Journal of Automata, Languages and Combinatorics **u** () w, x–y
© Institut für Informatik, Justus-Liebig-Universität Giessen

Non-Self-Embedding Linear Context-Free Tree Grammars Generate Regular Tree Languages

Mark-Jan Nederhof
School of Computer Science, University of St Andrews, North Haugh, KY16 9SX, UK
e-mail: markjan.nederhof@gmail.com

Markus Teichmann[1]
Department of Computer Science, Technische Universität Dresden, 01062 Dresden, Germany
e-mail: markus.teichmann@mailbox.tu-dresden.de

and

Heiko Vogler
Department of Computer Science, Technische Universität Dresden, 01062 Dresden, Germany
e-mail: heiko.vogler@tu-dresden.de

ABSTRACT

For the class of linear context-free tree grammars, we define a decidable property called self-embedding. We prove that each non-self-embedding grammar in this class generates a regular tree language and show how to construct the equivalent regular tree grammar.

Keywords: Context-Free Tree Grammar, Regular Tree Grammar, Self-Embedding, Natural Language Processing

1. Introduction

In natural language processing (NLP), formal string grammars are used to approximate the set of all syntactically valid sentences of a language. Two important and successful grammar classes are the regular grammars (REGs) and the context-free grammars (CFGs) [16]. For these two classes there is a clear trade-off between expressive power and cost of processing, e.g., for parsing. It is undecidable whether an arbitrary given CFG generates a regular language [13, Thm. 8.15], but one may approximate a given context-free language by a REG, for example, in order to achieve better parsing complexity [22]. Alternatively, one may restrict CFGs to satisfy a decidable property that guarantees that they generate regular languages. Chomsky [3] defined

[1] Financially supported by DFG Graduiertenkolleg 1763 (QuantLA).

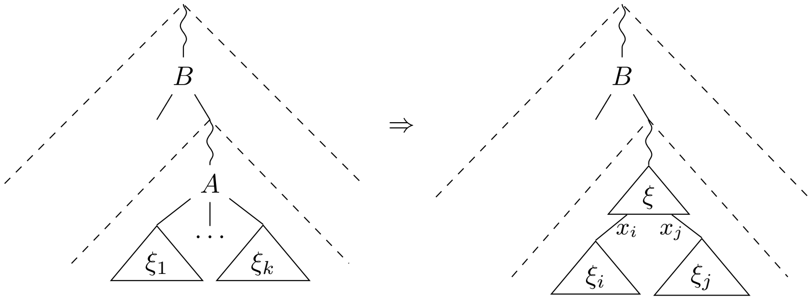


Figure 1:: An application of the rule $A(x_1, \ldots, x_k) \to \xi$.

such a property called non-self-embedding. A CFG is self-embedding if there are a nonterminal A and non-empty strings v and w over terminals and nonterminals such that $A \Rightarrow^* vAw$. He proved that each non-self-embedding CFG generates a regular language [3, Thm. 11]. In [22] self-embedding was expressed as a syntactic criterion, accompanied by a direct construction of a REG from a non-self-embedding CFG.

It was found that CFGs are inadequate to describe the syntax of all natural languages [27]. To remedy this, mildly context-sensitive formalisms were introduced [14], which can capture more linguistic phenomena than REGs and CFGs. Examples of such mildly context-sensitive formalisms are linear context-free rewriting systems [29] and macro grammars [7]. Related formalisms generate tree languages, in order to explicitly describe the internal structure of sentences. These grammars also generate string languages; a string is obtained as the yield (or frontier) of a generated tree. Examples of relevant tree-generating formalisms are tree adjoining grammars [15] and context-free tree grammars (CFTG) [26, 6, 12]. There is a close relationship between tree adjoining grammars and monadic linear CFTGs [10, 20, 11]. Monadic CFTGs have been investigated in [8, 9]. In [21] it was proved that CFTGs lexicalize tree adjoining grammars. Synchronous context-free tree grammars have been proposed and investigated as syntax-based translation models for natural languages [23, 24]. The class of linear nondeleting CFTGs is considered in [18, 17].

Context-free tree grammars generalize regular tree grammars (RTG) [1] by allowing nonterminals to have arguments (or: parameters), which contain trees over nonterminals and terminals. Thus, in a sentential form, nonterminals may occur nested as indicated in Figure 1, where A and B are nonterminals. This figure also illustrates the application of a rule $A(x_1, \ldots, x_k) \to \xi$, which proceeds as follows. The variables $x_1, \ldots, x_k$ are bound to the k subtrees $\xi_1, \ldots, \xi_k$, respectively. Let ξ' be the result of replacing all occurrences of these variables in ξ by the subtrees they are bound to. Then, A together with its subtrees is replaced by ξ'.

Much as for the string case, there is a trade-off between expressive power and processing cost when relating the classes of tree languages generated by CFTGs and RTGs. This motivates similar investigations as in the string case. In this paper we focus on the class of linear nondeleting CFTGs (lnCFTG). In each rule of a lnCFTG

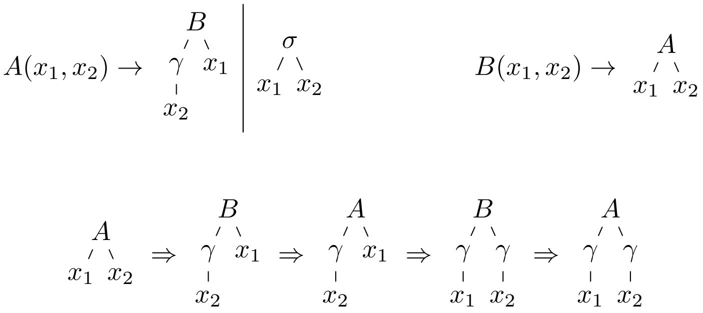


Figure 2:: Part of a lnCFTG G_1 and one of its derivations.

each variable from the left-hand side occurs exactly once in the right-hand side. We define the decidable property of self-embedding for lnCFTGs and as our main result we construct for each non-self-embedding lnCFTG an equivalent RTG, thus, in particular we show that each non-self-embedding lnCFTG induces a regular tree language. We will extend these results to linear CFTG in which variables may be deleted by a rule application. The extended results follow directly from the facts that (i) each linear CFTG can be transformed into an equivalent lnCFTG and (ii) this transformation preserves the property of being non-self-embedding.

The path language of (the tree language generated by) a lnCFTG is a context-free string language, while the path language of a RTG is a regular string language [5, Thm. 7.13]. It is therefore tempting to try to define the notion of self-embedding for lnCFTGs in terms of the familiar notion of self-embedding for CFGs, applied to path languages of lnCFTGs. However, there is an additional source of non-regularity in lnCFTG that cannot be captured solely in terms of path languages. To illustrate this we consider rules of the lnCFTG G_1 (cf. Figure 2) and one of its derivations. Note that the numbers of γ's in the two argument positions of A grow in a synchronized manner, so that the tree language generated from $A(x_1, x_2)$ is not regular, even though its path language is a regular string language. We say that a lnCFTG is self-embedding if at least one of the two properties illustrated in Figure 3 is satisfied: Property (1) generalizes self-embedding from the string case (applied to paths), while Property (2) captures potential non-regularity due to different branches growing in a synchronized manner. In Section 4 we will formally define the concept of self-embedding and show that this property can be decided for each lnCFTG.

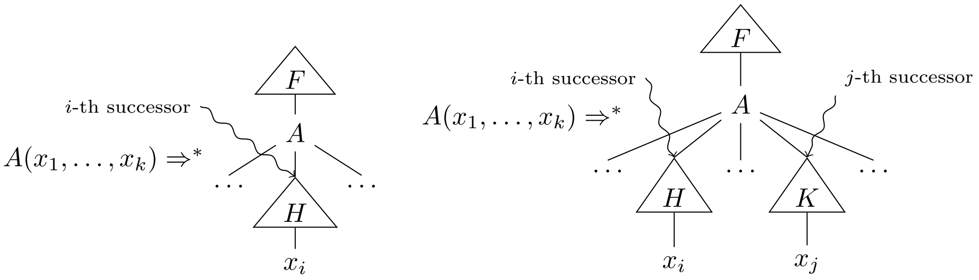
Our main result is the proof that a non-self-embedding lnCFTG generates a regular tree language. Our use of the term 'non-self-embedding' may already suggest this result, by analogy with the string case. However, due to the additional source of non-regularity (as described above), novel proof techniques are needed, which involve complications far beyond those of the string case. In the following, we describe the steps of the proof.

To simplify the proof, we transform G into an equivalent lnCFTG H that satisfies the novel property of being *unique in argument positions*. Roughly speaking, the effect of this transformation is that generation of symbols in distinct argument positions of one nonterminal of G is done through several newly introduced nonterminals in H. This transformation is possible because the negation of Property (2) guarantees that the generation of symbols in distinct argument positions of one and the same nonterminal is independent. In Section 5 we will formally define this property and provide the transformation with a proof of correctness.

Next, we analyze a non-self-embedding lnCFTG H which is unique in argument positions. We consider related nonterminals via analysis of a graph. We can see that, due to Property (1) of self-embedding, unboundedly many symbols can never be created synchronously above and below a nonterminal. Hence, we can divide the generation into two classes, namely top-recursion, which deals with unbounded generation below a nonterminal, and bottom-recursion, which deals with unbounded generation above a nonterminal.

