^v_{\Sigma_\varphi}$ is recognizable. Hence, if $[\![\varphi]\!]^{-1}(k)$ is recognizable, so is $[\![\varphi]\!]^{-1}_{\mathcal{V}}(k)$.
Conversely, let $[\![\varphi]\!]^{-1}_{\mathcal{V}}(k)$ be recognizable. Let $F \subseteq U^v_{\Sigma_{\mathcal{V}}}$ be the set of those trees $s = (t, \rho)$ such that $\rho(x) = \varepsilon$ (resp. $\rho(X) = \{\varepsilon\}$) for every variable x (resp. X) in $\mathcal{V} \setminus \mathrm{Free}(\varphi)$. Then F is a recognizable unranked tree language and for every $(t, \rho') \in U^v_{\Sigma_\varphi}$ there is a unique $(t, \rho) \in F$ such that $\pi((t,\rho)) = (t, \rho')$. Then $[\![\varphi]\!]^{-1}(k) = \pi([\![\varphi]\!]^{-1}_{\mathcal{V}}(k) \cap F)$ by our first assertion, showing that $[\![\varphi]\!]^{-1}(k)$ is recognizable. $\square$

Next we give an example (essentially from [DR06] for words) to show that for almost existential sentences φ, in general $[\![\varphi]\!] : U_\Sigma \to K$ is not recognizable.

Example 8.4. Let $K = (\mathbb{N}, +, \cdot, 0, 1)$ and $\varphi = \forall x.\exists y.1$. Clearly, φ is almost existential, and $([\![\varphi]\!], t) = |t|^{|t|}$ for any $t \in U_\Sigma$ where $|t|$ denotes the size (= the number of nodes) of t. Now choose any uwta M. Let us call a tree $t \in U_\Sigma$ word-like, if each node of t has at most one descendant. It follows that there is a constant $C \in \mathbb{N}$ such that for each word-like tree $t \in U_\Sigma$ we have $(S_M, t) \leq C^{|t|}$. Hence $(S_M, t) < ([\![\varphi]\!], t)$ for sufficiently large word-like $t \in U_\Sigma$, showing $[\![\varphi]\!] \notin K^{\mathrm{rec}}\langle\!\langle U_\Sigma \rangle\!\rangle$. For comparison with Theorems 4.7 and 8.10 given below we observe that φ is not restricted and that $(\mathbb{N}, +, 0)$ is not locally finite.

Next we will prove several closure properties of the class of recognizable unranked step functions. First we show that the semantics of the atomic formulas in $\mathrm{uMSO}^{\mathrm{ae}}(K, \Sigma)$ are recognizable unranked step functions.

Lemma 8.5. Let $\varphi \in \mathrm{uMSO}^{\mathrm{ae}}(K, \Sigma)$ be an atomic formula or the negation of an atomic formula. The $[\![\varphi]\!]$ is a recognizable unranked step function.

Proof. Assume that $\mathcal{V}$ contains $\mathrm{Free}(\varphi)$. Let $\varphi = k$ for some $k \in K$. Then $[\![\varphi]\!]_\mathcal{V} = k \cdot \mathbb{1}_{U_\Sigma}$ which is a recognizable unranked step function. Next, it is obvious that, for every $\sigma \in \Sigma$ and variables x, y, X, the following unranked tree languages are recognizable:

$$
\begin{array}{lcll}
\mathcal{L}_1 & = & \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid t(\rho(x)) = \sigma\} & (\text{for } \mathrm{label}_\sigma(x)) \\
\mathcal{L}_2 & = & \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid \rho(x).i \notin \mathrm{pos}(t) \text{ for every } i \geq 1\} & (\text{for } \mathrm{lf}(x)) \\
\mathcal{L}_3 & = & \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid \rho(x) = \varepsilon\} & (\text{for } \mathrm{rt}(x)) \\
\mathcal{L}_4 & = & \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid \rho(y) = \rho(x).i \text{ for some } i \geq 1\} & (\text{for } \mathrm{desc}(x,y)) \\
\mathcal{L}_5 & = & \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid \rho(x) = \rho(y) = \varepsilon \text{ or there are} & \\
 & & w \in \mathrm{pos}(t), i,j \geq 1 \text{ such that } i \leq j, \rho(x) = w.i, \rho(y) = w.j\} & (\text{for } x \leq y) \\
\mathcal{L}_6 & = & \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid \rho(y) = \rho(x).1\} & (\text{for } \mathrm{oldest.ch}(x,y)) \\
\mathcal{L}_7 & = & \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid \text{ there is an } m\text{: } \rho(y) = \rho(x).m & \\
 & & \text{and for every } m' > m\text{: } \rho(x).m' \notin \mathrm{pos}(t)\} & (\text{for } \mathrm{youngest.ch}(x,y)) \\
\mathcal{L}_8 & = & \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid \text{ there are } w \in \mathrm{pos}(t) \text{ and } i \geq 1\text{:} & \\
 & & \rho(x) = w.i, \rho(y).(i+1)\} & (\text{for } \mathrm{ne}(x,y)) \\
\mathcal{L}_9 & = & \{(t,\rho) \in U^v_{\Sigma_\mathcal{V}} \mid \rho(x) \in \rho(X)\} & (\text{for } x \in X)
\end{array}
$$

Then we have, e.g., $[\![\mathrm{label}_\sigma(x)]\!]_\mathcal{V} = 1 \cdot \mathbb{1}_{\mathcal{L}_1}$, and thus $[\![\mathrm{label}_\sigma(x)]\!]_\mathcal{V}$ is a recognizable unranked step function. Let $\overline{\mathcal{L}}$ denote the complement of the tree language $\mathcal{L}$. Then we have that $[\![\neg\mathrm{label}_\sigma(x)]\!]_\mathcal{V} = 1 \cdot \mathbb{1}_{\overline{\mathcal{L}_1} \cap U}$ where $U = U^v_{\Sigma_\mathcal{V}}$. Since the class of recognizable unranked tree languages is closed under complementation, we immediately obtain that also $[\![\neg\mathrm{label}_\sigma(x)]\!]_\mathcal{V}$ is a recognizable unranked step function. Similarly we obtain that $[\![\varphi]\!]$ is a recognizable unranked step function for the other atomic formulas and their negations. $\square$

Next we wish to show that the class of recognizable unranked step functions is closed under relabelings, if the additive monoid of K is locally finite. In its turn, the proof of this property will be based on the following backround result.

