Combinatory Categorial Grammars as Generators of Weighted Forests

Andreas Maletti, Lena Katharina Schiffer[1,*]

[a] *Institute for Computer Science, Universität Leipzig, P.O. box 100 920, D-04009 Leipzig, Germany*

Abstract

Combinatory Categorial Grammar (CCG) is an extension of categorial grammar that is well-established in computational linguistics. It is mildly context-sensitive, so it is efficiently parsable and reaches an expressiveness that is suitable for describing natural languages. Weighted CCG (wCCG) are introduced as a natural extension of CCG with weights taken from an arbitrary commutative semiring. Their expressive power is compared to other weighted formalisms with special emphasis on the weighted forests generated by wCCG since the ability to express the underlying syntactic structure of an input sentence is a vital feature of CCG in the area of natural language processing. Building on recent results for the expressivity in the unweighted setting, the corresponding results are derived for the weighted setting for any commutative semiring. More precisely, the weighted forests generatable by wCCG are also generatable by weighted simple monadic context-free tree grammar (wsCFTG). If the rule system is restricted to application rules and composition rules of first degree, then the generatable weighted forests are exactly the regular weighted forests. Finally, when only application rules are allowed, then a proper subset of the regular weighted forests is generatable.

Keywords:
Combinatory Categorial Grammar, Regular Tree Language, Linear Context-free Tree Language, Weighted Tree Language, Commutative Semiring

1. Introduction

Combinatory categorial grammar (CCG) [1, 2] uses rules that are inspired by combinatory logic [3] to extend classical categorial grammar [4], which has the same expressivity as context-free grammar and is based on notions from proof theory [5, 6]. These additional rules increase its expressivity beyond the context-free languages into the class of languages generated by mildly context-sensitive grammar formalisms [7]. These are formalisms that are efficiently parsable (i.e., in polynomial time), are able to express a limited amount of cross-serial dependencies, and have the constant growth property. Due to these features and its notion of syntactic categories, which are an intuitive way of representing constituents in natural languages, CCG has become well-established in computational linguistics [1]. The linguistically motivated need to easily express specific structures gave rise to a variety of different variants [1, 8, 2, 9]. Oftentimes it is not clear how subtle changes of the CCG definition influence its expressive power. To deal with this, our goal is to identify the principal structures expressible by a common core of the formalisms and consider CCG as generators of formal languages. As linguistic structure calls for a representation that goes beyond strings and motivated by an application-driven interest in a weighted variant of CCG [10, 11, 12, 13], in the present contribution we investigate the ability of CCG to generate weighted forests.

We briefly explain the basic operating principle of CCG. A CCG is essentially a lexicon together with a rule system. The lexicon assigns syntactic categories to the symbols of an input string, and the rule system describes how neighboring categories can be combined to new categories. Each category has a target, which is similar to the

[*]Corresponding author
Email addresses: maletti@informatik.uni-leipzig.de (Andreas Maletti), schiffer@informatik.uni-leipzig.de (Lena Katharina Schiffer)
[1]Supported by the German Research Foundation (DFG) Research Training Group GRK 1763 'Quantitative Logics and Automata'

return type of a function, and optionally, a number of arguments. Different from functions, each argument has a directionality that indicates whether the argument is expected on the left or right side. If repeated combination of categories leads to a binary derivation tree that comprises all input symbols and is rooted in an initial category, then the input string is generated. There exist several operators for combining categories, of which the most important are the *composition rules* (i.e., rules based on the **B**-combinator of combinatory logic [3]). In these rules, a *primary category*, which expects an argument, and a *secondary category*, which provides it, are combined. This is similar to function composition. The degree of a composition rule is the number of arguments of the secondary category that get transferred to the output category. If the secondary category gets completely consumed (i.e., it coincides with the relevant argument of the primary category), then we call the rule an *application rule*. Other common rules, which are not considered here, are substitution and type-raising rules [1].

Next we give an overview of the existing work on the expressive power of CCG. We start with the string languages generated by different variants of CCG. This is also called string (or weak) generative capacity. The famous equivalence result due to Vijay-Shanker and Weir [14] shows that CCG, tree-adjoining grammar (TAG) [15], and linear indexed grammar [16] are equivalent in expressive power. An equivalent automaton model is the embedded push-down automaton [17]. In the definition of CCG used by Vijay-Shanker and Weir, the lexicon allows ε-entries, which assign syntactic categories to the empty string ε, and their construction heavily depends on this feature. Their rule system restricts rules to specific categories and limits the rule degree. When these rule restrictions are omitted, i.e., all rules up to a limited degree are allowed, the thus obtained *pure* CCG is strictly less expressive than TAG [9]. On the other hand, when unbounded generalized composition rules are permitted, CCG is strictly more expressive than TAG [18]. It has been shown that CCG with unbounded composition rules, rule restrictions, and ε-entries in the lexicon is in fact Turing-complete [19]. Prefix-closed CCG without target restrictions, in which the rules obey special closure properties, is less powerful. This even holds for multi-modal CCG [20, 9], which allows various types of directionality indicators that permit more control over which categories can be combined.

Now we turn our attention to the expressiveness of CCG when going beyond the level of strings, which is often called strong generative power or capacity. For this, let us first mention that there exist different notions of this term. Hockenmaier and Young [21] regard two formalisms as strongly equivalent if they capture the same sets of dependencies. Then there exist specific scrambling cases whose dependencies can be expressed by their CCG (which allows type-raising [1] as an additional combinator) but not by Lexicalized TAG (LTAG). Note that the set of forests generated by LTAG is strictly smaller than that of TAG [22]. Koller and Kuhlmann [23] show that CCG without rule restrictions and TAG generate incomparable classes of *dependency trees*. On the other hand, in this contribution, we consider two formalisms as strongly equivalent if their generated derivation forests coincide modulo relabelings. An example for two strongly equivalent formalisms are the well-known local and regular tree grammars [24]. Already [18] asks for this tree-generative capacity or expressiveness of the CCG derivation forests. In this context, we would also like to mention the work of Tiede [25], who studied the strong generative capacity of Lambek-style categorial grammars [26]. The tree-generative capacity of CCG without ε-entries has been investigated in [27, 28]. The generated trees are always binary because the rules allow only combining exactly two input categories. The forests of CCG rule trees (i.e., trees labeled by applied rules instead of categories) are included in the forests generated by simple monadic context-free tree grammar (sCFTG). This inclusion is proper if rule restrictions are prohibited. CCG with application and first-degree composition rules generates exactly the regular forests. Without the composition rules, only a proper subset can be generated. The last result is analogous to [29, Theorem 1.1], where the focus is on classical categorial grammar, which has only application rules and no rule restrictions.

In our previous work, we proved the converse inclusion of the result concerning sCFTG mentioned above by showing that the forests generated by sCFTG can also be generated by CCG and thereby showed strong equivalence of both formalisms [30]. Since sCFTG and TAG are strongly equivalent [31], this result also shows strong equivalence of CCG and TAG. Additionally, it proves strong equivalence of CCG and linear top-down push-down tree automata [32]. Since the construction avoids ε-entries, the result shows that these entries can be removed without decreasing the expressive power of CCG. Another consequence of the construction is that rule degree 2 and first-order categories (i.e., all arguments are atomic) are sufficient to give CCG its full expressive power.

As in most theoretical work on CCG, the rule system of the variant we investigate is finite and includes only application and composition operators. Additionally, we allow rule restrictions that further constrain the categories the rules can be applied to. A weighted variant of CCG has made its appearance in various applied settings in the form of *probabilistic* CCG [10, 11, 12, 13]. It is thus evident that there is an application-driven interest in the extension of

CCG to *weighted* CCG (wCCG). However, while there is a wide range of theoretic work on the expressive power of unweighted CCG, there is an apparent lack of investigations into the weighted variants. Probabilistic CCG constitute only a single variant of weighted CCG, and in this contribution we allow weights from an arbitrary commutative semiring on both the rules as well as the lexicon entries. This general definition encompasses most existing definitions and allows a fundamental study of weighted CCG. We establish relationships to other well-understood weighted formalisms in order to understand the expressive power of wCCG. In particular, we are interested in the weighted forests generated by wCCG, since we consider the ability to express the underlying structure of an input sentence as a vital feature of CCG, whose most important area of application is natural language processing.

A forest is generatable by an unweighted CCG if it can be obtained by relabeling its set of derivation trees, so as mentioned above, all results refer to forests of binary trees. In the weighted case, each rule and each lexicon entry is associated with a weight. To obtain the weight of a given tree, we sum over the weights of all derivation trees that are relabeled to that particular tree, where the weight of a single derivation tree is the product of the weights of the applied rules and used lexicon entries (with each occurrence yielding a separate factor). On that basis, we extend the results shown in [27, 28] to the weighted scenario and investigate the classes of weighted forests that can be generated by three different variants of wCCG. Our main result is that the weighted forests generatable by wCCG are included in those generated by weighted simple monadic context-free tree grammar (wsCFTG). When restricting the rule system to application rules and composition rules of first degree, the generated weighted forests are exactly the regular weighted forests. Finally, when only application rules are allowed, a proper subclass of the regular weighted forests is generated.

2. Preliminaries

First, we introduce some general notation. We denote the set of nonnegative integers by $\mathbb{N}$ and the positive integers by $\mathbb{N}_+$. For every $k \in \mathbb{N}$, let $[k] = \{i \in \mathbb{N}_+ \mid i \leq k\}$ and $\mathbb{Z}_k = \{i \in \mathbb{N} \mid i \leq k-1\}$. For every $i, j \in \mathbb{N}$ with $i \leq j$, let $[i, j] = \{n \in \mathbb{N} \mid i \leq n \leq j\}$. The powerset (i.e., set of all subsets) of a set A is $\mathcal{P}(A) = \{A' \mid A' \subseteq A\}$, and the set of all nonempty subsets is denoted by $\mathcal{P}_+(A) = \mathcal{P}(A) \setminus \{\emptyset\}$. As usual, an alphabet is a finite set of symbols. The monoid $(\Sigma^*, \cdot, \varepsilon)$ consists of all strings (i.e., sequences) over a set Σ together with concatenation $\cdot$ and the empty string ε, where concatenation is often written by juxtaposition. The length of a string $w \in \Sigma^*$ is denoted by $|w|$. A language is a set $\mathcal{L} \subseteq \Sigma^*$ of strings. Languages form a monoid $(\mathcal{P}(\Sigma^*), \cdot, \{\varepsilon\})$, where concatenation is lifted to languages $\mathcal{L}$ and $\mathcal{L}'$ by $\mathcal{L} \cdot \mathcal{L}' = \{w \cdot w' \mid w \in \mathcal{L}, w' \in \mathcal{L}'\}$. Given two sets A and A', a relation from A to A' is a subset $\rho \subseteq A \times A'$. Its inverse is $\rho^{-1} = \{(a', a) \mid (a, a') \in \rho\}$. Further, for every $B \subseteq A$ we let $\rho(B) = \{a' \mid \exists b \in B \colon (b, a') \in \rho\}$.

Next we introduce notation for trees. In this contribution we restrict ourselves to binary trees. They are built over the set Σ_2 of binary internal symbols, the alphabet Σ_1 of unary internal symbols, and the alphabet Σ_0 of leaf symbols. Note that we explicitly allow an infinite set of internal binary symbols. Given sets T and T' we let

$$\Sigma_2(T, T') = \{\sigma(t, t') \mid \sigma \in \Sigma_2, t \in T, t' \in T'\}$$

and $\Sigma_1(T) = \{\sigma(t) \mid \sigma \in \Sigma_1, t \in T\}$. We write $T_{\Sigma_2,\Sigma_1}(\Sigma_0)$ for the set of binary (Σ_2, Σ_1)-trees indexed by Σ_0, and it is defined as the smallest set T such that (i) $\Sigma_0 \subseteq T$, (ii) $\Sigma_1(T) \subseteq T$, and (iii) $\Sigma_2(T, T) \subseteq T$. A forest is a subset $\mathcal{F} \subseteq T_{\Sigma_2,\Sigma_1}(\Sigma_0)$. We assign positions to a tree using the mapping $\text{pos} \colon T_{\Sigma_2,\Sigma_1}(\Sigma_0) \to \mathcal{P}_+(\{1, 2\}^*)$ and the leaf sequence, called yield, using the mapping $\text{yield} \colon T_{\Sigma_2,\Sigma_1}(\Sigma_0) \to \Sigma_0^*$. These mappings are defined by

$$\begin{aligned} \text{pos}(a) &= \{\varepsilon\} & \text{yield}(a) &= a \\ \text{pos}(n(t)) &= \{\varepsilon\} \cup \{1 \cdot w \mid w \in \text{pos}(t)\} & \text{yield}(n(t)) &= \text{yield}(t) \\ \text{pos}(c(t, t')) &= \{\varepsilon\} \cup \{1 \cdot w \mid w \in \text{pos}(t)\} \cup \{2 \cdot w \mid w \in \text{pos}(t')\} & \text{yield}(c(t, t')) &= \text{yield}(t)\, \text{yield}(t') \ , \end{aligned}$$

where $a \in \Sigma_0$, $n \in \Sigma_1$, $c \in \Sigma_2$, and $t, t' \in T_{\Sigma_2,\Sigma_1}(\Sigma_0)$. Given a tree t, let $\text{leaves}(t) = \{w \in \text{pos}(t) \mid w \cdot 1 \notin \text{pos}(t)\}$ be its leaf positions, and let $\text{ht}(t) = \max\{\, |w| \mid w \in \text{leaves}(t)\}$ be its height. We denote the subtree of t at position $w \in \text{pos}(t)$ by $t|_w$ and the label at position w by $t(w)$. Furthermore, $t[t']_w$ denotes the tree obtained from t by replacing the subtree at position w by the tree $t' \in T_{\Sigma_2,\Sigma_1}(\Sigma_0)$.

We reserve the use of the special symbol $\square$. We define the set of *contexts* as the set of all trees of $T_{\Sigma_2,\Sigma_1}(\Sigma_0 \cup \{\square\})$ in which the special symbol $\square$ occurs exactly once and denote this set by $C_{\Sigma_2,\Sigma_1}(\Sigma_0)$. A context can be interpreted as a structure containing a gap at the position of $\square$ such that a tree can be inserted at this position to obtain a (potentially different) tree. For a context $c \in C_{\Sigma_2,\Sigma_1}(\Sigma_0)$, we write $\mathrm{pos}_\square(c)$ to refer to the unique position $w \in \mathrm{pos}(c)$ with $c(w) = \square$. Given $t \in T_{\Sigma_2,\Sigma_1}(\Sigma_0) \cup C_{\Sigma_2,\Sigma_1}(\Sigma_0)$, we write $c[t]$ or even tc for $c[t]_w$ with $w = \mathrm{pos}_\square(c)$ for convenience. The order tc might seem unintuitive at this point, but will prove beneficial when dealing with argument contexts c, which will be introduced in the next section.

Given an alphabet Δ, a (deterministic) *relabeling* is a mapping $\rho\colon (\Sigma_2 \cup \Sigma_1 \cup \Sigma_0) \to \Delta$. It induces a mapping $\widehat{\rho}\colon T_{\Sigma_2,\Sigma_1}(\Sigma_0) \to T_{\Delta,\Delta}(\Delta)$ on trees such that $\widehat{\rho}(t) = u$ with $\mathrm{pos}(u) = \mathrm{pos}(t)$ and $u(w) = \rho(t(w))$ for all $w \in \mathrm{pos}(u)$. We again do not distinguish between the relabeling ρ and its induced mapping $\widehat{\rho}$ on trees.

Next we introduce the algebraic structure for our weights. A *commutative semiring* [33, 34] is a tuple $(H, +, \cdot, 0, 1)$ consisting of two commutative monoids $(H, +, 0)$ and $(H, \cdot, 1)$ such that $h \cdot 0 = 0$ for every $h \in H$ and multiplication $\cdot$ *distributes* over addition $+$; i.e., $h_1 \cdot (h_2 + h_3) = (h_1 \cdot h_2) + (h_1 \cdot h_3)$ for every $h_1, h_2, h_3 \in H$. For the rest of the contribution, let $(H, +, \cdot, 0, 1)$ be an arbitrary commutative semiring. The *support* $\mathrm{supp}(\varphi)$ of a mapping $\varphi\colon A \to H$ is given by $\mathrm{supp}(\varphi) = \{a \in A \mid \varphi(a) \neq 0\}$, where A is a set.

A *weighted forest* is a mapping $\varphi\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$; i.e., each tree $t \in T_{\Sigma_2,\emptyset}(\Sigma_0)$ is assigned a weight $\varphi(t) \in H$. For a thorough introduction to weighted string languages and weighted forests we refer to [35].

A *weighted simple monadic context-free tree grammar in normal form* (wsCFTG) [36, 37, 38] is a tuple

$$G = (N, \Sigma, S, P, \mathrm{wt})$$

such that

- $N = N_1 \cup N_0$ for alphabets N_1 and N_0 of unary and nullary *nonterminals*, respectively,
- $\Sigma = \Sigma_2 \cup \Sigma_0$ for alphabets Σ_2 and Σ_0 of internal and leaf *terminals*, respectively, with $N \cap \Sigma = \emptyset$,
- $S \subseteq N_0$ is a subset of nullary *start nonterminals*,
- $P = P_1 \cup P_0$ is a finite set of *productions* split into P_1 and P_0 for unary and nullary nonterminals, respectively, where

$$P_1 \subseteq N_1 \times \Big(\Sigma_2(\{\square\}, N_0) \cup \Sigma_2(N_0, \{\square\}) \cup N_1(N_1(\{\square\}))\Big)$$
$$P_0 \subseteq N_0 \times \Big(\Sigma_2(N_0, N_0) \cup \Sigma_0 \cup N_1(N_0)\Big)$$

- $\mathrm{wt}\colon P \to H$ is a *weight assignment* to each production.

If $N_1 = \emptyset$, then G is a *weighted tree automaton* (wTA) [39, 40].

A production $(n, r) \in P$ is also written as $n \to r$. The wsCFTG G is called *monadic* because the nonterminals are either nullary or unary, and it is *simple* because all productions are linear and nondeleting (i.e., the right-hand side r is a context for each production $n \to r \in P_1$). The set P of productions forms a ranked alphabet where the rank of a symbol $n \to r \in P$ is the number $k \in \mathbb{Z}_3$ of nonterminals in r. The set of *derivation trees* of G starting in $n \in N$, denoted by $\mathcal{D}_G^n$, are inductively defined for all $n \in N$ to be the smallest sets $(\mathcal{D}_G^n)_{n\in N}$ such that $\mathcal{D}_G^n$ contains all trees d over this ranked alphabet such that $d = (n \to r)(d_1, \dots, d_k)$, where $n \to r \in P$, right-hand side r contains the k ordered nonterminals $n_1, \dots, n_k \in N$, and $d_i \in \mathcal{D}_G^{n_i}$ for all $i \in [k]$. As usual, we let $\mathcal{D}_G = \bigcup_{n\in N} \mathcal{D}_G^n$. The *weight* of a derivation tree $d \in \mathcal{D}_G$ is $\mathrm{wt}_G(d) = \prod_{w\in\mathrm{pos}(d)} \mathrm{wt}(d(w))$.

Derivation trees evaluate to terminal trees or contexts via the map $\mathrm{eval}_G\colon \mathcal{D}_G \to T_{\Sigma_2,\emptyset}(\Sigma_0) \cup C_{\Sigma_2,\emptyset}(\Sigma_0)$, which is defined for every $n_0, n_1, n_2 \in N_0$, $c_0, c_1, c_2 \in N_1$, $\alpha \in \Sigma_0$, $\sigma \in \Sigma_2$, $d_1 \in \mathcal{D}_G^{n_1}$, $d_2 \in \mathcal{D}_G^{n_2}$, $d_1' \in \mathcal{D}_G^{c_1}$, and $d_2' \in \mathcal{D}_G^{c_2}$ by

$$\mathrm{eval}_G((n_0 \to \alpha)) = \alpha$$
$$\mathrm{eval}_G\Big((n_0 \to \sigma(n_1, n_2))(d_1, d_2)\Big) = \sigma(\mathrm{eval}_G(d_1), \mathrm{eval}_G(d_2))$$
$$\mathrm{eval}_G\Big((n_0 \to c_1(n_2))(d_1', d_2)\Big) = \mathrm{eval}_G(d_1')[\mathrm{eval}_G(d_2)]$$
$$\mathrm{eval}_G\Big((c_0 \to \sigma(\square, n_1))(d_1)\Big) = \sigma(\square, \mathrm{eval}_G(d_1))$$
$$\mathrm{eval}_G\Big((c_0 \to \sigma(n_1, \square))(d_1)\Big) = \sigma(\mathrm{eval}_G(d_1), \square)$$
$$\mathrm{eval}_G\Big((c_0 \to c_1(c_2(\square)))(d_1', d_2')\Big) = \mathrm{eval}_G(d_1')[\mathrm{eval}_G(d_2')] \ .$$

Note that $\text{eval}_G(d) \in T_{\Sigma_2,\emptyset}(\Sigma_0)$ for all $d \in \mathcal{D}_G^n$ with $n \in N_0$, and $\text{eval}_G(d') \in C_{\Sigma_2,\emptyset}(\Sigma_0)$ for all $d' \in \mathcal{D}_G^c$ with $c \in N_1$. For every $t \in T_{\Sigma_2,\emptyset}(\Sigma_0)$ and $n \in N_0$ we let $\mathcal{D}_G^n(t) = \{d \in \mathcal{D}_G^n \mid \text{eval}(d) = t\}$. The *weighted forest generated by G* is the mapping $\tau_G\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$, which is given for every $t \in T_{\Sigma_2,\emptyset}(\Sigma_0)$ by

$$\tau_G(t) = \sum_{n \in S,\, d \in \mathcal{D}_G^n(t)} \text{wt}_G(d) \enspace .$$

For convenience, we also introduce the run semantics [40] for weighted tree automata. Let $G = (N, \Sigma, S, P, \text{wt})$ be a wTA. A *run of G on* a tree $t \in T_{\Sigma_2,\emptyset}(\Sigma_0)$ is a mapping $\omega\colon \text{pos}(t) \to N_0$ such that (i) $\omega(w) \to t(w)(\omega(w1), \omega(w2)) \in P$ for all internal positions $w \in \text{pos}(t) \setminus \text{leaf}(t)$ and (ii) $\omega(w) \to t(w) \in P$ for all leaves $w \in \text{leaf}(t)$. The set of all runs of G on t is denoted by $R_G(t)$, and for every nonterminal $n \in N_0$ we let $R_G^n(t) = \{\omega \in R_G(t) \mid \omega(\varepsilon) = n\}$ be the set of all runs with nonterminal n at the root. The *weight* $\text{wt}_G(\omega)$ of a run $\omega \in R_G(t)$ is

$$\text{wt}_G(\omega) = \Bigg(\prod_{w \in \text{pos}(t) \setminus \text{leaf}(t)} \text{wt}\big(\omega(w) \to t(w)(\omega(w1), \omega(w2))\big)\Bigg) \cdot \Bigg(\prod_{w \in \text{leaf}(t)} \text{wt}(\omega(w) \to t(w))\Bigg) \enspace ,$$

so it coincides with the product of the weights of the productions used in the run (with each occurrence yielding a separate factor). Obviously, a run is just a slightly less verbose variant of a derivation tree yielding a weight-preserving bijection between $\mathcal{D}_G^n(t)$ and $R_G^n(t)$ for every $n \in N_0$ and $t \in T_{\Sigma_2,\emptyset}(\Sigma_0)$; i.e., bijection $\vartheta\colon \mathcal{D}_G^n(t) \to R_G^n(t)$ such that $\text{wt}_G(d) = \text{wt}_G(\vartheta(d))$ for all $d \in \mathcal{D}_G^n(t)$. With this knowledge it is trivial to show that $\tau_G(t) = \sum_{n \in S,\, \omega \in R_G^n(t)} \text{wt}_G(\omega)$ for all $t \in T_{\Sigma_2,\emptyset}(\Sigma_0)$, so the weight assigned by G to t coincides with the sum of the weights of the runs of $R_G^n(t)$ with $n \in S$. A weighted forest $\varphi\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$ is *regular* if there exists a wTA G such that $\varphi = \tau_G$. In the special case of the Boolean semiring $\mathbb{B} = (\{0, 1\}, \max, \min, 0, 1)$ we obtain the *regular forests* as the supports of the regular weighted forests [40]; i.e., for every regular forest $L \subseteq T_{\Sigma_2,\emptyset}(\Sigma_0)$ there exists a wTA G over the Boolean semiring such that $L = \text{supp}(\tau_G)$.