Relying on the properties of non-self-embedding and uniqueness in argument positions, top-recursion can be transformed to bottom-recursion. This is by a construction described in detail in Section 6.1. Subsequently, we show that a non-self-embedding lnCFTG which does not contain any top-recursion can be transformed into an equivalent RTG. This relies on the observation that the number of distinct values that may appear below a nonterminal is bounded. This is explained in detail in Section 6.2.

In Section 6.3 we prove our main theorem and in Section 7 we relate the definition of self-embedding for trees to the one used in the string case [22]. In Section 8 we show that our result can be extended to linear CFTG which may be deleting. In Section 9, we define the notion of weakly-self-embedding CFTG, i.e., for the full class of context-free tree grammars. This notion is inspired by self-embedding indexed grammars [25]. Each self-embedding lCFTG is a weakly-self-embedding lCFTG. We prove that each non-weakly-self-embedding CFTG induces a regular tree language. Section 10 concludes with a summary of the subclasses of CFTG relevant to this paper, and the inclusion relations between them.



(1): F and H are non-trivial trees.

(2): H and K are non-trivial trees.

Figure 3:: Properties for self-embedding ($i, j \in \{1, \ldots, k\}$).

2. Preliminaries

Mathematical Notions. The *set of natural numbers* $\{0, 1, \ldots\}$ is denoted by $\mathbb{N}$ and $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$. The set of finite sequences over $\mathbb{N}_+$ is denoted by $\mathbb{N}_+^*$ (including the empty sequence). For $n \in \mathbb{N}$, we let $[n] = \{1, \ldots, n\}$; hence $[0] = \emptyset$. An alphabet is a non-empty finite set. The *set of words* over the alphabet Σ is denoted by Σ^* with ε being the *empty word*. Let U be a set. Then $\mathcal{P}(U)$ denotes the *powerset of U*.

We fix an infinite list $x_1, x_2, \ldots$ of pairwise distinct *variables*. We let $X = \{x_1, x_2, x_3, \ldots\}$ and $X_k = \{x_1, \ldots, x_k\}$. Furthermore, we abbreviate $x_1, \ldots, x_k$ to $x_{1..k}$. We apply this abbreviation also to sequences of other objects. Sometimes we will also use symbols different from $x_1, x_2, \ldots$ to denote variables, such as $z, z_1, z_2, \ldots$.

Trees. A *ranked alphabet* is a pair $(\Delta, \mathrm{rk}_\Delta)$, where Δ is an alphabet and $\mathrm{rk}_\Delta \colon \Delta \to \mathbb{N}$ is a function. For every $\delta \in \Delta$, we call $\mathrm{rk}_\Delta(\delta)$ the *rank* of δ. Sometimes, we write $\delta^{(k)}$ to indicate that δ has rank k. We abbreviate the set $\mathrm{rk}_\Delta^{-1}(k)$ to $\Delta^{(k)}$ and $(\Delta, \mathrm{rk}_\Delta)$ to Δ assuming that rk_Δ is the rank function. In this paper, Δ denotes an arbitrary ranked alphabet. We assume that $\Delta \cap X = \emptyset$.

Let U be a set. We denote the set of *trees over Δ and U* by $\mathrm{T}_\Delta(U)$ and write T_Δ for $\mathrm{T}_\Delta(\emptyset)$. Each subset of T_Δ is called a *tree language*. Positions in trees are identified by Gorn addresses, represented as finite sequences over $\mathbb{N}_+$ as usual. Formally, for each $\xi \in \mathrm{T}_\Delta(U)$, the *set of positions of ξ*, denoted by $\mathrm{pos}(\xi)$, is defined inductively as follows: (i) if $\xi \in \Delta^{(0)} \cup U$, then $\mathrm{pos}(\xi) = \{\varepsilon\}$, and (ii) if $\xi = \delta(\xi_1, \ldots, \xi_k)$ for some $\delta \in \Delta^{(k)}$, $k \geq 1$ and $\xi_1, \ldots, \xi_k \in \mathrm{T}_\Delta(U)$, then $\mathrm{pos}(\xi) = \{\varepsilon\} \cup \{iv \mid 1 \leq i \leq k, v \in \mathrm{pos}(\xi_i)\}$. For a position $w \in \mathrm{pos}(\xi)$, the *label of ξ at w* and the *subtree of ξ at w* are denoted by $\xi(w)$ and $\xi|_w$, respectively. For every $V \subseteq \Delta \cup U$, we denote the *set of positions of ξ labeled by an element of V* by $\mathrm{pos}_V(\xi)$; if V is a singleton $\{v\}$, then we simply write $\mathrm{pos}_v(\xi)$. For $W \subseteq \mathrm{pos}(\xi)$ and $w \in W$, we say that w is *outermost in W* if there is no $u \in W$ such that $w = uv$ for some $v \in \mathbb{N}_+^* \setminus \{\varepsilon\}$.

Let U be a finite set with $\Delta \cap U = \emptyset$. A *context over Δ and U* is a tree in $\mathrm{T}_\Delta(U)$ in which each element $u \in U$ occurs exactly once. The set of all such contexts is denoted by $\mathrm{C}_\Delta(U)$.

Let $\xi \in \mathrm{T}_\Delta(X)$, $i \in \mathbb{N}$, $x_i \in X$, and $w \in \mathrm{pos}(\xi)$. We say that w is *x_i-dominating* if $\xi|_w$ contains a position labeled x_i. We call w *variable dominating* if it is x_i-dominating for some $x_i \in X$.

Tree concatenation. Let $k \in \mathbb{N}$, let $u_{1..k} \in U \cup \Delta^{(0)}$ be pairwise distinct symbols, and let $\xi \in \mathrm{T}_\Delta(U)$ and $\xi_{1..k} \in \mathrm{T}_\Delta(X)$. We define the *tree concatenation of ξ with $\xi_{1..k}$ at $u_{1..k}$*, denoted by $\xi[u_1/\xi_1, \ldots, u_k/\xi_k]$, inductively on the structure of ξ as follows:

(i) $u_i[u_1/\xi_1, \ldots, u_k/\xi_k] = \xi_i$ and

(ii) $\delta(\zeta_1, \ldots, \zeta_\ell)[u_1/\xi_1, \ldots, u_k/\xi_k] = \delta(\zeta_1[u_1/\xi_1, \ldots, u_k/\xi_k], \ldots, \zeta_\ell[u_1/\xi_1, \ldots, u_k/\xi_k])$ for each $\ell \geq 0$ and $\delta \in \Delta^{(\ell)} \setminus \{u_1, \ldots, u_k\}$.

Tree concatenation is associative [6, Cor 2.4.2].

For convenience, we will use the following abbreviations. For every $\xi \in \mathrm{T}_\Delta(X_k)$ and $\xi_1, \ldots, \xi_k \in \mathrm{T}_\Delta(X)$, we abbreviate $\xi[x_1/\xi_1, \ldots, x_k/\xi_k]$ by $\xi[\xi_1, \ldots, \xi_k]$ or $\xi[\xi_{1..k}]$.

This abbreviation is also used for $\xi \in \mathrm{T}_\Delta(U)$ when U is a finite set other than X_k, provided that elements in U are ordered explicitly, or if U is a singleton. Moreover, we may write $\xi[u_i/\xi_i \mid i \in [k]]$ instead of $\xi[u_1/\xi_1, \ldots, u_k/\xi_k]$.

Graphs. Let Σ be an alphabet. A Σ-*labeled directed graph* (for short: *graph*) is a pair (V, E) where V is a finite set of *vertices* and E is a finite set of *edges* satisfying $E \subseteq V \times \mathcal{P}(\Sigma) \times V$. For an edge $e = (v_1, U, v_2)$, we call U the label of e. Let K be a graph. Sometimes we denote the set of vertices by V_K and the set of edges by E_K or, if no confusion arises, by $\to$. Then $(v_1, U, v_2) \in E_K$ will also be abbreviated by $v_1 \overset{U}{\to} v_2$ or just by $v_1 \to v_2$.

We denote the set of *maximal strongly connected components (SCC) of* (V, E) by $\mathrm{scc}((V, E))$.

Let (V, E) be a graph and $M \subseteq V$. The M*-fragment of* (V, E), denoted by $(V, E)|_M$, is the graph $(M, E \cap (M \times \mathcal{P}(\Sigma) \times M))$.

3. Context-Free Tree Languages and Regular Tree Languages

A *linear nondeleting context-free tree grammar*[2] (lnCFTG) is a tuple $G = (N, \Delta, A_0, R)$, where N and Δ are ranked alphabets (of *nonterminals* and *terminals*, respectively) such that $N \cap \Delta = \emptyset$, $A_0 \in N^{(0)}$ (*initial nonterminal*), R is a finite set of *rules* of the form $A(x_{1..k}) \to \xi$ with $k \in \mathbb{N}$, $A \in N^{(k)}$, and $\xi \in \mathrm{C}_{N\cup\Delta}(X_k)$. In a rule $r\colon A(x_{1..k}) \to \xi$ the *left-hand side (LHS) of* r is $A(x_{1..k})$ and the *right-hand side (RHS) of* r is ξ, denoted by $\mathrm{lhs}(r)$ and $\mathrm{rhs}(r)$, respectively. If $\mathrm{lhs}(r) = A(x_{1..k})$, then we call A the *LHS-nonterminal of* r, also denoted by $\mathrm{lhs}(r)(\varepsilon)$. For each $A \in N$, we abbreviate $A(x_{1..\mathrm{rk}_N(A)})$ by $A(\overline{x})$.