Lemma 8.6. Let Σ be a ranked alphabet.

1. Let $S \in \mathbb{Z}^{\mathrm{rec}}\langle\!\langle U_\Sigma \rangle\!\rangle$ and $a, b \in \mathbb{Z}$ with $b \neq 0$. Then $S^{-1}(a + b\mathbb{Z})$ is a recognizable unranked tree language.

2. Let $S \in \mathbb{N}^{\mathrm{rec}}\langle\!\langle U_\Sigma \rangle\!\rangle$ and $a \in \mathbb{N}$. Then the language $S^{-1}(a)$ is recognizable.

Proof. Let K be $\mathbb{Z}$ or $\mathbb{N}$. Choose a uwta M over K and Σ which accepts S. By Lemma 6.1 there is a binary wta M' over $enc(\Sigma)$ such that $S_M = S_{M'} \circ enc$. Thus for any subset $Z \subseteq K$ we have $S_M^{-1}(Z) = enc^{-1}(S_{M'}^{-1}(Z))$. Let $Z = a + b\mathbb{Z}$ if $K = \mathbb{Z}$ and $Z = \{a\}$ if $K = \mathbb{N}$. Then by Lemma 6.3 of [DV06], $S_{M'}^{-1}(Z) \subseteq T_{enc(\Sigma)}$ is a recognizable tree language. Now apply the above equality and Proposition 5.2 to obtain that $S_M^{-1}(Z)$ is recognizable. □

Now we prove that the class of recognizable unranked step functions is closed under relabelings if the additive monoid of the underlying semiring is locally finite.

Lemma 8.7. Let $(K, +, 0)$ is locally finite. Moreover, let $S \in K\langle\langle U_\Sigma\rangle\rangle$ and $\tau : K\langle\langle U_\Sigma\rangle\rangle \to K\langle\langle U_\Delta\rangle\rangle$ be a relabeling. If S is a recognizable unranked step function, then also $\tau(S)$ is a recognizable unranked step function.

Proof. Let S have the form $S = \sum_{j=1}^{n} k_j \cdot \mathbb{1}_{\mathcal{L}_j}$ with $n \in \mathbb{N}$, $k_j \in K$, and recognizable unranked tree languages $\mathcal{L}_j \subseteq U_\Sigma$ $(j = 1, \ldots, n)$ which form a partition of U_Σ. Moreover, let τ be generated by the mapping $\tau : \Sigma \to \mathcal{P}(\Delta)$. We define the following abbreviation for repeated addition in K: for every $k \in K$ and $n \geq 0$, we define $0 \odot k = 0$ (of K) and $(n+1) \odot k = k + n \odot k$. Thus, $n \odot k = k + \ldots + k$ with n times k.

Now, for every $t \in U_\Delta$, let $m_j(t) := |\tau^{-1}(t) \cap \mathcal{L}_j\,|$. Then $(\tau(S), t) = \sum_{j=1}^{n} m_j(t) \odot k_j$. For each $j \in \{1, \ldots, n\}$, the submonoid of $(K, +, 0)$ generated by $\{k_j\}$ if finite. Choose a minimal $a_j \in \mathbb{N}$ such that $a_j \odot k_j = (a_j + x) \odot k_j$ for some $x > 0$ and let b_j be the smallest such $x > 0$. Then $\langle k_j\rangle = \{0, k_j, 2 \odot k_j, \ldots, (a_j + b_j - 1) \odot k_j\}$, and for each $t \in U_\Delta$, we have $m_j(t) \odot k_j = d_j(t) \odot k_j$ for some uniquely determined $d_j(t) \in \mathbb{N}$ with $0 \leq d_j(t) \leq a_j + b_j - 1$. Note that if $0 \leq d < a_j$, then $m_j(t) \odot k_j = d \odot k_j$ iff $m_j(t) = d$, and if $a_j \leq d < a_j + b_j$, then $m_j(t) \odot k_j = d \odot k_j$ iff $m_j(t) \in d + b_j\,\mathbb{N}$. For each $0 \leq d < a_j + b_j$ and $1 \leq j \leq n$ let $M_d^j := \{t \in U_\Delta \mid d_j(t) = d\}$. Then

$$(\tau(S), t) = \sum_{j=1}^{n} d_j(t) \odot k_j = \sum_{d_1,\ldots,d_n : 0 \leq d_j < a_j + b_j (j=1,\ldots,n)} (d_1 \odot k_1) \cdot \ldots \cdot (d_n \odot k_n) \cdot (\mathbb{1}_{M_{d_1}^1 \cap \cdots \cap M_{d_n}^n}, t).$$

For every $1 \leq j \leq n$, let $\mathbb{1}'_{\mathcal{L}_j} : U_\Sigma \to \mathbb{N}$ be the characteristic function of $\mathcal{L}_j$ with values $0, 1 \in \mathbb{N}$. By Lemma 3.5 the series $S_j := \tau(\mathbb{1}'_{\mathcal{L}_j}) : U_\Delta \to \mathbb{N}$ is recognizable, and $(S_j, t) = \sum_{s \in \tau^{-1}(t)} (\mathbb{1}'_{\mathcal{L}_j}, s) = m_j(t)$ $(t \in U_\Delta)$. Hence $M_d^j = \{t \in U_\Delta \mid m_j(t) = d\} = S_j^{-1}(d)$ if $0 \leq d < a_j$, and $M_d^j = \{t \in U_\Delta \mid m_j(t) \in d + b_j\,\mathbb{N}\} = S_j^{-1}(d + b_j\,\mathbb{N})$ if $a_j \leq d < a_j + b_j$. In each case, M_d^j is recognizable by Lemma 8.6. Hence $\tau(S)$ is a recognizable unranked step function. □

As a consequence, we can derive that the class of recognizable unranked step functions is closed under existential quantifications.

