Journal of Computer and System Sciences 146 (2024) 103563

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

journal homepage: www.elsevier.com/locate/jcss

The impact of state merging on predictive accuracy in probabilistic tree automata: Dietze's conjecture revisited

Johanna Björklund [1]

Dept. Computing Science, Umeå University, Umeå, 90187, Sweden

ARTICLE INFO

Article history:
Received 30 November 2023
Accepted 19 June 2024
Available online 27 June 2024

Keywords:
Tree automata
Statistical ML
Probability distributions

ABSTRACT

Dietze's conjecture concerns the problem of equipping a tree automaton M with weights to make it probabilistic, in such a way that the resulting automaton N predicts a given corpus $\mathcal{C}$ as accurately as possible. The conjecture states that the accuracy cannot increase if the states in M are merged with respect to an equivalence relation $\sim$ so that the result is a smaller automaton $M^{\sim}$. Put differently, merging states can never improve predictions. This is under the assumption that both M and $M^{\sim}$ are bottom-up deterministic and accept every tree in $\mathcal{C}$. We prove that the conjecture holds, using a construction that turns any probabilistic version $N^{\sim}$ of $M^{\sim}$ into a probabilistic version N of M, such that N assigns at least as great a weight to each tree in $\mathcal{C}$ as $N^{\sim}$ does.

© 2024 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Supervised learning is concerned with recovering or approximating a target function, based on a limited sample of input-output pairs called a *corpus*. The difficulty of the problem depends on the class of functions admitted and on various characteristics of the corpus. In this work, we focus on the supervised learning of *probabilistic tree languages* [5], in particular, on probability distributions computable by *weighted tree automata (wta)* over the probability semiring [9]. Some of the earliest results are due to Klein and Manning, who extracted a simple grammar from a corpus and then manually refined it based on linguistic patterns in the data [10]. Petrov et al. later translated their approach into an automated method that alternately splits and merges nonterminals to maximize the likelihood of the target corpus [11].

Dietze [6] formalised the ideas of Petrov et al. [11] into an algorithmic framework for supervised learning of weighted tree automata: First, the *read-off automaton* is extracted from a given corpus $\mathcal{C}$. This is a wta over the Boolean semiring whose states and transitions are in one-to-one correspondence with the subtrees of $\mathcal{C}$. The states of this automaton are then repeatedly merged or split, and after each pass, the resulting wta M is equipped with weights to optimise prediction accuracy with respect to the weight of the trees in $\mathcal{C}$. The resulting probabilistic automaton N is a *maximum-likelihood estimation* (mle) of M with respect to $\mathcal{C}$. Intuitively, this means that among all probabilistic automata that can be derived from M, the automaton N is that which makes the weights of the trees in C as likely as possible. The learning process terminates when the predictions produced by N are sufficiently well-aligned with $\mathcal{C}$, or when the number of merge-split cycles has reached some maximal limit.

E-mail address: johanna.bjorklund@umu.se.

[1] The project is supported by the Swedish Research Council under Grant Number 2020-03852, and by the Wallenberg AI, Autonomous Systems and Software Program (WASP).

https://doi.org/10.1016/j.jcss.2024.103563
0022-0000/© 2024 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

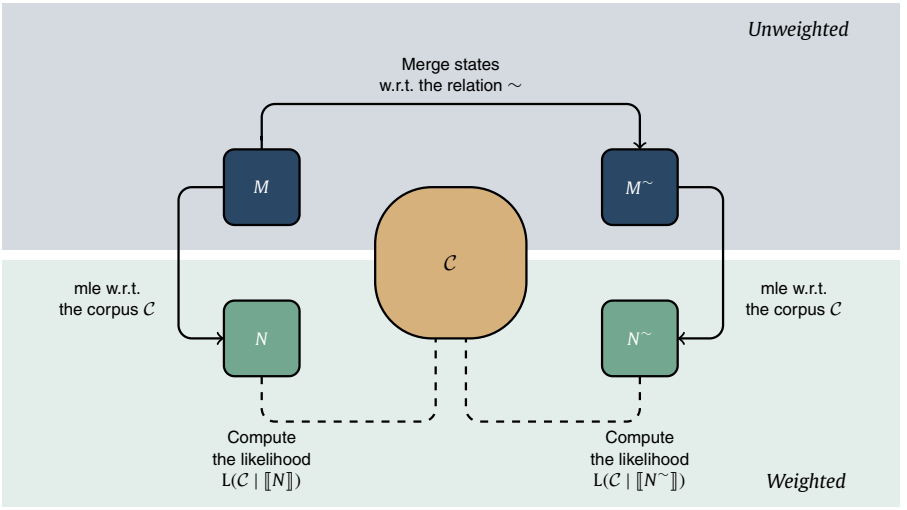


Fig. 1. By merging the states of the unweighted tree automaton M with respect to the equivalence relation $\sim$ on the states of M, we obtain the smaller automaton $M^{\sim}$. The probabilistic automata N and $N^{\sim}$ are derived from M and $M^{\sim}$, respectively, by computing the maximum likelihood estimation with respect to the corpus $\mathcal{C}$. The conjecture states that if all automata are deterministic and contain $\mathcal{C}$ in their support, then $\mathrm{L}(\mathcal{C} \mid [\![N^{\sim}]\!])$ cannot exceed $\mathrm{L}(\mathcal{C} \mid [\![N]\!])$.

A question that arises from this approach is how the split and merge operations affect the automaton's ability to predict the corpus. Merging states has the advantage that it allows the automaton to accept more trees, thereby generalising what has been learnt from data. However, Dietze conjectures that if we have a determinsitic wta (dta) M that already accepts every tree in the corpus and merge the states of M with respect to some equivalence relation $\sim$ to form a dta $M^{\sim}$, then the mle $N^{\sim}$ of $M^{\sim}$ cannot yield a better prediction of $\mathcal{C}$ than the mle N of M [6, Conjecture 6.2.2]. This means that further progress cannot be made by merging states alone (but possibly through a succession of splits and merges). Fig. 1 illustrates the relationship between the various weighted and unweighted automata involved in the conjecture.

We are now able to give a constructive proof of the veracity of Dietze's conjecture. It is based on the observation that the *support* of $M^{\sim}$, that is, the set of trees mapped to a nonzero value by $M^{\sim}$, is a superset of the support of M [6]. As a consequence, the probability mass assigned by the mle of $M^{\sim}$ is diluted over a greater number of trees. We find that it is possible to reshape this distribution into a distribution over the smaller support of the mle of M, in such a way that the weight of every tree in the support either increases or remains unchanged, but never decreases. Since $\mathcal{C}$ is contained in the support of the mle of M, the likelihood of $\mathcal{C}$ also increases.

In these times of deep learning, statistical approaches such as that outlined above remain relevant for a number of reasons. First, their training process is transparent, enabling a straightforward analysis of the impact of training data on the resultant model. Second, they are data efficient, rendering them suitable for deployment in low-data regimes. Finally, with an energy footprint that is generally smaller than corresponding neural approaches, they are a feasible choice for embedded systems and internet-of-things applications.

Finally, suppose that we are interested in supervised learning of structures that can be uniquely represented as terms in some algebra. We can then parse each element in the corpus into a term, i.e., a tree, and then apply the above learning framework. Examples of such structures are, e.g., strings, hedges [4], and certain types of directed acyclic graphs [2,1]. This means that learning results for tree languages have immediate consequences for countless other domains. For further reading on tree-based generation, see for example [7,8].

This is an extended version of the conference paper [3] presented at the 24th International Symposium on the Fundamentals of Computation Theory (FCT 2023) in Trier, Germany. It is organized as follows. Section 2 provides automata-theoretic definitions and fixes notation. Section 3 recalls Dietze's conjecture and proves its correctness. Section 4 discusses the limitations of the proposed approach, and Section 5 summarises our findings.

2. Preliminaries

Sets, numbers and strings. The set of *natural numbers* (excluding zero) is denoted by $\mathbb{N}$, and the set of *non-negative reals* is denoted by $\mathbb{R}_{\geq 0}$.

The *power set* of a set S is written 2^S, the *size* of S is denoted by $|S|$, and the *empty set* is written $\emptyset$. A *partition* of a set S is a set of pairwise disjoint, sets $\{S_1, \ldots, S_k\} \subseteq 2^S$ whose union is S. A *probability distribution* over S is a function $p\colon S \to \mathbb{R}_{\geq 0}$ such that $\sum_{s \in S} p(s) = 1$.