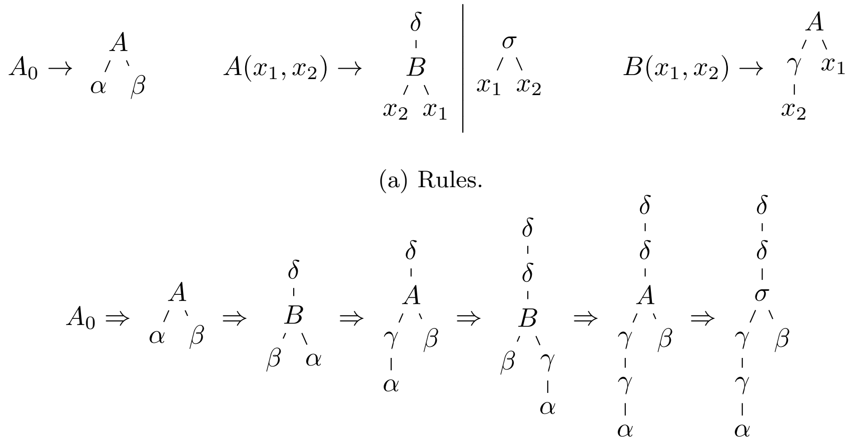
For technical convenience, we will also allow rules to use any finite combination of distinct variables instead of a prefix of the sequence $x_1, x_2, x_3, \ldots$, e.g., $A(x_2, x_5) \to \sigma(x_2, x_5)$. It is easy to see how to transform such a rule into the formally correct form (by renaming variables).

In the following let $G = (N, \Delta, A_0, R)$ be an arbitrary lnCFTG. The *derivation relation* $\Rightarrow$ is defined as follows. For trees $\zeta, \zeta' \in \mathrm{T}_{N\cup\Delta}(X)$ and a rule $r\colon A(x_{1..k}) \to \xi$ in R, we have $\zeta \Rightarrow_r \zeta'$ if there is a position $w \in \mathrm{pos}(\zeta)$ such that $\zeta(w) = A$ and ζ' is obtained from ζ by replacing the subtree at position w by $\xi[\zeta|_{w1}, \ldots, \zeta|_{wk}]$. Thus if ζ is a context, then so is ζ'. Note that we do not impose any restriction on the order in which nonterminals are derived (unrestricted derivation [7]). We write $\zeta \Rightarrow \zeta'$ if there is an $r \in R$ such that $\zeta \Rightarrow_r \zeta'$. We denote the reflexive, transitive closure of $\Rightarrow$ by $\Rightarrow^*$. For $n \in \mathbb{N}$ and $s \in R^*$ with $s = r_1 r_2 \ldots r_n$ and $r_i \in R$ for each $i \in [n]$, we write $\zeta \Rightarrow_s \zeta'$ if there are $\zeta_1, \ldots, \zeta_{n-1} \in \mathrm{T}_{N\cup\Delta}(X)$ such that $\zeta \Rightarrow_{r_1} \zeta_1 \Rightarrow_{r_2} \zeta_2 \Rightarrow_{r_3} \ldots \Rightarrow_{r_{n-1}} \zeta_{n-1} \Rightarrow_{r_n} \zeta'$. In this case, we call s a *derivation* in G.

For $k \in \mathbb{N}$ and $\zeta \in \mathrm{C}_{N\cup\Delta}(X_k)$, the *tree language induced by* ζ *on* G is

$$\mathcal{L}(G, \zeta) = \{\xi \in \mathrm{C}_\Delta(X_k) \mid \zeta \Rightarrow^* \xi\} \ .$$

[2] sometimes called *simple* context-free tree grammars in the literature



(a) Rules.

(b) Derivation.

Figure 4:: The lnCFTG G_2 and an example derivation.

The *tree language of* G, denoted by $\mathcal{L}(G)$, is defined by $\mathcal{L}(G) = \mathcal{L}(G, A_0)$. Note that $\mathcal{L}(G) \subseteq \mathrm{T}_\Delta$. Two lnCFTGs G and G' are *equivalent*[3] if $\mathcal{L}(G) = \mathcal{L}(G')$. Figure 4(a) presents an example lnCFTG G_2 and Figure 4(b) an example derivation. It can be seen that $\mathcal{L}(G_2) = \{\delta^n(\sigma(\gamma^n(\alpha), \beta)) \mid n \in \mathbb{N}\}$.

It is intuitively clear that rules of a lnCFTG G can be applied in any order without affecting the resulting tree (cf. the proof of [19, Lm. 4]). This fact goes back to the structural theorems for macro grammars [7]. More precisely, let $\zeta \in \mathrm{T}_{N \cup \Delta}(X)$ and w_1, w_2 be distinct positions of ζ at which rules $r_1, r_2 \in R$, respectively, apply. If neither w_1 is a prefix of w_2 nor vice versa, then clearly r_1 and r_2 can be applied in any order at w_1, w_2, respectively. If, e.g., w_1 is a prefix of w_2, i.e., $w_2 = w_1\, i\, u$ for some i and u, then the application of r_1 might change the position at which r_2 has to be applied. Let v be the uniquely determined position of x_i in $\mathrm{rhs}(r_1)$. Then the application of r_1 at w_1 followed by the application of r_2 at $w_1\, v\, u$ leads to the same tree as the application of r_2 at w_2 followed by the application of r_1 at w_1.

Sometimes, we fix an order where, in each derivation step, one of the nonterminals at an outermost position is derived. Such a derivation is called *outside-in* (cf. [7, pp. 2-15]).

Later we wish to analyze derivations in which, for some given subset $N' \subseteq N$, only rules may be used of which the LHS-nonterminal is in N'. This is achieved by considering all symbols from $N \setminus N'$ as terminal symbols. Formally, the *N'-fragment of* R, denoted by $R|_{N'}$, is defined to be the set $\{r \in R \mid \mathrm{lhs}(r)(\varepsilon) \in N'\}$. The *$N'$-fragment*

[3] in the NLP-community, equality of induced tree languages is more specifically called *strongly equivalent* to distinguish it from weak equivalence, which is the equality of induced string languages

of G is the lnCFTG

$$G|_{N'} = (N', \Delta \cup (N \setminus N'), \_, R|_{N'}) \ .$$

where the initial nonterminal of $G|_{N'}$ is irrelevant, and we only address its language using an explicitly given initial tree ζ via $\mathcal{L}(G|_{N'}, \zeta)$.

As an example, we consider the lnCFTG G using the set of nonterminals $\{A_0, A, B\}$, the set of terminals $\Delta = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}, \beta^{(0)}\}$, and the rules $A_0 \to A(\alpha, \beta)$, $A(x_1, x_2) \to A(B(x_1), x_2) \mid \sigma(x_1, x_2)$, and $B(x_1) \to \gamma(B(x_1))$. The $\{A\}$-fragment of G has only the two rules with LHS-nonterminal A. The tree language induced by $A(x_1, x_2)$ on the fragment is $\mathcal{L}\big(G|_{\{A\}}, A(x_1, x_2)\big) = \{\sigma(B^n(x_1), x_2) \mid n \in \mathbb{N}\} \subseteq \mathrm{C}_{\Delta \cup \{B\}}(X_2)$ where $B^n(x_1)$ is a tree consisting of n B's on top of each other followed by an x_1.

A *regular tree grammar* (RTG) is a lnCFTG in which each nonterminal has rank 0. A tree language $L \subseteq \mathrm{T}_\Delta$ is *regular* if there is a RTG $G = (N, \Delta, A_0, R)$ such that $\mathcal{L}(G) = L$.

4. Self-Embedding lnCFTG

In [3] it was proved that each context-free string grammar which is non-self-embedding generates a regular language, where self-embedding means the existence of a derivation of the form $A \Rightarrow^* vAw$ with $v \neq \varepsilon$ and $w \neq \varepsilon$ for some strings v and w over terminals and nonterminals. Here we generalize the notion of self-embedding to the tree case.

Let $G = (N, \Delta, A_0, R)$ be a lnCFTG. We say that G is *self-embedding* if there is a $k \geq 1$ and an $A \in N^{(k)}$ such that at least one of the following two properties holds (viewing the variables in X_k as symbols with rank 0):

(1) There is an $i \in [k]$ and there are $F, A', H \in \mathrm{C}_{N \cup \Delta \cup X_k}(\{z\})$ such that

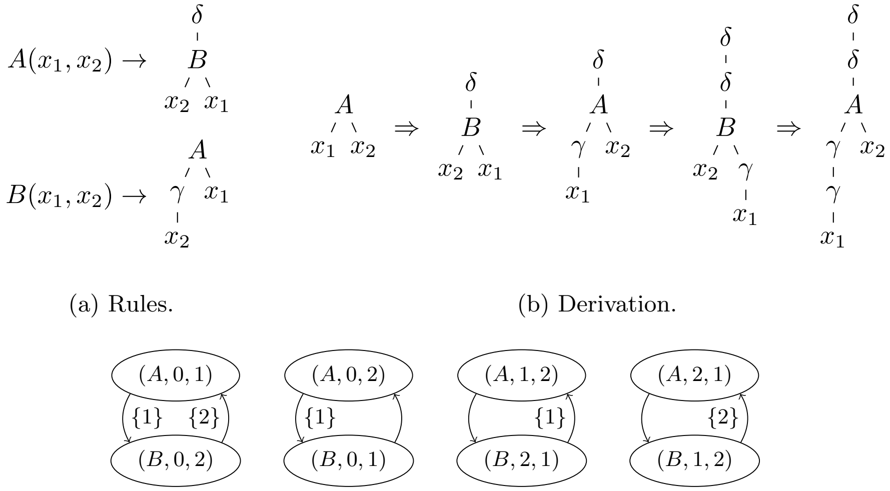
- $A(x_{1..k}) \Rightarrow^* F[A'[H[x_i]]]$,
- $A'(\varepsilon) = A$, $A'(i) = z$, and
- $F \neq z$ and $H \neq z$.