A wTA $G = (N, \Sigma, S, P, \text{wt})$ is *terminal-normalized* if there exists a mapping $\kappa\colon N_0 \to \Sigma$ such that $r(\varepsilon) = \kappa(n)$ for all productions $n \to r \in P$. In other words, in a terminal-normalized wTA each nonterminal $n \in N_0$ can only generate a single terminal symbol $\kappa(n)$. It is a routine matter to verify that for every wTA there exists an equivalent terminal-normalized wTA. Finally, if κ is injective, then G is a *weighted local tree grammar* [41] and for those, a production $n \to \sigma(n_1, n_2) \in P$ is simply written as $\frac{\sigma}{\kappa(n_1) \quad \kappa(n_2)}$ and $n' \to \alpha \in P$ is written as $\frac{\alpha}{}$. We may omit n, n', n_1, n_2 as these are uniquely determined by $n = \kappa^{-1}(\sigma)$, $n' = \kappa^{-1}(\alpha)$, $n_1 = \kappa^{-1}(\kappa(n_1))$, and $n_2 = \kappa^{-1}(\kappa(n_2))$, respectively. We similarly identify terminals and nonterminals in runs, so $R_G(t) \subseteq \{t\}$ for every $t \in T_{\Sigma_2,\emptyset}(\Sigma_0)$. For weighted local tree grammars, we thus let $R_G = \bigcup_{t \in T_{\Sigma_2,\emptyset}(\Sigma_0)} R_G(t)$ and $R_G^\sigma = \{t \in R_G \mid t(\varepsilon) = \sigma\}$. Finally, we present the weighted local tree grammar simply as $(\Sigma, \{\kappa(n) \mid n \in S\}, P, \text{wt})$. Finally, a weighted forest $\varphi\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$ is *local* if there exists a weighted local tree grammar G such that $\varphi = \tau_G$. The local forests are again the supports of local weighted forests over the Boolean semiring $\mathbb{B}$.

Example 1. *Let wsCFTG* $G_1 = (N, \Sigma, \{s_0\}, P, \text{wt})$ *be weighted over the semiring of nonnegative integers* $(\mathbb{N}, +, \cdot, 0, 1)$ *with ordinary addition and multiplication such that* $N_0 = \{s_0, a_0, b_0\}$, $N_1 = \{s_1, a_1, b_1\}$, $\Sigma_2 = \{\sigma\}$, $\Sigma_0 = \{\alpha, \beta\}$, *and*

$$P = \{\; s_0 \to s_1(s_0),\;\; s_0 \to \sigma(a_0, b_0),\;\; s_1 \to a_1(b_1(\square)),\;\; a_1 \to \sigma(a_0, \square),\;\; b_1 \to \sigma(\square, b_0),\;\; a_0 \to \alpha,\;\; b_0 \to \beta \;\}$$

with $\text{wt}(s_0 \to s_1(s_0)) = \text{wt}(s_0 \to \sigma(a_0, b_0)) = 2$ *and* $\text{wt}(p) = 1$ *for all other productions* $p \in P$. *Figure 1 shows a derivation tree* $d \in \mathcal{D}_G^{s_0}(t)$ *with* $\text{wt}(d) = 8$ *and the corresponding terminal tree* $t = \text{eval}_{G_1}(d)$.

Example 2. *Let wTA* $G_2 = (N, \Sigma, \{s\}, P, \text{wt})$ *be weighted over the semiring of nonnegative integers* $(\mathbb{N}, +, \cdot, 0, 1)$ *such that* $N_0 = \{s, n, a, b\}$, $\Sigma_2 = \{\sigma\}$, $\Sigma_0 = \{\alpha, \beta\}$, *and*

$$P = \{\; s \to \sigma(a, n),\;\; s \to \sigma(a, b),\;\; n \to \sigma(s, b),\;\; a \to \alpha,\;\; b \to \beta \;\}$$

with $\text{wt}(s \to \sigma(a, n)) = \text{wt}(s \to \sigma(a, b)) = 2$ *and* $\text{wt}(p) = 1$ *for all other productions* $p \in P$. *Figure 2 shows a derivation tree* $d' \in \mathcal{D}_{G_2}^s(t)$ *with* $\text{wt}(d') = 8$, *where t is the same terminal tree as in Example 1. Next to the derivation tree, the corresponding run* $b(d') \in R_{G_2}^s(t)$ *is depicted. The wTA* G_2 *is terminal-normalized with* $\kappa(s) = \kappa(n) = \sigma$, $\kappa(a) = \alpha$, *and* $\kappa(b) = \beta$. *However, since* $\kappa(s)$ *and* $\kappa(n)$ *coincide, it is not a weighted local tree grammar.*

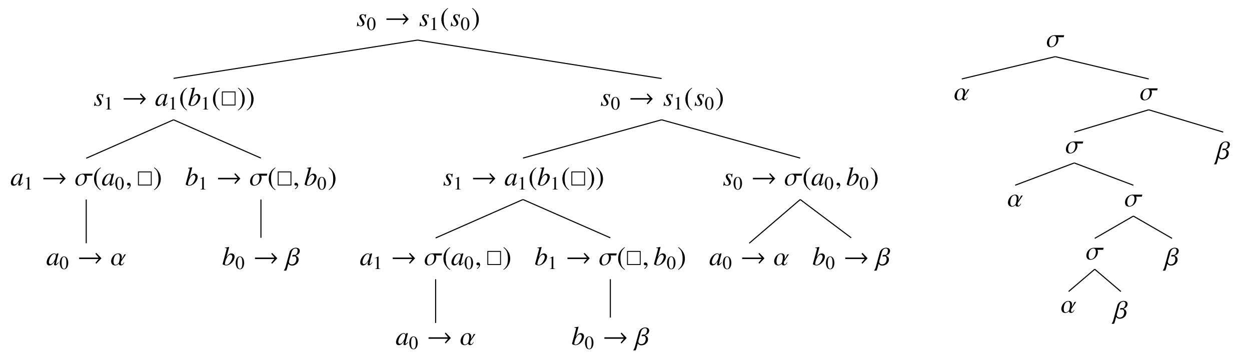

Figure 1: wsCFTG derivation tree and terminal tree resulting from evaluation (see Example 1)

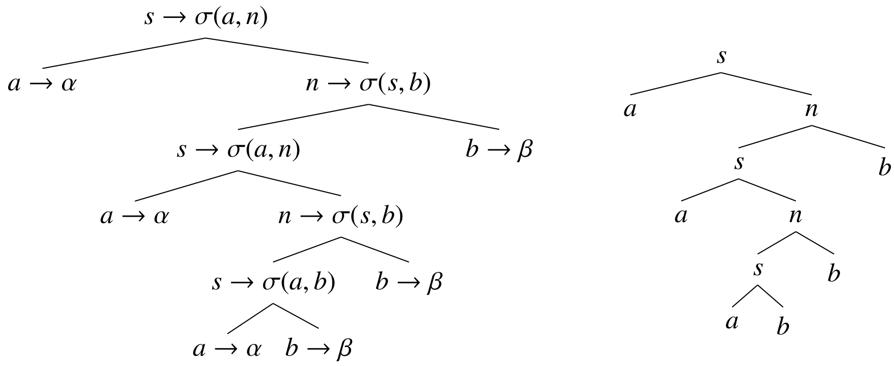

Figure 2: wTA derivation tree and the corresponding run (see Example 2)

3. Weighted Combinatory Categorial Grammar

We start with the definition of weighted combinatory categorial grammar, which is the main model under discussion in this contribution. It is a natural extension of combinatory categorial grammar [1, 2] to include weights in the rule system as well as the lexicon. Traditionally the rule system just permits or forbids certain category combinations via the presence or absence of certain rules, but in the extension each rule now carries a weight. Similarly, the association of a category to a lexical item is now weighted. The weights of all rule applications as well as all lexicon entries in a derivation tree are multiplied in our semiring H to obtain the weight of the derivation tree. If several derivation trees relabel to the same output tree (or yield the same output string), then the weights of those derivation trees are summed up to obtain the weight of the output tree (or output string). This follows the general principles of weighted automata [35]. We note that the rule system of our wCCG is restricted to application and composition rules, so it does not allow other combinators like substitution or type-raising rules [1].

First, we introduce some basic definitions and notations. Given an alphabet A and the set of slashes $S = \{/, \backslash\}$, the set of *categories* over A is given by $\mathcal{C}(A) = T_{S,\emptyset}(A)$, where the elements from $A \subseteq \mathcal{C}(A)$ are called *atomic*. We usually write categories in infix-notation and the slashes are left-associative by convention, so each category takes the form $c = a|_1 c_1 \cdots |_k c_k$ with $a \in A$, $|_i \in S$, and $c_i \in \mathcal{C}(A)$ for all $i \in [k]$. The atom a is called the *target* of c, and each slash-argument pair $|_i c_i$ is called an *argument* of c. We write target(c) to access the target a and argument(c, i) to access the i-th argument $|_i c_i$ of c. The number k of arguments is called *arity* of c and denoted by arity(c). We define argcats(c) $= \{c_i \mid i \in [k]\}$ as the set of all categories occurring as arguments in c. A category c with argcats(c) $\subseteq A$ is called a *first-order category*. A sequence of arguments $\alpha = \Box|_1 c_1 \cdots |_k c_k$ of length $|\alpha| = k$ can be seen as a context when viewed from the tree perspective, and we write $\mathcal{A}(A) \subseteq C_{S,\emptyset}(A)$ for the set of all such *argument contexts* over A of arbitrary length. When we restrict the categories to a limited arity $k \in \mathbb{N}$ (respectively, the argument contexts to a limited length k), we denote this set by $\mathcal{C}(A, k) = \{c \in \mathcal{C}(A) \mid \text{arity}(c) \leq k\}$ (resp.,

$\mathcal{A}(A, k) = \{ \alpha \in \mathcal{A}(A) \mid |\alpha| \leq k \})$.

We are now ready to describe how string-adjacent categories can be combined with each other. Intuitively, category a/c requires a category of the form $c\gamma$ with $\gamma \in \mathcal{A}(A)$ to its right and would then combine to the output category $a\gamma$. Similarly, $a\backslash c$ can be combined with $c\gamma$ to its left to obtain output category $a\gamma$ as well. The category a/c resp. $a\backslash c$, which takes an argument, is called *primary category*, whereas category $c\gamma$, which provides the argument, is called *secondary category*. Formally, given an alphabet A and $k \in \mathbb{N}$, a *rule of degree k over A* takes one of two possible forms [14]:

$$\text{forward rule: } \frac{ax\gamma}{ax/c \quad c\gamma} \qquad\qquad \text{backward rule: } \frac{ax\gamma}{c\gamma \quad ax\backslash c}$$

where $a \in A$, $c \in C(A)$, and $\gamma \in \mathcal{A}(A)$ with $|\gamma| = k$. The argument context variable x can match any argument context in $\mathcal{A}(A)$. We write $\mathcal{R}(A)$ for the set of all rules over A and $\mathcal{R}(A, k)$ for the set of all rules over A with degree at most $k \in \mathbb{N}$. Rules of degree 0 are called *application rules*, and rules of higher degree are called *composition rules*. A *ground instance* of a rule r is obtained by replacing the variable x by a concrete argument context. A *weighted rule system* is a tuple $\Pi = (A, R, \mathrm{wt})$ that consists of an alphabet A, a set $R \subseteq \mathcal{R}(A)$ of rules over A, and a weight function $\mathrm{wt}\colon R \to H$ such that $\mathrm{supp}(\mathrm{wt})$ is finite. Given a weighted rule system $\Pi = (A, R, \mathrm{wt})$, the set of all ground instances of R gives rise to a relation $\to_\Pi \subseteq C(A)^2 \times C(A)$, and we write $\frac{c''}{c \quad c'}\Pi$ instead of $(c, c') \to_\Pi c''$. The weight function on rules extends to a weight function $\widehat{\mathrm{wt}}\colon (\to_\Pi) \to H$ on ground instances by

$$\widehat{\mathrm{wt}}\Big(\frac{a\alpha\gamma}{a\alpha/c \quad c\gamma}\Big) = \mathrm{wt}\Big(\frac{ax\gamma}{ax/c \quad c\gamma}\Big) \qquad \text{and} \qquad \widehat{\mathrm{wt}}\Big(\frac{a\alpha\gamma}{c\gamma \quad a\alpha\backslash c}\Big) = \mathrm{wt}\Big(\frac{ax\gamma}{c\gamma \quad ax\backslash c}\Big)$$

for all $\alpha \in \mathcal{A}(A)$. We will not distinguish the two weight functions in the following. The relation $\to_\Pi$ extends to a relation $\Rightarrow_\Pi \subseteq C(A)^* \times C(A)^*$ on category sequences by

$$\Rightarrow_\Pi = \{ (\varphi\, c\, c'\, \psi,\ \varphi\, c''\, \psi) \mid \varphi, \psi \in C(A)^*;\ (c, c') \to_\Pi c'' \} \ .$$

Definition 3. *A* weighted combinatory categorial grammar *(wCCG) is a tuple $G = (\Sigma, A, R, I, L, \mathrm{wt})$ that consists of an alphabet Σ of* input symbols*, a weighted rule system (A, R, wt), a set $I \subseteq A$ of* initial (atomic) categories*, and a* weighted lexicon $L\colon \Sigma \times C(A) \to H$ *such that* $\mathrm{supp}(L)$ *is finite. For each $k \in \mathbb{N}$ it is called k-*wCCG *if each rule $r \in R$ has degree at most k; i.e., $R \subseteq \mathcal{R}(A, k)$.*

In the following, let $G = (\Sigma, A, R, I, L, \mathrm{wt})$ be a wCCG with its weighted rule system $\Pi = (A, R, \mathrm{wt})$.

Definition 4. *Let* $\mathrm{cat}\colon (C(A) \cup \mathrm{supp}(L)) \to C(A)$ *be such that* $\mathrm{cat}(c) = c$ *and* $\mathrm{cat}(\langle\sigma, c\rangle) = c$ *for every $c \in C(A)$ and $\sigma \in \Sigma$. A tree $d \in T_{C(A),\emptyset}(\mathrm{supp}(L))$ is called* derivation tree of *G if for every $w \in \mathrm{pos}(d) \setminus \mathrm{leaves}(d)$ there is a ground instance $\frac{\mathrm{cat}(d(w))}{\mathrm{cat}(d(w1)) \quad \mathrm{cat}(d(w2))}\Pi$. The set of all derivation trees is denoted by $\mathcal{D}_G$, and $\mathcal{D}_G^c = \{d \in \mathcal{D}_G \mid \mathrm{cat}(d(\varepsilon)) = c\}$ for every $c \in C(A)$. The weight of a derivation tree $d \in \mathcal{D}_G$ is*

$$\mathrm{wt}_G(d) = \Bigg(\prod_{w \in \mathrm{pos}(d) \setminus \mathrm{leaves}(d)} \mathrm{wt}\Big(\frac{\mathrm{cat}(d(w))}{\mathrm{cat}(d(w1)) \quad \mathrm{cat}(d(w2))}\Big)\Bigg) \cdot \Bigg(\prod_{w \in \mathrm{leaves}(d)} L(d(w))\Bigg) \ .$$

The wCCG G generates *the weighted string language $\mathcal{L}_G\colon \Sigma^* \to H$ given for every string $s \in \Sigma^*$ by*

$$\mathcal{L}_G(s) = \sum_{\substack{c \in I,\, d \in \mathcal{D}_G^c \\ \pi_1(\mathrm{yield}(d)) = s}} \mathrm{wt}_G(d) \ ,$$

where $\pi_1(\langle\sigma, c\rangle) = \sigma$ for every $\langle\sigma, c\rangle \in \mathrm{supp}(L)$, which uniquely extends to sequences of $\mathrm{supp}(L)^*$ *as a homomorphism.*

The labels of the derivation trees of a wCCG are (or contain, in the case of leaf labels) always specific categories. To allow less specific labels, we will allow a mapping, called relabeling, that relabels the specific categories to the desired output symbols. Already the classical definition of CCG [1, 2] contains such a (nondeterministic) relabeling, the lexicon, which in essence relabels the categories at the leaves to the symbols of the desired language. In the same spirit we now introduce a specific form of relabeling for the remaining symbols in the tree that is compatible with the lexicon and its application to the weighted derivation forest of a wCCG. More precisely, our specific form of relabeling is deterministic and requires that categories relabel to the same symbol if they coincide on the target category as well as the final argument.

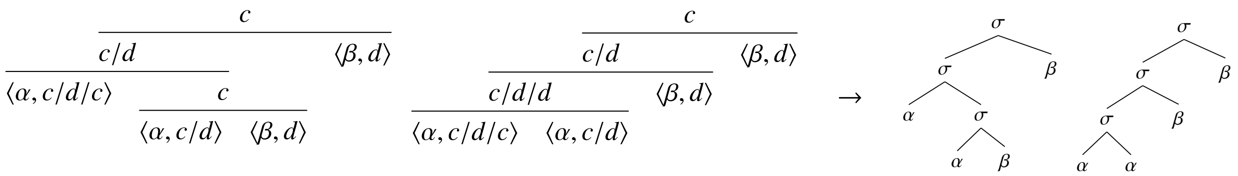

Figure 3: wCCG derivation trees and their respective relabeled trees (see Example 7)

Definition 5. *Let $G = (\Sigma, A, R, I, L, \mathrm{wt})$ be a wCCG. Given an alphabet $\Delta \supseteq \Sigma$, a mapping $\rho\colon (\mathcal{C}(A) \cup \mathrm{supp}(L)) \to \Delta$ is a* (deterministic) category Δ-relabeling *for G if*

(i) $\rho(\langle \sigma, c\rangle) = \sigma$ for every $\langle \sigma, c\rangle \in \mathrm{supp}(L)$ and

(ii) $\rho(c) = \rho(c')$ for all $c, c' \in \mathcal{C}(A)$ with $\mathrm{target}(c) = \mathrm{target}(c')$ and $\mathrm{argument}(c, \mathrm{arity}(c)) = \mathrm{argument}(c', \mathrm{arity}(c'))$.