Lemma 8.8. Let $(K, +, 0)$ be locally finite and let $\varphi \in \mathrm{uMSO}^{\mathrm{ae}}(K, \Sigma)$ such that $[\![\varphi]\!]$ is a recognizable unranked step function. Then $[\![\exists X.\varphi]\!]$ and $[\![\exists x.\varphi]\!]$ are also recognizable unranked step functions.

Proof. $\underline{[\![\exists X.\varphi]\!]}$: Let $\mathcal{V} = \mathrm{Free}(\exists X.\varphi)$; note that $X \notin \mathcal{V}$. Consider the relabeling $\pi : U_{\Sigma_{\mathcal{V} \cup \{X\}}} \to U_{\Sigma_\mathcal{V}}$ defined by erasing the X-row, i.e., $\pi(t, \rho) = (t, \rho|_\mathcal{V})$. Let $s \in U_{\Sigma_\mathcal{V}}$. We note that s is valid iff $s[X \to I]$ is valid for every $I \subseteq \mathrm{pos}(s)$. Hence, by definition of $[\![\exists X.\varphi]\!]$ and of relabeling, we have for every $s \in U_{\Sigma_\mathcal{V}}$:

$$([\![\exists X.\varphi]\!], s) = \sum_{I \subseteq \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}, s[X \to I]) = \sum_{t \in \pi^{-1}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}, t) = (\pi([\![\varphi]\!]_{\mathcal{V}\cup\{X\}}), s).$$

Thus $[\![\exists X.\varphi]\!] = \pi([\![\varphi]\!]_{\mathcal{V}\cup\{X\}})$. Since $\mathrm{Free}(\varphi) \subseteq \mathcal{V} \cup \{X\}$ and $[\![\varphi]\!]$ is recognizable, by Lemma 4.3 also $[\![\varphi]\!]_{\mathcal{V}\cup\{X\}}$ is recognizable. Hence by Lemma 8.7, also $[\![\exists X.\varphi]\!]$ is a recognizable step function.

$[\![\exists x.\varphi]\!]$: Let $\mathcal{V} = \mathrm{Free}(\exists x.\varphi)$; note that $x \notin \mathcal{V}$. Consider the relabeling $\pi : U_{\Sigma_{\mathcal{V}\cup\{x\}}} \to U_{\Sigma_\mathcal{V}}$ which is defined by erasing the x-row, as above. Let $s \in U_{\Sigma_\mathcal{V}}$. We note that s is valid iff $s[x \to w]$ is valid for every $w \in \mathrm{pos}(s)$. Then we have for every $s \in U_{\Sigma_\mathcal{V}}$:

$$\begin{aligned}
& ([\![\exists x.\varphi]\!], s) \\
= \;& \textstyle\sum_{w \in \mathrm{pos}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, s[x \to w]) \\
=^{(1)} \;& \textstyle\sum_{\substack{t \in \pi^{-1}(s) \\ t \text{ is valid}}} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, t) \\
=^{(2)} \;& \textstyle\sum_{t \in \pi^{-1}(s)} ([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, t) \\
= \;& (\pi([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}), s).
\end{aligned}$$

At (1) observe that, if s is valid, there is a bijection between the index sets $\mathrm{pos}(s)$ and $U^v_{\Sigma_{\mathcal{V}\cup\{x\}}} \cap \pi^{-1}(s)$; if s is not valid, then the set $U^v_{\Sigma_{\mathcal{V}\cup\{x\}}} \cap \pi^{-1}(s)$ is empty. For (2), note that $([\![\varphi]\!]_{\mathcal{V}\cup\{x\}}, t) = 0$ for every non-valid t. Thus $[\![\exists x.\varphi]\!] = \pi([\![\varphi]\!]_{\mathcal{V}\cup\{x\}})$. Since $\mathrm{Free}(\varphi) \subseteq \mathcal{V} \cup \{x\}$ and $[\![\varphi]\!]$ is a recognizable unranked step function, by Lemma 4.3 also $[\![\varphi]\!]_{\mathcal{V}\cup\{x\}}$ is a recognizable unranked step function. Hence by Lemma 8.7, also $[\![\exists x.\varphi]\!]$ is a recognizable unranked step function. ☐

We say that the formula $\varphi \in \mathrm{uMSO}^{\mathrm{ae}}(K, \Sigma)$ is *restricted* if for every subformula $\forall x.\psi$ of φ, $[\![\psi]\!]$ is a recognizable step function. Now we show that this is always the case if $(K, +, 0)$ is locally finite.

Lemma 8.9. Let $(K, +, 0)$ locally finite and let $\varphi \in \mathrm{uMSO}^{\mathrm{ae}}(K, \Sigma)$. Then φ is restricted.

Proof. Let $\forall x.\psi$ be a subformula of φ. Then ψ does not contain a universal quantifier. We show that $[\![\psi]\!]$ is a recognizable unranked step function by induction of the structure of φ. The case that ψ is an atomic formula or the negation of an atomic formula holds by Lemma 8.5.

Now let $\psi = \psi_1 \vee \psi_2$ such that $[\![\psi]\!]$ and $[\![\psi_2]\!]$ are recognizable unranked step functions. Put $\mathcal{V} = \mathrm{Free}(\psi_1) \cup \mathrm{Free}(\psi_2)$. By Lemma 8.3, $[\![\psi_1]\!]_\mathcal{V}$ and $[\![\psi_2]\!]_\mathcal{V}$ are recognizable unranked step functions. Hence $[\![\psi]\!] = [\![\psi_1]\!]_\mathcal{V} + [\![\psi_2]\!]_\mathcal{V}$ is a recognizable unranked step function by Proposition 3.2. The case of conjunction can be dealt with analoguously.