(2) There are $i, j \in [k]$ with $i \neq j$ and there are $F, H, K \in \mathrm{C}_{N \cup \Delta \cup X_k}(\{z\})$ and $A' \in \mathrm{C}_{N \cup \Delta \cup X_k}(\{z_1, z_2\})$ such that

- $A(x_{1..k}) \Rightarrow^* F[A'[H[x_i], K[x_j]]]$,
- $A'(\varepsilon) = A$, $A'(i) = z_1$, and $A'(j) = z_2$, and
- $H \neq z$ and $K \neq z$.

These two properties were depicted in Figure 3 and we illustrate them by three examples. Simultaneously we will motivate the introduction of a particular finite graph which allows us to check these properties.

As a first example, we consider the two rules of the lnCFTG G_2 shown in Figure 5(a). The derivation in Figure 5(b) shows that G_2 satisfies Requirement (1) of self-embedding: terminals are created above and below the nonterminal A in a synchronized manner (the numbers of δ's and γ's are equal). In order to detect this phenomenon it suffices to consider a finite directed graph, called position pair graph, of which each vertex is a triple: a nonterminal and two of its argument positions. We include argument position



(a) Rules.

(b) Derivation.

(c) Position pair graph.

Figure 5:: Part of the lnCFTG G_2.

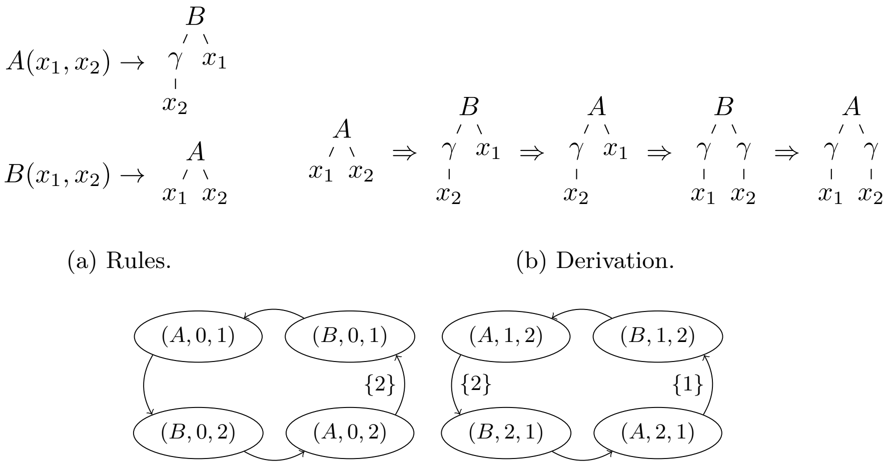
0, which represents the generation happening above the nonterminal; it may only occur in the first of the two argument positions. An edge from $(A, 0, j)$ to $(B, 0, m)$ indicates that there is a rule r with LHS-nonterminal A such that x_j appears in the argument position m of an occurrence of B in the RHS. An edge can be labeled by any subset of $\{1, 2\}$, where we drop the label $\emptyset$ in the figures. If there is at least one symbol above the occurrence of B, then the label contains a 1; if at least one symbol occurs between the occurrences of B and x_m, then the label of this edge contains a 2. The 1 pertains to the first of the two argument positions in $(A, 0, j)$ and $(B, 0, m)$, which are both 0, while 2 pertains to the second argument positions, which are j and m, respectively.

The leftmost two SCCs in Figure 5(c) show part of the position pair graph of G_2 dealing with this combination of generation above and below a nonterminal. (The rightmost two SCCs will become clear soon.) The first two steps of the derivation in Figure 5(b) induce in particular the path

$$(A, 0, 1) \xrightarrow{\{1\}} (B, 0, 2) \xrightarrow{\{2\}} (A, 0, 1)$$

through the position pair graph. Since (i) this path is cyclic, (ii) the union of the edge labels contains 1 and 2, and (iii) the position 0 is involved, Property (1) of self-embedding is satisfied.

As a second example, we consider two rules (cf. Figure 6(a)) of the lnCFTG G_1 in Figure 2. The derivation in Figure 6(b) shows that G_1 generates terminals in a synchronized manner in two argument positions below a nonterminal: the numbers of



(a) Rules.

(b) Derivation.

(c) Position pair graph.

Figure 6:: Part of the lnCFTG G_1.

γ's are equal after each fourth step. To capture this in the position pair graph, we employ vertices with two argument positions different from 0. An edge from (A, i, j) to (B, ℓ, m) (with $i \neq j$ and $\ell \neq m$) indicates that there is a rule r with LHS-nonterminal A such that x_i appears in the argument position ℓ of an occurrence of B in the RHS and x_j appears in the argument position m of the same occurrence of B. If at least one symbol occurs between the occurrences of B and x_i, then the edge label contains a 1. Likewise, if at least one symbol occurs between the occurrences of B and x_j, then the edge label contains a 2.

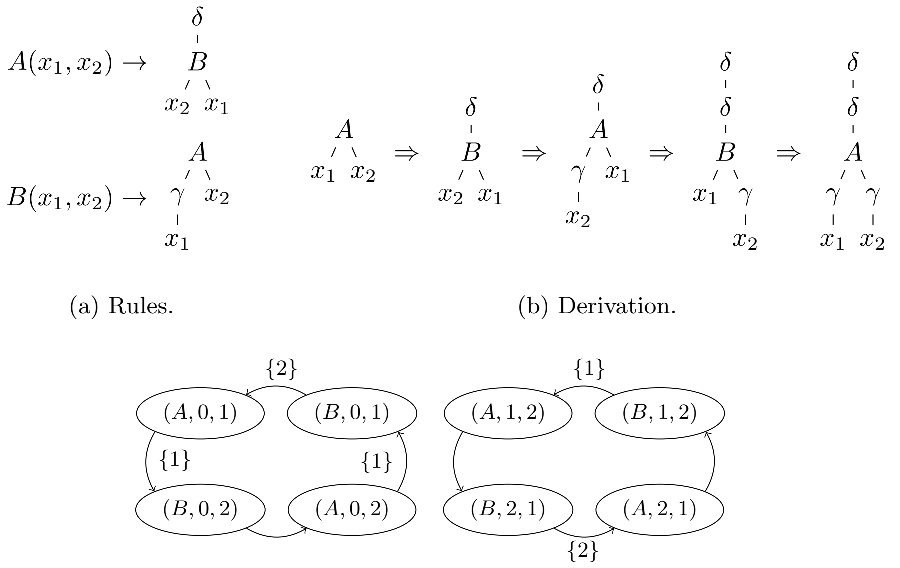
Thus, the position pair graph of G_1 is the one shown in Figure 6(c). The derivation in Figure 6(b) induces in particular the path

$$(A, 1, 2) \xrightarrow{\{2\}} (B, 2, 1) \longrightarrow (A, 2, 1) \xrightarrow{\{1\}} (B, 1, 2) \longrightarrow (A, 1, 2)$$

through the rightmost SCC of the position pair graph. Since (i) this path is cyclic, (ii) its union of edge labels contains 1 and 2, and (iii) the position 0 is not involved, Property (2) of self-embedding is satisfied.

In the third and last example, we show part of a lnCFTG G_3 in Figure 7(a), which simultaneously satisfies Properties (1) and (2) of self-embedding. The lnCFTG G_3 looks similar to G_1, but the order of the variables in the second rule is swapped. Figure 7(b) shows an example derivation and Figure 7(c) the position pair graph. Note that there are two cycles and in each cycle the union of edge labels contains 1 and 2.

Next we will formally define the notion of position pair graph for an arbitrary lnCFTG $G = (N, \Delta, A_0, R)$.



(a) Rules.

(b) Derivation.

(c) Position pair graph.

Figure 7:: Part of a lnCFTG G_3.