A binary relation on S is an equivalence relation if it is reflexive, symmetric and transitive. Let $\sim$ be such a relation. The *equivalence class* of an element s in S with respect to s is the set $[s]_{\sim} = \{s' \in S \mid s \sim s'\}$. Since $[s]_{\sim}$ and $[s']_{\sim}$ are equal if

$s \sim s'$ and disjoint otherwise, the relation $\sim$ induces a partition $(S/\sim) = \{[s]_\sim \mid s \in S\}$ of S. In the following, we often define equivalence relations on by stating the partition on S that they induce.

An *alphabet* Σ is a finite nonempty set. A *string* is an ordered sequence of zero or more symbols from Σ. The set of all strings over Σ is denoted by Σ^* and the *empty string* is denoted by ϵ.

Semirings. A *monoid* is a structure $(A, \oplus, \mathbb{0})$ where A is a set, $\oplus$ a binary operation on A, and $\mathbb{0}$ is an identity element with respect to $\oplus$ (so $a \oplus \mathbb{0} = \mathbb{0} \oplus a = a$ for every $a \in A$). If, for every $a, b \in A$, we have that $a \oplus b = b \oplus a$ then the monoid is *commutative*.

A *(commutative) semiring* is a structure $\mathcal{A} = (A, \oplus, \odot, \mathbb{0}, \mathbb{1})$ where $(A, \oplus, \mathbb{0})$ and $(A, \odot, \mathbb{1})$ are commutative monoids, $\odot$ distributes over $\oplus$, and $\mathbb{0}$ is an absorbing element with respect to $\odot$ (that is, $a \odot \mathbb{0} = \mathbb{0} \odot a = \mathbb{0}$ for every $a \in A$). The *Boolean semiring* is the semiring $\mathbb{B} = (\{0, 1\}, \vee, \wedge, 0, 1)$ and the *probability semiring* is the semiring $\mathbb{P} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, +, \cdot, 0, 1)$. Let S be a set, $\mathcal{A} = (A, \oplus, \odot, \mathbb{0}, \mathbb{1})$ a semiring, and $f : S \to A$ a function. The *support* of f is the set $supp(f) = \{a \in A \mid f(a) \neq \mathbb{0}\}$.

Trees. An *unranked tree* over the alphabet Σ is a function $t : D \to \Sigma$ where $D \subseteq \mathbb{N}^*$ is such that $\epsilon \in D$ and, for every $v \in D$, there exists a $k \in \mathbb{N}$ with $\{i \in \mathbb{N} \mid vi \in D\} = [k]$. We call D the *domain* of t and denote it by $dom(t)$. An element v of $dom(t)$ is called a *node of* t, and k is the *rank of* v. The *subtree* of $t \in T_\Sigma$ rooted at v is the tree t/v defined by $dom(t/v) = \{u \in \mathbb{N}^* \mid vu \in dom(t)\}$ and $t/v(u) = t(vu)$ for every $u \in D$. If $t(\epsilon) = f$ and $t/i = t_i$ for all $i \in [k]$, where k is the rank of ϵ in t, then we denote t by $f[t_1, \ldots, t_k]$. If $k = 0$, then $f[]$ is usually abbreviated as f. The *height* of t is $height(t) = \max\{|v| \mid v \in dom(t)\}$.

A *ranked alphabet* is an alphabet $\Sigma = \bigcup_{k \in \mathbb{N}} \Sigma_k$ which is partitioned into subsets Σ_k. For every $k \in \mathbb{N}$ and $f \in \Sigma_k$, the *rank* of f is $rank(f) = k$. We write $f^{(k)}$ to indicate that the rank of f is k. A *tree* over Σ is an unranked tree t over Σ such that the rank of every node $v \in dom(t)$ coincides with the rank of $t(v)$. The set of all (ranked) trees over Σ is denoted by T_Σ. A tree language over Σ is a subset of T_Σ. A *weighted tree language* over Σ and a semiring $\mathcal{A}$ is a mapping $T_\Sigma \to \mathcal{A}$.

Weighted automata. A *weighted tree automaton* (wta) over the semiring $\mathcal{A}$ is a tuple $M = (Q, \Sigma, \mathcal{A}, F, \delta)$ where

- Q is an alphabet of *states*,
- Σ is a ranked alphabet of *input symbols*,
- $\mathcal{A}$ is a commutative semiring,
- $F : Q \to \mathcal{A}$ is a *final weight mapping*,
- $\delta = (\delta_k)_{k \in \mathbb{N}}$ is a family of functions such that $\delta_k : Q^k \times \Sigma_k \times Q \to \mathcal{A}$, for every $k \in \mathbb{N}$.

The wta is *(bottom-up) deterministic* if for every choice of $q_1 \cdots q_k \in Q^k$ and $\sigma \in \Sigma$, there is at most one $q \in Q$ such that $\delta_k(q_1 \cdots q_k, \sigma, q) \neq \mathbb{0}$.

A *run* of M on $t \in T_\Sigma$ is a function $dom(t) \to Q$. The set of all runs of M on t is denoted by $runs_M(t)$. The weight function $w_M : runs_M(t) \to \mathcal{A}$ is defined for every $r \in runs_M(t)$ by

$$w_M(r) = \bigodot_{v \in dom(t)} \delta_k(r(v1) \cdots r(vk), t(v), r(v)) \ ,$$

where $k = rank(v)$. If M is deterministic, then there is at most one run $r \in runs_M(t)$ such that $w_M(t) \neq 0$. If it exists, we denote by q_t the state $r(\epsilon)$.

The *$\mathcal{A}$-weighted tree language* computed by M is the function $[\![M]\!] : T_\Sigma \to \mathcal{A}$ that is defined, for every $t \in T_\Sigma$, by

$$[\![M]\!](t) = \bigoplus_{r \in runs_M(t)} w_M(r) \odot F(r(\epsilon)) \ .$$

For every $q \in Q$, we denote by M_q the wta $(Q, \Sigma, \mathcal{A}, F', \delta)$, where $F'(q) = \mathbb{1}$ and $F'(p) = \mathbb{0}$ for every $p \in Q \setminus \{q\}$.

Definition 2.1 *(Probabilistic automata, cf Section 4.3 of [6]).* Consider the wta $M = (Q, \Sigma, \mathbb{P}, F, \delta)$ over the probability semiring. The automaton M is:

- *out-probabilistic* if for every $q \in Q$,
$$\sum_{k \in \mathbb{N}, \sigma \in \Sigma_k, q_1 \cdots q_k \in Q^k} \delta_k(q_1 \cdots q_k, \sigma, q) = 1 \ ,$$
- *end-probabilistic* if
$$\sum_{q \in Q} F(q) = 1 \ , \text{ and}$$
- *consistent* if
$$\sum_{t \in T_\Sigma} [\![M]\!](t) = 1 \ .$$

The wta M is *probabilistic* if it is out- and end-probabilistic, and consistent. $\Diamond$

For wta over the Boolean semiring, we treat the transition function and final weight mappings as sets, and only explicitly report those transitions and states that are assigned the value `true`. See, e.g., Example 3.2.

Corpora. In [6], a *corpus* is a mapping $\mathcal{C}\colon T_\Sigma \to \mathbb{R}_{\geq 0}$ such that $supp(\mathcal{C})$ is finite and nonempty. The size of $\mathcal{C}$ is $|\mathcal{C}| = \sum_{t\in T_\Sigma} \mathcal{C}(t)$. Let p be a probability distribution over T_Σ. The *likelihood* of $\mathcal{C}$ under p is:

$$\mathrm{L}(\mathcal{C} \mid p) = \prod_{t\in T_\Sigma} p(t)^{\mathcal{C}(t)} \ .$$

Let $M = (Q, \Sigma, \mathbb{B}, F, \delta)$ be a wta and let $\mathcal{C}$ be a corpus. We denote by $prob(M)$ the set of probabilistic wta $(Q, \Sigma, \mathbb{P}, F', \delta')$ such that $supp(F') \subseteq supp(F)$ and $supp(\delta') \subseteq supp(\delta)$. Finally,

$$mle_{\mathcal{C}}(M) = \operatorname*{argmax}_{N\in prob(M)} \mathrm{L}(\mathcal{C} \mid [\![N]\!]) \ .$$

3. Dietze's conjecture

Dietze's conjecture concerns the impact of state merging on the inference of probabilistic tree languages from corpora. In essence, the conjecture says that when attempting to transform a dta M into a probabilistic wta N to maximise predictive accuracy with respect to a given corpus $\mathcal{C}$, where M already accepts every tree in $\mathcal{C}$, no predictive power can be gained by merging states of M to obtain a smaller dta $M^{\sim}$ and subsequently endowing $M^{\sim}$ with weights. To formalise this assertion, we recall the concept of merging states in a weighted tree automaton based on an equivalence relation.