The case of existential quantifiers is proved in Lemma 8.8. ☐

The goal of this section is to prove the third main result of this paper.

Theorem 8.10. *Let K be a commutative semiring such that the monoid $(K, +, 0)$ is locally finite. Then $K^{\mathrm{rec}}\langle\!\langle U_\Sigma \rangle\!\rangle = K^{\mathrm{aemso}}\langle\!\langle U_\Sigma \rangle\!\rangle$.*

Proof. First let M be a uwta such that $S = S_M$. By Lemma 6.1 there is a binary wta M' over K and $enc(\Sigma)$ such that $S_M = S_{M'} \circ enc(s)$. Next, using Corollary 4.10, there is an REMSO$(K, enc(\Sigma))$-formula φ such that $S_{M'} = [\![\varphi]\!]$ and φ has the form $\exists X_1 \ldots \exists X_n.\forall x.\theta$ where θ contains the subformula $\mathrm{root}(z) = \forall z. \bigwedge_{1\leq i\leq \max_\Sigma} \neg\mathrm{edge}_i(z,x)$ and only first order existential quantifications, and $[\![\theta]\!]$ is a recognizable step function.

Now we translate φ according to Definition 6.3 into the sentence $\varphi_\emptyset \in \mathrm{uMSO}(K,\Sigma)$ but we keep subformulas $((\mathrm{root}(z))_V$ obtained during the translation process, for any set V of variables. By Corollary 6.5, we have that $[\![\varphi_\emptyset]\!] = [\![\varphi]\!] \circ enc$.

Finally, we transform $\varphi_\emptyset$ equivalently into a $\mathrm{uMSO}^{\mathrm{ae}}(K,\Sigma)$-formula ψ as follows. In $\varphi_\emptyset$ we replace the macros $\mathrm{leaf}(x)$, $\mathrm{oldest.child}(x,y)$, $\mathrm{youngest.child}(x,y)$, and $\mathrm{next}(x,y)$ by the atomic formulas $\mathrm{lf}(x)$, $\mathrm{oldest.ch}(x,y)$, $\mathrm{youngest.ch}(x,y)$, and $\mathrm{ne}(x,y)$, respectively. Moreover, for every variable set V, we replace $(\mathrm{root}(z))_V$ by $\mathrm{rt}(z)$ if $z \notin V$, and by 0 if $z \in V$. Due to the definitions of the semantics of the atomic formulas $\mathrm{lf}(x)$, $\mathrm{oldest.ch}(x,y)$, $\mathrm{youngest.ch}(x,y)$, and $\mathrm{ne}(x,y)$, and by a computation for $\mathrm{root}(z)$ similar to the ones in Lemma 6.4 for the atomic formulas, we obtain that $[\![\varphi_\emptyset]\!] = [\![\psi]\!]$.

The sentence ψ resulting from this replacement, has the form $\exists X_1 \ldots \exists X_{2\cdot n}.\forall x.\theta'$ where θ' contains only first order existential quantifications. In particular, ψ is an element of $\mathrm{uMSO}^{\mathrm{ae}}(K,\Sigma)$ and hence S is $\mathrm{uMSO}^{\mathrm{ae}}(K,\Sigma)$-definable.

For the converse let $\varphi \in \mathrm{uMSO}^{\mathrm{ae}}(K,\Sigma)$. Since $(K,+,0)$ is locally finite, by Lemma 8.9 φ is restricted. Now we translate φ into a sentence $\varphi' \in \mathrm{uMSO}(K,\Sigma)$ as follows. Replace every occurrence of the atomic formulas $\mathrm{lf}(x)$, $\mathrm{oldest.ch}(x,y)$, $\mathrm{youngest.ch}(x,y)$, $\mathrm{ne}(x,y)$, and $\mathrm{rt}(x)$ by the macros $\mathrm{leaf}(x)$, $\mathrm{oldest.child}(x,y)$, $\mathrm{youngest.child}(x,y)$, $\mathrm{next}(x,y)$, and $\forall y.\neg\mathrm{desc}(y,x)$, respectively, which each have the same semantics as the original atomic formulas. Hence, $[\![\varphi]\!] = [\![\varphi']\!]$ and φ' is also restricted. By Theorem 7.6, $[\![\varphi']\!]$ is recognizable. □

Next we wish to derive a similar equivalence result for series over ranked alphabets Σ. We extend the syntax of $\mathrm{MSO}(K,\Sigma)$ by adding an atomic formula $\mathrm{root}(z)$, its semantics is defined by $\big([\![\mathrm{root}(z)]\!]_\mathcal{V}, (t,\rho)\big) = 1$ if (t,ρ) is valid and $\rho(z) = \varepsilon$, and $\big([\![\mathrm{root}(z)]\!]_\mathcal{V}, (t,\rho)\big) = 0$ otherwise. We call a formula φ over this extended syntax *almost existential*, if φ contains no universal set quantifier and whenever $\forall x.\psi$ is a subformula of φ, then ψ contains no universal quantifier. We let $K^{\mathrm{aemso}}\langle\langle T_\Sigma\rangle\rangle$ comprise all series $S : T_\Sigma \to K$ such that $S = [\![\varphi]\!]$ for some almost existential sentence φ over K and Σ. Then we have:

Theorem 8.11. *Let K be a commutative semiring such that the monoid $(K,+,0)$ is locally finite, and let Σ be a ranked alphabet. Then $K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle = K^{\mathrm{aemso}}\langle\langle T_\Sigma\rangle\rangle$.*

Proof. First, let $S \in K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle$. By Corollary 4.10, we have $S = [\![\varphi]\!]$ for a sentence φ of the form $\varphi = \exists X_1 \ldots \exists X_n.\forall x.\Theta$ where Θ is a formula over our extended syntax containing no universal quantifications. Hence φ is almost existential.