Together the wCCG G and the category Δ-relabeling ρ generate the weighted forest $\mathrm{wt}^{\rho}_{G}\colon T_{\Delta,\emptyset}(\Delta) \to H$ given for every tree $t \in T_{\Delta,\emptyset}(\Delta)$ by

$$\mathrm{wt}^{\rho}_{G}(t) = \sum_{\substack{c \in I,\, d \in \mathcal{D}^{c}_{G} \\ \rho(d) = t}} \mathrm{wt}_G(d) \enspace .$$

A weighted forest $\varphi\colon T_{\Delta,\emptyset}(\Delta) \to H$ is generatable *by G if there exists a category Δ-relabeling ρ' such that $\varphi = \mathrm{wt}^{\rho'}_{G}$. Finally, φ is generatable by a class $\mathcal{C}$ of wCCG if φ is generatable by some $G \in \mathcal{C}$.*

Lemma 6. *The weighted forests generatable by k-wCCG are closed under deterministic relabelings for all $k \in \mathbb{N}$.*

Proof. Let $\varphi\colon T_{\Delta,\emptyset}(\Delta) \to H$ be generatable by the k-wCCG $G = (\Sigma, A, R, I, L, \mathrm{wt})$ together with the category Δ-relabeling ρ for G; i.e., $\varphi = \mathrm{wt}^{\rho}_{G}$. Moreover, let $\rho'\colon \Delta \to \Gamma$ be a deterministic relabeling. Unfortunately, $\rho \,;\, \rho'$ is not necessarily a category Γ-relabeling for G because we might have $\rho'(\rho(\langle \sigma, c\rangle)) = \rho'(\sigma) \neq \sigma$ for some $\langle \sigma, c\rangle \in \mathrm{supp}(L)$. However, we can move the leaf symbol relabeling into the k-wCCG and construct a k-wCCG $G' = (\Gamma, A, R, I, L', \mathrm{wt})$ with

$$L'(\langle \gamma, c\rangle) = \sum_{\substack{\sigma \in \Sigma \\ \rho'(\sigma) = \gamma}} L(\langle \sigma, c\rangle)$$

for every $\gamma \in \Gamma$ and $c \in \mathcal{C}(A)$. Note that $\mathrm{supp}(L')$ is trivially finite. The relabeling $\overline{\rho}\colon (\mathcal{C}(A) \cup \mathrm{supp}(L')) \to \Gamma$ given for every $c \in \mathcal{C}(A)$ and $\langle \gamma, c\rangle \in \mathrm{supp}(L')$ by $\overline{\rho}(c) = \rho'(\rho(c))$ and $\overline{\rho}(\langle \gamma, c\rangle) = \gamma$ is a category Γ-relabeling for G', and it is a routine matter to verify that $\rho'(\varphi) = \mathrm{wt}^{\overline{\rho}}_{G'}$, which proves that $\rho'(\varphi)$ is generatable by k-wCCG. □

Example 7. *Let $G_3 = (\Sigma, A, \mathcal{R}(A, 2), I, L, \mathrm{wt})$ be the 2-wCCG that is weighted over the semiring of nonnegative integers $(\mathbb{N}, +, \cdot, 0, 1)$ with $\Sigma = \{\alpha, \beta\}$, $A = \{c, d\}$, $I = \{c\}$. Its weighted rule system $(A, \mathcal{R}(A, 2), \mathrm{wt})$ is given by*

$$\mathrm{wt}\Big(\frac{cx}{cx/d \quad d}\Big) = \mathrm{wt}\Big(\frac{cx}{cx/c \quad c}\Big) = 1 \qquad \textit{and} \qquad \mathrm{wt}\Big(\frac{cx/d}{cx/c \quad c/d}\Big) = \mathrm{wt}\Big(\frac{cx/d/c}{cx/c \quad c/d/c}\Big) = 2 \enspace .$$

The weight of all other rules is 0. The weighted lexicon $L\colon \Sigma \times \mathcal{C}(A) \to \mathbb{N}$ is given by

$$L(\langle \alpha, c/d/c\rangle) = 1 \qquad L(\langle \alpha, c/d\rangle) = 3 \qquad L(\langle \beta, d\rangle) = 1 \enspace .$$

Finally, we define the category relabeling $\rho(\mathcal{C}(A) \cup \mathrm{supp}(L)) \to \Delta$ with $\Delta = \{\sigma, \alpha, \beta\}$ as

$$\rho(c) = \rho(c\alpha/d) = \sigma \qquad \rho(\langle \alpha, c/d/c\rangle) = \rho(\langle \alpha, c/d\rangle) = \alpha \qquad \rho(\langle \beta, d\rangle) = \beta$$

for all $\alpha \in \mathcal{A}(A)$, and the relabeling is irrelevant for all other labels since they cannot occur in derivation trees of G_3. Figure 3 depicts two derivation trees d_1 and d_2 that are generated by G_3 together with their respective relabeled trees t_1 and t_2. Their weights are $\mathrm{wt}_{G_3}(d_1) = 3$ and $\mathrm{wt}_{G_3}(d_2) = 6$, respectively. These are the only derivation trees with yield $\alpha^2\beta^2$, so we get $\mathcal{L}_{G_3}(\alpha^2\beta^2) = 9$.

4. 0-wCCGs

0-wCCG can only utilize application rules. It is known from the unweighted setting [29, 27] that the forests generatable by 0-CCGs are universally minheight-bounded. The forests generatable by 0-CCGs are obtained as relabeled derivation forests, so this structural property also applies to the derivation trees. In the weighted setting, exactly the same derivation trees are additionally equipped with weights and again yield the generatable weighted forests via a (specific) relabeling. Thus the derivation trees are again universally minheight-bounded and the relabeling preserves this structural property. Hence it stands to reason that universal minheight again plays a major role in our weighted setting, so let us recall the relevant notions from [29, 27]. The *minheight* $\mathrm{mht}(t)$ of a tree t is the minimal length of a path from the root to a leaf. Formally, for all alphabets Σ_2 and Σ_0, let $\mathrm{mht}\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to \mathbb{N}$ be such that $\mathrm{mht}(\alpha) = 0$ and $\mathrm{mht}(\sigma(t_1, t_2)) = 1 + \min(\mathrm{mht}(t_1), \mathrm{mht}(t_2))$ for all $\alpha \in \Sigma_0$, $\sigma \in \Sigma_2$, and $t_1, t_2 \in T_{\Sigma_2,\emptyset}(\Sigma_0)$. Both trees displayed in Figure 4 have minheight 2 (along path 2.1). A tree $t \in T_{\Sigma_2,\emptyset}(\Sigma_0)$ is *universally minheight-bounded by* $\ell \in \mathbb{N}$ if $\mathrm{mht}(t|_w) \leq \ell$ for every $w \in \mathrm{pos}(t)$. Indeed both trees of Figure 4 are also universally minheight-bounded by 2. Additionally, the right direct subtree of the root (i.e., the subtree rooted in VP or $\langle \mathrm{VP}, 1\rangle$) is in both cases universally minheight-bounded by 1. A forest $\mathcal{F} \subseteq T_{\Sigma_2,\emptyset}(\Sigma_0)$ is *universally minheight-bounded by* ℓ if every $t \in \mathcal{F}$ is universally minheight-bounded by ℓ, and it is *universally minheight-bounded* if there exists $\ell \in \mathbb{N}$ such that it is universally minheight-bounded by ℓ. Finally, we extend the notion to a weighted forest $\varphi\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$ by calling it *universally minheight-bounded (by ℓ)* if $\mathrm{supp}(\varphi)$ is universally minheight-bounded (by ℓ).

Let φ be a regular weighted forest that is universally minheight-bounded by some $\ell \in \mathbb{N}$. Hence there exists a wTA $G = (N, \Sigma, S, P, \mathrm{wt})$ such that $\tau_G = \varphi$. In the unweighted setting it is immediately clear that $R_G(t) = \emptyset$ for all trees t that are not universally minheight-bounded by ℓ. To simplify this discussion, we call a tree *improper* for the moment if it is not universally minheight-bounded by ℓ. In other words, in the unweighted case there are no runs for improper trees. However, this property need not be true in the weighted setting since different runs might cancel each other out. For example, using weights from the commutative semiring $(\mathbb{Z}, +, \cdot, 0, 1)$ of integers, an improper tree t might have two runs ω_1 and ω_2 with $\mathrm{wt}_G(\omega_1) = 2$ and $\mathrm{wt}_G(\omega_2) = -2$. Overall, we might thus have $\tau_G(t) = \mathrm{wt}_G(\omega_1) + \mathrm{wt}_G(\omega_2) = 0$, which yields $t \notin \mathrm{supp}(\varphi)$. Hence φ might be universally minheight-bounded by ℓ, whereas $R_G(t) \neq \emptyset$ for an improper tree t. Fortunately, we can obtain a normal form that restores the desired property. Additionally, we will annotate to each symbol the direction along which the minheight is achieved with the first successor taking priority in case of a tie. We will use this annotation later to decompose trees into sets of short paths that follow the annotated direction and are simulated in a 0-wCCG along sequences of primary categories. Formally, we define the mapping $\mathrm{dir}\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to T_{\Sigma_2\times[2],\emptyset}(\Sigma_0)$ by $\mathrm{dir}(\alpha) = \alpha$ for every $\alpha \in \Sigma_0$ and

$$\mathrm{dir}(\sigma(t_1, t_2)) = \begin{cases} \langle \sigma, 1\rangle(\mathrm{dir}(t_1), \mathrm{dir}(t_2)) & \text{if } \mathrm{mht}(t_1) \leq \mathrm{mht}(t_2) \\ \langle \sigma, 2\rangle(\mathrm{dir}(t_1), \mathrm{dir}(t_2)) & \text{otherwise} \end{cases}$$

for every $\sigma \in \Sigma_2$ and trees $t_1, t_2 \in T_{\Sigma_2,\emptyset}(\Sigma_0)$. Figure 4 displays a tree t on the left and the corresponding tree $\mathrm{dir}(t)$ on the right. Note that 'dir' is injective on trees. Next, we lift this mapping to weighted forests $\varphi\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$ by $\mathrm{dir}_\varphi(u) = \sum_{t \in \mathrm{dir}^{-1}(u)} \varphi(t)$ for all $u \in T_{\Sigma_2\times[2],\emptyset}(\Sigma_0)$. Note that the presented sum always has at most one summand due to the injectivity of 'dir' and that $\mathrm{dir}_\varphi\colon T_{\Sigma_2\times[2],\emptyset}(\Sigma_0) \to H$ is a weighted forest. Finally, we note that φ and dir_φ are equivalent modulo the obvious relabeling that removes the direction indicators.

Lemma 8. *Let $\varphi\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$ be a regular weighted forest that is universally minheight-bounded by $\ell \in \mathbb{N}$. Then there exists a wTA G' such that $\tau_{G'} = \mathrm{dir}_\varphi$ and $R_{G'}(u) = \emptyset$ for all trees $u \in T_{\Sigma_2\times[2],\emptyset}(\Sigma_0)$ that are not universally minheight-bounded by ℓ.*

Proof. Since φ is regular, there exists a wTA $G = (N, \Sigma, S, P, \mathrm{wt})$ such that $\tau_G = \varphi$. We construct the wTA

$$G' = (N \times [0, \ell],\ \Sigma',\ S \times [0, \ell],\ P',\ \mathrm{wt}')$$

for dir_φ as follows:

- $\Sigma'_2 = \Sigma_2 \times [2]$ and $\Sigma'_0 = \Sigma_0$,

- the set P' of productions is given by

$$P' = \Big\{ \langle n, i+1\rangle \to \langle \sigma, 1\rangle(\langle n_1, i\rangle, \langle n_2, j\rangle) \;\Big|\; n \to \sigma(n_1, n_2) \in P,\ i, j \in [0,\ell],\ i = \min(i,j) < \ell \Big\} \cup$$
$$\Big\{ \langle n, j+1\rangle \to \langle \sigma, 2\rangle(\langle n_1, i\rangle, \langle n_2, j\rangle) \;\Big|\; n \to \sigma(n_1, n_2) \in P,\ i, j \in [0,\ell],\ j < i \Big\} \cup$$
$$\Big\{ \langle n, 0\rangle \to \alpha \;\Big|\; n \to \alpha \in P \Big\}$$

- $\mathrm{wt}'\big(\langle n,k\rangle \to \langle \sigma, d\rangle(\langle n_1, i\rangle, \langle n_2, j\rangle)\big) = \mathrm{wt}(n \to \sigma(n_1,n_2))$ for all $\langle n,k\rangle \to \langle \sigma, d\rangle(\langle n_1, i\rangle, \langle n_2, j\rangle) \in P'$ and $\mathrm{wt}'\big(\langle n, 0\rangle \to \alpha\big) = \mathrm{wt}(n \to \alpha)$ for all $\langle n, 0\rangle \to \alpha \in P'$.

For correctness, let us consider an arbitrary tree $t \in T_{\Sigma_2,\emptyset}(\Sigma_0)$. Given a run $\omega \in R_G(t)$ let $\vartheta_\omega \colon \mathrm{pos}(t) \to N \times \mathbb{N}$ be given by $\vartheta_\omega(w) = \langle \omega(w), \mathrm{mht}(t|_w)\rangle$ for all $w \in \mathrm{pos}(t)$. If t is universally minheight-bounded by ℓ, then we can easily verify by induction that ϑ bijectively maps runs of G on t onto runs of G' on $\mathrm{dir}(t)$; i.e., $\vartheta \colon R_G(t) \to R_{G'}(\mathrm{dir}(t))$. Alongside we can also observe that $\mathrm{wt}(\omega) = \mathrm{wt}'(\vartheta(\omega))$ for all $\omega \in R_G(t)$. Thus, we identified our weight-preserving bijection. To complete the proof we can also show using the same arguments that $R_{G'}(u) = \emptyset$ if u is not universally minheight-bounded by ℓ. Hence $\tau_{G'} = \mathrm{dir}_\varphi$. □

Now we are ready to characterize the expressive power of 0-wCCG. We start by showing that the weighted derivation forest of each 1-wCCG is regular, which applies to 0-wCCG as well because they are special 1-wCCG. Let $G = (\Sigma, A, R, I, L, \mathrm{wt})$ be a 1-wCCG. As in the unweighted case [28, Lemma 10], the main property of 1-wCCGs is that each category that occurs in a derivation tree has arity at most $\mathrm{arity}(L) = \max\{\mathrm{arity}(c) \mid \langle \sigma, c\rangle \in \mathrm{supp}(L)\}$. Thus, the derivation trees of $\mathcal{D}_G$ are built over a finite set of symbols, and the ground instances of the weighted rule system essentially specify the weighted branchings.

Lemma 9. *For every* 1*-wCCG* $G = (\Sigma, A, R, I, L, \mathrm{wt})$ *the weighted derivation forest* $\mathrm{wt}_G \colon \mathcal{D}_G \to H$ *is regular.*

Proof. We show that the weighted derivation forest wt_G is actually local, which also proves that it is regular by [41, Theorem 1]. The weighted tree automata of [41] have root weights, but it is well-known [42, Theorem 6.2.2] that weighted tree automata with root weights are as expressive as our wTA, and the same remarks apply to the weighted local systems of [41] and our weighted local tree grammars. Let $k = \mathrm{arity}(L)$. To prove that wt_G is local, we construct the weighted local tree grammar $G' = (\Delta, I', P, \mathrm{wt}')$ as follows.

- The terminals $\Delta = \Delta_2 \cup \Delta_0$ are given by $\Delta_2 = C(A, k)$, which are all the categories of $C(A)$ that have arity at most k, and the leaf labels are $\Delta_0 = \mathrm{supp}(L)$.
- The root labels $I' = I \cup \{\langle \alpha, c\rangle \in \mathrm{supp}(L) \mid c \in I\}$ are the initial categories and the leaf labels containing them.
- The permitted branchings $P = \{\frac{c}{c_1 \quad c_2} \mid c \in \Delta_2,\ c_1, c_2 \in \Delta,\ \frac{\mathrm{cat}(c)}{\mathrm{cat}(c_1) \quad \mathrm{cat}(c_2)} \Pi\} \cup \{\frac{c}{\quad} \mid c \in \Delta_0\}$ are essentially the valid ground instances of the weighted rule system $\Pi = (A, R, \mathrm{wt})$.
- The weight $\mathrm{wt}'(\frac{c}{c_1 \quad c_2}) = \mathrm{wt}(\frac{\mathrm{cat}(c)}{\mathrm{cat}(c_1) \quad \mathrm{cat}(c_2)})$ of branching $\frac{c}{c_1 \quad c_2} \in P$ is inherited from $\Pi = (A, R, \mathrm{wt})$, and $\mathrm{wt}'(\frac{c}{\quad}) = L(c)$ for every $c \in \Delta_0$ is taken from the lexicon.

Obviously, $\mathcal{D}_G = R_{G'}$; i.e., each derivation tree of G is a run of G' and vice versa. Moreover, the weights assigned to a given derivation tree $d \in \mathcal{D}_G$ by G and G' coincide, so the identity is a suitable weight-preserving bijection that proves $\mathrm{wt}_{G'} = \mathrm{wt}_G$. Thus wt_G is local and regular. □

We already know by [28, Lemma 10] that the derivation trees $\mathcal{D}_G$ of a 0-wCCG G are universally minheight-bounded by some grammar-specific constant $\ell \in \mathbb{N}$. Hence also the weighted derivation forest wt_G is universally minheight-bounded. Together with Lemma 9 we obtain that the weighted forests generatable by 0-wCCGs are regular and universally minheight-bounded. In the rest of this section we prove that every universally minheight-bounded regular weighted forest is generatable by a 0-wCCG. The construction utilizes the annotation provided by 'dir' and decomposes the tree into paths that start at a given node v and then repeatedly proceed to the child node indicated by the annotation. These paths, called *spines*, lead from v to a leaf and are never longer than the universal minheight. The (nontrivial) spines of the right tree in Figure 4 are indicated by a special background. In the constructed 0-wCCG the primary categories for the applications are placed along those spines and each spine terminates in an atomic category that can be combined with the category from another spine. This is achieved by simulating each spine by a lexical category, where the arguments store the information needed for relabling and get successively removed (see Figure 5).

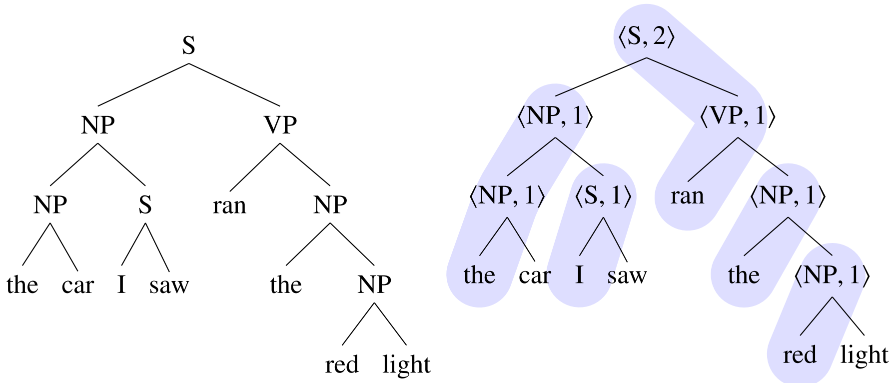

Figure 4: Example tree t and its direction-annotated variant $\mathrm{dir}(t)$, in which we also indicated nontrivial spines.

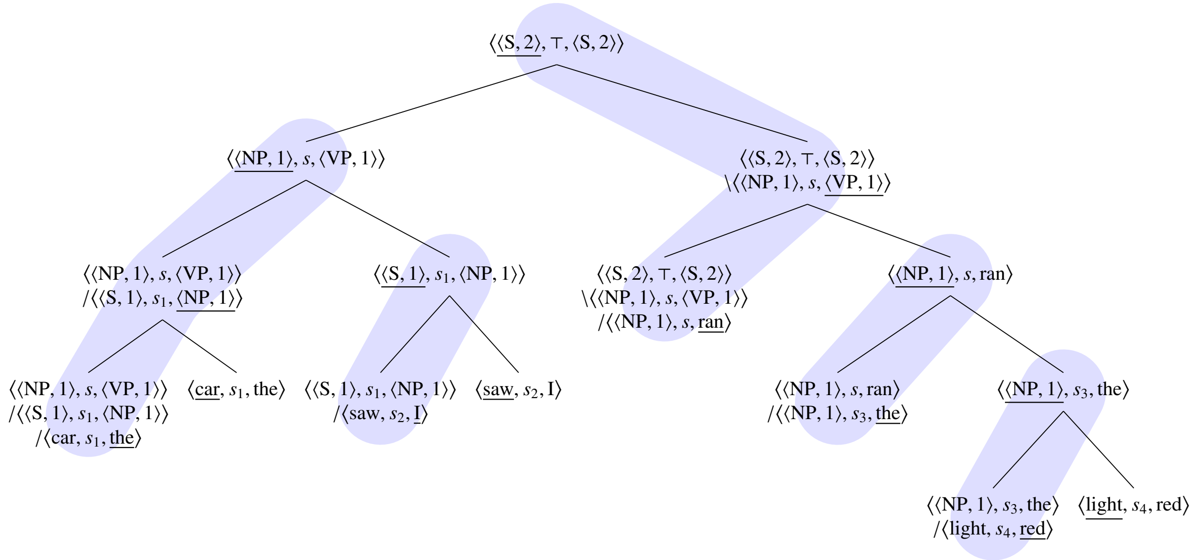

Figure 5: Derivation tree of the 0-wCCG corresponding to tree t of Figure 4 with nontrivial spines indicated and with the symbol resulting after relabeling underlined. Input symbols of leaf labels are omitted.

In the unweighted case, each tree could be decomposed into a set of spines in multiple ways. To avoid this ambiguity, the annotation 'dir' enforces for each tree a unique decomposition into spines. Additionally, the set of initial categories is more restricted than in the unweighted case.

Theorem 10. *Let $\varphi\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$ be a weighted forest. Then φ is generatable by some 0-wCCG if and only if it is regular and universally minheight-bounded.*

Proof. The direction from left to right is clear by [28, Lemma 10] and Lemma 9 as already discussed. For the only-if direction, let $\ell \in \mathbb{N}$ be such that φ is universally minheight-bounded by ℓ. Since φ is regular and universally minheight-bounded by ℓ, we can use Lemma 8 to conclude that there exists a wTA $\widetilde{G}$ such that $\tau_{\widetilde{G}} = \mathrm{dir}_\varphi$ and $R_{\widetilde{G}}(u) = \emptyset$ for all trees $u \in T_{\Sigma_2\times[2],\emptyset}(\Sigma_0)$ that are not universally minheight-bounded by ℓ. Since the weighted forests generatable by 0-wCCG are closed under deterministic relabelings by Lemma 6, it suffices to prove that dir_φ is generatable by 0-wCCG. Utilizing this approach once more, we know that according to [41, Theorem 1] there also exists a weighted local tree grammar $G = (\Delta, I, P, \mathrm{wt})$ and a deterministic relabeling $\rho\colon \Delta \to (\Sigma_2 \times [2]) \cup \Sigma_0$ such that $\rho(\mathrm{wt}_G) = \mathrm{dir}_\varphi$ and every $d \in R_G$ is universally minheight-bounded by ℓ. The latter property can easily be seen from the constructions used in [41, Theorem 1] and the fact that $R_{\widetilde{G}}(u) = \emptyset$ for all trees $u \in T_{\Sigma_2\times[2],\emptyset}(\Sigma_0)$ that are not universally minheight-bounded

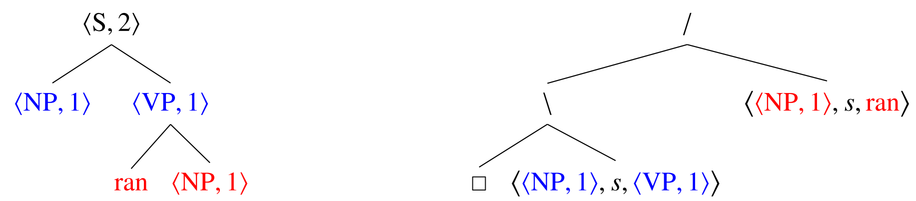

Figure 6: Spinal run s together with its argument tree $\arg(s, s)$.

by ℓ. Overall, it is thus sufficient to prove that there exists a 0-wCCG $G' = (\Delta_0, A, R, I', L, \mathrm{wt}')$ and a category Δ-relabeling ρ' such that $\mathrm{wt}_{G'}^{\rho'} = \mathrm{wt}_G$. As already remarked, R_G is universally minheight-bounded by ℓ.