Definition 3.1. Let $M = (Q, \Sigma, \mathcal{A}, F, \delta)$ be a wta and let $\sim$ be an equivalence relation on Q. We denote by $M^{\sim}$ the wta $(Q^{\sim}, \Sigma, \mathcal{A}, F^{\sim}, \delta^{\sim})$ with the components

- $Q^{\sim} = (Q/{\sim})$,
- $F^{\sim}\colon Q^{\sim} \to \mathcal{A}$, where for every $p = [q]_{\sim} \in Q^{\sim}$,

$$F^{\sim}(p) = \bigoplus_{q'\in[q]_{\sim}} F(q') \ , \text{and}$$

- $\delta^{\sim} = (\delta_k^{\sim})_{k\in\mathbb{N}}$ where $\delta_k^{\sim}\colon (Q/{\sim})^k \times \Sigma \times (Q/{\sim}) \to \mathcal{A}$ and for every $p = [q]_{\sim}, p_1 = [q_1]_{\sim}, \ldots, p_k = [q_k]_{\sim} \in Q^{\sim}$ and $\sigma \in \Sigma$, we have

$$\delta_k^{\sim}(p_1 \cdots p_k, \sigma, p) = \bigoplus_{\substack{q'\in[q]_{\sim},\\ q'_i\in[q_i]_{\sim}, i\in[k]}} \delta_k(q'_1 \cdots q'_k, \sigma, q') \ . \quad \Diamond$$

Recall from [6, Thm 5.1.1.] that if M is a wta over the Boolean semiring, then $supp([\![M]\!]) \subseteq supp([\![M^{\sim}]\!])$. Dietze's conjecture now reads as follows.

Theorem 3.1 *(cf. Conjecture 6.2.2 of [6]).* Let $M = (Q, \Sigma, \mathbb{B}, F, \delta)$ be a wta over the Boolean semiring, $\sim$ an equivalence relation on Q, and $\mathcal{C}$ a corpus such that $supp(\mathcal{C}) \subseteq supp([\![M]\!])$. If M and $M^{\sim}$ are bottom-up deterministic, then

$$\mathrm{L}(\mathcal{C} \mid [\![mle_{\mathcal{C}}(M)]\!]) \geq \mathrm{L}(\mathcal{C} \mid [\![mle_{\mathcal{C}}(M^{\sim})]\!]) \ .$$

To prove Theorem 3.1, we show that for every $N^{\sim} = ((Q/{\sim}), \Sigma, \mathbb{P}, F^{\sim}, \delta^{\sim})$ in $prob(M^{\sim})$, we can construct a wta $N = (Q, \Sigma, \mathbb{P}, F', \delta')$ in $prob(M)$ that has the following property:

$$\text{For every } t \in supp([\![M]\!]) \cap supp([\![N^{\sim}]\!]), \text{ we have } [\![N]\!](t) \geq [\![N^{\sim}]\!](t). \tag{P1}$$

Since $\mathcal{C} \subseteq supp([\![M]\!])$, this is enough to validate the conjecture.

The construction of N is based on the following observation. Recall that every state p in $N^{\sim}$ is a set of states $\{q_1, \ldots, q_k\}$ in M, and let T_p be the set of trees that are mapped by $N^{\sim}$ to p with nonzero weight. Since $N^{\sim}$ is probabilistic, by a straightforward induction the weights assigned to the trees in T_p by $N^{\sim}$ sum to 1. When p is split into its constituent states $q_1, \ldots, q_k$, the set T_p is partitioned over the states $q_1, \ldots, q_k$ into sets $T_{q_1}, \ldots, T_{q_k}$, such that the state q_i is reached by M by precisely the trees in T_{q_i}. When we add weights to M to form N, the weights assigned by N to the trees in T_{q_i} must also sum to 1 for every $i \in [k]$. We can therefore compute a value $\theta(q_i)$ for each q_i that corresponds to the probability

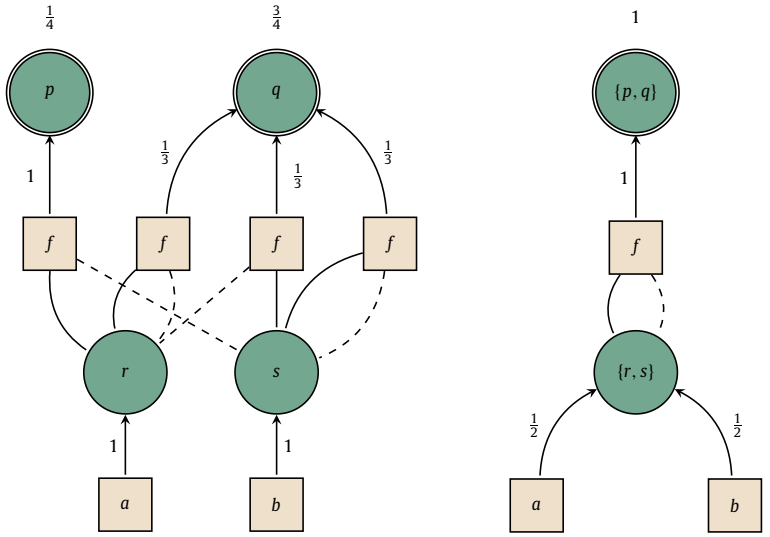


Fig. 2. The automata discussed in Example 3.2. The circles and boxes denote states and transitions, respectively. In a transition of the form (rs, f, p), the first and second argument (r and s) are indicated with a solid and dashed line, respectively, while the target state p is indicated with an arrow. Accepting states are drawn with double strokes, and final weights are written above them. The automaton on the left, taken without weights, corresponds to the wta M over the Boolean semiring in the example; the automaton on the right, again without weights, is the automaton $M^{\sim}$ resulting from merging the states of M with respect to the simulation relation $\sim = \{(r,s),(s,r),(p,q),(q,p)\}$. The automaton on the right with weights is the probabilistic wta $N^{\sim} \in prob(M^{\sim})$. Finally, the automaton on the right, now with weights, is a probabilistic wta $N \in prob(M)$ obtained from $N^{\sim}$ by following the construction sketched in the example and formalised over the following pages. It is easy to verify that the two probabilistic wta represent the same distribution.

mass given by $N^{\sim}$ to the trees in T_{q_i}, relative to the probability mass given by $N^{\sim}$ to all of T_p. Note that in Definition 3.2 below, we do not explicitly divide by the sum of the latter because it is always 1.

The mapping $\theta : Q \to \mathbb{P}$ allows us in turn to compute the weights of the transitions entering each $q_i \in Q$, by scaling the weights of the transitions entering p in $N^{\sim}$ by a factor of $\theta(q_i)^{-1}$. If we also scale the accepting weights of $N^{\sim}$, then we obtain a wta N with the property (P1).

Definition 3.2 *(The mapping θ).* Let $\theta\colon Q \to \mathbb{P}$ be a mapping, defined for every $q \in Q$ by

$$\theta(q) = \sum_{t \in supp([\![M_q]\!])} [\![N^{\sim}_{[q]_{\sim}}]\!](t) \quad . \quad \Diamond$$

Before giving a formal definition of the scaled automaton N and establishing the correctness of construction, we provide a practical example to illustrate the role of θ in the construction.