Conversely, let φ be an almost existential sentence; we claim that $[\![\varphi]\!] \in K^{\mathrm{rec}}\langle\langle T_\Sigma\rangle\rangle$. For this, we could proceed completely analogously to the argument for this implication of Theorem 8.10. Alternatively, we can argue as follows. We express the atomic formulas $\mathrm{edge}_i(x,y)$ and their negations in $\mathrm{uMSO}(K,\Sigma)$ as described after Definition 4.8; then we replace the formulas $\mathrm{oldest.child}(x,y)$, $\mathrm{next}(x,y)$ and $\mathrm{root}(z)$ by their semantic equivalents $\mathrm{oldest.ch}(x,y)$, $\mathrm{ne}(x,y)$ and $\mathrm{rt}(z)$, respectively. We obtain a sentence $\varphi \in \mathrm{uMSO}^{\mathrm{ae}}(K,\Sigma)$ whose semantics restricted to T_Σ coincides with $[\![\varphi]\!]$. By Theorem 8.10, there as a uwta M' over K and the (unranked) alphabet Σ such that $[\![\varphi']\!] = S_{M'}$. But now it is easy to define a wta M over K and the ranked alphabet Σ such that $S_{M'}$ and S_M agree on ranked trees. Hence $[\![\varphi]\!] = S_{M'}$, as needed. □

Next we turn to constructibility and decision problems.

Corollary 8.12. *Let K be a computable commutative semiring such that $(K, +, 0)$ is locally finite. Given a* uMSO$^{\mathrm{ae}}(K,\Sigma)$*-formula φ, we can effectively compute a uwta M such that $[\![\varphi]\!] = S_M$. If Σ is ranked, then we can effentively compute a wta M with $[\![\varphi]\!] = S_M$.*

Proof. We proceed by induction on the structure of φ. During the first steps where we are dealing with formulae ψ such that $[\![\psi[\![$ is a recognizable step function, say $[\![\psi]\!] = \sum_{i=1}^{n} k_i \cdot \mathbb{1}_{L_i}$, we always try to compute all values k_i and automata for the languages L_i $(i = 1, \ldots, n)$. For atomic formulae and their negations, this is clear. Assume, ψ, ψ' are of this form. By our constructions, we also obtain this form and the needed automata for $\psi \vee \psi', \psi \wedge \psi', \exists x.\psi$ and $\exists X.\psi$. Now consider $\forall x.\psi$. If Σ is ranked, since we know the values k_i and the automata for the languages L_i, we can apply the constructions of Lemma 5.5 of [DV06] to obtain a wta for $[\![\forall x.\psi]\!]$. If Σ is unranked, apply the construction of Definition 7.1 (also cf. Corollary 7.3), then the construction of the ranked case, followed by Lemma 7.5.

Now assume that ψ, ψ' are subformulas of φ and we have constructed uwta resp. wta M, M' with $[\![\psi]\!] = S_M$ and $[\![\psi']\!] = S_{M'}$. Then we also obtain uwta resp. wta for the formulas $\psi \vee \psi', \psi \wedge \psi', \exists x.\psi$ and $\exists X.\psi$ effectively by our constructions. □

Unfortunately, for such semirings K with $(K, +, 0)$ locally finite, the equality $[\![\varphi]\!] = [\![\varphi']\!]$ for two sentences $\varphi, \varphi' \in$ uMSO$^{\mathrm{ae}}(K,\Sigma)$ is in general undecidable. Consider $K = \mathit{Trop}$, the tropical semiring, and suppose there was a decision procedure for this equality. By the first part of the proof of Theorem 8.11, we would obtain a decision procedure for uwta M, M' of whether $S_M = S_{M'}$. But this is already impossible for weighted automata on words by Krob [Kro94].

9 The full weighted uMSO-logic

As noted before, easy examples like Example 8.4 show that there are MSO(K,Σ)-definable series which are not recognizable (compare [DV06]). In this section, we wish to show that for a large class of semirings K, all uMSO(K,Σ)-definable tree series are recognizable. Again let Σ denote an arbitrary alphabet.

If K is a semiring and $A \subseteq K$, we denote by $\langle A \rangle$ the subsemiring of K generated by A. Since multiplication is distributive over addition, we may construct $\langle A \rangle$ by first taking the closure A' of $A \cup \{0, 1\}$ under multiplication, then $\langle A \rangle$ is the closure of A' under addition. A semiring K is called *locally finite*, if each finitely generated subsemiring of K is finite. Clearly, a semiring $(K, +, \cdot, 0, 1)$ is locally finite iff both monoids $(K, +, 0)$ and $(K, \cdot, 1)$ are locally finite (apply the remark on $\langle A \rangle$ above). Example of locally finite semiring are the following:

- Any Boolean algebra $(B, \vee, \wedge, 0, 1)$ is locally finite.
- The max-min semiring $\mathbb{R}_{\max,\min} = (\mathbb{R}_+ \cup \{\infty\}, \max, \min, 0, \infty)$ of positive reals, used for maximum capacity problems of networks, is locally finite. This semiring is isomorphic to the semiring $[0, 1], \max, \min, 0, 1)$ used for fuzzy automata.

- More generally, any distributive lattice $(L, \vee, \wedge, 0, 1)$ with smallest element 0 and largest element 1 is locally finite.
- The Lukasiewicz semiring $([0,1], \max, \otimes, 0, 1)$ where $x \otimes y = \max\{0, x+y-1\}$ occurring in multivalued logics is locally finite.
- Examples of infinite but locally finite fields are provided by the algebraic closures of the finite fields $\mathbb{Z}/p\mathbb{Z}$, for any prime p [Rom95].