Definition 4.1 *The* position pair graph *of* G *is the* $\{1,2\}$*-labeled directed graph* $\mathrm{ppg}(G) = (V, E)$ *where*

$$V = \{(A, i, j) \mid A \in N^{(k)},\ i \in ([k] \cup \{0\}),\ j \in [k],\ i \neq j\}$$

and E *is defined as follows. Let* $(A, 0, j), (B, 0, m) \in V$ *and* $r \in R|_{\{A\}}$ *such that there is a* $w \in \mathrm{pos}_B(\mathrm{rhs}(r))$ *for which* wm *is* x_j*-dominating. We let* $((A, 0, j), U, (B, 0, m)) \in E$ *where* $U \subseteq \{1, 2\}$ *is defined as follows:*

- $w \neq \varepsilon$ *iff* $1 \in U$*,*
- $\mathrm{rhs}(r)(wm) \neq x_j$ *iff* $2 \in U$*.*

Furthermore, let $(A, i, j), (B, \ell, m) \in V$ *with* $i \neq 0$*,* $\ell \neq 0$*, and let* $r \in R|_{\{A\}}$ *be such that there exists a* $w \in \mathrm{pos}_B(\mathrm{rhs}(r))$ *for which* $w\ell$ *is* x_i*-dominating and* wm *is* x_j*-dominating. We let* $((A, i, j), U, (B, \ell, m)) \in E$ *where* $U \subseteq \{1, 2\}$ *is defined as follows:*

- $\mathrm{rhs}(r)(w\ell) \neq x_i$ *iff* $1 \in U$*,*
- $\mathrm{rhs}(r)(wm) \neq x_j$ *iff* $2 \in U$*.*

Observation 4.2 *Let $Q \in \mathrm{scc}(\mathrm{ppg}(G))$. Then exactly one of the following two statements holds:*

- *For each vertex $(A, i, j) \in V_Q$, we have $i = 0$.*
- *For each vertex $(A, i, j) \in V_Q$, we have $i \neq 0$.*

Now we can characterize the property of a lnCFTG G being self-embedding in terms of the position pair graph of G.

Theorem 4.3 *A lnCFTG G is self-embedding iff $\mathrm{ppg}(G)$ contains a vertex (A, i, j) and a path from (A, i, j) to (A, i, j) such that the union of all its labels contains 1 and 2.*

Proof. [⇒]: If G is self-embedding, then Property (1) or (2) holds. If Property (1) holds, there is a derivation starting from $A(x_{1..k})$ resulting in a tree with an x_i-dominating occurrence of A which is not at the root, and x_i occurs in its i-th argument position but not as its direct descendant. This derivation corresponds to a cycle in $\mathrm{ppg}(G)$ of the same length. Since symbols are generated both above A and between A and x_i, the union of the edge labels contains 1 and 2.

Similarly, if Property (2) holds, there is a derivation that corresponds to a cycle in $\mathrm{ppg}(G)$. The union of the edge labels of this path contains 1 and 2, because symbols are synchronously generated under two different argument positions.

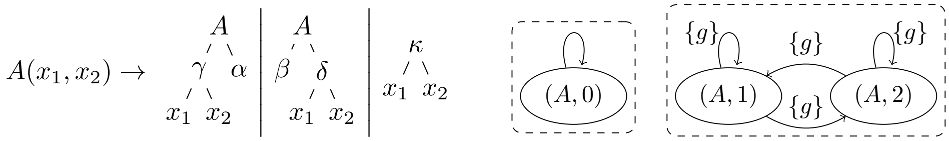
[⇐]: Suppose that there is a cycle in the position pair graph and the union of its edge labels contains 1 and 2. Then we can construct a derivation in G which satisfies Property (1) or (2): For each edge in the cycle, we apply a rule that gave rise to this edge in the construction of $\mathrm{ppg}(G)$. □

Corollary 4.4 *For each lnCFTG G, it is decidable in polynomial time whether G is self-embedding.*

Proof. The position pair graph of G can be constructed in polynomial time in the following parameters of G: number of nonterminals, the maximal rank of the nonterminals, the number of rules, and the maximal number of occurrences of nonterminals in the RHS of any rule. One can enumerate all SCCs of $\mathrm{ppg}(G)$ in linear time [4, p. 617]. For a SCC in $\mathrm{ppg}(G)$ it can be determined in linear time whether the union of all its edge labels is $\{1, 2\}$. By Theorem 4.3 this is all that is required to decide whether G is self-embedding. □

The reader might have realized that none of our examples contains nested nonterminals. This choice is reasonable, because the grammars remain self-embedding even if one replaces any occurrence of a terminal by a nonterminal (with arbitrary rules). However, applying this replacement to a non-self-embedding lnCFTG might lead to a non-self-embedding lnCFTG or a self-embedding lnCFTG. For instance, if in the non-self-embedding lnCFTG with the two rules

$$A(x) \to A(G(x)) \qquad \text{and} \qquad G(x) \to \gamma(x)$$



(a) Rules.

(b) Position graph.

Figure 8:: Part of a lnCFTG G_4.

we replace the second rule by

$$G(x) \to A(x) \enspace .$$

then the resulting grammar is self-embedding. Our formal investigation allows for nested nonterminals, but they will not be needed in our examples to illustrate various constructions and proofs.

5. Uniqueness in Argument Positions

In this section, we introduce a property called *unique in argument positions.* We prove that each non-self-embedding lnCFTG can be transformed into an equivalent lnCFTG which is unique in argument positions. This syntactic restriction will turn out to be useful to construct an equivalent RTG for each non-self-embedding lnCFTG. The definition of the property is based on the concept of the position graph, which we describe in the following. Subsequently, we will give a formal definition of the property.

In this section, we let $G = (N, \Delta, A_0, R)$ be a non-self-embedding lnCFTG.

Position Graph. The graph contains one vertex for each pair of nonterminal and argument position (including the special argument position 0 as in the case of the position pair graph). Its edges represent the movement of values across argument positions, and are labeled with $\{g\}$ if new symbols are generated at the same time, and with $\emptyset$ otherwise. For instance, Figure 8(b) depicts the position graph of the lnCFTG G_4 in Figure 8(a).

Definition 5.1 *The* position graph of G *is the* $\{g\}$*-labeled directed graph* $\mathrm{pg}(G) = (V, E)$ *where*

$$V = \{(A, i) \mid A \in N^{(k)}, i \in ([k] \cup \{0\})\} \enspace .$$

In order to obtain E*, we first define the auxiliary mapping* $\mathrm{edg}\colon R \to \mathcal{P}(V \times \mathcal{P}(\{g\}) \times V)$ *as follows. Let* $A, B \in N$ *and* $r \in R|_{\{A\}}$ *such that there exists a variable dominating position* $w \in \mathrm{pos}_B(\mathrm{rhs}(r))$*.*

- *If* $w \neq \varepsilon$*, then* $\mathrm{edg}(r)$ *contains the edge* $((A,0),\{g\},(B,0))$ *and*
- *if* $w = \varepsilon$*, then* $\mathrm{edg}(r)$ *contains the edge* $((A,0),\emptyset,(B,0))$*.*

Moreover, let $i \in [\mathrm{rk}_N(A)]$ *and* $j \in [\mathrm{rk}_N(B)]$ *such that* wj *is* x_i*-dominating.*

- *If* $\mathrm{rhs}(r)(wj) \neq x_i$*, then* $\mathrm{edg}(r)$ *contains the edge* $((A,i),\{g\},(B,j))$ *and*
- *if* $\mathrm{rhs}(r)(wj) = x_i$*, then* $\mathrm{edg}(r)$ *contains the edge* $((A,i),\emptyset,(B,j))$*.*

Furthermore, $\mathrm{edg}(r)$ *does not contain any other elements. Then we define* $E = \bigcup_{r\in R} \mathrm{edg}(r)$*.*

If an edge is labeled by $\{g\}$, then we call it *generating*. We call $P \in \mathrm{scc}(\mathrm{pg}(G))$ *generating* if P contains a generating edge. Sometimes we will also be interested in the set of rules which have induced edges in a particular $P \in \mathrm{scc}(\mathrm{pg}(G))$. Formally, we define the set of *rules of* P, denoted by $\mathrm{rules}(P)$, to be the set

$$\mathrm{rules}(P) = \{r \in R \mid E_P \cap \mathrm{edg}(r) \neq \emptyset\} \ .$$

Let $n \in \mathbb{N}$ and $r_1, r_2, \ldots, r_n \in R$. We say that the sequence $r_1 r_2 \ldots r_n$ *induces a path*

$$p : (A_0, i_0) \to (A_1, i_1) \to \ldots \to (A_n, i_n)$$

in $\mathrm{pg}(G)$ if for each $k \in [n]$ the set $\mathrm{edg}(r_k)$ contains the edge $(A_{k-1}, i_{k-1}) \to (A_k, i_k)$. For instance consider the rules in Figure 8(a), which we denote by r_1, r_2, and r_3, respectively. The sequence of rules $s = r_1 r_2 r_2$ induces the paths

$$\begin{aligned}
p_1 :&\quad (A,0) \overset{\emptyset}{\to} (A,0) \overset{\emptyset}{\to} (A,0) \overset{\emptyset}{\to} (A,0) \ , \\
p_2 :&\quad (A,1) \overset{\{g\}}{\to} (A,1) \overset{\{g\}}{\to} (A,2) \overset{\{g\}}{\to} (A,2) \ , \text{ and} \\
p_3 :&\quad (A,2) \overset{\{g\}}{\to} (A,1) \overset{\{g\}}{\to} (A,2) \overset{\{g\}}{\to} (A,2) \ .
\end{aligned}$$

We make three observations concerning the position graph, which will help us later.