Example 3.2. Let $M = (Q, \Sigma, \mathbb{B}, F, \delta)$ be the wta over the Boolean semiring shown on the left in Fig. 2, momentarily ignoring the weights. As we can see, $Q = \{p, q, r, s\}$, $\Sigma = \{f^{(2)}, a^{(0)}, b^{(0)}\}$, $F = \{p, q\}$, and

$$\delta = \{ \quad (rs, f, p), \quad (ss, f, q), \quad (sr, f, q), \quad (rr, f, q), \quad (\epsilon, a, r), \quad (\epsilon, b, s) \quad \} \ .$$

Now, let $\sim$ be the equivalence relation $\{\{s,r\},\{p,q\}\}$ on Q. When merging M with respect to $\sim$, the resulting automaton is $M^{\sim} = (Q^{\sim}, \Sigma, \mathbb{B}, F^{\sim}, \delta^{\sim})$, which is shown on the right in Fig. 2, again ignoring the weights. Here, $Q^{\sim} = \{\{p,q\},\{r,s\}\}$, $F^{\sim} = \{\{p,q\}\}$, and

$$\delta^{\sim} = \{ \quad (\{r,s\}\{r,s\}, f, \{p,q\}), \quad (\epsilon, a, \{r,s\}), \quad (\epsilon, b, \{r,s\}) \quad \}$$

Assume then that $M^{\sim}$ is equipped with weights to turn it into a probabilistic automaton, for example the wta $N^{\sim} = (Q^{\sim}, \Sigma, \mathcal{A}, F^{\sim}, \delta^{\sim})$ shown on the right in Fig. 2 (but now observing the weights). For this wta, $F^{\sim}(\{p,q\}) = 1$, $F^{\sim}(\{r,s\}) = 0$, and

$$\delta^{\sim} = \{ \quad ((\{r,s\}\{r,s\}, f, \{p,q\}), 1), \quad ((\epsilon, a, \{r,s\}), \tfrac{1}{2}), \quad ((\epsilon, b, \{r,s\}), \tfrac{1}{2}) \quad \} \ .$$

We may now ask ourselves how we can derive weights for the transition function and final states of M, to obtain a probabilistic wta $N = (Q, \Sigma, \mathcal{A}, F', \delta')$ that assigns at least as great a weight to every tree that is both in $supp([\![M]\!])$ and $supp([\![N^{\sim}]\!])$ as N does. A first attempt may be to assign $\delta'(\epsilon, a, r)$ the weight $\delta^{\sim}(\epsilon, a, \{r,s\}) = 1/2$. However, since there

is only one run that ends in r in M (namely, that on the tree a, which has weight $1/2$), the resulting automaton would fail to be out-probabilistic. For this reason, we divide the transition weight by a factor $\theta(r) = 1/2$, meaning that we let $\delta'(\epsilon, a, r) = \delta^{\sim}(\epsilon, a, r) \cdot \theta(r)^{-1} = 1$. The transition (ϵ, b, s) is treated in the same way and is therefore also assigned the weight 1 by δ'.

Moving on to the transition (rs, f, p) we generalise the argument: The runs reaching p only make up $1/4$ of the probability mass assigned to trees reaching $\{p, q\}$ by $N^{\sim}$. We would thus want to normalise the weights of the transitions leading to p by dividing by $\theta(p) = 1/4$. However, due to the normalisations used to obtain the weights $\delta'(\epsilon, b, r)$ and $\delta'(\epsilon, b, s)$, this would give the unique run ending in p the weight 4, which means that the automaton would not be consistent. To preserve consistency, the weight given to (rs, f, p) must undo the normalisations made in previous steps. In this particular case, this means multiplying by $1/2$ twice; once for the normalisation with respect to $\theta(r)$ and once for that with respect to $\theta(s)$. This gives us $\delta'(rs, f, p) = \theta(p)^{-1} \cdot \delta(\{r, s\}\{r, s\}, f, \{p, q\}) \cdot \theta(r) \cdot \theta(s) = 1$.

Looking then at the weight of the transitions reaching p, we note that together they are responsible for $3/4$ of the probability mass assigned by $N^{\sim}$. To compute the weight of, e.g., $\delta'(rs, f, q)$, we therefore divide the weight of $\delta^{\sim}(\{r, s\}, \{r, s\}, f, \{p, q\})$ by $\theta(q) = 3/4$ and multiply in the factor $1/2$ twice to compensate for the normalisation with respect to $\theta(r)$ and $\theta(s)$. This gives us

$$\delta'(sr, f, q) = \theta(q)^{-1} \cdot \delta(\{r, s\}, \{r, s\}, f, \{p, q\}) \cdot \theta(r) \cdot \theta(s) = 1/3 \ .$$

Similar computations yield the same weight $1/3$ for the remaining two transitions ending in q. To summarise, this means that

$$\begin{aligned} \delta' = \{ \quad & ((rs, f, p), 1), \quad ((ss, f, q), \tfrac{1}{3}), \quad ((sr, f, q), \tfrac{1}{3}), \quad ((rr, f, q), \tfrac{1}{3}), \\ & ((\epsilon, a, r), 1), \quad ((\epsilon, b, s), 1) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \} \end{aligned}$$

To compute the final weight of p, we divide the probability given by $N^{\sim}$ to trees in $supp(M)$ that reach p, by the total weight of all trees in $supp(M)$ with respect to $N^{\sim}$, which is equal to $\theta(p)/(\theta(p) + \theta(q)) = 1/4$. The analogue computation gives us the final weight $3/4$ for p, and 0 for r and s.

It is easy to verify that the automaton N' thus constructed is out-probabilistic and end-probabilistic. To give an idea of why it also consistent, let us look at the weight N assigns to one of the trees in $supp([\![N']\!])$, for example $f[b, a]$. There is only one run with non-zero weight on this tree, and its weight is computed as

$$[\![N]\!](f[b, a]) = \delta'(\epsilon, b, s) \cdot \delta'(\epsilon, a, r) \cdot \delta'(sr, f, q) \cdot F'(q) \ ,$$

which expands to

$$\begin{aligned} [\![N]\!](f[b, a]) = \ & \delta^{\sim}(\epsilon, b, \{r, s\}) \cdot \theta(s)^{-1} \cdot \\ & \delta^{\sim}(\epsilon, a, \{r, s\}) \cdot \theta(r)^{-1} \cdot \\ & \theta(s) \cdot \theta(r) \cdot \delta^{\sim}(\{r, s\}\{r, s\}, f, \{p, q\}) \cdot \theta(q)^{-1} \cdot \\ & \theta(q) \cdot (\theta(q) + \theta(q))^{-1} \cdot F^{\sim}(\{p, q\}) \ . \end{aligned}$$

Notice here how the normalisation factors cancel out, so that the result is

$$\begin{aligned} [\![N]\!](f[b, a]) = \ & \delta^{\sim}(\epsilon, b, \{r, s\}) \cdot \\ & \delta^{\sim}(\epsilon, a, \{r, s\}) \cdot \\ & \delta^{\sim}(\{r, s\}\{r, s\}, f, \{p, q\}) \cdot \\ & (\theta(q) + \theta(q))^{-1} \cdot F^{\sim}(\{p, q\}) \ . \end{aligned}$$

This means that

$$[\![N]\!](f[b, a]) = (\theta(q) + \theta(q))^{-1} \cdot [\![N^{\sim}]\!](f[b, a]) \ .$$

Since the sum $\theta(p) + \theta(q) = 1$, we have that

$$[\![N]\!](f[b, a])) = [\![N^{\sim}]\!](f[b, a]) \ .$$

The reader is encouraged to compute the probabilities assigned by N to other trees in $supp([\![N]\!]) \cap supp([\![N^{\sim}]\!])$, and observe a similar cancellation pattern.

In short, the purpose of θ is thus to transfer weights from $N^{\sim}$ to M so that the resulting wta N is still a probabilistic automaton, while ensuring that probabilities of the trees in $supp([\![N]\!]) \cap supp([\![N^{\sim}]\!])$ do not decrease. $\diamondsuit$

In the upcoming proofs we need an expanded form of θ. To this end, we introduce a sequence of functions $\theta_0, \theta_1, \theta_2, \ldots$ which converge to θ. In the following recursive definition, the value of θ_0 is zero for all states, while the value of θ_1 is nonzero only for states that can be reached on symbols of rank 0.