As a consequence of our preceding results and of the corresponding result of [DV06] for ranked alphabets we obtain:

Corollary 9.1. Let K be a locally finite commutative semiring and let $S \in K^{\mathrm{rec}}\langle\langle U_\Sigma \rangle\rangle$. Then S is a recognizable step function.

Proof. Let M be a uwta over K and Σ which accepts S. Then by Lemma 6.1 there is a wta M' over $enc(\Sigma)$ such that $S_M = S_{M'} \circ enc$. By Lemma 6.1 of [DV06], the tree language $S_{M'}$ is a recognizable step function. Hence S_M assumes, as $S_{M'}$, only finitely many values. Moreover, for any $k \in K$ we have $(S_M)^{-1}(k) = enc^{-1}((S_{M'})^{-1}(k))$ and $(S_{M'})^{-1}(k)$ is a recognizable tree language. So by Proposition 5.2, $S_M^{-1}(k) \subseteq U_\Sigma$ is recognizable, and the result follows. $\square$

Now we will show for locally finite and commutative semirings that the semantics of all sentences of our unranked weighted MSO-logic are recognizable series.

Theorem 9.2. *Let K be a locally finite commutative semiring. Then $K^{\mathrm{rec}}\langle\langle U_\Sigma \rangle\rangle = K^{\mathrm{mso}}\langle\langle U_\Sigma \rangle\rangle$.*

Proof. The inclusion $K^{\mathrm{rec}}\langle\langle U_\Sigma \rangle\rangle \subseteq K^{\mathrm{mso}}\langle\langle U_\Sigma \rangle\rangle$ is immediate by Theorem 6.6. For the converse, we follow the proof of Theorem 7.6. Let $S = [\![\psi]\!]$ for some sentence $\psi \in \mathrm{uMSO}(K, \Sigma)$. Construct the sentence $\psi' \in \mathrm{MSO}(K, \mathrm{enc}(\Sigma))$ as in Lemma 7.2. Since K is locally finite, by Theorem 6.5 of [DV06] there is a wta M' over $\mathrm{enc}(\Sigma)$ and K such that $[\![\psi']\!] = S_{M'}$. Now construct the uwta M over Σ and K as in Lemma 7.5. Then $S = S_M$ is recognizable. $\square$

Next we turn to constructibility and decision problems for computable locally finite semirings K.

Corollary 9.3. *Let K be a computable locally finite commutative semiring. Given a* $\mathrm{uMSO}(K, \Sigma)$*-formula ψ, we can effectively compute a uwta M such that $[\![\psi]\!] = S_M$.*

Proof. We follow the above proof of Theorem 9.2. Since the constructions for Lemma 7.2 and Lemma 7.5 are obviously effective, we only have to show that given a sentence $\psi' \in \mathrm{MSO}(K, \mathrm{enc}(\Sigma))$, we can find the equivalent wta M' effectively. But this follows directly from Corollary 6.5 of [DV06]. $\square$

Corollary 9.4. *Let K be a computable locally finite commutative semiring. It is decidable whether two given* $\mathrm{uMSO}(K, \Sigma)$*-formulas φ and ψ satisfy $[\![\varphi]\!] = [\![\psi]\!]$.*

Proof. We may assume that $\mathrm{Free}(\varphi) = \mathrm{Free}(\psi)$. We form the corresponding formulas φ' and ψ' over $\mathrm{enc}(\Sigma)$ and K as in Lemma 7.2. By Lemma 7.2, we have $[\![\varphi]\!] = [\![\psi]\!]$ iff $[\![\varphi' \wedge \mathrm{Enc}_\varphi]\!] = [\![\psi' \wedge \mathrm{Enc}_\psi]\!]$. But by Corollary 6.7 of [DV06], the latter equality is decidable. $\square$

10 An XML example

In this section we show how formulas of weighted uMSO-logic can be used to specify quantitative queries in XML-oriented databases. In fact, every used formula is almost existential. For this purpose, let us assume that we want to maintain our private database in which we have stored all the bibtex entries which are relevant for our work. The syntax of such a database `bibliography` can be easily expressed by the following document type definition (DTD) [ABS00] (here we only show a fragment):

```
<!DOCTYPE biblography [
   <!ELEMENT bibliography (entry)*>
   <!ELEMENT entry        ((key)+,article|book)>
   <!ELEMENT key          (#PCDATA)>
   <!ELEMENT article      (mandatory,(empty|optional)>
   <!ELEMENT mandatory    (author*,title,journal,year)>
   <!ELEMENT author       (#PCDATA)>
   <!ELEMENT title        (#PCDATA)>
   ...
   <!ELEMENT empty         EMPTY>
   <!ELEMENT optional      (volume,pages)>
   ...
]>
```

In the DTD the comma, vertical bar, and star represent the sequence, alternative, resp. iteration of syntactic constructs; at `(#PCDATA)` an arbitrary text is allowed. This DTD specifies a set of XML-documents, in particular, a set of databases with bibtex entries; note that an entry can have any number of keys. An example of such a (very small) database with three entries is as follows:

```
<biblography>
   <entry>
       <key> car71 </key> <key> Carre71 </key> <key> BAC71 </key>
       <article>
           <mandatory>
              <author> B.A. Carr\'e </author>
              <title> An algebra for network routing problems </title>
              <journal> J. Inst. Math. Applic. </journal>
              <year> 1971 </year>
           </mandatory>
           <empty> EMPTY </empty>
       </article>
   </entry>
   <entry>
       <key> nevsch02 </key>
       <article>
           <mandatory>
              <author>  F. Neven </author> <author> Th. Schwentick </author>
              <title>  Query automata </title>
              <journal> Theoret. Comput. Sci. </journal>
              <year> 2002 </year>
           </mandatory>
           <empty> EMPTY </empty>
       </article>
   </entry>
```

```
    <entry>
        <key> knu77 </key> <key> Knuth77 </key>
        <article>
            <mandatory>
               <author> D.E. Knuth </author>
               <title> A generalization of Dijkstra's algorithm </title>
               <journal> Information Processing Letters </journal>
               <year> 1977 </year>
            </mandatory>
            <optional>
               <volume> 6 </volume>
               <pages> 1-5 </pages>
            </optional>
         </article>
    </entry>
 </bibliography>
```

In the usual way we can view XML-documents of our DTD `bibliography` as unranked trees. In Figure 11 the unranked tree which corresponds to our example XML-database is shown partially.