Observation 5.2 *Let* $P \in \mathrm{scc}(\mathrm{pg}(G))$*. Then exactly one of the following two statements holds:*

- *For each vertex* $(A,i) \in V_P$*, we have* $i = 0$*.*
- *For each vertex* $(A,i) \in V_P$*, we have* $i \neq 0$*.*

Observation 5.3 *Let* $A, B \in N$ *and* $r \in R$*. Then* $(A,i) \to (B,j)$ *is in* $\mathrm{edg}(r)$ *for some* $i \in [\mathrm{rk}_N(A)]$ *and* $j \in [\mathrm{rk}_N(B)]$ *iff* $(A,0) \to (B,0)$ *is in* $\mathrm{edg}(r)$*.*

Observation 5.4 *Let* $M \subseteq N$ *and* $M' = \{(A,i) \mid A \in M, i \in ([\mathrm{rk}_N(A)] \cup \{0\})\}$*. By definition of the respective fragments, we have that* $\mathrm{pg}(G|_M) = \mathrm{pg}(G)|_{M'}$*.*

For $P \in \mathrm{scc}(\mathrm{pg}(G))$, we denote *the set of all nonterminals occurring in* P by M_P, i.e., $M_P = \{A \mid k \in \mathbb{N}, (A,k) \in V_P\}$.

Since G is non-self-embedding, the rules of a generating SCC of $\mathrm{pg}(G)$ which does not contain references to 0 have a particular form. This will be crucial while transforming the grammar.

Lemma 5.5 *Let $P \in \mathrm{scc}(\mathrm{pg}(G))$ be generating such that $(C,0) \notin V_P$ for each $C \in N$. Then each rule in* $\mathrm{rules}(P)$ *has the form $A(x_{1..k}) \to B(\zeta_{1..\ell})$ for some $A, B \in M_P$ and $\zeta_{1..\ell} \in \mathrm{T}_{N\cup\Delta}(X_k)$.*

Proof. The proof is by contradiction.

Assume that there is a rule r in $\mathrm{rules}(P)$ such that $\mathrm{rhs}(r)(\varepsilon) \notin M_P$. Since $r \in \mathrm{rules}(P)$ and $(C,0) \notin V_P$ for each $C \in N$, it follows that there are $A, B \in M_P$, $i \in [\mathrm{rk}_N(A)]$, $j \in [\mathrm{rk}_N(B)]$, and $U \subseteq \{g\}$ such that $((A,i), U, (B,j)) \in E_P \cap \mathrm{edg}(r)$. From Definition 5.1, we get that there is a position $w \in \mathrm{pos}_B(\mathrm{rhs}(r))$ such that wj is x_i-dominating. By the assumption, we have that $w \neq \varepsilon$.

Since (A,i) and (B,j) are vertices in the same generating SCC P, there are $C, D \in M_P$, $m \in [\mathrm{rk}_N(C)]$, and $n \in [\mathrm{rk}_N(D)]$ such that $((C,m), \{g\}, (D,n)) \in E_P$ and thus the following path exists in P:

$$p : (A,i) \to (B,j) \to \ldots \to (C,m) \overset{\{g\}}{\to} (D,n) \to \ldots \to (A,i)$$

where the first edge is induced by r.

We will now use p and construct a cycle in $\mathrm{ppg}(G)$. For each edge $(A',i') \to (B',j')$ in p, there are $r' \in \mathrm{rules}(P)$ and $w' \in \mathrm{pos}_{B'}(\mathrm{rhs}(r'))$ such that $w'j'$ is $x_{i'}$-dominating. Then, by Definition 4.1, there is an edge $((A',0,i'), U', (B',0,j'))$ in $\mathrm{ppg}(G)$ for some $U' \subseteq \{1,2\}$. Hence, we obtain the cycle

$$p' : (A,0,i) \overset{U}{\to} (B,0,j) \to \ldots \to (C,0,m) \overset{U'}{\to} (D,0,n) \to \ldots \to (A,0,i) \ .$$

We now investigate U and U'. First consider the edge $((A,0,i), U, (B,0,j))$. This edge is induced using the rule r at position w in $\mathrm{rhs}(r)$. By Definition 4.1, we have that $1 \in U$, because $w \neq \varepsilon$. Second, consider the edge $((C,0,m), U', (D,0,n))$. It was constructed on the basis of $((C,m), \{g\}, (D,n))$, for a rule $r' \in \mathrm{rules}(P)$ and a position $w' \in \mathrm{pos}_D(\mathrm{rhs}(r'))$ such that $w'n$ is x_m-dominating and $\mathrm{rhs}(r')(w'n) \neq x_m$. Hence, by Definition 4.1, we have $2 \in U'$.

Since p' is a cycle in $\mathrm{ppg}(G)$ such that the union of its labels contains 1 and 2, the lnCFTG G is self-embedding by Theorem 4.3. This contradicts the assumption on G. □

Uniqueness in argument positions. Consider the rules of the lnCFTG G_4 in Figure 8(a). It can be seen that G_4 can generate arbitrarily large trees, involving both the first and the second argument position of A. At first sight, this seems difficult to rhyme with the fact that G_4 is non-self-embedding, which we can prove using Theorem 4.3. Upon closer inspection however, we see that generation of γ's and δ's in different argument positions is not synchronized. Using the first rule repeatedly, an unbounded number of γ's can be generated in the first argument position. Likewise,

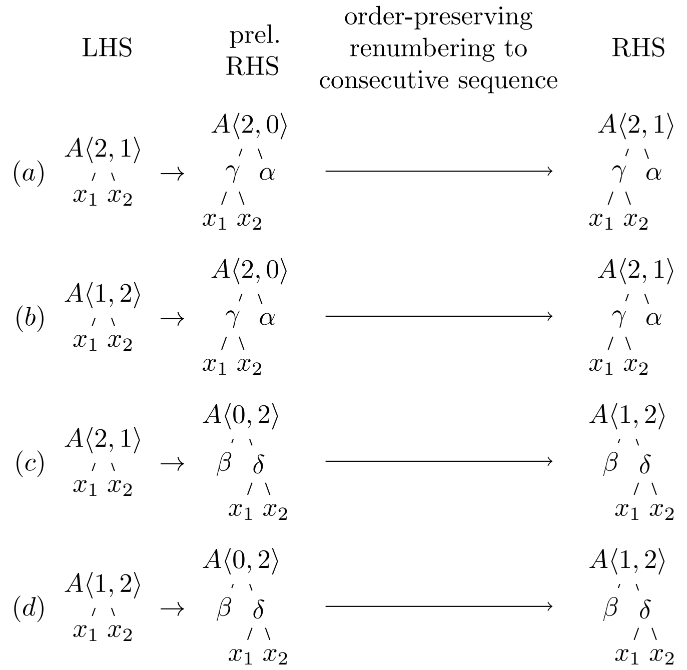


Figure 9:: Rules from G_4 with annotated relative ages.

using the second rule repeatedly, an unbounded number of δ's can be generated in the second argument position. But, by switching from the generation in argument position i (with $i \in \{1, 2\}$) to the generation in the other argument position, the value of argument position i is reset to a constant tree (β if $i = 1$; α if $i = 2$). Hence, there is no *synchronized* generation of symbols in different argument positions of one nonterminal.

In the following, we transform a non-self-embedding lnCFTG in such a way that such unsynchronized generation below one nonterminal is distributed over distinct nonterminals and thus, each nonterminal generates unbounded material in at most one argument position. Before we present our method of transformation, we first give a formal characterization of the desired property.

Definition 5.6 *Let* $P \in \text{scc}(\text{pg}(G))$ *be generating. We call* P unique in argument positions *if* $(A, j) \in V_P$ *and* $(A, j') \in V_P$ *implies* $j = j'$. *We call* G unique in argument positions *if each generating* $P \in \text{scc}(\text{pg}(G))$ *is unique in argument positions.*

The transformation involves the notion of *relative age* of an argument position. For an occurrence of a nonterminal A with rank k, the relative ages of the argument positions are expressed by a permutation of $[k]$. For example, if $k = 2$, then the sequence $\langle 2,1\rangle$, which is a permutation of $[2]$, states that the first position has relative age 2 and the second one has relative age 1. The intuition is that the value in the second position was 'born' after the value in the first position.

Upon applying a rule, the variables determine how relative ages are transferred from the LHS to positions of the root nonterminal in the RHS, say an occurrence of B with rank ℓ. This is subject to the following two constraints. First, if a new value is 'born' in argument position j of B, which is when there are no variables in that position, it receives a unique relative age that is smaller than any position that does have variables; we assign 0 as a preliminary value. This is exemplified by 0 in $\langle 4,0\rangle$ in the preliminary RHS in Figure 9(a), and 0 in $\langle 0,4\rangle$ in the preliminary RHS of Figure 9(c). Second, if an argument position of B contains several variables, we take the maximum of the relative ages of the corresponding LHS positions as preliminary value. This is exemplified by 2 in $\langle 2,0\rangle$ in the preliminary RHS in Figure 9(a), where $2 = \max\{2,1\}$, and similarly 2 in $\langle 0,2\rangle$ in Figure 9(c). We then assign the final ages represented as a permutation according to the following rules. The higher the preliminary value of an argument position is, the higher its final age will be. The argument positions with preliminary value 0 are assigned decreasing values from left to right. Thus, the newly born argument positions obtain the smallest relative ages, whereas the argument positions obtain the smallest relative ages, whereas the argument positions that contain older subtrees obtain strictly higher relative ages. Hence, $\langle 0,2\rangle$ is turned into $\langle 1,2\rangle$. This transformation yields the RHS of the newly constructed rule, as depicted in Figure 9.

Lemma 5.7 *For each non-self-embedding lnCFTG G, there is an equivalent lnCFTG H that is non-self-embedding and unique in argument positions.*

Proof. Assume that G is not unique in argument positions. Then there is a generating $P \in \mathrm{scc}(\mathrm{pg}(G))$ such that P is not unique in argument positions. We note that, for each $A \in N$, the vertex $(A,0)$ is not in P (cf. Observation 5.2).