Definition 3.3. For every $q \in Q$, $\theta_0(q) = 0$ and for every $i \geq 1$,

$$\theta_i(q) = \sum_{(q_1 \cdots q_k, \sigma, q) \in supp(\delta)} \delta^{\sim}([q_1]_{\sim} \cdots [q_k]_{\sim}, \sigma, [q]_{\sim}) \cdot \prod_{j \in [k]} \theta_{i-1}(q_j) \ . \quad \Diamond \tag{1}$$

Lemma 3.3. For every $q \in Q$ and $i = 0, 1, 2, \ldots$

$$\theta_i(q) = \sum_{\substack{t \in supp([\![M_q]\!]) \\ \text{and } height(t) < i}} [\![N^{\sim}_{[q]_{\sim}}]\!](t) \ .$$

Proof. The proof is by induction on i. The base case, where $i = 0$, is trivially true. Assume that $i \geq 1$ and that Lemma 3.3 holds for $i - 1$. We compute as follows:

$\theta_i(q)$

$=$ by Definition 3.3 of θ_i

$$\sum_{\substack{(q_1 \cdots q_k, \sigma, q) \\ \in supp(\delta)}} \delta^{\sim}_k([q_1]_{\sim} \cdots [q_k]_{\sim}, \sigma, [q]_{\sim}) \cdot \prod_{j \in [k]} \theta_{i-1}(q_j)$$

$=$ by the induction hypothesis

$$\sum_{\substack{(q_1 \cdots q_k, \sigma, q) \\ \in supp(\delta)}} \delta^{\sim}_k([q_1]_{\sim} \cdots [q_k]_{\sim}, \sigma, [q]_{\sim}) \cdot \prod_{j \in [k]} \sum_{\substack{t_j \in supp(M_{q_j}), \\ height(t_j) < i-1}} [\![N^{\sim}_{[q_j]_{\sim}}]\!](t_j)$$

$=$ because multiplication distributes over addition

$$\sum_{\substack{(q_1 \cdots q_k, \sigma, q) \\ \in supp(\delta)}} \delta^{\sim}_k([q_1]_{\sim} \cdots [q_k]_{\sim}, \sigma, [q]_{\sim}) \cdot \sum_{\substack{t_1 \cdots t_k \in supp(M_{q_1}) \times \cdots \times supp([\![M_{q_k}]\!]) \\ \text{and } height(t_j) < i-1, \text{ for } j \in [k]}} [\![N^{\sim}_{[q_1]_{\sim}}]\!](t_1) \cdot \cdots \cdot [\![N^{\sim}_{[q_k]_{\sim}}]\!](t_k)$$

$=$ because multiplication distributes over addition

$$\sum_{\substack{(q_1 \cdots q_k, \sigma, q) \in supp(\delta), \\ t_1 \cdots t_k \in supp(M_{q_1}) \times \cdots \times supp([\![M_{q_k}]\!]) \\ \text{and } height(t_j) < i-1, \text{ for } j \in [k]}} \delta^{\sim}_k([q_1]_{\sim} \cdots [q_k]_{\sim}, \sigma, [q]_{\sim}) \cdot [\![N^{\sim}_{[q_1]_{\sim}}]\!](t_1) \cdot \cdots \cdot [\![N^{\sim}_{[q_k]_{\sim}}]\!](t_k)$$

$=$ by the definition of wta semantics in Section 2

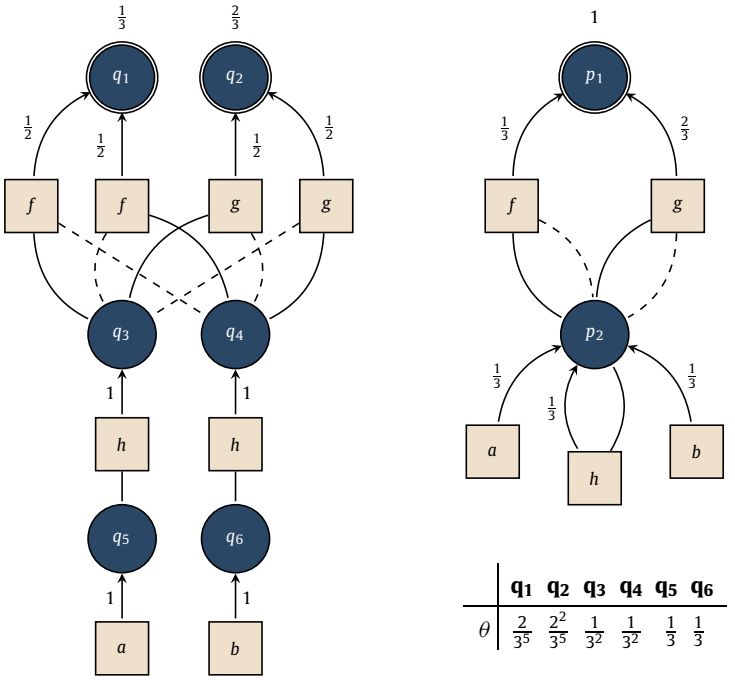
$$\sum_{\substack{t \in supp(M_q) \\ \text{and } height(t) < i}} [\![N^{\sim}_{[q]_{\sim}}]\!](t) \quad \square$$

Lemma 3.4. For every $q \in Q$, $\theta(q) = \lim_{i \to \infty} \theta_i(q)$.

Proof. To show that the value of $\theta_i(q)$ converges, we argue that there is a constant c such that for all $i > c$, there is a $j < i$ such that

$$\max\{[\![N^{\sim}]\!](t) \mid height(t) = i\} < \max\{[\![N^{\sim}]\!](t) \mid height(t) = j\} \ .$$

The claim holds because the products of the weight used in each loop in $N^{\sim}$ must be strictly less than 1. If this was not the case, then $N^{\sim}$ would not be probabilistic, because at the state marking the beginning and end of the loop, there must be one transition leading into the loop and one transition exiting the loop, both of which must have non-zero weight. Once



	q_1	q_2	q_3	q_4	q_5	q_6
θ	$\frac{2}{3^5}$	$\frac{2^2}{3^5}$	$\frac{1}{3^2}$	$\frac{1}{3^2}$	$\frac{1}{3}$	$\frac{1}{3}$

Fig. 3. The automata discussed in Example 3.5. The circles and boxes denote states and transitions, respectively. In a transition of the form $(q_3 q_4, f, q_1)$, the first and second argument (q_3 and q_4) are indicated with a solid and dashed line, respectively, while the target state q_1 is indicated with an arrow. Accepting states are drawn with double strokes, and final weights are written above them. On the left in the figure is a wta M over the Boolean semiring, here equipped with the weights that turns it into a wta N in $prob(M)$. On the right is the automaton $M^{\sim}$, where $(Q/{\sim}) = \{p_1, p_2\}$, $p_1 = \{q_1, q_2\}$, and $p_2 = \{q_2, q_4, q_5, q_6\}$, equipped with weights that turns it into a wta $N^{\sim}$ in $prob(M^{\sim})$. In the lower right corner is a table of θ values that allow us to derive the weights of N in $prob(M)$ from those of $N^{\sim}$.

a tree has been constructed by pumping the loop several times, there is a smaller tree with strictly greater weight that was built by pumping the loop fewer times. In combination with Lemma 3.3, this means that the sequence $\theta_i(q)$ converges to $\theta(q)$ for all $q \in Q$. □

We can now use the function θ to scale the weight mappings used in $N^{\sim}$ to compute suitable weights for M, such that the resulting probabilistic tree automaton N assigns at least as great a weight to every tree in $\mathcal{C}$ as $N^{\sim}$ does.

Definition 3.4 *(The automaton N).* Let $M = (Q, \Sigma, \mathbb{B}, F, \delta)$ be a wta, $\sim$ an equivalence relation on Q, and $N^{\sim} = ((Q/{\sim}), \Sigma, \mathbb{P}, F^{\sim}, \delta^{\sim}) \in prob(M^{\sim})$. We define N as the wta $(Q, \Sigma, \mathbb{P}, F', \delta')$, where for every $q \in Q$

$$F'(q) = F^{\sim}([q]_{\sim}) \cdot \frac{\theta(q)}{\sum_{p \in [q]_{\sim}} \theta(p)} \quad , \tag{2}$$

and for every $(q_1 \cdots q_k, \sigma, q) \in supp(\delta)$,

$$\delta'_k(q_1 \cdots q_k, \sigma, q) = \frac{\delta^{\sim}_k([q_1]_{\sim} \cdots [q_k]_{\sim}, \sigma, [q]_{\sim}) \cdot \prod_{i \in [k]} \theta(q_i)}{\theta(q)} \quad , \tag{3}$$

and 0 otherwise.