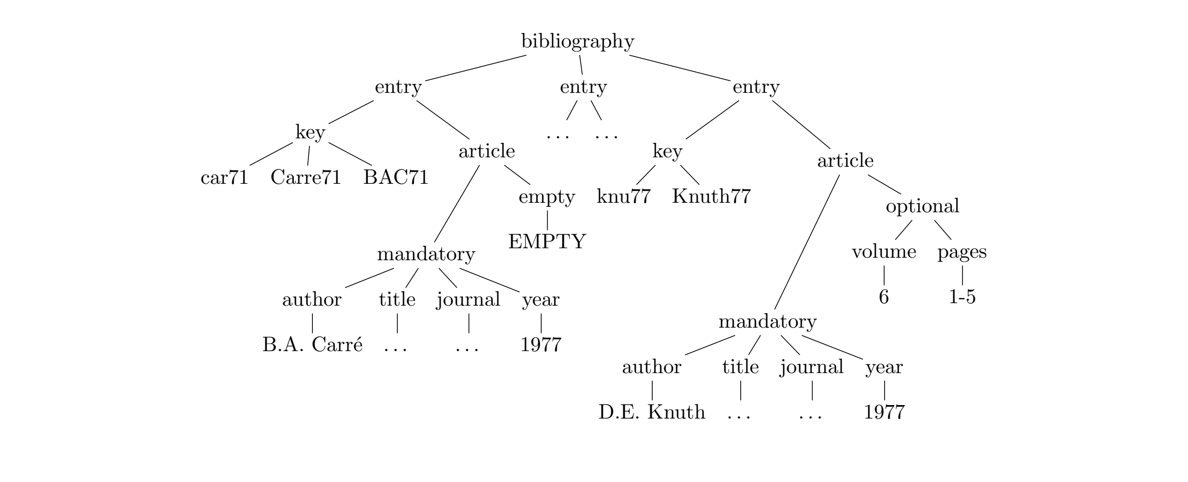

Figure 11: The unranked tree corresponding to the XML-database.

Our example database shows two article entries which contain the mandatory pieces of information only (*incomplete entries*), and another one with the mandatory and the optional pieces of information (*complete entry*). Now let us assume that we would like to complete our database in the sense of adding the optional pieces of information to every incomplete article entry. However, before doing so, we would like to estimate the effort for this maintenance, i.e., count the number of incomplete bibtex entries. For this purpose we can use the following formula of weighted uMSO-logic:

$$\texttt{how-many-incomplete?} \;=\; \exists x.\ \mathrm{label}_{\texttt{article}}(x) \wedge \texttt{incomplete}(x)$$

$$\texttt{incomplete}(x) \;=\; \exists y.\ \big(\mathrm{desc}(x,y) \;\wedge\; \mathrm{label}_{\texttt{empty}}(y)\big).$$

If we interpret our formula `how-many-incomplete?` on the unranked tree of Figure 11, then we obtain the value 2 (assuming that we use the usual semiring of natural numbers).

To show another use of weighted uMSO-logic, let us now assume that the document type definition as shown above is extended as follows:

```
<!ELEMENT book         (mandatory-b,(empty-b|optional-b)>
<!ELEMENT mandatory-b  (author*,title,publisher,year)>
<!ELEMENT empty-b      EMPTY>
<!ELEMENT optional-b   (volume,edition,summary)>
<!ELEMENT summary      (#PCDATA)>
```

In the part `optional-b` a short summary of the book should occur. Now let us again try to estimate the effort for completing bibtex entries. Clearly, to write a summary of a book takes much more effort than just to add the optional information pieces of an article. So let us describe the corresponding efforts by k and m (time units), respectively, for some natural numbers k and m. Then the estimation of the total effort is computed by the following formula of uMSO-logic:

$$\texttt{how-much-effort?} \;=$$

$$\exists x.\ \big(\mathrm{label}_{\texttt{book}}(x)\ \wedge\ \texttt{incomplete}(x)\ \wedge\ k\big)\ \vee\ \big(\mathrm{label}_{\texttt{article}}(x)\ \wedge\ \texttt{incomplete}(x)\ \wedge\ m\big).$$

If, e.g., our database contains 20 incomplete book entries and 500 incomplete article entries, then the interpretation of `how-much-effort?` on the corresponding unranked tree would yield the effort $20 \cdot k + 500 \cdot m$ (time units).

As as final example, we would like to count the number of `article`-entries with k authors where k is any natural number. This can be achieved by interpreting the following formula of $\mathrm{uMSO}(\Sigma, K)$ over the unranked tree:

$$\texttt{number-k-authored?} \;=\; \exists x.\, \mathrm{label}_{\texttt{mandatory}}(x) \wedge \texttt{has-k-authors}(x)$$

$$\texttt{has-k-authors}(x) \;=$$

$$\exists y_1.\ldots\exists y_k.\, \mathrm{oldest.child}(x, y_1) \wedge \bigwedge\nolimits_{1\le i\le k-1}(\mathrm{label}_{\texttt{author}}(y_i) \wedge \mathrm{next}(y_i, y_{i+1}))$$
$$\wedge\, \mathrm{label}_{\texttt{author}}(y_k) \wedge \mathrm{youngest.child}(x, y_k).$$

References

[ABS00] S. Abiteboul, P. Bunemann, and D. Suciu. *Data on the Web*. Morgan Kaufmann Publishers, San Francisco, California, 2000.