Our goal is to establish a weight-preserving bijection between the runs of the weighted local tree grammar G and the derivation trees of the 0-wCCG G' to be constructed. To this end, we follow the basic strategy that was already used in the unweighted case [29, 28]. First we note that all categories $\{c \in C(A) \mid \langle \sigma, c\rangle \in \mathrm{supp}(L)\}$ that we include in the lexicon L are left-spinal (i.e., all right children are leaves — see argument context displayed right in Figure 6). Together with the fact that we can only use application rules, we obtain that all categories that can occur in derivations of $\mathcal{D}_{G'}$ must be subtrees of the categories in $\{c \in C(A) \mid \langle \sigma, c\rangle \in \mathrm{supp}(L)\}$. Consequently, let $C_0(A) \subseteq C(A)$ be that subset of all categories.

For every $\gamma \in \Delta_2$, let $\pi(\gamma)$ be the annotated direction; i.e., $\pi \colon \Delta_2 \to \{1, 2\}$ such that $\pi(\gamma) = z$, where $\rho(\gamma) = \langle \sigma, z\rangle$, for all $\gamma \in \Delta_2$. We denote the opposite direction by overlining, so $\overline{2} = 1$ and $\overline{1} = 2$. For every terminal $\alpha \in \Delta_0$, we define the set $\mathrm{Spines}_G(\alpha)$ of *spines of G terminating in α* to be the smallest set $M \subseteq T_{\Delta_2,\emptyset}(\Delta)$ such that (i) $\alpha \in M$, and (ii) $\gamma(s_1, s_2) \in M$ for all $\gamma \in \Delta_2$, $s_{\overline{\pi(\gamma)}} \in \Delta$, and $s_{\pi(\gamma)} \in M$ such that $\frac{\gamma}{s_1(\varepsilon) \quad s_2(\varepsilon)} \in P$. In other words, the spines are obtained from the runs R_G by selecting a run $d \in R_G$ and a position $p \in \mathrm{pos}(d)$ in it, taking the subtree $d|_p$, and replacing the off-direction direct subtree d' of each node in $d|_p$ by just a leaf with the same root label $d'(\varepsilon)$. This process is illustrated in Figure 6 (left), and we note that each run $d \in R_G$ decomposes uniquely into spines as illustrated in Figure 4 (right). Since the runs of G are universally mht-bounded by ℓ along the spine direction π, we obtain that $\mathrm{ht}(s) \leq \ell$ for all $s \in \mathrm{Spines}_G(\alpha)$, which shows that $\mathrm{Spines}_G(\alpha)$ is finite.

Let $A = \Delta \times \mathrm{Spines}_G \times \Delta$ be the set of atomic categories, where $\mathrm{Spines}_G = \bigcup_{\alpha \in \Delta_0} \mathrm{Spines}_G(\alpha)$. Each atomic category $\langle \gamma', s, \gamma\rangle \in A$ thus stores two symbols γ' and γ, which are required for the relabeling ρ', and a spine s to enforce consistency. Next, we define the category Δ-relabeling $\rho' \colon (C_0(A) \cup \mathrm{supp}(L)) \to \Delta$ by $\rho'(\langle \gamma', s, \gamma\rangle) = \gamma'$ and $\rho'(c|\langle \gamma', s, \gamma\rangle) = \gamma$ for all $\langle \gamma', s, \gamma\rangle \in A$, $c \in C_0(A)$, and $| \in \{/, \backslash\}$ as well as $\rho'(\langle \sigma, c\rangle) = \sigma$ for every $\langle \sigma, c\rangle \in \mathrm{supp}(L)$. Each lexical category is based on a spine, such that each argument records the label of a spinal node, the label of its sibling, and the spine it belongs to. The outermost argument corresponds to the bottom of the spine because it gets removed first in the wCCG derivation. Therefore, for all spines $s, s' \in \mathrm{Spines}_G(\alpha)$ we inductively construct the argument context $\arg(s, s') \in \mathcal{A}(A, \ell)$, also illustrated in Figure 6, as follows:

- $\arg(\alpha, s') = \square$, and
- $\arg(\gamma(s_1, s_2), s') = \arg(s_{\pi(\gamma)}, s')[\square|\langle s_{\overline{\pi(\gamma)}}, s', s_{\pi(\gamma)}(\varepsilon)\rangle]$ for all $\gamma \in \Delta_2$, $s_{\overline{\pi(\gamma)}} \in \Delta$, and $s_{\pi(\gamma)} \in \mathrm{Spines}_G(\alpha)$, where $| = /$ if $\pi(\gamma) = 1$ and $| = \backslash$ otherwise.[2]

For an argument context based on some $s' \in \mathrm{Spines}_G$ we can use as a target all atoms that relabel to $s'(\varepsilon)$. Thus, let

$$C_\alpha = \Big\{\arg(s', s')\big[\langle s'(\varepsilon), s, \gamma\rangle\big] \;\Big|\; s' \in \mathrm{Spines}_G(\alpha),\ s \in \mathrm{Spines}_G,\ \gamma \in \Delta\Big\}$$

for all $\alpha \in \Delta_0$. Note that $C_\alpha \cap C_\beta = \emptyset$ for all $\alpha, \beta \in \Delta_0$ with $\alpha \neq \beta$ and $\rho'(c) = \alpha$ for every $c \in C_\alpha$, so $\rho'(y) = \rho'(\mathrm{cat}(y))$ for every $y \in C_0(A) \cup \mathrm{supp}(L)$ provided that $\mathrm{supp}(L) \subseteq \bigcup_{\alpha \in \Delta_0} \{\alpha\} \times C_\alpha$. Finally, we fix any spine $\top \in \mathrm{Spines}_G$ and construct the 0-wCCG $G' = (\Delta_0, A, \mathcal{R}(A, 0), I', L, \mathrm{wt}')$ with initial atoms I' and lexicon L given by

$$I' = \{\langle a, \top, a\rangle \mid a \in I\}$$

$$L(\langle \alpha, c\rangle) = \begin{cases} \mathrm{wt}\big(\frac{\alpha}{}\big) & \text{if } c \in C_\alpha \text{ and } \frac{\alpha}{} \in P \\ 0 & \text{otherwise} \end{cases}$$

[2]For better readability, we write $|(\square, c)$ using the infix notation $\square \mid c$.

for all $\alpha \in \Delta_0$ and $c \in C(A)$. Clearly, $\operatorname{supp}(L) \subseteq \bigcup_{\alpha\in\Delta_0}\{\alpha\} \times C_\alpha$. Finally, for all atoms $a \in A$ and categories $c \in C(A)$

$$\mathrm{wt}'\Big(\frac{ax}{ax/c \quad c}\Big) = \begin{cases} \mathrm{wt}\Big(\frac{\rho'(ax)}{\rho'(ax/c) \quad \rho'(c)}\Big) & \text{if } c \in A \text{ and } \frac{\rho'(ax)}{\rho'(ax/c) \quad \rho'(c)} \in P \\ 1 & \text{otherwise} \end{cases}$$
$$\mathrm{wt}'\Big(\frac{ax}{c \quad ax\backslash c}\Big) = \begin{cases} \mathrm{wt}\Big(\frac{\rho'(ax)}{\rho'(c) \quad \rho'(ax\backslash c)}\Big) & \text{if } c \in A \text{ and } \frac{\rho'(ax)}{\rho'(c) \quad \rho'(ax\backslash c)} \in P \\ 1 & \text{otherwise.} \end{cases}$$

The weight of application rules without corresponding productions in P is in fact irrelevant since by the definition of the lexicon L such rules can never occur in derivations of G'.

It remains to prove that $\mathrm{wt}_{G'}^{\rho'} = \mathrm{wt}_G$.

For every $s \in \mathrm{Spines}_G$ and $\gamma \in \Delta$, let $\mathcal{D}_{G'}^{s,\gamma} = \{d' \in \mathcal{D}_{G'} \mid \exists\gamma' \in \Delta\colon \operatorname{target}(d'(\varepsilon)) = \langle\gamma', s, \gamma\rangle\}$ be those derivation trees of G' whose root is labeled by a category that has a target with s and γ in the second and third component, respectively. Additionally, let $\mathfrak{D}_{G'}^{s,\gamma} = \{d' \in \mathcal{D}_{G'}^{s,\gamma} \mid d'(\varepsilon) \in A\}$ be those derivation trees of $\mathcal{D}_{G'}^{s,\gamma}$ that are labeled by an atom at the root. We first prove the auxiliary statement that for every $s \in \mathrm{Spines}_G$ and $\gamma \in \Delta$ the mapping $\rho'\colon \mathfrak{D}_{G'}^{s,\gamma} \to R_G$ is a weight-preserving bijection; i.e., $\mathrm{wt}_{G'}(d') = \mathrm{wt}_G(\rho'(d'))$ for every $d' \in \mathfrak{D}_{G'}^{s,\gamma}$. To this end, we need to prove four subgoals: (i) $\rho'(\mathcal{D}_{G'}^{s,\gamma}) \subseteq R_G$, (ii) $\rho'\colon \mathfrak{D}_{G'}^{s,\gamma} \to R_G$ is injective, (iii) $R_G \subseteq \rho'(\mathfrak{D}_{G'}^{s,\gamma})$, which shows that $\rho'\colon \mathfrak{D}_{G'}^{s,\gamma} \to R_G$ is surjective, and finally subgoal (iv) $\mathrm{wt}_{G'}(d') = \mathrm{wt}_G(\rho'(d'))$ for every $d' \in \mathfrak{D}_{G'}^{s,\gamma}$. Given the definition of wt', subgoal (iv) is automatically fulfilled once we establish subgoal (i).

We start with subgoal (i) $\rho'(\mathcal{D}_{G'}^{s,\gamma}) \subseteq R_G$ for every $s \in \mathrm{Spines}_G$ and $\gamma \in \Delta$ using induction on the height of $d' \in \mathcal{D}_{G'}^{s,\gamma}$. In the induction base d' is a lexicon entry, so $d' = \langle\alpha, c\rangle \in \operatorname{supp}(L)$ with $c \in C_\alpha$ and $\frac{\alpha}{\quad} \in P$. Clearly, $\rho'(d') = \alpha$. Since $\frac{\alpha}{\quad} \in P$, tree α is a valid run of R_G, which completes the induction base.

In the induction step, let $d' = c(d'_1, d'_2) \in \mathcal{D}_{G'}^{s,\gamma}$ for some $c \in C_0(A)$, and subtrees $d'_1 \in \mathcal{D}_{G'}^{s_1,\gamma_1}$, $d'_2 \in \mathcal{D}_{G'}^{s_2,\gamma_2}$ with $\gamma_1, \gamma_2 \in \Delta$, and $s_1, s_2 \in \mathrm{Spines}_G$. By the induction hypothesis $d_1 = \rho'(d'_1) \in R_G$ as well as $d_2 = \rho'(d'_2) \in R_G$. It remains to prove that $\frac{\rho'(c)}{d_1(\varepsilon) \quad d_2(\varepsilon)} \in P$, which would prove that $\rho'(d') \in R_G$. Since we only utilize first-order categories and application rules we have $\{\operatorname{cat}(d'_1(\varepsilon)), \operatorname{cat}(d'_2(\varepsilon))\} = \{c|a,\, a\}$ for some $| \in \{/, \backslash\}$ and $a \in A$. Moreover, let $a = \langle\gamma'', s', \gamma'\rangle$ for some $\gamma'', \gamma' \in \Delta$ and $s' \in \mathrm{Spines}_G$. We assume that $\operatorname{cat}(d'_1(\varepsilon)) = a$ and $\operatorname{cat}(d'_2(\varepsilon)) = c\backslash a$. The remaining case, in which $\operatorname{cat}(d'_1(\varepsilon)) = c/a$ and $\operatorname{cat}(d'_2(\varepsilon)) = a$, is analogous. By the definition of ρ', we obtain that $d_1(\varepsilon) = \gamma''$ and $d_2(\varepsilon) = \gamma'$. Moreover, $c\backslash a$ must be a subtree of the category $\arg(s', s')[\langle s'(\varepsilon), s_2, \gamma_2\rangle]$ because the spine s' is annotated to a right child.

Now we distinguish two cases. If c is atomic, then $c = \langle s'(\varepsilon), s_2, \gamma_2\rangle$ and $\rho'(c) = s'(\varepsilon)$ by the definition of ρ'. By the construction of the argument context '$\arg(s', s')$' we have

$$s'(\varepsilon) = \rho'(c) \qquad \text{and} \qquad s'(1) = \gamma'' = d_1(\varepsilon) \qquad \text{and} \qquad s'(2) = \gamma' = d_2(\varepsilon) \ .$$

Since $s' \in \mathrm{Spines}_G$ we have $\frac{s'(\varepsilon)}{s'(1) \quad s'(2)} \in P$, which yields $\frac{\rho'(c)}{d_1(\varepsilon) \quad d_2(\varepsilon)} \in P$ as desired with the help of the equations above. In the remaining case c is not atomic. Let $c = c' \,|\, \langle\overline{\gamma}', s', \overline{\gamma}\rangle$ for some $c' \in C_0(A)$, $| \in \{/, \backslash\}$, and $\overline{\gamma}', \overline{\gamma} \in \Delta$. The definition of ρ' yields that $\rho'(c) = \overline{\gamma}$. Since 'arg' reverses the order (see Figure 6), our subtree $c\backslash a$ corresponds to an initial fragment of s'. Let $s' = C[\overline{s}']$ with $C \in C_{\Delta_2,\emptyset}(\Delta)$ and $\overline{s}' \in \mathrm{Spines}_G$ such that $c\backslash a = \arg(C[\overline{s}'(\varepsilon)], s')[\langle s'(\varepsilon), s_2, \gamma_2\rangle]$. Moreover, let $w = \mathrm{pos}_\square(C)$. Since we have at least two arguments in $c\backslash a$, the definition of 'arg' yields $|w| \geq 2$, so let $w = w'z2$ with $w' \in [2]^*$ and $z \in [2]$. Then $\square \,|\, \langle\overline{\gamma}', s', \overline{\gamma}\rangle \backslash \langle\gamma'', s', \gamma'\rangle = \arg(C|_{w'}[\overline{s}'(\varepsilon)], s')$ constructs the last two arguments and thus

$$s'(w'z) = \overline{\gamma} = \rho'(c) \qquad \text{and} \qquad s'(w'z1) = \gamma'' = d_1(\varepsilon) \qquad \text{and} \qquad s'(w'z2) = \gamma' = d_2(\varepsilon) \ .$$

Since $s' \in \mathrm{Spines}_G$ we have $\frac{s'(w'z)}{s'(w'z1) \quad s'(w'z2)} \in P$, which together with the equalities above yields the existence of the production $\frac{\rho'(c)}{d_1(\varepsilon) \quad d_2(\varepsilon)} \in P$ as required. This concludes the induction and establishes subgoal (i) $\rho'(\mathcal{D}_{G'}^{s,\gamma}) \subseteq R_G$ and together with it also subgoal (iv) as already argued.

For subgoal (ii), let $s \in \mathrm{Spines}_G$ and $\gamma \in \Delta$ and consider derivation trees $d', d'' \in \mathfrak{D}_{G'}^{s,\gamma}$ with $\rho'(d') = d = \rho'(d'')$ and $\operatorname{cat}(d'(\varepsilon)) = \operatorname{cat}(d''(\varepsilon))$. The latter follows from the facts that the root label of any derivation tree of $\mathfrak{D}_{G'}^{s,\gamma} \setminus \operatorname{supp}(L)$ that gets relabeled to d via ρ' is $\langle d(\varepsilon), s, \gamma\rangle$ and that $d' \in \mathfrak{D}_{G'}^{s,\gamma} \cap \operatorname{supp}(L)$ with $\rho'(d') = d$ implies $d' = \langle d(\varepsilon), \langle d(\varepsilon), s, \gamma\rangle\rangle$. Obviously, we have $\mathrm{pos}(d') = \mathrm{pos}(d) = \mathrm{pos}(d'')$. If $\mathrm{pos}(d) = \{\varepsilon\}$, then trivially $d' = d''$ as already argued. Otherwise we prove that $\text{cat}(d'(1)) = \text{cat}(d''(1))$ and $\text{cat}(d'(2)) = \text{cat}(d''(2))$, which can then be used inductively to show that $d' = d''$. Let $d'(\varepsilon) = \langle \gamma', s, \gamma\rangle\beta$ for some $\gamma' \in \Delta$ and suffix β. The child categories $\{\text{cat}(d'(1)), \text{cat}(d'(2))\}$ are $\{a, \langle \gamma', s, \gamma\rangle\beta \,|\, a\}$ for some $| \in \{/, \backslash\}$ and atom $a \in A$. Indeed, we show that $|$ and a are uniquely determined, which also settles which category labels which child. If $\beta \neq \varepsilon$, then it determines the spine $s' \in \text{Spines}_G$, which occurs in all second components of atoms in the suffix β by the construction of L. In turn, this uniquely determines $|$ and a to be those that make $\langle \gamma', s, \gamma\rangle\beta \,|\, a$ a subtree of $\text{arg}(s', s')[\langle \gamma', s, \gamma\rangle]$. It remains to consider the case $\beta = \varepsilon$. Every useful non-atomic category is a prefix of a category of $\bigcup_{\alpha\in\Delta_0} C_\alpha$ and each that starts with $\langle \gamma', s, \gamma\rangle$ is a prefix of $\text{arg}(s', s')[\langle \gamma', s, \gamma\rangle]$ for some unknown $s' \in \text{Spines}_G(\alpha)$ and $\alpha \in \Delta_0$ with $s'(\varepsilon) = \gamma'$ by the construction of L. Since $s'(\varepsilon) = \gamma'$, we conclude that $| = /$ if $\pi(\gamma') = 1$ and $| = \backslash$ otherwise. Due to the relabeling ρ' we conclude that $a = \langle d(\overline{\pi(\gamma')}),\, s',\, d(\pi(\gamma'))\rangle$ and thus $s'(1) = d(1)$ and $s'(2) = d(2)$. Continuing the same process with symbol $d(\pi(\gamma'))$, we obtain all of s'. Since $|$ and a are uniquely determined, we have proved $d'(1) = d''(1)$ as well as $d'(2) = d''(2)$ as desired. As already mentioned iterating the argument then yields $d' = d''$, which proves injectivity of $\rho' \colon \mathfrak{D}^{s,\gamma}_{G'} \to R_G$ because as noted above the root label category of any derivation tree $d' \in \mathfrak{D}^{s,\gamma}_{G'}$ that relabels to $\rho'(d') = d$ is $\text{cat}(d'(\varepsilon)) = \langle d(\varepsilon), s, \gamma\rangle$.

For the final subgoal (iii), which is the converse inclusion $R_G \subseteq \rho'(\mathfrak{D}^{s,\gamma}_{G'})$, we first prove an auxiliary statement. Let $d' \in \mathfrak{D}^{s,\gamma}_{G'}$ be a derivation tree. Then for every $s' \in \text{Spines}_G$ and $\gamma' \in \Delta$ there exists a derivation tree $d'_{s',\gamma'} \in \mathfrak{D}^{s',\gamma'}_{G'}$ such that $\rho'(d'_{s',\gamma'}) = \rho'(d')$. In other words, in any derivation tree with an atomic category at the root we can adjust the derivation tree such that the root label contains any spine $s' \in \text{Spines}_G$ and third component $\gamma' \in \Delta$. The obtained tree is still a derivation tree and relabels to the same run as d'. This statement is very easy to prove using [14, Lemma 3.1], which shows that $d'(\varepsilon)$ is the target of a category of $\bigcup_{\alpha\in\Delta_0} C_\alpha$. However, by the construction of C_α those targets always allow each spine s' as second component and each γ' as third component. A detailed proof is left to the reader.