Example 3.5. Consider the automata in Fig. 3. If we momentarily ignore the weights, we can take these to be the wta M and $M^{\sim}$ over the Boolean semiring. We see that $M^{\sim}$ can be obtained from M by merging the states with respect to the equivalence relation $\sim$, which is such that $(Q/{\sim}) = \{p_1, p_2\}$, $p_1 = \{q_1, q_2\}$, and $p_2 = \{q_3, q_4, q_5, q_6\}$. Assume then that $N^{\sim}$ is the wta in $prob(M^{\sim})$ that results from equipping $M^{\sim}$ with weights as shown on the right in the figure. The table in the lower right of the figure lists the values of θ for the states in M, and from these the weights that turn M into the wta N are computed according to Definition 3.4. As we can see,

$$supp([\![M]\!]) = \{f[h[a], h[b]],\ f[h[b], h[a]],\ g[h[a], h[b]],\ g[h[b], h[a]]\} \ .$$

It is easy to verify that for every tree t in this set $N(t) \geq N^{\sim}(t)$. For example, $N(f[h[a], h[b]]) = \frac{1}{2 \cdot 3}$ and $N^{\sim}(f[h[a], h[b]]) = \frac{1}{3^5}$. $\Diamond$

We note that if a tree in $supp([\![M]\!])$ is assigned a nonzero weight by $N^{\sim}$, then it is also assigned a nonzero weight by N.

Lemma 3.6. If $t \in supp([\![M]\!]) \cap supp([\![N^{\sim}]\!])$, then $t \in supp([\![N]\!])$.

Proof. Let $t \in supp([\![M]\!]) \cap supp([\![N^{\sim}]\!])$. There is then a unique run $r \in runs_M(t)$. Intuitively, $\theta(q)$ records the quotient of the weight assigned by $N^{\sim}$ to the trees that reach q in M, divided by the weight assigned by $N^{\sim}$ to the trees that reach $[q]$. It follows that all nominators and denominators in the equations (2) and (3) of Definition 3.4 are nonzero when the weights are computed for every transition $r(v)$, $v \in dom(t)$, and the final weight $F'(r(\epsilon))$. Hence, $t \in supp([\![N]\!])$. $\square$

For N be useful in the main proof, we need to verify that it is a probabilistic wta as per Definition 2.1.

Lemma 3.7. The automaton N is an end- and out-probabilistic wta.

Proof. The definition of the final weight function F' contains a normalisation factor in the denominator which ensures that the weights $F'(q), q \in Q$, sum up to 1. This means that N is end-probabilistic.

Let us now verify that N is out-probabilistic. For every $q \in Q$,

$$\sum_{k \in \mathbb{N}, \sigma \in \Sigma_k, q_1 \cdots q_k \in Q^k} \delta'_k(q_1 \cdots q_k, \sigma, q)$$

$= $ by Definition 3.4 of the automaton N

$$\sum_{(q_1 \cdots q_k, \sigma, q) \in supp(\delta)} \delta^{\sim}_k([q_1]_{\sim} \cdots [q_k]_{\sim}, \sigma, [q]_{\sim}) \cdot \theta(q)^{-1} \cdot \prod_{i \in [k]} \theta(q_i)$$

$= $ because multiplication distributes over addition

$$\frac{\sum_{(q_1 \cdots q_k, \sigma, q) \in supp(\delta)} \delta^{\sim}_k([q_1]_{\sim} \cdots [q_k]_{\sim}, \sigma, [q]_{\sim}) \cdot \prod_{i \in [k]} \theta(q_i)}{\theta(q)}$$

$= $ by Definition 3.2 of θ (which occurs in the denominator), Definition 3.3 of θ_i (which corresponds to the nominator), and Lemma 3.4 which states that for every $q \in Q$, $\theta(q) = \lim_{i \to \infty} \theta_i(q)$

1 . $\square$

The following Lemma 3.8 relates the weights assigned by the automata N and $N^{\sim}$ to the trees in $supp([\![M]\!])$.

Lemma 3.8. For every $t \in supp([\![M]\!]) \cap supp([\![N^{\sim}]\!])$,

$$[\![N]\!](t) = \frac{1}{\sum_{p \in [q]_{\sim}} \theta(p)} \cdot [\![N^{\sim}]\!](t) \ .$$

Proof. Let $t \in supp([\![M]\!]) \cap supp([\![N^{\sim}]\!])$. By Lemma 3.6, there is exactly one run $r \in runs_N(t)$ and one run $r' \in runs_{N^{\sim}}(t)$ with non-zero weights. Let $q = r(\epsilon)$. We compute the weight of r through the equations below. (We refer to this computation as the series of equations *.)

$w_N(r)$

$=$ by the definition of w_N in Section 2

$$\prod_{v \in dom(t)} \delta'_k(r(v1)\cdots r(vk), t(v), r(v))$$

$=$ by Definition 3.4 of δ'

$$\prod_{v \in dom(t)} \frac{\delta^{\sim}_k([r(v1)]_{\sim}\cdots[r(vk)]_{\sim}, t(v), [r(v)]_{\sim}) \cdot \prod_{i\in[k]} \theta(r(vi))}{\theta(r(v))}$$

$=$ by simplification through cancellation at each node v, where the weights divided away at vi, $i \in [k]$, are multiplied back in at the node v

$$\frac{\prod_{v \in dom(t)} \delta^{\sim}_k([r(v1)]_{\sim}\cdots[r(vk)]_{\sim}, t(v), [r(v)]_{\sim})}{\theta(q)}$$

$=$ by Definition 3.1 of merged wta, and because $N^{\sim}$ is deterministic

$$\frac{w_{N^{\sim}}(r')}{\theta(q)}$$

Using this relation between the weight functions, we find that:

$[\![N]\!](t)$

$=$ by the definition of wta semantics in Section 2

$F'(q) \cdot w_N(r)$

$=$ by Definition 3.4 of F'

$$F^{\sim}([q]_{\sim}) \cdot \frac{\theta(q)}{\sum_{p\in[q]_{\sim}} \theta(p)} \cdot w_N(r)$$

$=$ by the sequence of equations (*) above

$$F^{\sim}([q]_{\sim}) \cdot \frac{\theta(q)}{\sum_{p\in[q]_{\sim}} \theta(p)} \cdot \frac{w_{N^{\sim}}(r')}{\theta(q)}$$

$=$ by cancelling and reordering terms

$$\frac{1}{\sum_{p\in[q]_{\sim}} \theta(p)} \cdot F^{\sim}([q]_{\sim}) \cdot w_{N^{\sim}}(r')$$

$=$ by the definition of wta semantics in Section 2

$$\frac{1}{\sum_{p\in[q]_{\sim}} \theta(p)} \cdot [\![N^{\sim}]\!](t) \quad . \quad \square$$

To ensure that N is a probabilistic wta, it remains to verify consistency.

Lemma 3.9. The automaton N is consistent.

Proof.

$$\sum_{t\in T_\Sigma} [\![N]\!](t)$$

$=$ by associativity and commutativity

$$\sum_{P\in Q^{\sim}} \sum_{p\in P} \sum_{t\in supp([\![N_p]\!])} [\![N]\!](t)$$

$=$ by Lemma 3.8 which relates $[\![N]\!](t)$ to $[\![N^{\sim}]\!](t)$

$$\sum_{P\in Q^{\sim}} \sum_{p\in P} \sum_{t\in supp([\![N_p]\!])} \frac{1}{\sum_{p\in P}\theta(p)} \cdot [\![N^{\sim}]\!](t)$$

$=$ by distributivity

$$\sum_{P\in Q^{\sim}} \frac{1}{\sum_{p\in P}\theta(p)} \sum_{p\in P} \sum_{t\in supp([\![N_p]\!])} [\![N^{\sim}]\!](t)$$

$=$ by construction of θ and by changing the range of the final sum

$$\sum_{P\in Q^{\sim}} \frac{1}{\sum_{p\in P}\theta(p)} \sum_{p\in P}\theta(p) \sum_{t\in supp([\![N^{\sim}_P]\!])} [\![N^{\sim}]\!](t)$$

$=$ by distributivity

$$\sum_{P\in Q^{\sim}} \frac{1}{\sum_{p\in P}\theta(p)} \left(\sum_{p\in P}\theta(p)\right) \sum_{t\in supp([\![N^{\sim}_P]\!])} [\![N^{\sim}]\!](t)$$

$=$ by cancellation

$$\sum_{P\in Q^{\sim}} \sum_{t\in supp([\![N^{\sim}_P]\!])} [\![N^{\sim}]\!](t)$$

$=$ since $N^{\sim}$ is consistent

1 . □

We now connect the intermediate results into a proof of Dietze's conjecture.