[BMW01] A. Brüggemann-Klein, M. Murata, and D. Wood. Regular tree and regular hedge languages over unranked alphabets: Version 1, April 2001. Technical Report HKUST-TCSC-2001-0, The Hongkong University of Science and Technology, 2001.

[Bor04] B. Borchardt. A Pumping Lemma and Decidability Problems for Recognizable Tree Series. *Acta Cybernet.*, 16(4):509–544, 2004.

[Boz91] S. Bozapalidis. Effective construction of the syntactic algebra of a recognizable series on trees. *Acta Informatica*, 28:351–363, 1991.

[Boz99] S. Bozapalidis. Equational elements in additive algebras. *Theory of Computing Systems*, 32(1):1–33, 1999.

[BPSM98] T. Bray, J. Paoli, and C.M. Sperberg-McQueen. Extensible Markup Language (XML 1.0 W3C Recommendation. `http://www.w3.org/TR/REC-xml/.`, 1998.

[BR82] J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoret. Comput. Sci.*, 18(2):115–148, 1982.

[BR88] J. Berstel and Ch. Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS-Monographs*. Springer-Verlag, 1988.

[Büc60] J. R. Büchi. Weak second–order arithmetic and finite automata. *Z. Math. Logik Math.*, 6:66–92, 1960.

[BW98] A. Brüggemann-Klein and D. Wood. Regular tree languages over non-ranked alphabets. available at `http://citeseer.ist.psu.edu/br98regular.html`, 1998.

[Car71] B. A. Carré. An algebra for network routing problems. *J. Inst. Maths. Applics.*, 7:273–294, 1971.

[CDG+97] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata, 1997.

[DG05] M. Droste and P. Gastin. Weighted automata and weighted logics. In *Automata, Languages and Programming – 32nd International Colloquium, ICALP 2005, Lisbon, Portugal*, volume 3580 of *LNCS*, pages 513–525. Springer-Verlag, 2005. Full version: Theoret. Comput. Sci. in print.

[Don70] J. Doner. Tree acceptors and some of their applications. *J. Comput. System Sci.*, 4:406–451, 1970.

[DPV05] M. Droste, Chr. Pech, and H. Vogler. A Kleene theorem for weighted tree automata. *Theory of Computing Systems*, 38:1–38, 2005.

[DR06] M. Droste and G. Rahonis. Weighted automata and weighted logics with discounting, 2006. preprint.

[DV06] M. Droste and H. Vogler. Weighted tree automata and weighted logics. *Theoret. Comput. Sci.*, 366:228–247, 2006.

[Eil74] S. Eilenberg. *Automata, Languages, and Machines – Volume A*, volume 59 of *Pure and Applied Mathematics*. Academic Press, 1974.

[GS84] F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.

[GS97] F. Gécseg and M. Steinby. Tree languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, chapter 1, pages 1–68. Springer-Verlag, 1997.

[Kro94] D. Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *Intern. J. of Information and Computation*, 4:405–425, 1994.

[KS86] W. Kuich and A. Salomaa. *Semirings, Automata, Languages.* EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1986.

[KSS03] N. Klarlund, Th. Schwentick, and D. Suciu. XML: Models, Schemas, Types, Logics, and Queries. In *Logics for Engineering Applications on Databases*, pages 1–41. Springer-Verlag, 2003.

[Küh06] A. Kühnemann. personal communication. TU Dresden, 2006.

[Kui97] W. Kuich. Semirings and formal power series: Their relevance to formal languages and automata. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Vol. 1*, chapter 9, pages 609–677. Springer-Verlag, 1997.

[Kui98] W. Kuich. Formal power series over trees. In S. Bozapalidis, editor, *3rd International Conference on Developments in Language Theory, DLT 1997, Thessaloniki, Greece, Proceedings*, pages 61–101. Aristotle University of Thessaloniki, 1998.

[Lib05] L. Libkin. Logics for unranked trees: an overview. In L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata, Languages and Programming: 32nd International Colloquium, ICALP 2005, Lisbon, Portugal*, volume 3580 of *LNCS*, pages 35–50. Springer-Verlag, 2005.

[Nev02] F. Neven. Automata, logic, and XML. In J. Bradfield, editor, *Computer Science Logic: 16th International Workshop, CSL 2002, Edinburgh, Scotland, UK*, volume 2471 of *LNCS*, pages 2–26. Springer-Verlag, 2002.

[NS02] F. Neven and Th. Schwentick. Query automata over finite trees. *Theoret. Comput. Sci.*, 275:633–674, 2002.

[Rom95] S. Roman. *Field Theory.* Springer-Verlag, 1995.

[Sch61] M.P. Schützenberger. On the definition of a family of automata. *Inf. and Control*, 4:245–270, 1961.

[Sei90] H. Seidl. Deciding equivalence of finite tree automata. *SIAM Journal of Computing*, 19(3):424–437, 1990.

[SS78] A. Salomaa and M. Soittola. *Automata-Theoretic Aspects of Formal Power Series.* Texts and Monographs in Computer Science, Springer-Verlag, 1978.

[SSM03] H. Seidl, T. Schwentick, and A. Muscholl. Numerical document queries. In *IEEE Symposium on Principles of Database Systems*, pages 155–166. ACM Press, 2003.

[SSMH04] H. Seidl, T. Schwentick, A. Muscholl, and P. Habermehl. Counting in trees for free. In *In Proc. 31st International Colloq. on Automata, Languages, and Programming (ICALP 2004)*, volume LNCS 3142, pages 1136–1149. Springer-Verlag, 2004.

[Suc02] D. Suciu. Typechecking for semistructured data. In *8th Workshop on Data Bases and Programming Languages (DBPL 2001)*, volume LNCS 2397, pages 1–20. Springer-Verlag, 2002.

[TW68] J.W. Thatcher and J.B. Wright. Generalized finite automata theory with application to a decision problem of second-order logic. *Math. Syst. Theory*, 2(1):57–81, 1968.