Before we proceed we define the functions $\text{prune} \colon R_G \to \text{Spines}_G$ and $\text{spine-path} \colon R_G \to [2]^*$ inductively by

$$\text{prune}(\alpha) = \alpha \qquad\qquad \text{spine-path}(\alpha) = \varepsilon$$

$$\text{prune}(\gamma(d_1, d_2)) = \begin{cases} \gamma(\text{prune}(d_1),\, d_2(\varepsilon)) & \text{if } \pi(\gamma) = 1 \\ \gamma(d_1(\varepsilon),\, \text{prune}(d_2)) & \text{otherwise} \end{cases} \qquad \text{spine-path}(\gamma(d_1, d_2)) = \pi(\gamma)\,\text{spine-path}(d_{\pi(\gamma)})$$

for every $\alpha \in \Delta_0$, $\gamma \in \Delta_2$, and $d_1, d_2 \in R_G$. It is straightforward to show that $\text{prune}(d) \in \text{Spines}_G$ as well as $\text{spine-path}(d) \in \text{pos}(d)$ for all $d \in R_G$. We return to the main proof that $R_G \subseteq \rho'(\mathfrak{D}^{s,\gamma}_{G'})$, which is achieved by induction on $d \in R_G$. In the induction base, let $d \in \Delta_0$. Then $\langle d, \langle d, s, \gamma\rangle\rangle \in \text{supp}(L) \cap \mathfrak{D}^{s,\gamma}_{G'}$ and thus $d \in \rho'(\mathfrak{D}^{s,\gamma}_{G'})$ by the definition of ρ'. In the induction step, we have $d \notin \Delta_0$ and the desired property $\overline{d} \in \rho'(\mathfrak{D}^{s,\gamma}_{G'})$ is true for all proper subtrees $\overline{d}$ of d that are located next to the spine. We let $s' = \text{prune}(d)$ and $w = \text{spine-path}(d)$. More precisely, let $w = z_1 \cdots z_k$ with $z_1, \dots, z_k \in [2]$. We first deal with the positions outside the spine. For every $i \in [k]$, let $\overline{w_i} = z_1 \cdots z_{i-1}\overline{z_i}$, which refer to the positions outside the spine s'. Similarly, for every $i \in [0, k]$, let $w_i = z_1 \cdots z_i$ be the i-th position on the spine s'. Trivially, $s'(\overline{w_i}) = d(\overline{w_i})$ for all $i \in [k]$ by the construction of s'. For every $i \in [k]$ we know that $d|_{\overline{w_i}} \in \rho'(\mathfrak{D}^{s,\gamma}_{G'})$ by the induction hypothesis and together with the auxiliary statement we obtain that there exists a derivation tree $d'_i \in \mathfrak{D}^{s',d(w_i)}_{G'}$ such that $\rho'(d'_i) = d|_{\overline{w_i}}$. Let $c = \text{arg}(s', s')[\langle d(\varepsilon), s, \gamma\rangle] \in C_{d(w_k)}$. More precisely, let $c = \langle d(\varepsilon), s, \gamma\rangle |_1 a_1 |_2 \cdots |_k a_k$ for some $|_1, \dots, |_k \in \{/, \backslash\}$ and $a_1, \dots, a_k \in A$. Note that the categories $a_1, \dots, a_k$ are atomic. By the construction of c we know that (i) $|_i = /$ if and only if $z_i = 1$, and (ii) $a_i = \langle s'(\overline{w_i}), s', s'(w_i)\rangle$ for every $i \in [k]$. Additionally, note that $s'(\overline{w_i}) = d(\overline{w_i})$ and $s'(w_i) = d(w_i)$ for all $i \in [k]$, which yields $a_i = d'_i(\varepsilon)$. Now we can construct the required derivation $d' \in \mathfrak{D}^{s,\gamma}_{G'}$ by combining this category c with the subderivations $d'_i \in \mathfrak{D}^{s',d(w_i)}_{G'}$. For every $i \in \mathbb{Z}_k$ we let

$$t'_k = \langle d(w_k), c\rangle \qquad \text{and} \qquad t'_i(\varepsilon) = \langle d(\varepsilon), s, \gamma\rangle |_1 a_1 |_2 \cdots |_i a_i \qquad t'_i|_{z_{i+1}} = t'_{i+1} \qquad t'_i|_{\overline{z_{i+1}}} = d'_{i+1} \enspace .$$

Finally, we set $d' = t'_0$. A straightforward check shows that $d' \in \mathfrak{D}^{s,\gamma}_{G'}$. It remains to show that $\rho'(d') = d$. Obviously, $\text{pos}(d') = \text{pos}(d)$, so we need to show that $d(v) = \rho'(\text{cat}(d'(v)))$ for every $v \in \text{pos}(d)$. If $v = \overline{w_i}v'$ for some $i \in [k]$ and $v' \in \text{pos}(d|_{\overline{w_i}})$, then this is trivially true because $d'|_{\overline{w_i}} = d'_i$ and we already observed that $\rho'(d'_i) = d|_{\overline{w_i}}$. Consequently, we only need to prove the property for all the prefixes w_i of w with $i \in \mathbb{Z}_k$. By the construction of d' we have $\text{cat}(d'(w_i)) = \langle d(\varepsilon), s, \gamma\rangle |_1 a_1 |_2 \cdots |_i a_i$. For $i = 0$, we thus obtain $\rho'(\text{cat}(d'(w_0))) = \rho'(\langle d(\varepsilon), s, \gamma\rangle) = d(\varepsilon)$. For all $i \in [k]$ we have

$$\rho'(\text{cat}(d'(w_i))) = \rho'(\langle d(\varepsilon), s, \gamma\rangle |_1 a_1 |_2 \cdots |_i a_i) = d(w_i)$$

since $a_i = \langle d(\overline{w_i}), s', d(w_i)\rangle$. This completes the proof of $R_G \subseteq \rho'(\mathfrak{D}^{s,\gamma}_{G'})$.

It remains to prove $\mathrm{wt}^{\rho'}_{G'} = \mathrm{wt}_G$ using the proved subgoals. Let $t \in T_{\Delta_2,\emptyset}(\Delta_0)$ be arbitrary. Then

$$\mathrm{wt}^{\rho'}_{G'}(t) = \sum_{\substack{d' \in (\rho')^{-1}(t) \\ d'(\varepsilon) = \langle t(\varepsilon), \top, t(\varepsilon)\rangle}} \mathrm{wt}_{G'}(d') = \sum_{d' \in (\rho')^{-1}(t) \cap \mathfrak{D}^{\top,t(\varepsilon)}_{G'}} \mathrm{wt}_{G'}(d') = \sum_{\substack{d' \in \mathfrak{D}^{\top,t(\varepsilon)}_{G'} \\ t = \rho'(d')}} \mathrm{wt}_G(t)$$

because the weight of non-derivations is 0 and we proved that $\rho' \colon \mathfrak{D}^{\top,t(\varepsilon)}_{G'} \to R_G$ is a weight-preserving bijection. Thus, $\mathrm{wt}^{\rho'}_{G'}(t) = \mathrm{wt}_G(t)$ if $t \in R_G$. Otherwise, $\mathrm{wt}^{\rho'}_{G'}(t) = 0 = \mathrm{wt}_G(t)$ if $t \notin R_G$, which completes the proof. □

5. 1-wCCGs

In this section we show that the weighted forests generatable by 1-wCCG are exactly the regular weighted forests. 1-wCCG can use composition rules of degree at most 1. The inclusion of the class of weighted forests generatable by 1-wCCG in the class of regular weighted forests has already been shown in Lemma 9, so we only have to prove the remaining direction. Lemma 6 shows that the weighted forests generatable by 1-wCCG are closed under deterministic relabelings, so we only need to prove that the weighted forest of each weighted local tree grammar $G = (\Sigma, S, P, \mathrm{wt})$ is generatable by some 1-wCCG. Without loss of generality, we can assume that (i) $\Sigma = \mathbb{Z}_m = \{0, \dots, m-1\}$ for some $m \in \mathbb{N}$, (ii) $\frac{\sigma}{\sigma_1 \quad \sigma_2} \in P$ for every $\sigma, \sigma_1, \sigma_2 \in \Sigma$, and (iii) $\frac{\alpha}{} \in P$ for every $\alpha \in \Sigma_0$. The latter 2 properties can be achieved by setting $\mathrm{wt}(\frac{\sigma}{\sigma_1 \quad \sigma_2}) = 0$ and $\mathrm{wt}(\frac{\alpha}{}) = 0$ for all undesired productions.

The main idea of the construction is taken from the unweighted setting [27]. Our goal is to construct a 1-wCCG G' and a deterministic category Σ-relabeling ρ such that for all productions $\frac{\sigma}{\sigma_1 \quad \sigma_2} \in P$ there is a rule $\frac{c}{c_1 \quad c_2}\Pi$ such that $\rho(c) = \sigma$, $\rho(c_1) = \sigma_1$, and $\rho(c_2) = \sigma_2$. In this way we model the runs of G as the derivation trees of the constructed wCCG G', where the weights that control which productions are admissible are transferred to the rule system Π. We use only first-order categories with at most one argument which has to start with a forward-slash.

In the following, we lay out some preliminary considerations. The productions P permit to derive all ordered pairs of terminals $(\sigma_1, \sigma_2) \in \Sigma^2$ from $\sigma \in \Sigma_2$. Clearly, there are m^2 such pairs. Given a set A of atoms and a category a/a' with $a, a' \in A$, there are $|A|$ pairs of input categories a/a'' and a''/a' with $a'' \in A$ whose composition results in a/a'. This is depicted in Figure 7 (left). To cover all terminal pairs (σ_1, σ_2) that can be generated by $\sigma = \rho(a/a')$, the constructed 1-wCCG G' needs m^2 atoms. The relabeling ρ is defined such that all pairs (σ_1, σ_2) are actually covered. The strategy for categories of arity 1 is illustrated in Figure 7 (right). The set I of initial atomic categories is restricted such that for each start terminal $s \in S$ only a single category a_s with $\rho(a_s) = s$ is contained in I. This restriction is necessary to ensure that for each run $d \in \mathcal{D}^s_G$ only a single derivation tree $d' \in \mathcal{D}^{a_s}_{G'}$ exists. Finally, we let $\preceq_{\mathrm{lex}}$ be the total lexicographic order on $\mathbb{N}^2$.

Definition 11. *Let $G = (\mathbb{Z}_m, S, P, \mathrm{wt})$ be a weighted local tree grammar with $\frac{\sigma}{\sigma_1 \quad \sigma_2} \in P$ for every $\sigma, \sigma_1, \sigma_2 \in \mathbb{Z}_m$ and $\frac{\sigma'}{} \in P$ for every $\sigma' \in (\mathbb{Z}_m)_0$. We construct the 1-wCCG $C_G = (\mathbb{Z}_m, A, R, I, L, \mathrm{wt}')$ with $A = \mathbb{Z}_m^2$ and the deterministic category $\mathbb{Z}_m$-relabeling $\rho \colon (\mathcal{C}(A) \cup \mathrm{supp}(L)) \to \mathbb{Z}_m$ such that*

$$I = \left\{ \min\nolimits_{\preceq_{\mathrm{lex}}} (\rho^{-1}(\sigma) \cap A) \;\middle|\; \sigma \in S \right\}$$

$$R = \left\{ \Big(\frac{ax}{ax/c \quad c}\Big) \;\middle|\; a, c \in A \right\} \cup \left\{ \Big(\frac{ax/b}{ax/c \quad c/b}\Big) \;\middle|\; a, b, c \in A \right\}$$

$$\mathrm{wt}'\Big(\frac{ax}{ax/c \quad c}\Big) = \mathrm{wt}\Big(\frac{\rho(a)}{\rho(a/c) \quad \rho(c)}\Big) \quad \textit{and} \quad \mathrm{wt}'\Big(\frac{ax/b}{ax/c \quad c/b}\Big) = \mathrm{wt}\Big(\frac{\rho(a/b)}{\rho(a/c) \quad \rho(c/b)}\Big) \quad \textit{for all } a, b, c \in A$$

$$L(\langle \sigma, c\rangle) = \begin{cases} \mathrm{wt}\Big(\frac{\rho(c)}{}\Big) & \textit{if } \rho(c) = \sigma \\ 0 & \textit{otherwise,} \end{cases}$$

and $\rho(\langle i, j\rangle) = i$ and $\rho(\langle i, j\rangle\, \alpha \,/\langle i', j'\rangle) = i + j' \bmod m$ for all $i, i', j, j' \in \mathbb{Z}_m$ and argument contexts $\alpha \in \mathcal{A}(A)$ as well as $\rho(\langle \sigma, c\rangle) = \sigma$ for every $\langle \sigma, c\rangle \in \mathrm{supp}(L)$. The relabeling is irrelevant for all other categories.

	a_0	a_1	a_2	a_3
a_0	a_0/a_0	a_0/a_1	a_0/a_2	a_0/a_3
a_1	a_1/a_0	a_1/a_1	a_1/a_2	a_1/a_3
a_2	a_2/a_0	a_2/a_1	a_2/a_2	a_2/a_3
a_3	a_3/a_0	a_3/a_1	a_3/a_2	a_3/a_3

	$\langle 0,0\rangle$	$\langle 0,1\rangle$	$\langle 0,2\rangle$	$\langle 1,0\rangle$	$\langle 1,1\rangle$	$\langle 1,2\rangle$	$\langle 2,0\rangle$	$\langle 2,1\rangle$	$\langle 2,2\rangle$
$\langle 0,0\rangle$	0	1	2	0	1	2	0	1	2
$\langle 0,1\rangle$	0	1	2	0	1	2	0	1	2
$\langle 0,2\rangle$	0	1	2	0	1	2	0	1	2
$\langle 1,0\rangle$	1	2	0	1	2	0	1	2	0
$\langle 1,1\rangle$	1	2	0	1	2	0	1	2	0
$\langle 1,2\rangle$	1	2	0	1	2	0	1	2	0
$\langle 2,0\rangle$	2	0	1	2	0	1	2	0	1
$\langle 2,1\rangle$	2	0	1	2	0	1	2	0	1
$\langle 2,2\rangle$	2	0	1	2	0	1	2	0	1

Figure 7: The category matrix (left) contains all first-order categories of arity 1 with only forward slashes in a 1-wCCG with four atoms. Each category is the result of the forward composition of a category taken from the same row and one from the same column, respectively. The i-th entry of each row can be combined with the i-th entry of each column. Thus, each category a/a' is the result of four different forward compositions combining a/a'' and a''/a' (with four choices for a''). The relabeling matrix (right) shows a 1-wCCG with nine atomic categories after relabeling using category relabeling $\rho\colon C(\mathbb{Z}_3^2) \to \mathbb{Z}_3$, obtained from a local tree grammar G with three terminals by applying Definition 11. Note that when slicing the matrix evenly into blocks of size 3×3, the entries in the rows cycle through the terminals, whereas in a single column, each block has only a single terminal in all three entries. Relabeling in this manner ensures that each category can be obtained as the composition of two input categories that are relabeled to $\sigma_1, \sigma_2 \in \mathbb{Z}_3$ for all ordered pairs (σ_1, σ_2) of terminals. Suppose we want to find two categories relabeled to terminals $(\sigma_1, \sigma_2) = (0, 2)$ whose composition yields $\langle i, j\rangle/\langle i', j'\rangle = \langle 0, 1\rangle/\langle 0, 1\rangle$. These are categories $\langle 0, 1\rangle/\langle 1, 0\rangle$ and $\langle 1, 0\rangle/\langle 0, 1\rangle$ since $\langle k, \ell\rangle = \langle \sigma_2 - j' \mod 3, \sigma_1 - i \mod 3\rangle = \langle 2 - 1 \mod 3, 0 - 0 \mod 3\rangle = \langle 1, 0\rangle$.

Lemma 12. *Every regular weighted forest $\varphi\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$ is generatable by some 1-wCCG.*

Proof. As argued above, we use [41, Theorem 1], due to which for each regular weighted forest $\varphi\colon T_{\Sigma_2,\emptyset}(\Sigma_0) \to H$, there exists a weighted local tree grammar $G = (\Delta, S, P, \mathrm{wt})$ and a deterministic relabeling $\rho'\colon \Delta \to \Sigma_2 \cup \Sigma_0$ such that $\rho'(\mathrm{wt}_G) = \varphi$. Without loss of generality, let $\Delta = \mathbb{Z}_m$, and let G comply with the conditions described before, so $\frac{\sigma}{\sigma_1 \quad \sigma_2} \in P$ for every $\sigma, \sigma_1, \sigma_2 \in \mathbb{Z}_m$ and $\frac{\sigma'}{} \in P$ for every $\sigma' \in (\mathbb{Z}_m)_0$. Based on the weighted local tree grammar G, we use Definition 11 to define the 1-wCCG $G' = C_G = (\mathbb{Z}_m, A, R, I, L, \mathrm{wt}')$ with $A = \mathbb{Z}_m^2$ and deterministic category $\mathbb{Z}_m$-relabeling $\rho\colon (C(A) \cup \mathrm{supp}(L)) \to \mathbb{Z}_m$. As 1-wCCG is closed under deterministic relabelings, it suffices to show that $\mathrm{wt}^{\rho}_{G'} = \mathrm{wt}_G$ to prove the main statement. To this end, we establish that ρ is a weight-preserving bijection between the derivation trees $\mathcal{D}^I_{G'} = \bigcup_{a \in I} \mathcal{D}^a_{G'}$ of G' and the runs $\rho(\mathcal{D}^I_{G'}) \subseteq R^S_G = \bigcup_{\sigma \in S} R^{\sigma}_G$ containing runs of G. It is clear that the image of $\mathcal{D}^I_{G'}$ under ρ results in a run, since G permits all productions. By the choice of I, this run is rooted in an element $\sigma \in S$.

First we observe that the categories that occur in $\mathcal{D}_{G'}$ are restricted to the set $C_1 = A \cup \{a/b \mid a, b \in A\}$ of categories that are either atomic or have at most one atomic argument with a forward slash. This follows because all categories can have only arguments that are already present in the lexicon by [14, Lemma 3.1] and because in a 1-wCCG the limited rule degree prevents that arities of categories grow larger than the maximal arity found in the lexicon.

We start by showing the following auxiliary statement: Given a production $\frac{\sigma}{\sigma_1 \quad \sigma_2} \in P$ and a category $c \in C_1$ with $\rho(c) = \sigma$, there are unique categories $c_1, c_2 \in C_1$ with $\rho(c_1) = \sigma_1$ and $\rho(c_2) = \sigma_2$ such that $\frac{c}{c_1 \quad c_2}\Pi$ is permitted by the rule system. Additionally, the weight of this rule is $\mathrm{wt}'(\frac{c}{c_1 \quad c_2}) = \mathrm{wt}(\frac{\sigma}{\sigma_1 \quad \sigma_2})$.

We distinguish two cases. If c is atomic, then the desired rule instance $\frac{c}{c_1 \quad c_2}\Pi$ of G' has the shape $\frac{\langle i,j\rangle}{\langle i,j\rangle/\langle k,\ell\rangle \quad \langle k,\ell\rangle}$, where $i, j, k, \ell \in \mathbb{Z}_m$. Relabeling the input categories c_1 and c_2 yields $\rho(c_1) = \rho(\langle i, j\rangle/\langle k, \ell\rangle) = i + \ell \mod m$ and $\rho(c_2) = \rho(\langle k, \ell\rangle) = k$. From $\rho(c_2) = \sigma_2$ follows $k = \sigma_2$, and from $\rho(c_1) = \sigma_1$ follows $i + \ell \mod m = \sigma_1$, thus $\ell = \sigma_1 - i \mod m$. Clearly, k and ℓ and therefore also c_1 and c_2 are uniquely determined by the choice of $c = \langle i, j\rangle$, σ_1, and σ_2, as required. If c has arity 1, then the desired rule instance $\frac{c}{c_1 \quad c_2}\Pi$ of G' is of the form $\frac{\langle i,j\rangle/\langle i',j'\rangle}{\langle i,j\rangle/\langle k,\ell\rangle \quad \langle k,\ell\rangle/\langle i',j'\rangle}$, where $i, i', j, j', k, \ell \in \mathbb{Z}_m$. The input categories are relabeled to $\rho(c_1) = \rho(\langle i, j\rangle/\langle k, \ell\rangle) = i + \ell \mod m = n_1$ and $\rho(c_2) = \rho(\langle k, \ell\rangle/\langle i', j'\rangle) = k + j' \mod m = \sigma_2$. Hence, we have $\ell = \sigma_1 - i \mod m$ and $k = \sigma_2 - j' \mod m$. Again, it is easy to see that k and ℓ are determined by the choice of c, σ_1, and σ_2. We conclude that this choice uniquely fixes $c_1, c_2 \in C_1$ as well. The approach is illustrated in Figure 7. In both cases,

$$\mathrm{wt}'\Big(\frac{c}{c_1 \quad c_2}\Big) = \mathrm{wt}\Big(\frac{\rho(c)}{\rho(c_1) \quad \rho(c_2)}\Big) = \mathrm{wt}\Big(\frac{\sigma}{\sigma_1 \quad \sigma_2}\Big)$$

holds by definition of wt′.

Next, we show injectivity of ρ using the auxiliary statement. Let $d \in \rho(\mathcal{D}_G^I)$ with $d(\varepsilon) = \sigma \in S$. It follows that each $d' \in \rho^{-1}(d)$ has root category $c = \mathrm{cat}(d'(\varepsilon)) = \min_{\leq_{\mathrm{lex}}}\{a \in \rho^{-1}(\sigma) \cap A\}$. There are two cases. If d consists of a single node, then a production $(\frac{\sigma}{})$ is used. In that case, there is a lexicon entry $L(\langle\sigma, c\rangle) = \mathrm{wt}(\frac{\sigma}{})$ with $\rho(c) = \sigma$. Since c and σ are fixed, this is the only lexicon entry that we can use. As a consequence, there is only a single derivation tree $d' \in \rho^{-1}(d)$. Now assume that d has more than one node. Let $\sigma_1 = d(1)$ and $\sigma_2 = d(2)$. Because $c = d'(\varepsilon)$ and σ_1, σ_2 are fixed and $\rho(d'(1)) = \sigma_1$ and $\rho(d'(2)) = \sigma_2$ have to hold, there are unique categories $c_1, c_2 \in C_1$ such that $\frac{c}{c_1 \quad c_2}\Pi$ is a valid rule instance of G'. Thus, we have $d'(1) = c_1$ and $d'(2) = c_2$. We can then successively apply the auxiliary statement for each position $w \in \mathrm{pos}(d)$ in a top-down manner and use $d'(w)$, $d(w1)$, and $d(w2)$ to infer the categories $d'(w1)$ and $d'(w2)$. Because the root of d' is fixed, all other categories are fixed as well. The leaves of d correspond to productions $\frac{\sigma}{}$ that in turn correspond to lexicon entries. Since the leaf categories are fixed by their parent nodes, there is only a single choice for each lexicon entry. We can conclude that $|\rho^{-1}(d)| = 1$, so $\rho\colon \mathcal{D}_{G'}^I \to \rho(\mathcal{D}_{G'}^I)$ is a bijection.

To see that ρ is weight-preserving we can use the auxiliary statement as well, namely

$$\mathrm{wt}'\Big(\frac{d'(w)}{d'(w1) \quad d'(w2)}\Big) = \mathrm{wt}\Big(\frac{\rho(d'(w))}{\rho(d'(w1)) \quad \rho(d'(w2))}\Big) = \mathrm{wt}\Big(\frac{d(w)}{d(w1) \quad d(w2)}\Big)\ ,$$

giving a one-to-one correspondence between weights of applied rules and branching productions. For the leaf symbols the weights of lexicon entries are given by $L(\langle\sigma, c\rangle) = \mathrm{wt}(\frac{\sigma}{})$ in the same manner, so there we have a direct correspondence to productions of G as well.