Proof of Theorem 3.1. By Lemma 3.7 and Lemma 3.9, the wta N is a probabilistic wta and by construction, it is in $prob(M)$. For every $t \in supp([\![M]\!]) \cap supp([\![N^{\sim}]\!])$,

$$[\![N]\!](t) = \frac{1}{\sum_{p\in[q_t]_{\sim}}\theta(q_t)} \cdot [\![N^{\sim}]\!](t) \ .$$

By Definition 3.2 of θ, the sum $\sum_{p\in[q_t]_{\sim}}\theta(p)$ cannot exceed 1, and it is assumed that $\mathcal{C} \subseteq supp([\![M]\!])$, so for every $t \in \mathcal{C}$,

$$[\![N]\!](t) \geq [\![N^{\sim}]\!](t) \ .$$

By definition of the maximum likelihood estimation,

$$\mathrm{L}(\mathcal{C} \mid [\![mle_{\mathcal{C}}(M)]\!]) \geq \mathrm{L}(\mathcal{C} \mid [\![N]\!])$$

from which it follows that

$$\mathrm{L}(\mathcal{C} \mid [\![mle_{\mathcal{C}}(M)]\!]) \geq \mathrm{L}(\mathcal{C} \mid [\![N^{\sim}]\!]) \ ,$$

including when $[\![N^{\sim}]\!] = mle_{\mathcal{C}}(M^{\sim})$. □

4. Limitations

Let us now see what happens if a subset of the assumptions made in Theorem 3.1 are relaxed. The first assumption that we will consider is that the support of the corpus $\mathcal{C}$ is contained in the support of the wta M. This is indeed necessary, because without it, state-merging can be used to expand the support of M so that it encompasses a larger part of the corpus $\mathcal{C}$. Consequently, the likelihood of $\mathcal{C}$ with respect to probabilistic wta derived from the smaller automaton may increase [6]. Let us give a concrete example:

Example 4.1. Fig. 4 shows the wta $M = (Q, \Sigma, \mathbb{B}, F, \delta)$ over the Boolean semiring, where $Q = \{q_1, q_2, r_1, r_2\}$, $\Sigma = \{a^{(1)}, b^{(1)}, c^{(0)}, d^{(0)}\}$, $F = \{q_1, r_1\}$, and

$$\delta = \{ \quad (q_2, a, q_1), \quad (r_2, b, r_1), \quad (\epsilon, d, q_2), \quad (\epsilon, c, r_2) \quad \} \ .$$

Suppose that the corpus is $\mathcal{C} = \{(a[c], 1)\}$. Clearly, the automaton M is deterministic, but as it only accepts the trees $a[d]$ and $b[c]$, $supp(\mathcal{C}) \not\subseteq supp([\![M]\!])$. Now, let $\sim = \{\{q_1, r_1\}, \{q_2, r_2\}\}$. Then, $M^{\sim} = (\{\{q_1, r_1\}, \{q_2, r_2\}\}, \Sigma, \mathbb{B}, F^{\sim}, \delta^{\sim})$, where $F^{\sim} = \{\{q_1, r_1\}\}$, and

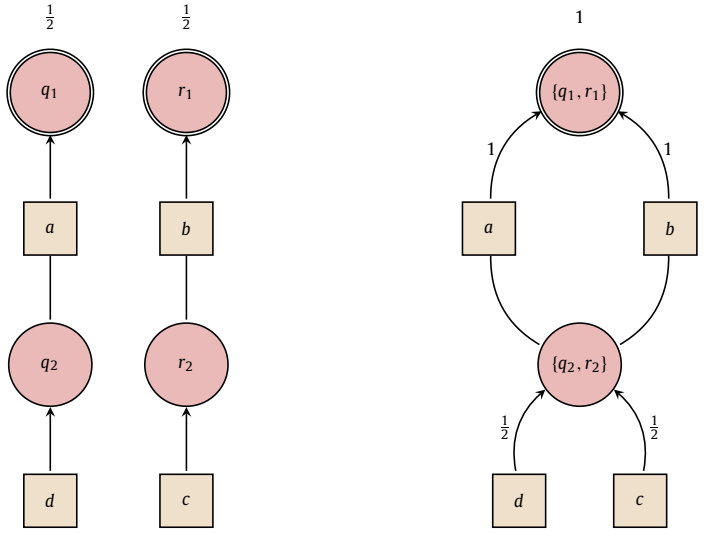


Fig. 4. The wta M on the left over the Boolean semiring does not cover the corpus $\mathcal{C} = \{a[c]\}$. However, by merging M with respect to an equivalence relation $\sim = \{\{q_1, r_1\}, \{q_2, r_2\}\}$ we obtain the wta $M^{\sim}$ (on the right, disregarding the weights). We can now find the wta $N^{\sim} \in prob(M^{\sim})$ (again on the right, but now including the weights) which does cover the corpus. It is however not possible to find a corresponding $N \in prob(M)$, which shows that the requirement that $supp(\mathcal{C}) \subseteq supp(\llbracket M \rrbracket)$ in Dietze's conjecture is indeed necessary.

$$\delta^{\sim} = \{ \quad (\{q_2, r_2\}, a, \{q_1, r_1\}), \quad (\{q_2, r_2\}, b, \{q_1, r_1\}), \\ (\epsilon, d, \{q_2, r_2\}), \qquad (\epsilon, c, \{q_2, r_2\}) \qquad \} \ .$$

Also the wta $M^{\sim}$ (on the right in Fig. 4) is deterministic, but now $supp(\mathcal{C}) \subseteq supp(\llbracket M^{\sim} \rrbracket)$. This allows us to find an $N^{\sim} = (\{\{q_1, r_1\}, \{q_2, r_2\}\}, \Sigma, \mathcal{A}, \delta'', F'') \in prob(M^{\sim})$ such that $F''(\{q_1, r_1\}) = 1$, $F''(\{q_2, r_2\}) = 0$, and

$$\delta'' = \{ \quad ((\{q_2, r_2\}, a, \{q_1, r_1\}), \tfrac{1}{2}), \quad ((\{q_2, r_2\}, b, \{q_1, r_1\}), \tfrac{1}{2}), \\ ((\epsilon, d, \{q_2, r_2\}), \tfrac{1}{2}), \qquad ((\epsilon, c, \{q_2, r_2\}), \tfrac{1}{2}) \qquad \} \ .$$

It follows that $\mathrm{L}(\mathcal{C} \mid \llbracket M^{\sim} \rrbracket) = \frac{1}{4}$, whereas we have $\mathrm{L}(\mathcal{C} \mid \llbracket N \rrbracket) = 0$ for every wta $N \in prob(M)$. $\Diamond$

To the best of our knowledge, it is still an open question whether Theorem 3.1 can be extended to nondeterministic automata. However, as the following example shows, the particular construction used to prove Theorem 3.1 in this article fails when the merging introduces nondeterminism.

Example 4.2. Let us now consider an extension of the wta in Example 4.1. Let $M = (Q, \Sigma, \mathbb{B}, F, \delta)$ be a wta over the Boolean semiring (shown on the left of Fig. 5), where we have $Q = \{p_1, p_2, q_1, q_2, r_1, r_2\}$, $\Sigma = \{a^{(1)}, b^{(1)}, c^{(0)}, d^{(0)}\}$, $F = \{p_1, q_1, r_1\}$, and

$$\delta = \{ \quad (p_2, a, p_1), \quad (q_2, a, q_1), \quad (r_2, b, r_1), \\ (\epsilon, c, p_2), \quad (\epsilon, d, q_2), \quad (\epsilon, c, r_2) \qquad \} \ .$$

Again, let $\mathcal{C} = \{(a[c], 1)\}$, but this time, M is nondeterministic (because c can reach both p_2 and r_2) and $supp(\mathcal{C}) \subseteq supp(\llbracket M \rrbracket)$. Now, we let

$$\sim = \{\{p_1\}, \{p_2\}, \{q_1\}, \{r_1\}, \{q_2, r_2\}\} \ .$$

This yields $M^{\sim} = (Q^{\sim}, \Sigma, \mathbb{B}, \delta^{\sim}, F^{\sim})$, where $Q^{\sim} = \{\{p_1\}, \{p_2\}, \{q_1\}, \{r_1\}, \{q_2, r_2\}\}$, $F^{\sim} = \{\{p_1\}, \{q_1\}, \{r_2\}\}$, and

$$\delta^{\sim} = \{ \quad (\{p_2\}, a, \{p_1\}), \quad (\{q_2, r_2\}, a, \{q_1\}), \quad (\{q_2, r_2\}, b, \{r_1\}), \\ (\epsilon, c, p_2), \qquad (\epsilon, d, \{q_2, r_2\}), \qquad (\epsilon, c, \{q_2, r_2\}) \qquad \} \ .$$