The following construction splits each nonterminal involved in P into new nonterminals of the form $A\langle\pi\rangle$ where π is a permutation of the argument positions of A. Each number in π represents the relative age of the corresponding argument with respect to the other arguments. The lower the number of an argument, the more recently its corresponding value was introduced, as explained above Lemma 5.7.

Formally, let $r\colon A(x_{1..k}) \to B(\zeta_{1..\ell})$ be a rule in $\mathrm{rules}(P)$ for some $A, B \in M_P$ (cf. Lemma 5.5). Furthermore, let π be a permutation of $[k]$. This determines a permutation π_r of the argument positions of B. For this, we define the auxiliary mapping $\rho\colon [\ell] \to \mathbb{N}$ as follows. For each $j \in [\ell]$, let V_j denote the set of all $i \in [k]$ such that x_i occurs in ζ_j and let $\rho(j) = \max(\{\pi(i) \mid i \in V_j\})$ where $\max(\emptyset) = 0$. Then, we define the permutation π_r of $[\ell]$ as the unique permutation such that $\pi_r(j) < \pi_r(j')$ if

(i) $\rho(j) < \rho(j')$, or

(ii) $\rho(j) = \rho(j')$ and $j > j'$.

Note that, in case (ii), we have that $\rho(j) = \rho(j') = 0$ because of linearity.

We construct a lnCFTG $H = (N', \Delta, A_0, R')$ where

- $N' = (N \setminus M_P) \cup \tilde{N}$ and $\tilde{N} = \{A\langle\pi\rangle^{(k)} \mid A \in M_P^{(k)}, \pi \text{ is a permutation of } [k]\}$,
- $R' = \text{enr}((R \setminus R|_{M_P}) \cup \tilde{R}_1 \cup \tilde{R}_2)$, and $\tilde{R}_1$, $\tilde{R}_2$, and enr are defined as follows. For each rule $r\colon (A(x_{1..k}) \to B(\zeta_{1..\ell}))$ in rules(P) and for each permutation π of $[k]$ and π_r as constructed above, let $A\langle\pi\rangle(x_{1..k}) \to B\langle\pi_r\rangle(\zeta_{1..\ell})$ be in $\tilde{R}_1$. Furthermore, for each rule $r \in (R|_{M_P} \setminus \text{rules}(P))$ with $\text{lhs}(r)(\varepsilon) = A^{(k)}$, let $A\langle\pi\rangle(x_{1..k}) \to \text{rhs}(r)$ be in $\tilde{R}_2$ for each permutation π.

  The set of rules $(R \setminus R|_{M_P}) \cup \tilde{R}_1 \cup \tilde{R}_2$ is 'enriched' by the function enr, which replaces each occurrence of a nonterminal $A \in M_P^{(k)}$ in the RHS of each rule by $A\langle\tilde{\pi}\rangle$ where $\tilde{\pi}$ is the reversal of $[k]$, i.e., for each $i \in [k]$ we have $\pi(i) = k - i + 1$.

Due to the use of the maximum in the definition of the permutations, we have that if there is a path from $(B\langle\pi\rangle^{(k)}, i)$ to $(B'\langle\pi'\rangle^{(\ell)}, i')$ in pg(H), then $k - \pi(i) \geq \ell - \pi'(i')$.

Claim 1: G and H are equivalent. Moreover, H is non-self-embedding.

Proof of Claim 1: For each derivation of G, there is precisely one way to add permutations to turn it into a derivation of H, as LHS permutations uniquely determine RHS permutations, and all permutations are allowed. Thus, G and H are equivalent.

Furthermore, since each derivation in H can be projected onto a derivation in G, the lnCFTG H is non-self-embedding. ◇

Claim 2: $H|_{\tilde{N}}$ is unique in argument positions.

Proof of Claim 2: We already stated that if there is an edge from $(B\langle\pi\rangle^{(k)}, i)$ to $(B\langle\pi'\rangle^{(\ell)}, i')$, then $k - \pi(i) \geq \ell - \pi'(i')$. Recall that this property holds due to the use of the maximum in the construction of π'.

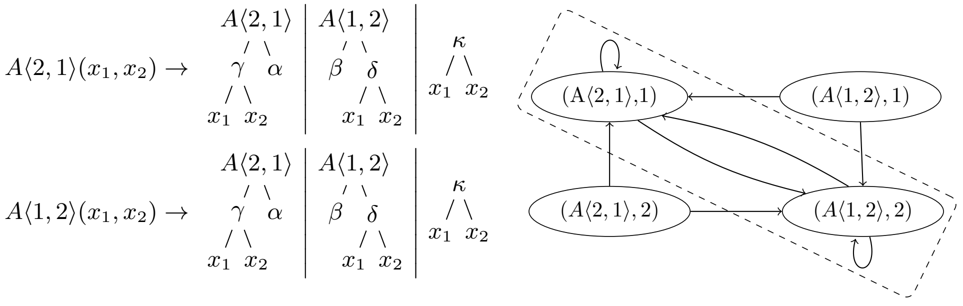
Consider any generating $P' \in \text{scc}(\text{pg}(H|_{\tilde{N}}))$, $A^{(k)} \in M_P$, a permutation π of $[k]$, and $i, j \in [k]$ such that $(A\langle\pi\rangle, i) \in V_{P'}$ and $(A\langle\pi\rangle, j) \in V_{P'}$. Since $(A\langle\pi\rangle, i)$ and $(A\langle\pi\rangle, j)$ are in the same SCC P' we can deduce that (i) there is a path from $(A\langle\pi\rangle, i)$ to $(A\langle\pi\rangle, j)$ and (ii) there is a path in the other direction. We can thus conclude that $k - \pi(i) \geq k - \pi(j)$ and $k - \pi(j) \geq k - \pi(i)$. Since π is a permutation, we have $i = j$. Therefore, $H|_{\tilde{N}}$ is unique in argument positions. ◇

The above process can be repeated until the resulting grammar is unique in argument positions. Termination is guaranteed, because in the transformation we only introduce SCCs which are unique in argument positions and thus, the total number of SCCs which are not unique in argument positions decreases in each step. □

If we apply the construction of Lemma 5.7 to the rules of G_4 in Figure 8(a), then we obtain the lnCFTG depicted in Figure 10(a). Figure 10(b) shows the relevant part of the corresponding position graph where each edge is generating (edge labels were omitted). The non-trivial SCC of Figure 10(b) is marked by a dashed box. It can be seen that each generating SCC contains, for each involved nonterminal, a unique argument position.

6. Proving Regularity of Non-self-embedding lnCFTG

In this section, let $H = (N, \Delta, A_0, R)$ be a non-self-embedding lnCFTG which is unique in argument positions.


$A\langle 2,1\rangle(x_1,x_2) \to$
$A\langle 1,2\rangle(x_1,x_2) \to$
$A\langle 2,1\rangle$
$A\langle 1,2\rangle$
γ
α
β
δ
κ
x_1
x_2
$(A\langle 2,1\rangle,1)$
$(A\langle 1,2\rangle,1)$
$(A\langle 2,1\rangle,2)$
$(A\langle 1,2\rangle,2)$

(a) Rules. (b) Part of the position graph.

Figure 10:: The lnCFTG obtained by applying the construction of Lemma 5.7 to G_4.

Consider the position graph of H. We classify a generating SCC $P \in \mathrm{scc}(\mathrm{pg}(H))$ according to whether nonterminals involved in P generate unbounded material below them, or above them. Formally, for each SCC $P \in \mathrm{scc}(\mathrm{pg}(H))$ we say that P is

- *bottom-recursive* if P is generating and contains $(A, 0)$ for some $A \in N$,
- *top-recursive* if P is generating and does not contain $(A, 0)$ for every $A \in N$.

As a running example, we consider the non-self-embedding lnCFTG

$$H_1 = (\{A_0^{(0)}, A^{(3)}, B^{(2)}\}, \{\alpha^{(0)}, \beta^{(0)}, \sigma^{(2)}, \kappa^{(3)}\}, A_0, R)$$

where R consists of the following four rules:

$$A_0 \to A(\alpha, \alpha, \alpha) \ ,$$

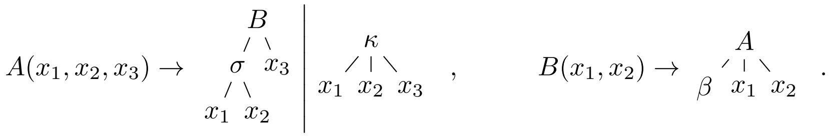


Figure 11 depicts the position graph of H_1, which shows that H_1 is unique in argument positions. Furthermore, $\mathrm{pg}(H_1)$ contains five SCCs, from which three are non-trivial. The non-trivial SCCs are marked by dashed boxes. The SCC P is top-recursive, since it is generating and does not contain $(A, 0)$, $(B, 0)$, or $(A_0, 0)$. All other SCCs are not generating and thus neither top-recursive nor bottom-recursive.

6.1. Transforming a Top-Recursive SCC into Bottom-Recursive SCCs

We present a construction which transforms a top-recursive SCC into at least one bottom-recursive SCC and a number (possibly 0) of non-generating SCCs. We repeat this process, until no more top-recursive SCCs remain in the position graph of the grammar. To be able to reason about termination of the process, we count the number of vertices in top-recursive SCCs.