Note that the only runs $d \in R_G^S$ that are not in $\rho(\mathcal{D}_{G'}^I)$ are those containing positions $w \in \mathrm{leaves}(d)$ such that $\langle d(w), c\rangle \notin \mathrm{supp}(L)$ for some $c \in \rho^{-1}(d(w))$. These runs have weight 0. Preimages of all other runs $d \in R_G^S$ can be constructed with the approach described above. In summary, for every $d' \in \mathcal{D}_{G'}^I$ and $d = \rho(d')$ we have $\mathrm{wt}(d) = \mathrm{wt}'(d')$, and for every $d \in R_G^S$ with $d \notin \rho(\mathcal{D}_{G'}^I)$ we have $\mathrm{wt}(d) = 0$, which shows $\mathrm{wt}_{G'}^{\rho} = \mathrm{wt}_G$. □

Theorem 13. *The weighted forests generatable by* 1*-wCCG are exactly the regular weighted forests.*

6. Inclusion in the Context-Free Weighted Forests

Finally, we aim to settle the relation between the weighted derivation forests of wCCGs and the simple monadic context-free weighted forests, which are exactly those weighted forests that are generated by wsCFTG. To this end, let us fix a wCCG $G = (\Sigma, A, R, I, L, \mathrm{wt})$. However, the derivation forest $\mathcal{D}_G$ might potentially contain infinitely many categories, whereas infinitely many symbols are impossible in any context-free weighted forest. An illustration of this problem can be found in [28, Example 21] for the unweighted case. As in the unweighted case, we circumvent this problem by considering an alternative notion, called *rule trees*, which plainly record the used rules instead of the categories. The leaves continue to record the used lexicon entry, so full category information is available at the leaves. An example rule tree is depicted in Figure 9 (top left). The category of the root of each subtree can be uniquely determined from the subtree supposing that the subtree is well-formed. Let us formally define this category evaluation and the associated well-formedness condition. To simplify the notation, let $\mathrm{T} = T_{R,\emptyset}(\mathrm{supp}(L))$.

Definition 14 (see [28, Definition 22])**.** *A tree* $t \in \mathrm{T}$ *is a* rule tree *of* G *if* $\mathrm{eval}_G(t)$ *is defined, where* $\mathrm{eval}_G\colon \mathrm{T} \dashrightarrow C(A)$ *is the partial mapping that is inductively defined by*

- $\mathrm{eval}_G(\langle\sigma, c\rangle) = c$ *for all* $\langle\sigma, c\rangle \in \mathrm{supp}(L)$,
- $\mathrm{eval}_G(\frac{ax\gamma}{ax/c \quad c\gamma}(t_1, t_2)) = a\alpha\gamma$ *for all* $\frac{ax\gamma}{ax/c \quad c\gamma} \in R$ *and* $t_1, t_2 \in \mathrm{T}$ *with* $\mathrm{eval}_G(t_1) = a\alpha/c$ *and* $\mathrm{eval}_G(t_2) = c\gamma$, *and*
- $\mathrm{eval}_G(\frac{ax\gamma}{c\gamma \quad ax\backslash c}(t_1, t_2)) = a\alpha\gamma$ *for all* $\frac{ax\gamma}{c\gamma \quad ax\backslash c} \in R$ *and* $t_1, t_2 \in \mathrm{T}$ *with* $\mathrm{eval}_G(t_1) = c\gamma$ *and* $\mathrm{eval}_G(t_2) = a\alpha\backslash c$.

Let $\mathcal{R}_G$ *be the set of all rule trees of* G*, and* $\mathcal{R}_G^c = \mathrm{eval}_G^{-1}(\{c\})$ *for all* $c \in C(A)$*. The weight of a rule tree* $t \in \mathcal{R}_G$ *is*

$$\mathrm{wt}_G(t) = \Bigg(\prod_{w \in \mathrm{pos}(d)\setminus\mathrm{leaves}(d)} \mathrm{wt}(t(w))\Bigg) \cdot \Bigg(\prod_{w \in \mathrm{leaves}(d)} L(t(w))\Bigg)$$

and the weighted rule forest $\tau_G\colon \mathrm{T} \to H$ *is given for every* $t \in \mathrm{T}$ *by*

$$\tau_G(t) = \begin{cases} \mathrm{wt}_G(t) & \text{if } \exists a \in I\colon t \in \mathcal{R}_G^a \\ 0 & \text{otherwise.} \end{cases}$$

We note that there is an (obvious) bijection between the derivation trees $\mathcal{D}_G$ and the rule trees $\mathcal{R}_G$. This correspondence extends to a bijection between the subsets $\mathcal{D}_G^c$ and $\mathcal{R}_G^c$ for every category $c \in C(A)$. Finally, we note that all those bijections are trivially weight-preserving, so we have found a suitable representation that will permit the comparison to wsCFTG.

Our next goal is to construct a wsCFTG that computes exactly the weighted rule forest τ_G. To this end, we essentially follow the unweighted approach of [28], but need to modify the construction slightly as in the previous sections to correctly account for the weight. We start by limiting the number of categories. Let $k \in \mathbb{N}_+$ be the maximal arity of a category in

$$\operatorname{cat}(\operatorname{supp}(L)) \cup \left\{c\gamma \mid \frac{ax\gamma}{ax/c \quad c\gamma} \in R\right\} \cup \left\{c\gamma \mid \frac{ax\gamma}{c\gamma \quad ax\backslash c} \in R\right\} ,$$

i.e., the maximal arity of the categories occurring in the lexicon or as secondary category in a rule of R. Additionally, let $C_L(A, k) = \{c \in C(A, k) \mid \operatorname{argcats}(c) \subseteq \operatorname{argcats}(L)\}$, and define the sets $C_L(A)$, $\mathcal{A}_L(A, k)$, and $\mathcal{A}_L(A)$ analogously. Obviously, the derivation trees $\mathcal{D}_G$ only contain categories whose arguments appear as arguments of lexical entries because each argument of an output category exists in an input category [14, Lemma 3.1]. Hence we can safely restrict ourselves to only categories using these arguments. Intuitively, the nullary nonterminals of the constructed wsCFTG are the "short" categories $C_L(A, k)$ and the unary nonterminals are tuples $\langle a, |c, \gamma\rangle$ consisting of an atom $a \in A$, a single argument $|c$ with $| \in \{/, \backslash\}$ and "short" category $c \in C_L(A, k)$, and a "short" argument context $\gamma \in \mathcal{A}_L(A, k)$. Unary nonterminals can be interpreted as CCG rules in the following sense: Given a primary category with target a, the output category is obtained by replacing the outermost argument $|c$ by the argument context γ. Recall that we write substitutions $\alpha[t]$ as $t\alpha$ for $\alpha \in \mathcal{A}(A)$ and $t \in C(A) \cup \mathcal{A}(A)$.

Definition 15. *We construct the wsCFTG* $G' = (N, R \cup \operatorname{supp}(L), I, P, \mathrm{wt}')$ *with*

- $N = N_1 \cup N_0$ *such that* $N_0 = C_L(A, k)$ *and* $N_1 = \{\langle a, |c, \gamma\rangle \mid a \in A,\ | \in \{/, \backslash\},\ c \in \mathit{argcats}(L),\ \gamma \in \mathcal{A}_L(A, k)\}$,
- *the following set* $P = P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5 \cup P_6 \cup P_7$ *of productions*

$$P_1 = \{c \to \langle\sigma, c\rangle \mid \langle\sigma, c\rangle \in \operatorname{supp}(L)\} \tag{1}$$

$$P_2 = \left\{\langle a, /c, \gamma\rangle \to s(\square, c\gamma) \,\middle|\, s = \frac{ax\gamma}{ax/c \quad c\gamma} \in R\right\} \tag{2}$$

$$P_3 = \left\{\langle a, \backslash c, \gamma\rangle \to s(c\gamma, \square) \,\middle|\, s = \frac{ax\gamma}{c\gamma \quad ax\backslash c} \in R\right\} \tag{3}$$

$$P_4 = \left\{a\alpha\gamma \to s(a\alpha/c, c\gamma) \,\middle|\, s = \frac{ax\gamma}{ax/c \quad c\gamma} \in R,\ \alpha \in \mathcal{A}_L(A),\ |\alpha\gamma| < k\right\} \tag{4}$$

$$P_5 = \left\{a\alpha\gamma \to s(c\gamma,\ a\alpha\backslash c) \,\middle|\, s = \frac{ax\gamma}{c\gamma \quad ax\backslash c} \in R,\ \alpha \in \mathcal{A}_L(A),\ |\alpha\gamma| < k\right\} \tag{5}$$

$$P_6 = \left\{a\alpha\gamma \to \langle a, |c, \gamma\rangle(a\alpha|c) \,\middle|\, a \in A,\ \alpha, \gamma \in \mathcal{A}_L(A),\ | \in \{/, \backslash\}, c \in \mathit{argcats}(L),\ |\alpha| < k,\ |\alpha\gamma| = k\right\} \tag{6}$$

$$P_7 = \left\{\langle a, |c, \gamma\rangle \to \langle a, |'c', \square\rangle(\langle a, |c, \gamma|'c'\rangle(\square)) \,\middle|\, a \in A,\ |, |' \in \{/, \backslash\},\ c, c' \in \mathit{argcats}(L),\ \gamma \in \mathcal{A}_L(A, k-1)\right\} \tag{7}$$

- *and the weight assignment* $\mathrm{wt}' \colon P \to H$ *given for every* $(n \to r) \in P$ *by*

$$\mathrm{wt}'(n \to r) = \begin{cases} L(r(\varepsilon)) & \text{if } r(\varepsilon) \in \operatorname{supp}(L) \\ \mathrm{wt}(r(\varepsilon)) & \text{if } r(\varepsilon) \in R \\ 1 & \text{otherwise.} \end{cases}$$

A nonterminal $c \in N_0$ can be treated in three different ways. First, it can be replaced by an element of $\operatorname{supp}(L)$ and thus produce a nullary terminal (P_1). Second, if its arity is lower than k, it can be replaced by a CCG rule with output category c, producing a binary terminal, and two new nullary nonterminals representing input categories (P_4, P_5). Third, if its arity is exactly k, instead of directly producing a CCG rule, a unary nonterminal is generated as a suitable rule placeholder, together with a nullary nonterminal representing a matching primary category for that rule (P_6). This is necessary because the arity of categories might rise above k in the CCG derivation that is produced next, such that the involved categories can no longer be represented using elements of N_0. In the unweighted case, the productions of P_6 were used regardless of arity. However, this leads to ambiguity and therefore, P_4 and P_5 were introduced here to avoid this. Each unary nonterminal $\langle a, |c, \gamma\rangle \in N_1$ eventually produces a part of the rule tree (or more precisely, a context) where, when starting at a primary category at the bottom (i.e., at the position of $\square$) and following the nodes towards the root, the overall effect on that category is that of the rule represented by $\langle a, |c, \gamma\rangle$. This can be achieved either by means of a single rule performing that operation (P_2, P_3), or by gradually splitting into placeholders for rules that, when applied successively, have the same effect (P_7).

Naturally we still need to convince ourselves that the construction is correct, so we need to prove that $\tau_{G'} = \tau_G$. This will be achieved by five auxiliary lemmas, which, when combined, establish correctness. We start by proving that every derivation tree of $\bigcup_{c \in N_0} \mathcal{D}^c_{G'}$ evaluates to a rule tree of G. More precisely, we claim that $\mathrm{eval}_{G'}(d) \in \mathcal{R}^c_G$ for every derivation tree $d \in \mathcal{D}^c_{G'}$ with $c \in N_0$.

Lemma 16. $\mathrm{eval}_{G'}(\mathcal{D}^c_{G'}) \subseteq \mathcal{R}^c_G$ *for every* $c \in N_0$.

Proof. Let $\mathrm{eval} = \mathrm{eval}_{G'} ; \mathrm{eval}_G$. We start with the following auxiliary statement, which is important enough to warrant special labels. For every nonterminal $n \in N$ and derivation tree $d \in \mathcal{D}^n_{G'}$

(c1) if $n \in N_0$, then $\mathrm{eval}(d) = n$, and

(c2) if $n = \langle a, |c, \gamma\rangle \in N_1$, then $\mathrm{eval}_G(\mathrm{eval}_{G'}(d)[t]) = a\alpha\gamma$ for every $t \in \mathrm{T}$ such that $\mathrm{eval}_G(t) = a\alpha|c$ with $\alpha \in \mathcal{A}_L(A)$.

We distinguish seven cases based on the production $p = d(\varepsilon) \in P$ used at the root of d.

(1) If $p = c \to \langle\sigma, c\rangle \in P_1$ is a production of type (1), then $n = c \in N_0$ and statement (c1) needs to be proved. By the definition of P_1 we get $\langle\sigma, c\rangle \in \mathrm{supp}(L)$. Moreover, $\mathrm{eval}_{G'}(d) = \langle\sigma, c\rangle$ and thus $\mathrm{eval}(d) = c = n$ as required.

(2) If $p = \langle a, /c, \gamma\rangle \to s(\square, c\gamma) \in P_2$ is a production of type (2), then $n = \langle a, /c, \gamma\rangle \in N_1$ and $s = \frac{ax\gamma}{ax/c \quad c\gamma} \in R$. We need to prove statement (c2). Consequently, let $t \in \mathrm{T}$ be such that $\mathrm{eval}_G(t) = a\alpha/c$ with $\alpha \in \mathcal{A}_L(A)$. Obviously, we have $d|_1 \in \mathcal{D}^{c\gamma}_{G'}$ and by the induction hypothesis applied to $d|_1$ we conclude that $\mathrm{eval}(d|_1) = c\gamma$. Hence

$$\mathrm{eval}_G(\mathrm{eval}_{G'}(d)[t]) = \mathrm{eval}_G\Big(\frac{ax\gamma}{ax/c \quad c\gamma}\big(t, \mathrm{eval}_{G'}(d|_1)\big)\Big) = a\alpha\gamma \enspace .$$

(3) If $p = \langle a, \backslash c, \gamma\rangle \to s(c\gamma, \square) \in P_3$ is a production of type (3), then we prove statement (c2) in the same way as in the previous case (2).

(4) If $p = a\alpha\gamma \to s(a\alpha/c, c\gamma) \in P_4$ is a production of type (4), then $n = a\alpha\gamma \in N_0$ and $s = \frac{ax\gamma}{ax/c \quad c\gamma} \in R$. We need to prove statement (c1). Obviously, we have $d|_1 \in \mathcal{D}^{a\alpha/c}_{G'}$ as well as $d|_2 \in \mathcal{D}^{c\gamma}_{G'}$ and by the induction hypothesis applied to both we conclude that $\mathrm{eval}(d|_1) = a\alpha/c$ and $\mathrm{eval}(d|_2) = c\gamma$. Hence $\mathrm{eval}(d) = a\alpha\gamma = n$ as required.

(5) If $p = a\alpha\gamma \to s(c\gamma, a\alpha\backslash c) \in P_5$ is a production of type (5), then we prove statement (c1) in the same way as in the previous case (4).

(6) If $p = a\alpha\gamma \to \langle a, |c, \gamma\rangle(a\alpha|c) \in P_6$ is a production of type (6), then $n = a\alpha\gamma \in N_0$ and we again need to prove statement (c1). Obviously we have $d|_1 \in \mathcal{D}^{\langle a,|c,\gamma\rangle}_{G'}$ as well as $d|_2 \in \mathcal{D}^{a\alpha|c}_{G'}$ and by the induction hypothesis applied to $d|_2$ we obtain that $\mathrm{eval}(d|_2) = a\alpha|c$. Moreover, $\mathrm{eval}_{G'}(d) = \mathrm{eval}_{G'}(d|_1)[\mathrm{eval}_{G'}(d|_2)]$ and by the induction hypothesis applied to $d|_1$ with $t = \mathrm{eval}_{G'}(d|_2)$ we obtain

$$\mathrm{eval}(d) = \mathrm{eval}_G\big(\mathrm{eval}_{G'}(d)\big) = \mathrm{eval}_G\big(\mathrm{eval}_{G'}(d|_1)[\mathrm{eval}_{G'}(d|_2)]\big) = \mathrm{eval}_G\big(\mathrm{eval}_{G'}(d|_1)[t]\big) = a\alpha\gamma = n \enspace .$$

(7) If $p = \langle a, |c, \gamma\rangle \to \langle a, |'c', \square\rangle(\langle a, |c, \gamma|'c'\rangle(\square)) \in P_7$ is a production of type (7), then $n = \langle a, |c, \gamma\rangle \in N_1$ and we need to prove statement (c2). Let $t \in \mathrm{T}$ be such that $\mathrm{eval}_G(t) = a\alpha|c$ with $\alpha \in \mathcal{A}_L(A)$. Clearly,

$$\mathrm{eval}_{G'}(d) = \mathrm{eval}_{G'}(d|_1)[\mathrm{eval}_{G'}(d|_2)] \qquad \text{and} \qquad \mathrm{eval}_{G'}(d)[t] = \mathrm{eval}_{G'}(d|_1)\big[\mathrm{eval}_{G'}(d|_2)[t]\big] \enspace .$$

By the induction hypothesis applied to $d|_2$, we obtain $\mathrm{eval}_G(\mathrm{eval}_{G'}(d|_2)[t]) = a\alpha\gamma|'c'$. Now we apply the induction hypothesis to $d|_1$ with $t' = \mathrm{eval}_{G'}(d|_2)[t]$, which is suitable due to $\mathrm{eval}_G(t') = \mathrm{eval}_G(\mathrm{eval}_{G'}(d|_2)[t]) = a\alpha\gamma|'c'$, and obtain

$$\mathrm{eval}_G\big(\mathrm{eval}_{G'}(d)[t]\big) = \mathrm{eval}_G\big(\mathrm{eval}_{G'}(d|_1)[t']\big) = a\alpha\gamma \enspace .$$

This completes the proof of the auxiliary statement, which yields $\mathrm{eval}(d) = c$ for all $d \in \mathcal{D}^c_{G'}$ with $c \in N_0$ and thus $\mathrm{eval}_{G'}(d) \in \mathcal{R}^c_G$. $\square$

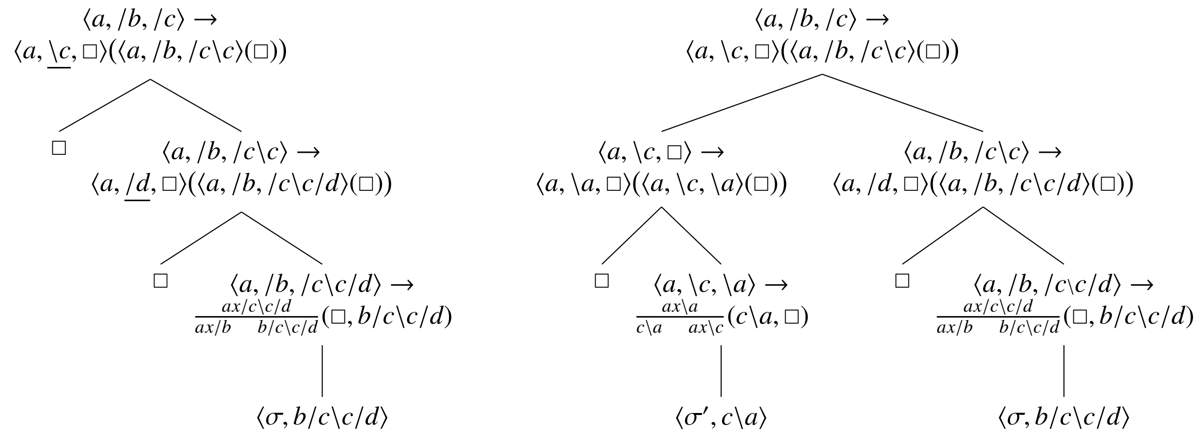

Figure 8: The left tree is a partial derivation tree of $\underline{\mathcal{D}}^{\langle a,/b,/c\rangle}_{G'}(\backslash c/d)$, in which we underlined the open arguments, whereas the right tree is not a partial derivation tree.

The previous lemma already shows that any derivation tree of G' for a nullary nonterminal of N_0 evaluates to a rule tree of G. This immediately proves that $\operatorname{supp}(\tau_{G'}) \subseteq \mathcal{R}_G$ since $\tau_{G'}$ is determined by the derivation trees for the nullary nonterminals of I. Additionally, we observe that G' assigns weights to productions based solely on the generated terminal of the production. All productions that do not generate any terminal are assigned weight 1. Additionally, the weight assigned to a terminal by G' is exactly the weight assigned to that terminal by G. By definition the weight assigned by G to a rule tree of G is simply the product of the weights assigned by G to the lexicon entries and rules (i.e., terminals) that constitute the rule tree. These observations make the following statement self-evident.

Lemma 17. $\mathrm{wt}_{G'}(d) = \mathrm{wt}_G(\mathrm{eval}_{G'}(d))$ *for every* $d \in \mathcal{D}^c_{G'}$ *with* $c \in N_0$.

To enable the proof of the converse we first introduce (special) partial derivation trees for our wsCFTG G'. The proof of this direction differs considerably from the one in the unweighted case, which, instead of utilizing partial derivation trees, decomposes the rule tree into spines that are derived separately and then assembled to obtain the complete rule tree. The approach employed here accounts for the slightly different construction of G' while facilitating the proof of the existence of a *unique* (partial) derivation tree for each rule tree. Recall that the productions P form a ranked alphabet, so we can consider trees of $T_{P_4 \cup P_5 \cup P_6 \cup P_7,\, P_2 \cup P_3}(P_1 \cup \{\square\})$ of productions together with $\square$ as a nullary symbol. The *partial derivation trees* of G' that start in nonterminal $n \in N$ and have open arguments $\beta \in \mathcal{A}_L(A)$, denoted by $\underline{\mathcal{D}}^n_{G'}(\beta)$, are inductively defined for all $n \in N$ and $\beta \in \mathcal{A}_L(A)$ to be the smallest sets $(\underline{\mathcal{D}}^n_{G'}(\beta))_{n \in N}$ such that

- $\mathcal{D}^n_{G'} \subseteq \underline{\mathcal{D}}^n_{G'}(\square)$,
- $p_7(\square, d_2) \in \underline{\mathcal{D}}^n_{G'}(|c\beta)$ for every $p_7 = n \to \langle a, |c, \square\rangle(n_2(\square)) \in P_7$ of type (7), $| \in \{/, \backslash\}$, $c \in C_L(A)$, and $d_2 \in \underline{\mathcal{D}}^{n_2}_{G'}(\beta)$,
- $p_6(d_1, d_2) \in \underline{\mathcal{D}}^n_{G'}(\beta)$ for every $p_6 = n \to n_1(n_2) \in P_6$ of type (6), $d_1 \in \underline{\mathcal{D}}^{n_1}_{G'}(\beta)$, and $d_2 \in \mathcal{D}^{n_2}_{G'}$, and
- $p_7(d_1, d_2) \in \underline{\mathcal{D}}^n_{G'}(\beta)$ for every $p_7 = n \to n_1(n_2(\square)) \in P_7$ of type (7), $d_1 \in \underline{\mathcal{D}}^{n_1}_{G'}(\beta)$, and $d_2 \in \mathcal{D}^{n_2}_{G'}$.