The automaton $M^{\sim}$ (shown on the right of Fig. 5) is also nondeterministic as there are two distinct runs on the tree $a[c]$, one that passes through the states $\{p_2\}$ and $\{p_1\}$, and one that instead goes through $\{q_2, r_2\}$ and $\{q_1\}$. We can now find an automaton $N^{\sim} \in prob(M^{\sim})$ where $N^{\sim} = (Q^{\sim}, \Sigma, \mathcal{A}, \delta'', F'') \in prob(M^{\sim})$ such that $F''(\{q_1\}) = F''(\{r_1\}) = 1/2$ and 0 for all other arguments (in particular, for q_1), and

$$\delta'' = \{ (((\{p_2\}, a, \{p_1\}), 1), \ (((\{q_2, r_2\}, a, \{q_1\}), 1), \ (((\{q_2, r_2\}, b, \{r_1\}), 1), \\ ((\epsilon, c, \{p_2\}), 1), \qquad ((\epsilon, d, \{q_2, r_2\}), \tfrac{1}{2}), \qquad ((\epsilon, c, \{q_2, r_2\}), \tfrac{1}{2}) \qquad \} \ .$$

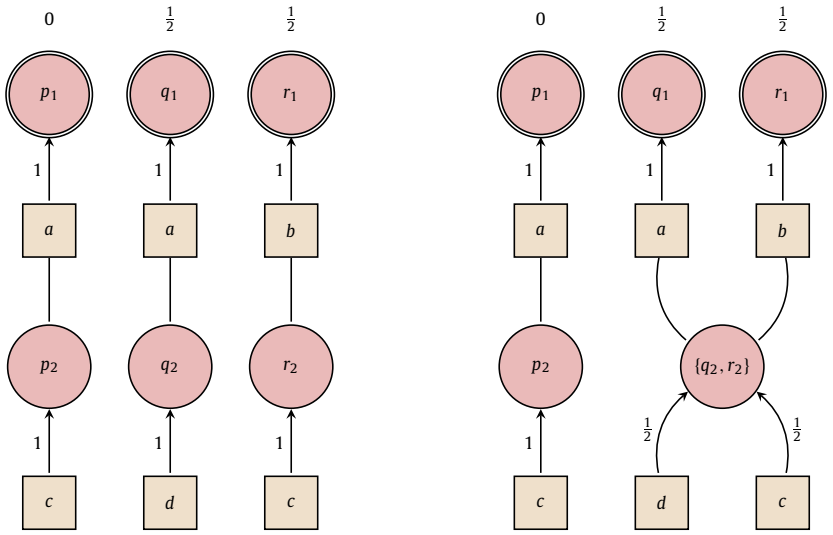


Fig. 5. The nondeterministic wta M on the left (disregarding the weights) covers the corpus $\mathcal{C} = \{a[c]\}$. By merging M with respect to an equivalence relation $\sim = \{\{q_1\}, \{r_1\}, \{q_2, r_2\}\}$ we obtain the wta $M^{\sim}$ on the right (again disregarding the weights). We can now find the wta $N^{\sim} \in prob(M^{\sim})$ (also on the right, but now including the weights) which covers the corpus. Although it is possible to find an $N \in prob(M)$ which makes the corpus equally likely, the construction presented here is insufficient, as it yields the wta on the left (now with weights).

If we compute the θ values for M from $N^{\sim}$ we get

	p_1	p_2	q_1	q_2	r_1	r_2
θ	1	1	1/2	1/2	1/2	1/2

Based on these, we build the wta $N = (Q, \Sigma, \mathbb{P}, \delta', F')$ in $prob(M)$ where Q and Σ are as above, and $F'(q_1) = F'(r_1) = 1/2$ and 0 for all other arguments, and the weight of every transition in δ' is 1. We see that $\mathrm{L}(\mathcal{C} \mid [\![M^{\sim}]\!]) = \frac{1}{4}$, whereas $\mathrm{L}(\mathcal{C} \mid [\![N]\!]) = 0$. In this particular case it is however easy to see that

$$\mathrm{L}(\mathcal{C} \mid [\![mle_{\mathcal{C}}(M)]\!]) \geq \mathrm{L}(\mathcal{C} \mid [\![mle_{\mathcal{C}}(M^{\sim})]\!])$$

because if we let $F'(p_1) = 1$ and 0 for all other states, and keep δ' as it is, then the entire probability mass of 1 will be given to the unique tree $a[c] \in \mathcal{C}$, and this is clearly the optimal fit between any automaton and the corpus. The above example thus shows that the construction provided in this paper fails to generalise to nondeterministic wta, but it is still open whether there is another construction that accomplishes this. ◊

5. Conclusion

We have seen that given an automaton M and an equivalence relation $\sim$ on the states of M, we can derive from every wta $N^{\sim} \in prob(M^{\sim})$ a wta $N \in prob(M)$ such that N assigns a weight to every tree in $supp([\![M]\!]) \cap supp([\![N^{\sim}]\!])$ that is greater or equal to that assigned by $N^{\sim}$. From this it follows that if $\mathcal{C} \subseteq supp([\![M]\!])$, then

$$\mathrm{L}(\mathcal{C} \mid [\![mle_{\mathcal{C}}(M)]\!]) \geq \mathrm{L}(\mathcal{C} \mid [\![mle_{\mathcal{C}}(M^{\sim})]\!]) \ .$$

As discussed in Section 4, it remains an open question whether Theorem 3.1 can be generalised to non-deterministic automata.

CRediT authorship contribution statement

Johanna Björklund: Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Johanna Bjoerklund reports financial support was provided by Swedish Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used ChatGPT to improve grammar and readability. The author only consulted the tool to check grammatical constructions, get suggestions on how to simplify complex sentences, and vary the language to make it less repetitive. She takes full responsibility for the content of the publication.

Acknowledgments

I would like to thank the reviewers of FCT 2023 for their constructive criticism that helped improve this article. I am also grateful to Frank Drewes for his willingness to act as a sounding board throughout the work.

References

[1] Henrik Björklund, Johanna Björklund, Petter Ericson, On the regularity and learnability of ordered DAG languages, in: Arnaus Carayol, Cyril Nicaud (Eds.), 22nd International Conference on the Implementation and Application of Automata (CIAA 2017), Marne-la-Vallée, in: Lecture Notes in Computer Science, vol. 10329, Springer International Publishing, France, 2017, pp. 27–39.
[2] Henrik Björklund, Frank Drewes, Petter Ericson, Between a rock and a hard place – uniform parsing for hyperedge replacement DAG grammars, in: 10th International Conference on Language and Automata Theory and Applications (LATA 2016), Prague, Czech Republic, 2016, 2016, pp. 521–532.
[3] Johanna Björklund, The impact of state merging on predictive accuracy in probabilistic tree automata: Dietze's conjecture revisited, in: Henning Fernau, Klaus Jansen (Eds.), Fundamentals of Computation Theory - 24th International Symposium, FCT 2023, Trier, Germany, September 18-21, 2023, Proceedings, in: Lecture Notes in Computer Science, vol. 14292, Springer Verlag, Trier, Germany, 2023, pp. 74–87.
[4] Anne Bruggemann-Klein, Makoto Murata, Derick Wood, Regular tree and regular hedge languages over unranked alphabets, Technical report, The Hong Kong University of Science and Technology, 2001.
[5] Michael Collins, Head-driven statistical models for natural language parsing, Comput. Linguist. 29 (4) (2003) 589–637.
[6] T. Dietze, A Formal View on Training of Weighted Tree Automata by Likelihood-Driven State Splitting and Merging, Technische Universität Dresden, 2004.
[7] Frank Drewes, Grammatical Picture Generation: A Tree-Based Approach, Springer, 2006.
[8] Zoltán Fülöp, Heiko Vogler, Syntax-Directed Semantics: Formal Models Based on Tree Transducers, Springer Science & Business Media, 2012.
[9] Zoltan Fülöp, Heiko Vogler, Weighted tree automata and tree transducers, in: Handbook of Weighted Automata, Springer Verlag, 2009, pp. 313–403.
[10] Dan Klein, Christopher D. Manning, Accurate unlexicalized parsing, in: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Sapporo, Japan, July 2003, pp. 423–430.
[11] Slav Petrov, Leon Barrett, Romain Thibaux, Dan Klein, Learning accurate, compact, and interpretable tree annotation, in: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 2006, pp. 433–440.