It is obvious from the definition of $\underline{\mathcal{D}}^n_{G'}(\beta)$ that open positions, indicated by $\square$, can only occur for nonterminals of the form $\langle a, |c, \square\rangle$. Moreover, they occur left-most in the sense that an open position forces all occurrences of nonterminals of N_1 to the left of it (but not ancestors of it) to be open as well. It is a routine matter to verify that $\underline{\mathcal{D}}^n_{G'}(\square) = \mathcal{D}^n_{G'}$ for all $n \in N$. Consequently, we call a partial derivation tree $d \in \underline{\mathcal{D}}^n_{G'}(\beta)$ *complete* if $\beta = \square$ and *incomplete* otherwise. We note that for every incomplete $d \in \underline{\mathcal{D}}^n_{G'}(\beta)$ we have (i) $d(\varepsilon) \in P_6$ if $n \in N_0$ and (ii) $d(\varepsilon) \in P_7$ if $n \in N_1$. We illustrate a partial derivation tree in Figure 8.

Next, we extend our evaluation to the partial derivation trees of $\underline{\mathcal{D}}_{G'} = \bigcup_{n \in N, \beta \in \mathcal{A}_L(A)} \underline{\mathcal{D}}^n_{G'}(\beta)$. Similar to standard derivation trees our partial derivation trees also evaluate to terminal trees or contexts via the map

$$\mathrm{eval}'_{G'} \colon \underline{\mathcal{D}}_{G'} \to \mathrm{T} \cup C_{R,\emptyset}(\operatorname{supp}(L)) \enspace ,$$

which is defined for every $d \in \mathcal{D}_{G'}$, $p_6 = n \to n_1(n_2) \in P_6$, $p_7 = n \to n_1(n_2(\square)) \in P_7$, $d'_1 \in \underline{\mathcal{D}}^{n_1}_{G'}(\beta)$, $d_2 \in \mathcal{D}^{n_2}_{G'}$, and $d'_2 \in \underline{\mathcal{D}}^{n_2}_{G'}(\beta)$ by

$$\mathrm{eval}'_{G'}(d) = \mathrm{eval}_{G'}(d) \qquad \mathrm{eval}'_{G'}(p_7(\square, d'_2)) = \mathrm{eval}'_{G'}(d'_2)$$
$$\mathrm{eval}'_{G'}(p_6(d'_1, d_2)) = \mathrm{eval}'_{G'}(d'_1)[\mathrm{eval}_{G'}(d_2)] \qquad \mathrm{eval}'_{G'}(p_7(d'_1, d_2)) = \mathrm{eval}'_{G'}(d'_1)[\mathrm{eval}_{G'}(d_2)] \enspace .$$

In other words, the evaluation is performed normally ignoring the open positions completely. We simply write $\mathrm{eval}_{G'}$ instead of $\mathrm{eval}'_{G'}$, which should not lead to confusion since we have $\mathrm{eval}'_{G'}(d) = \mathrm{eval}_{G'}(d)$ for all complete derivation trees $d \in \mathcal{D}_{G'}$. Note that partial derivation trees of G' still generate complete rule trees of G, since each open position represents a context that gets "wrapped around" the rule tree of the sibling tree, removing the last argument of its root category. This is ensured in the definition of partial derivation trees by restricting which positions can be open. Next, we show that straightforward variants of the two main properties (c1) and (c2) in the proof of Lemma 16 also hold for partial derivation trees.

Lemma 18. *Let* $\mathrm{eval} = \mathrm{eval}_{G'}$; eval_G. *For every nonterminal $n \in N$, $\beta \in \mathcal{A}_L(A)$, and derivation tree $d \in \underline{\mathcal{D}}^n_{G'}(\beta)$*
(d1) *if $n \in N_0$, then* $\mathrm{eval}(d) = n\beta$*, and*
(d2) *if $n = \langle a, |c, \gamma\rangle \in N_1$, then* $\mathrm{eval}_G(\mathrm{eval}_{G'}(d)[t]) = a\alpha\gamma\beta$ *for every $t \in \mathrm{T}$ with* $\mathrm{eval}_G(t) = a\alpha|c$ *for some $\alpha \in \mathcal{A}_L(A)$.*

Proof. We distinguish four cases corresponding to the definition of $\underline{\mathcal{D}}^n_{G'}(\beta)$.
(1) If $d \in \mathcal{D}^n_{G'}$ (i.e., $\beta = \square$), then statements (d1) and (d2) hold by the corresponding statements (c1) and (c2) in the proof of Lemma 16.
(2) If $d = p_6(d_1, d_2)$ for some $p_6 = a\alpha\gamma \to \langle a, |c, \gamma\rangle(a\alpha|c) \in P_6$ of type (6), $d_1 \in \underline{\mathcal{D}}^{\langle a,|c,\gamma\rangle}_{G'}(\beta)$, and $d_2 \in \mathcal{D}^{a\alpha|c}_{G'}$, then $n = a\alpha\gamma$ and $\mathrm{eval}_{G'}(d) = \mathrm{eval}_{G'}(d_1)[\mathrm{eval}_{G'}(d_2)]$. Let $t = \mathrm{eval}_{G'}(d_2)$, which is suitable because $\mathrm{eval}(d_2) = a\alpha|c$ by (c1) in the proof of Lemma 16. Hence by the induction hypothesis applied to d_1 with t, we obtain

$$\mathrm{eval}_G(\mathrm{eval}_{G'}(d)) = \mathrm{eval}_G(\mathrm{eval}_{G'}(d_1)[t]) = a\alpha\gamma\beta$$

as required to prove statement (d1).
(3) If $d = p_7(\square, d_2)$ for some $p_7 = \langle a, |c, \gamma\rangle \to \langle a, |'c', \square\rangle(\langle a, |c, \gamma|'c'\rangle(\square)) \in P_7$ of type (7) and $d_2 \in \underline{\mathcal{D}}^{\langle a,|c,\gamma|'c'\rangle}_{G'}(\beta_2)$, then $n = \langle a, |c, \gamma\rangle$, $\beta = |'c'\beta_2$, and $\mathrm{eval}_{G'}(d) = \mathrm{eval}_{G'}(d_2)$. We need to prove statement (d2), so let $t \in \mathrm{T}$ be such that $\mathrm{eval}_G(t) = a\alpha|c$ for some $\alpha \in \mathcal{A}_L(A)$. By the induction hypothesis applied to d_2 with t, we obtain the desired

$$\mathrm{eval}_G(\mathrm{eval}_{G'}(d)[t]) = \mathrm{eval}_G(\mathrm{eval}_{G'}(d_2)[t]) = a\alpha\gamma|'c'\beta_2 = a\alpha\gamma\beta \enspace .$$

(4) If $d = p_7(d_1, d_2)$ for some $p_7 = \langle a, |c, \gamma\rangle \to \langle a, |'c', \square\rangle(\langle a, |c, \gamma|'c'\rangle(\square)) \in P_7$ of type (7), $d_1 \in \underline{\mathcal{D}}^{\langle a,|'c',\square\rangle}_{G'}(\beta)$, and $d_2 \in \mathcal{D}^{\langle a,|c,\gamma|'c'\rangle}_{G'}$, then $n = \langle a, |c, \gamma\rangle$. Moreover, $\mathrm{eval}_{G'}(d) = \mathrm{eval}_{G'}(d_1)[\mathrm{eval}_{G'}(d_2)]$. We need to prove statement (d2), so let $t \in \mathrm{T}$ be such that $\mathrm{eval}_G(t) = a\alpha|c$ for some $\alpha \in \mathcal{A}_L(A)$. By (c2) in the proof of Lemma 16 applied to d_2 with t, we obtain $\mathrm{eval}_G(\mathrm{eval}_{G'}(d_2)[t]) = a\alpha\gamma|'c'$. Now we can apply the induction hypothesis to d_1 with $t' = \mathrm{eval}_{G'}(d_2)[t]$ to obtain the required equality

$$\mathrm{eval}_G(\mathrm{eval}_{G'}(d)[t]) = \mathrm{eval}_G(\mathrm{eval}_{G'}(d_1)[t']) = a\alpha\gamma\beta \enspace . \qquad \square$$

Let $t \in \mathrm{T}$, $n \in N_0$, and $\beta \in \mathcal{A}_L(A)$. Moreover, let

$$\underline{\mathcal{D}}^n_{G'}(t, \beta) = \{d \in \underline{\mathcal{D}}^n_{G'}(\beta) \mid \mathrm{eval}_{G'}(d) = t\} \enspace .$$

Lemma 18 yields the nice property $\underline{\mathcal{D}}^n_{G'}(t, \beta) = \emptyset$ unless $\mathrm{eval}_G(t) = n\beta$. Moreover, if $\beta \neq \square$ and $n \in C_L(A, k-1)$, then $\underline{\mathcal{D}}^n_{G'}(t, \beta) = \emptyset$ because every incomplete derivation tree $d \in \underline{\mathcal{D}}^n_{G'}(\beta)$ with $n \in N_0$ utilizes a production of P_6 at the root as we already remarked, which yields $n = a\alpha\gamma$ with $|\alpha\gamma| = k$ and thus $n \notin C_L(A, k-1)$. Hence if the category $n\beta$ has arity $k' \geq k$, then $\underline{\mathcal{D}}_{G'}(t, \beta) = \emptyset$ unless n has arity k. This motivates the following definition. For every $c \in C_L(A)$ let $\mathrm{nonterm}(c) = c$ if $c \in N_0$ and $\mathrm{nonterm}(c) = c'$ otherwise, where c' is the unique category of arity k such that $c = c'\beta$ for some $\beta \in \mathcal{A}_L(A)$. In other words, $\mathrm{nonterm}(c)$ is either directly the category c if c is "short" (i.e., c has arity at most k) or the prefix of c of arity k. Given a rule tree $t \in \mathcal{R}^c_{G'}$ we know that the only partial derivation trees that may evaluate to t are to be found in $\underline{\mathcal{D}}^n_{G'}(\beta)$ with $n = \mathrm{nonterm}(c)$ and $n\beta = c$. We use this property without explicit mention in the following.

Finally, we define substitution of a partial derivation tree into another partial derivation tree by replacing the right-most open position; i.e., right-most occurrence of $\square$. Formally, for every $d \in \underline{\mathcal{D}}^n_{G'}(\beta|c)$ with $n \in N_0$ and $d' \in \underline{\mathcal{D}}^{n'}_{G'}(\beta')$ with $n' = \langle \mathrm{target}(n), |c, \square\rangle$ we define $d[d'] \in \underline{\mathcal{D}}^n_{G'}(\beta\beta')$ by

- $p(\square, d_2)[d'] = p(d', d_2)$ for every $p = n \to n'(n_2(\square)) \in P_7$ of type (7), and $d_2 \in \mathcal{D}_{G'}^{n_2}$,
- $p(\square, d_2)[d'] = p(\square, d_2[d'])$ for every $p = n \to n_1(n_2(\square)) \in P_7$ of type (7), and $d_2 \in \underline{\mathcal{D}}_{G'}^{n_2}(\beta_2)$ with $\beta_2 \neq \square$,
- $p(d_1, d_2)[d'] = p(d_1[d'], d_2)$ for every $p = n \to n_1(n_2) \in P_6$ of type (6), $d_1 \in \underline{\mathcal{D}}_{G'}^{n_1}(\beta_1)$ with $\beta_1 \neq \square$, and $d_2 \in \mathcal{D}_{G'}^{n_2}$, and
- $p(d_1, d_2)[d'] = p(d_1[d'], d_2)$ for every $p = n \to n_1(n_2(\square)) \in P_7$ of type (7), $d_1 \in \underline{\mathcal{D}}_{G'}^{n_1}(\beta_1)$, and $d_2 \in \mathcal{D}_{G'}^{n_2}$.

This overloads the current substitution into contexts, but replacing the right-most open position is the only option to obtain another partial derivation tree. As mentioned before, this ensures that, when evaluated to rule trees, the context corresponding to the inserted partial derivation tree properly wraps around the rule tree corresponding to the other partial derivation tree. Thus, we obtain the following useful property.

Lemma 19. $\mathrm{eval}_{G'}(d[d']) = \mathrm{eval}_{G'}(d')[\mathrm{eval}_{G'}(d)]$ *for every partial derivation* $d \in \underline{\mathcal{D}}_{G'}^{n}(\beta|c)$ *with* $n \in N_0$ *and partial derivation* $d' \in \underline{\mathcal{D}}_{G'}^{n'}(\beta')$ *with* $n' = \langle \mathrm{target}(n), |c, \square\rangle$.

Proof. We prove the statement by induction on the size of d and case distinction.

- If $d = p(\square, d_2)$ for some $p = n \to n'(n_2(\square)) \in P_7$ of type (7) and $d_2 \in \mathcal{D}_{G'}^{n_2}$, then

$$\mathrm{eval}_{G'}(d[d']) = \mathrm{eval}_{G'}\big(p(d', d_2)\big) = \mathrm{eval}_{G'}(d')[\mathrm{eval}_{G'}(d_2)] = \mathrm{eval}_{G'}(d')[\mathrm{eval}_{G'}(d)] \enspace .$$

- If $d = p(\square, d_2)$ for some $p = n \to n_1(n_2(\square)) \in P_7$ of type (7) and $d_2 \in \underline{\mathcal{D}}_{G'}^{n_2}(\beta_2)$ with $\beta_2 \neq \square$, then

$$\mathrm{eval}_{G'}(d[d']) = \mathrm{eval}_{G'}\big(p(\square, d_2[d'])\big) = \mathrm{eval}_{G'}(d_2[d']) \overset{\mathrm{IH}}{=} \mathrm{eval}_{G'}(d')[\mathrm{eval}_{G'}(d_2)] = \mathrm{eval}_{G'}(d')[\mathrm{eval}_{G'}(d)] \enspace .$$

- If $d = p(d_1, d_2)$ for some $p = n \to n_1(n_2) \in P_6$ of type (6), $d_1 \in \underline{\mathcal{D}}_{G'}^{n_1}(\beta_1)$ with $\beta_1 \neq \square$, and $d_2 \in \mathcal{D}_{G'}^{n_2}$, then

$$\begin{aligned}\mathrm{eval}_{G'}(d[d']) &= \mathrm{eval}_{G'}\big(p(d_1[d'], d_2)\big) = \mathrm{eval}_{G'}(d_1[d'])[\mathrm{eval}_{G'}(d_2)] \\ &\overset{\mathrm{IH}}{=} \Big(\mathrm{eval}_{G'}(d')[\mathrm{eval}_{G'}(d_1)]\Big)[\mathrm{eval}_{G'}(d_2)] = \mathrm{eval}_{G'}(d')\Big[\mathrm{eval}_{G'}(d_1)[\mathrm{eval}_{G'}(d_2)]\Big] \\ &= \mathrm{eval}_{G'}(d')[\mathrm{eval}_{G'}(d)] \enspace .\end{aligned}$$

- If $d = p(d_1, d_2)$ for some $p = n \to n_1(n_2(\square)) \in P_7$ of type (7), $d_1 \in \underline{\mathcal{D}}_{G'}^{n_1}(\beta_1)$, and $d_2 \in \mathcal{D}_{G'}^{n_2}$, then

$$\begin{aligned}\mathrm{eval}_{G'}(d[d']) &= \mathrm{eval}_{G'}\big(p(d_1[d'], d_2)\big) = \mathrm{eval}_{G'}(d_1[d'])[\mathrm{eval}_{G'}(d_2)] \\ &\overset{\mathrm{IH}}{=} \Big(\mathrm{eval}_{G'}(d')[\mathrm{eval}_{G'}(d_1)]\Big)[\mathrm{eval}_{G'}(d_2)] = \mathrm{eval}_{G'}(d')\Big[\mathrm{eval}_{G'}(d_1)[\mathrm{eval}_{G'}(d_2)]\Big] \\ &= \mathrm{eval}_{G'}(d')[\mathrm{eval}_{G'}(d)] \enspace .\end{aligned}$$

□

We now have the ingredients for the converse of Lemma 16, namely we prove that for every rule tree $t \in \mathcal{R}_G^c$ there exists a unique partial derivation tree that evaluates to t. If $c \in N_0$, then the unique derivation tree is complete. Before we prove this final lemma, let us illustrate the construction. Suppose that the rule tree t displayed top left in Figure 9 belongs to $\mathcal{R}_G$ and its corresponding derivation tree d displayed top right in Figure 9 belongs to $\mathcal{D}_G$. Moreover, suppose that $k = 2$. Let us construct the corresponding partial derivation tree of G' step-by-step.

The derivation tree of G' for a leaf $\langle \sigma, c\rangle$ is obviously $c \to \langle \sigma, c\rangle$. Next, we investigate the subtree $t|_{12}$, which has root label $s = \frac{ax/c\backslash c}{ax/b \quad b/c\backslash c}$ and $\mathrm{eval}_G(t|_{12}) = a/b/c\backslash c$ as indicated in the top right derivation tree of Figure 9. Since we have $\mathrm{eval}_G(t|_{12}) \notin N_0$, we expect to find the derivation tree in $\underline{\mathcal{D}}_{G'}^{a/b/c}(\backslash c)$. Clearly it has to start with the production $a/b/c \to \langle a, /b, /c\rangle(a/b/b) \in P_6$ of type (6), so that we can attach the unique derivation tree for subtree $t|_{121}$ that delivers the primary category $\mathrm{eval}_G(t|_{121}) = a/b/b$ for s. The two nonterminals $a/b/c \in N_0$ and $a/b/b \in N_0$ uniquely determine $\langle a, /b, /c\rangle \in N_1$. Next, we need to generate the root label s, which will also allow us to attach the derivation tree for subtree $t|_{122}$ that delivers the secondary category $\mathrm{eval}_G(t|_{122}) = b/c\backslash c$ for s. Since we need to generate the root label from nonterminal $\langle a, /b, /c\rangle$ we need to utilize a production of type (2). Additionally, we may only generate a single open position (and no additional terminals), so we need to extend the third component $/c$ of nonterminal $\langle a, /b, /c\rangle$ in a single step to the full arguments of $\mathrm{eval}_G(t|_{122})$; i.e., $/c\backslash c$. This is achieved with the production $\langle a, /b, /c\rangle \to \langle a, \backslash c, \square\rangle(\langle a, /b, /c\backslash c\rangle(\square)) \in P_7$ of type (7). Clearly, the nonterminal $\langle a, \backslash c, \square\rangle$ remains open (and has the right argument for the single open position), and the nonterminal $\langle a, /b, /c\backslash c\rangle$ now allows the

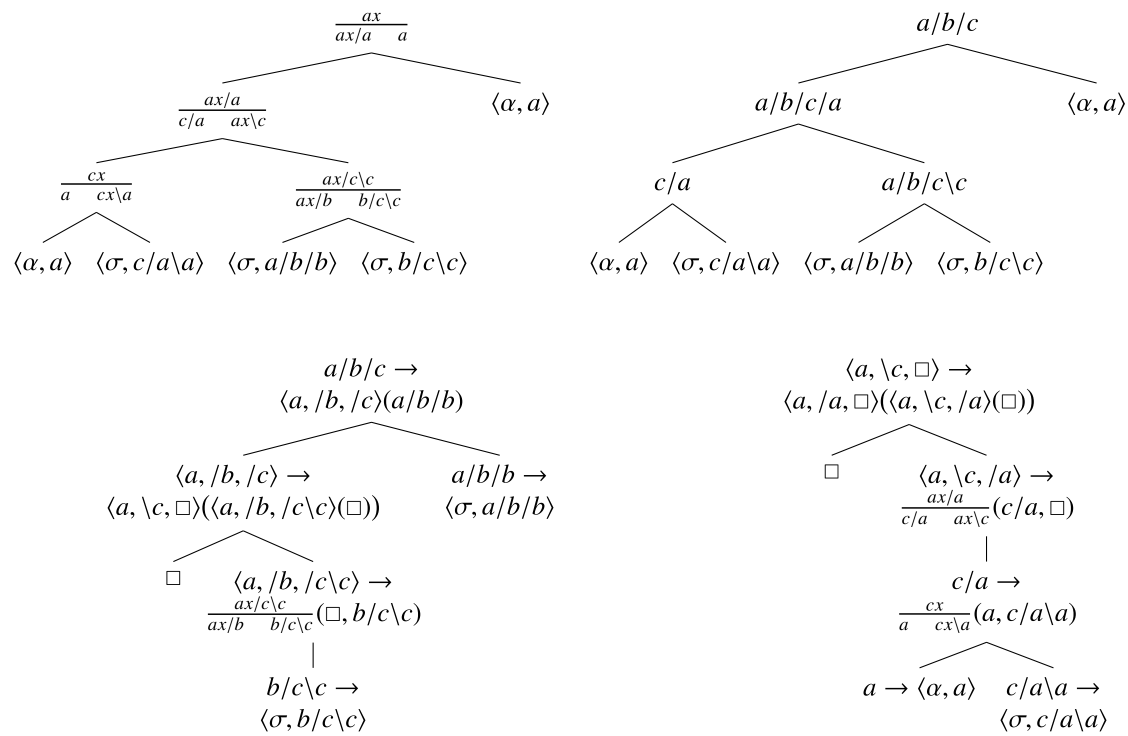

Figure 9: Example rule tree t (top left) of a wCCG G, its corresponding derivation tree of G (top right), the derivation tree of G' corresponding to $t|_{12}$ (bottom left), and the insert into that derivation tree that yields the derivation tree of G' corresponding to $t|_1$ (bottom right).

production $\langle a, /b, /c\backslash c \rangle \to s(\square, b/c\backslash c) \in P_2$ of type (2). This successfully generates the root terminal s and allows the attachment of the derivation tree for $t|_{122}$. The resulting derivation tree of G' is displayed bottom left in Figure 9.

To derive the unique derivation tree of G' for $t|_1$ we now utilize the only derivation tree of G' for $t|_{12}$ and insert additional material at the top via the open position. The root label of $t|_1$ is $s' = \frac{ax/a}{c/a \quad ax\backslash c}$, so with the same rationale as above, we extend the nonterminal $\langle a, \backslash c, \square \rangle$ corresponding to the open position in a single step to the full arguments $/a$ of the secondary category for s'. This is achieved by the production $\langle a, \backslash c, \square \rangle \to \langle a, /a, \square \rangle(\langle a, \backslash c, /a \rangle(\square)) \in P_7$ of type (7). The first nonterminal in that production remains open and the second nonterminal generates s' via the production $\langle a, \backslash c, /a \rangle \to s'(c/a, \square) \in P_2$ of type (2). The unique derivation tree of G' for $t|_{11}$ clearly attaches to the nonterminal c/a thus generated. The complete insert is also displayed bottom right in Figure 9.

Lemma 20. *For every category $c \in C_L(A)$ and rule tree $t \in \mathcal{R}^c_G$ there exists a unique partial derivation tree $d \in \underline{\mathcal{D}}^n_{G'}(t, \beta)$, where $n = \text{nonterm}(c)$ and $\beta \in \mathcal{A}_L(A)$ is such that $c = n\beta$.*

Proof. We prove the statement by induction on $t \in \mathcal{R}^c_G$. In the induction base, we have $t = \langle \sigma, c \rangle \in \text{supp}(L)$ and thus $n = \text{nonterm}(c) = c \in N_0$ and $\beta = \square$. Let $d = n \to \langle \sigma, n \rangle \in \mathcal{D}^n_{G'}$, for which we observe that $\text{eval}_{G'}(d) = \langle \sigma, c \rangle$. Hence $d \in \underline{\mathcal{D}}^n_{G'}(t, \beta)$ as desired. Obviously, the chosen d is unique because productions of types (4) and (5) generate terminal symbols of R and productions of type (6) generate a nonterminal of the form $\langle a, |'c', \gamma \rangle$ with $\gamma \neq \square$ that will eventually generate at least one terminal symbol of R. This is because only nonterminals of the form $\langle a, |'c', \square \rangle$ may remain open. Obviously, the production of type (1) that generates $\langle \sigma, c \rangle$ is unique.

In the induction step, let $t = s(t_1, t_2)$ with $s = \frac{a\alpha\gamma}{a\alpha/c' \quad c'\gamma} \in R$, $t_1 \in \mathcal{R}^{c_1}_G$, and $t_2 \in \mathcal{R}^{c_2}_G$, where $c = a\alpha\gamma$, $c_1 = a\alpha/c'$, and $c_2 = c'\gamma$ for some $\alpha \in \mathcal{A}_L(A)$. By the induction hypothesis applied to both t_1 and t_2 there exist unique $d_1 \in \underline{\mathcal{D}}^{n_1}_{G'}(t_1, \beta_1)$ and $d_2 \in \underline{\mathcal{D}}^{n_2}_{G'}(t_2, \beta_2)$, where $n_1 = \text{nonterm}(c_1)$, $n_2 = \text{nonterm}(c_2)$, and $\beta_1, \beta_2 \in \mathcal{A}(A)$ such that $c_1 = n_1\beta_1$ and $c_2 = n_2\beta_2$. Obviously, $c_2 \in N_0$, so $n_2 = \text{nonterm}(c_2) = c_2$ and $\beta_2 = \square$. Now we distinguish five cases.

(i) If $c, c_1 \in N_0$ with $|\alpha\gamma| < k$, then $n = \text{nonterm}(c) = c = a\alpha\gamma$, $n_1 = \text{nonterm}(c_1) = c_1 = a\alpha/c'$, and $\beta = \beta_1 = \square$.

We construct the derivation tree $d = p(d_1, d_2)$ with $p = n \to s(n_1, n_2) \in P_4$. Clearly,

$$\mathrm{eval}_{G'}(d) = s(\mathrm{eval}_{G'}(d_1), \mathrm{eval}_{G'}(d_2)) = s(t_1, t_2) = t \enspace ,$$

which proves that $d \in \underline{\mathcal{D}}^n_{G'}(t, \beta)$. Obviously only productions of type (4) are applicable to n and thus the production must generate s. Productions of type (6) are not applicable because $n = c \in C_L(A, k-1)$. Moreover, the terminal s and the category n uniquely determine $p \in P_4$. The subderivations d_1 and d_2 are unique by the induction hypothesis.

(ii) If $c, c_1 \in N_0$ with $|\alpha\gamma| = k$, then $n = \mathrm{nonterm}(c) = c = a\alpha\gamma$, $n_1 = \mathrm{nonterm}(c_1) = c_1 = a\alpha/c'$, and $\beta = \beta_1 = \square$. We construct the derivation tree $d = p(p'(d_2), d_1)$ with $p = a\alpha\gamma \to \langle a, /c', \gamma\rangle(a\alpha/c') \in P_6$ of type (6) and $p' = \langle a, /c', \gamma\rangle \to s(\square, c'\gamma) \in P_2$ of type (2). Clearly,

$$\mathrm{eval}_{G'}(d) = \mathrm{eval}_{G'}(p'(d_2))[\mathrm{eval}_{G'}(d_1)] = s(\mathrm{eval}_{G'}(d_1), \mathrm{eval}_{G'}(d_2)) = s(t_1, t_2) = t \enspace ,$$

which proves $d \in \underline{\mathcal{D}}^n_{G'}(t, \beta)$ as desired. Obviously only productions of type (6) are applicable to n since productions of type (1) generate the wrong terminal. Note that $|\alpha\gamma| = k$ and $|\alpha/c'| \le k$ yield $|\alpha| \le k - 1$ and thus $\gamma \ne \square$. The categories c and c_2 uniquely determine the initial production p, which generates the nonterminal $n' = \langle a, /c', \gamma\rangle$. Because $\gamma \ne \square$, it generates at least one terminal symbol and thus must generate s and only s. Clearly, n' uniquely determines the production p' of type (2) needed to generate s.

(iii) If $c_1 \in N_0$, but $c \notin N_0$, then $n = \mathrm{nonterm}(c) = a\alpha\gamma'$, $n_1 = \mathrm{nonterm}(c_1) = c_1 = a\alpha/c$, and $\beta_1 = \square$, where $\gamma = \gamma'\beta$. Let $\beta = |_1 c'_1 \cdots |_\ell c'_\ell$ for some $\ell \in \mathbb{N}_+$, $|_i \in \{/, \backslash\}$ and $c'_i \in \mathrm{argcats}(L)$ for all $i \in [\ell]$. Additionally, let $\gamma'_i = \gamma' |_1 c'_1 \cdots |_i c'_i$ for all $0 \le i \le \ell$. We construct the partial derivation tree

$$d = p\Big(p_1(\square, \cdots (p_\ell(\square, p'(d_2))) \cdots), d_1\Big)$$

with $p = a\alpha\gamma' \to \langle a, /c', \gamma'\rangle(a\alpha/c') \in P_6$ of type (6), $p_i = \langle a, /c', \gamma'_{i-1}\rangle \to \langle a, |_i c'_i, \square\rangle(\langle a, /c', \gamma'_i\rangle(\square)) \in P_7$ of type (7) for every $i \in [\ell]$, and $p' = \langle a, /c', \gamma\rangle \to s(\square, c'\gamma) \in P_2$ of type (2). Note that $\gamma'_\ell = \gamma$. Clearly,

$$\begin{aligned}\mathrm{eval}_{G'}(d) &= \mathrm{eval}_{G'}\Big(p_1(\square, \cdots (p_\ell(\square, p'(d_2))) \cdots)\Big)[\mathrm{eval}_{G'}(d_1)] = \mathrm{eval}_{G'}(p'(d_2))[t_1] \\ &= s(t_1, \mathrm{eval}_{G'}(d_2)) = s(t_1, t_2) = t \enspace ,\end{aligned}$$

which proves that $d \in \underline{\mathcal{D}}^n_{G'}(t, \beta)$ as desired. Obviously we need to start with a production of type (6) and it is uniquely determined by n and n_2. It generates the nonterminal $n' = \langle a, /c', \gamma'\rangle$, which will generate at least one terminal symbol since $\gamma' \ne \square$. Hence it needs to generate s, which forces the chain of productions of type (7). Each is uniquely determined by the next argument on the path from γ' to γ in the third component. Additionally, the first arguments of all those productions need to remain open as we otherwise would eventually generate additional terminals. Finally, once we generated the nonterminal $\langle a, /c', \gamma\rangle$, the production that generates s is clearly p'.

(iv) If $c, c_1 \notin N_0$, then $n = \mathrm{nonterm}(c) = a\alpha' = \mathrm{nonterm}(c_1) = n_1$ for some $\alpha', \alpha'' \in \mathcal{A}_L(A)$ such that $\alpha = \alpha'\alpha''$. Moreover, $\beta = \alpha''\gamma$ and $\beta_1 = \alpha''/c'$. Let $\gamma = |_1 c'_1 \cdots |_\ell c'_\ell$ for some $\ell \in \mathbb{N}$, $|_i \in \{/, \backslash\}$ and $c'_i \in \mathrm{argcats}(L)$ for all $i \in [\ell]$. Additionally, let $\gamma_i = |_1 c'_1 \cdots |_i c'_i$ for all $0 \le i \le \ell$. We construct the partial derivation tree

$$d = d_1\Big[p_1(\square, \cdots (p_\ell(\square, p'(d_2))) \cdots)\Big]$$

with production $p_i = \langle a, /c', \gamma_{i-1}\rangle \to \langle a, |_i c'_i, \square\rangle(\langle a, /c', \gamma_i\rangle(\square)) \in P_7$ of type (7) for every $i \in [\ell]$ and production $p' = \langle a, /c', \gamma\rangle \to s(\square, c\gamma) \in P_2$ of type (2). Note that $\gamma_\ell = \gamma$. We verify

$$\begin{aligned}\mathrm{eval}_{G'}(d) &= \mathrm{eval}_{G'}\Big(p_1(\square, \cdots (p_\ell(\square, p'(d_2))) \cdots)\Big)[\mathrm{eval}_{G'}(d_1)] = \mathrm{eval}_{G'}(p'(d_2))[t_1] \\ &= s(t_1, \mathrm{eval}_{G'}(d_2)) = s(t_1, t_2) = t\end{aligned}$$

by Lemma 19, which proves that $d \in \underline{\mathcal{D}}^n_{G'}(t, \beta)$ as desired. Obviously, we need the derivation d_1 to generate the subtree t_1 and we can only extend it by replacing the innermost open argument as it otherwise is no longer a partial derivation tree. Additionally, we can only close a single open argument as each will generate a terminal symbol. Thus we extend the open argument for nonterminal $n' = \langle a, /c', \square\rangle$ with the goal to generate the terminal s. It is uniquely extended as described in the previous case from $\square$ all the way to γ in the third component using productions of type (7), which then allows us to use the uniquely determined rule of type (2) to generate s. Once again, all the first arguments of the productions of type (7) need to remain open to avoid generating additional terminals.

(v) If $c \in N_0$, but $c_1 \notin N_0$, then $n = \mathrm{nonterm}(c) = c = a\alpha\gamma$ and $n_1 = \mathrm{nonterm}(c_1) = a\alpha$. Additionally, $\gamma = \beta = \square$ and $\beta_1 = /c'$. We construct the derivation tree $d = d_1[p'(d_2)]$ with production $p' = \langle a, /c', \square\rangle \to s(\square, c') \in P_2$ of type (2). We verify

$$\mathrm{eval}_{G'}(d) = \mathrm{eval}_{G'}(p'(d_2))[\mathrm{eval}_{G'}(d_1)] = s(t_1, \mathrm{eval}_{G'}(d_2)) = s(t_1, t_2) = t$$

by Lemma 19, which proves that $d \in \underline{\mathcal{D}}^n_{G'}(t, \beta)$ as desired. As in the previous case, we need derivation d_1 to generate t_1 and we can only extend it at its single open argument for nonterminal $\langle a, /c', \square\rangle$, which we need to use to generate the terminal s. Clearly, no extension via productions of type (7) is possible since this would generate additional terminals, so we directly utilize the uniquely determined production p' of type (2) to generate s.

The induction step for root symbol $s = \frac{ax\gamma}{c'\gamma \quad ax\backslash c'} \in R$ is analogous. □

Theorem 21. $\tau_{G'} = \tau_G$

Proof. We claim that $\mathrm{eval}_{G'}$ is a weight-preserving bijection between $\mathcal{D}^n_{G'}$ and $\mathcal{R}^n_G$ for all $n \in N_0$. By Lemma 16 we have $\mathrm{eval}_{G'} \colon \mathcal{D}^n_{G'} \to \mathcal{R}^n_G$. Moreover, by Lemma 17 it is weight-preserving. Additionally, Lemma 20 shows that for every $t \in \mathcal{R}^n_G$ there exists a unique $d \in \mathcal{D}^n_{G'}$ such that $\mathrm{eval}_{G'}(d) = t$, which proves surjectivity and injectivity. Let $a \in I$, $t \in \mathcal{R}^a_G$, and $d \in \mathcal{D}^a_{G'}$ such that $\mathrm{eval}_{G'}(d) = t$. Then

$$\tau_G(t) = \mathrm{wt}_G(t) = \mathrm{wt}_G(\mathrm{eval}_{G'}(d)) = \mathrm{wt}_{G'}(d) = \sum_{n \in I, d \in \mathcal{D}^n_{G'}(t)} \mathrm{wt}_{G'}(d) = \tau_{G'}(t) \enspace .$$

Finally, let $t \in \mathrm{T}$ be such that $t \notin \bigcup_{a \in I} \mathcal{R}^a_G$. In this case we have

$$\tau_G(t) = 0 = \sum_{a \in I, d \in \mathcal{D}^a_{G'}(t)} \mathrm{wt}_{G'}(d) = \tau_{G'}(t)$$

because $\mathcal{D}^a_{G'}(t) = \emptyset$ for all $a \in I$ due to Lemma 16. □

Corollary 22. *Weighted forests generatable by wCCG can also be generated by wsCFTG.*

Acknowledgements

We thank the anonymous reviewers for carefully reading this article and for their detailed and helpful suggestions on how to improve it.

References

[1] M. Steedman, The Syntactic Process, MIT Press, 2000. `doi:10.7551/mitpress/6591.001.0001`.

[2] M. Steedman, J. Baldridge, Combinatory categorial grammar, in: R. D. Borsley, K. Börjars (Eds.), Non-Transformational Syntax: Formal and Explicit Models of Grammar, Blackwell, 2011, Ch. 5, pp. 181–224. `doi:10.1002/9781444395037.ch5`.

[3] H. B. Curry, R. Feys, W. Craig, Combinatory Logic, no. 22 in Studies in Logic and the Foundations of Mathematics, North-Holland, 1958. URL `https://www.sciencedirect.com/bookseries/studies-in-logic-and-the-foundations-of-mathematics/vol/22`

[4] Y. Bar-Hillel, H. Gaifman, E. Shamir, On categorial and phrase structure grammars, in: Y. Bar-Hillel (Ed.), Language and Information: Selected Essays on Their Theory and Application, Addison Wesley, 1964, pp. 99–115.

[5] K. Ajdukiewicz, Die syntaktische Konnexität, Studia Philosophica 1 (1935) 1–27.

[6] Y. Bar-Hillel, A quasi-arithmetical notation for syntactic description, Language 29 (1) (1953) 47–58. `doi:10.2307/410452`.

[7] A. K. Joshi, Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions?, in: D. R. Dowty, L. Karttunen, A. M. Zwicky (Eds.), Natual Language Parsing, Cambridge University Press, 1985, Ch. 6, pp. 206–250.

[8] J. Baldridge, Lexically specified derivational control in combinatory categorial grammar, Ph.D. thesis, University of Edinburgh (2002).
URL http://hdl.handle.net/1842/562
[9] M. Kuhlmann, A. Koller, G. Satta, Lexicalization and generative power in CCG, Computational Linguistics 41 (2) (2015) 187–219. doi:10.1162/COLI_a_00219.
[10] T. Kwiatkowksi, L. Zettlemoyer, S. Goldwater, M. Steedman, Inducing probabilistic CCG grammars from logical form with higher-order unification, in: Proc. 2010 Conf. Empirical Methods in Natural Language Processing, ACL, 2010, pp. 1223–1233.
URL https://aclanthology.org/D10-1119
[11] L. S. Zettlemoyer, M. Collins, Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars, arXiv preprint arXiv:1207.1420 (2012). doi:10.48550/arXiv.1207.1420.
[12] L. Zettlemoyer, M. Collins, Online learning of relaxed CCG grammars for parsing to logical form, in: Proc. 2007 Conf. Empirical Methods in Natural Language Processing, ACL, 2007, pp. 678–687.
URL https://aclanthology.org/D07-1071
[13] T. Kwiatkowski, S. Goldwater, L. Zettlemoyer, M. Steedman, A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings, in: Proc. 13th Conf. European Chapter of Association for Computational Linguistics, ACL, 2012, pp. 234–244.
URL https://aclanthology.org/E12-1024
[14] K. Vijay-Shanker, D. J. Weir, The equivalence of four extensions of context-free grammars, Mathematical Systems Theory 27 (6) (1994) 511–546. doi:10.1007/BF01191624.
[15] A. K. Joshi, Y. Schabes, Tree-adjoining grammars, in: Rozenberg and Salomaa [43], Ch. 2, pp. 69–123. doi:10.1007/978-3-642-59126-6_2.
[16] J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison Wesley, 1979.
[17] K. Vijay-Shanker, A study of tree adjoining grammars, Ph.D. thesis, University of Pennsylvania (1988).
[18] K. Vijay-Shanker, D. J. Weir, Combinatory categorial grammars: Generative power and relationship to linear context-free rewriting systems, in: Proc. 26th Ann. Meeting Association for Computational Linguistics, ACL, 1988, pp. 278–285. doi:10.3115/982023.982057.
[19] M. Kuhlmann, G. Satta, P. Jonsson, On the complexity of CCG parsing, Computational Linguistics 44 (3) (2018) 447–482. doi:10.1162/coli_a_00324.
[20] M. Kuhlmann, A. Koller, G. Satta, The importance of rule restrictions in CCG, in: Proc. 48th Ann. Meeting Association for Computational Linguistics, ACL, 2010, pp. 534–543.
URL https://aclanthology.org/P10-1055
[21] J. Hockenmaier, P. Young, Non-local scrambling: The equivalence of TAG and CCG revisited, in: Proc. 9th Int. Workshop Tree Adjoining Grammar and Related Formalisms, ACL, 2008, pp. 41–48.
URL https://aclanthology.org/W08-2306
[22] M. Kuhlmann, G. Satta, Tree-adjoining grammars are not closed under strong lexicalization, Computational Linguistics 38 (3) (2012) 617–629. doi:10.1162/COLI_a_00090.
[23] A. Koller, M. Kuhlmann, Dependency trees and the strong generative capacity of CCG, in: Proc. 12th Conf. European Chapter of Association for Computational Linguistics, ACL, 2009, pp. 460–468.
URL https://aclanthology.org/E09-1053
[24] F. Gécseg, M. Steinby, Tree languages, in: Rozenberg and Salomaa [43], Ch. 1, pp. 1–68. doi:10.1007/978-3-642-59126-6_1.
[25] H.-J. Tiede, Deductive systems and grammars: Proofs as grammatical structures, Ph.D. thesis, Indiana University (1999).
[26] J. Lambek, The mathematics of sentence structure, The American Mathematical Monthly 65 (3) (1958) 154–170. doi:10.1080/00029890.1958.11989160.
[27] M. Kuhlmann, A. Maletti, L. K. Schiffer, The tree-generative capacity of combinatory categorial grammars, in: Proc. 39th Foundations of Software Technology and Theoretical Computer Science, Vol. 150 of LIPIcs, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2019, pp. 44:1–44:14. doi:10.4230/LIPIcs.FSTTCS.2019.44.
[28] M. Kuhlmann, A. Maletti, L. K. Schiffer, The tree-generative capacity of combinatory categorial grammars, Journal of Computer and System Sciences 124 (2022) 214–233. doi:10.1016/j.jcss.2021.10.005.
[29] W. Buszkowski, Generative power of categorial grammars, in: R. T. Oehrle, E. Bach, D. Wheeler (Eds.), Categorial Grammars and Natural Language Structures, Vol. 32 of Studies in Linguistics and Philosophy, Springer, 1988, Ch. 4, pp. 69–94. doi:10.1007/978-94-015-6878-4_4.
[30] L. K. Schiffer, A. Maletti, Strong equivalence of TAG and CCG, Transactions of the Association for Computational Linguistics 9 (2021) 707–720. doi:10.1162/tacl_a_00393.
[31] S. Kepser, J. Rogers, The equivalence of tree adjoining grammars and monadic linear context-free tree grammars, Journal of Logic, Language and Information 20 (3) (2011) 361–384. doi:10.1007/s10849-011-9134-0.
[32] A. Fujiyoshi, T. Kasai, Spinal-formed context-free tree grammars, Theory of Computing Systems 33 (1) (2000) 59–83. doi:10.1007/s002249910004.
[33] U. Hebisch, H. J. Weinert, Semirings — Algebraic Theory and Applications in Computer Science, World Scientific, 1998. doi:10.1142/3903.
[34] J. S. Golan, Semirings and their Applications, Kluwer Academic Publishers, 1999. doi:10.1007/978-94-015-9333-5.
[35] M. Droste, W. Kuich, H. Vogler (Eds.), Handbook of Weighted Automata, EATCS Monographs in Theoretical Computer Science, Springer, 2009. doi:10.1007/978-3-642-01492-5.
[36] W. C. Rounds, Context-free grammars on trees, in: Proc. 1st Symposium on Theory of Computing, ACM, 1969, pp. 143–148. doi:10.1145/800169.805428.
[37] M. Teichmann, Expressing context-free tree languages by regular tree grammars, Ph.D. thesis, Technische Universität Dresden (2017).
URL https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-224756
[38] J. Engelfriet, A. Maletti, S. Maneth, Multiple context-free tree grammars: Lexicalization and characterization, Theoretical Computer Science 728 (2018) 29–99. doi:10.1016/j.tcs.2018.03.014.

[39] J. Berstel, C. Reutenauer, Recognizable formal power series on trees, Theoretical Computer Science 18 (2) (1982) 115–148. doi:10.1016/0304-3975(82)90019-6.
[40] Z. Fülöp, H. Vogler, Weighted tree automata and tree transducers, in: Droste et al. [35], Ch. 9, pp. 313–403. doi:10.1007/978-3-642-01492-5_9.
[41] Z. Fülöp, Local weighted tree languages, Acta Cybernetica 22 (2) (2015) 393–402. doi:10.14232/actacyb.22.2.2015.10.
[42] B. Borchardt, The theory of recognizable tree series, Ph.D. thesis, Technische Universität Dresden (2005).
[43] G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 3, Springer, 1997. doi:10.1007/978-3-642-59126-6.