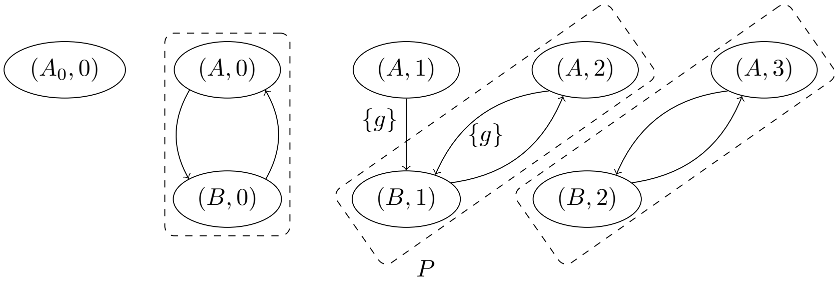


Figure 11:: The position graph of H_1.

Definition 6.1 *The* top-recursive rank *of H, denoted by* $\text{topRank}(H)$*, is the number of vertices in top-recursive SCCs in* $\text{pg}(H)$*, or formally*

$$\text{topRank}(H) = \sum_{\substack{P \in \text{scc}(\text{pg}(H)) \\ P \textit{ is top-recursive}}} |P| \ .$$

For the running example lnCFTG H_1 (cf. Figure 11 for its position graph), we have $\text{topRank}(H_1) = 2$, since $(A, 2)$ and $(B, 1)$ are in the only top-recursive SCC P.

We recall Lemma 5.5 stating that, for each top-recursive SCC P, each rule $r \in \text{rules}(P)$ has the form $A(\overline{x}) \to B(\xi_{1..\ell})$, where $A, B \in M_P$. The following two observations will be needed later.

Observation 6.2 *We let $r \in R$ be of the form $A(x_{1..k}) \to B(\xi_{1..\ell})$. Then, for each $i \in [k]$, there is a unique $j_i \in [\ell]$ such that x_i occurs in ξ_{j_i}. Thus, there is an edge $(A, i) \to (B, j_i)$ in* $\text{edg}(r)$*.*

Given an outside-in derivation s consisting exclusively of rules from a top-recursive SCC P and given a vertex $(A, i) \in V_{\text{pg}(H)}$ where $A \in M_P$ and $A(\overline{x}) \Rightarrow_s \xi$ for some $\xi \in \mathrm{T}_{N \cup \Delta}(X_{\text{rk}_N(A)})$, there are uniquely determined $B \in M_P$, $j \in [\text{rk}_N(B)]$, and $p\colon (A, i) \to^* (B, j)$ such that each edge along the path p is determined according to Observation 6.2. In this case, we say that s *top-induces* the path p.

Observation 6.3 *Let $r \in R$ be of the form $A(\overline{x}) \to B(\xi_{1..\ell})$. If there are $i, j \in [\text{rk}_N(A)]$ with $i \neq j$ and $k \in [\ell]$ such that x_i and x_j both occur in ξ_k, then* $\text{edg}(r)$ *contains the edges $((A, i), \{g\}, (B, k))$ and $((A, j), \{g\}, (B, k))$.*

The proof of the following lemma incorporates two constructions and is rather lengthy. A full example can be found after the proof and may be consulted alongside.

Lemma 6.4 *Let H be a non-self-embedding lnCFTG which is unique in argument positions and* $\text{topRank}(H) \geq 1$*. Then we can construct a non-self-embedding lnCFTG H' which is unique in argument positions such that $\mathcal{L}(H') = \mathcal{L}(H)$ and* $\text{topRank}(H') < \text{topRank}(H)$*.*

Proof. Since $\mathrm{topRank}(H) \geq 1$, there is a $P \in \mathrm{scc}(\mathrm{pg}(H))$ which is top-recursive and not reachable from any other top-recursive SCC. Let P now be fixed. For each $B \in M_P$, we denote the unique index $j \in [\mathrm{rk}_N(B)]$ such that $(B, j) \in V_P$ by j_B.

We will construct a set K of items of the form $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle$. Each item represents the fact that there is an ξ_{j_B} such that $A(\overline{x}) \Rightarrow^* B(\xi_{1..\ell})$. Intuitively, an item in K captures a context in which trees ξ_{j_B} can be generated. We use $\mathbf{d}$ as a placeholder for the *dynamic position*. We will show that K is finite and use the elements of K as nonterminals for new rules that will replace the rules from $\mathrm{rules}(P)$ and thereby decrease the top-recursive rank.

Formally, we define K through a family $(K_i \mid i \in \mathbb{N})$ as follows.

- $K_0 = \{\langle A, A, x_{1..(j_A-1)}, \mathbf{d}, x_{(j_A+1)..k}\rangle \mid k \in \mathbb{N}, A \in M_P^{(k)}\}$.
- We let $i \in \mathbb{N}$. Then K_{i+1} is the smallest set K' satisfying the following condition. If there are $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K_i$, $r \in \mathrm{rules}(P)|_{\{B\}}$, $m \in \mathbb{N}$, $C \in M_P^{(m)}$ and $\xi'_{1..m} \in \mathrm{T}_{N\cup\Delta}(X_{\mathrm{rk}_N(A)})$ such that $B(\xi_{1..(j_B-1)}, x_{j_B}, \xi_{(j_B+1)..\ell}) \Rightarrow_r C(\xi'_{1..m})$ is an outside-in derivation, then $\langle A, C, \xi'_{1..(j_C-1)}, \mathbf{d}, \xi'_{(j_C+1)..m}\rangle$ is in K'.
- $K = \bigcup_{i\in\mathbb{N}} K_i$.

Claim 1: Let $n \in \mathbb{N}$. Furthermore, let $A \in M_P^{(k)}$ and $B \in M_P^{(\ell)}$ with $k, \ell \in \mathbb{N}_+$, and $\xi_{1..(j_B-1)}, \xi_{(j_B+1)..\ell} \in \mathrm{T}_{N\cup\Delta}(X_k)$. The following are equivalent.

(i) There are a $\xi_{j_B} \in \mathrm{T}_{N\cup\Delta}(X_k)$ and an outside-in derivation s such that $|s| = n$ and $A(\overline{x}) \Rightarrow_s B(\xi_{1..\ell})$.

(ii) $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K_n$.

Proof of Claim 1: (i) ⇒ (ii): We can show this claim by well-founded induction on n.

Clearly, for each $A \in M_P$, we have that $A(\overline{x})$ derives to $A(\overline{x})$ within zero rule application steps and also by definition $\langle A, A, x_{1..(j_A-1)}, \mathbf{d}, x_{(j_A+1)..k}\rangle \in K_0$.

Now assume that the claim holds for derivations of length n for some $n \in \mathbb{N}$. Assume a derivation s of length n and a rule $r \in R$ such that $A(\overline{x}) \Rightarrow_s B(\xi_{1..\ell}) \Rightarrow_r C(\xi'_{1..m})$. By the induction hypothesis, we have $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K_n$ and thus, by definition of K_{n+1}, it follows that $\langle A, C, \xi'_{1..(j_C-1)}, \mathbf{d}, \xi'_{(j_C+1)..m}\rangle \in K_{n+1}$.

(ii) ⇒ (i): For each $A \in M_P$ we have $\langle A, A, x_{1..(j_A-1)}, \mathbf{d}, x_{(j_A+1)..k}\rangle \in K_0$ and it holds that $A(\overline{x})$ derives to $A(\overline{x})$ within zero rule application steps. Now let $n \in \mathbb{N}$ and assume that the claim holds for each element in K_n. Furthermore, assume $\langle A, C, \xi'_{1..(j_C-1)}, \mathbf{d}, \xi'_{(j_C+1)..m}\rangle \in K_{n+1}$. By definition of K_{n+1}, there are $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K_n$, a rule $r : B(\overline{x}) \to C(\zeta_{1..m})$, and some $\xi'_{j_C} \in \mathrm{T}_{N\cup\Delta}(X_k)$ such that $B(\xi_{1..(j_B-1)}, x_{j_B}, \xi_{(j_B+1)..\ell}) \Rightarrow_r C(\xi'_{1..m})$. By the induction hypothesis, there are $\xi_{j_B} \in \mathrm{T}_{N\cup\Delta}(X_k)$ and a derivation $A(\overline{x}) \Rightarrow_s B(\xi_{1..\ell})$ of length n. Then, we extend s with r and obtain

$$A(\overline{x}) \Rightarrow_s B(\xi_{1..\ell}) \Rightarrow_r C(\xi'_{1..(j_C-1)}, \xi'_{j_C}[x_{j_B}/\xi_{j_B}], \xi'_{(j_C+1)..m}) \ .$$

◇

Claim 2: The set K is finite.

Proof of Claim 2: In this proof let $n_1 = \max\{|\mathrm{pos}_{\Delta \cup (N \setminus M_P)}(\mathrm{rhs}(r))| \mid r \in \mathrm{rules}(P)\}$, $n_2 = |M_P|$, and $n_3 = \max\{\mathrm{rk}_N(C) \mid C \in M_P\}$. We prove the claim by contradiction.

Assume that K is an infinite set. Then there are $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K$, $i, m \in [\ell] \setminus \{j_B\}$, and rule sequences s, s_1, and s_2 of rules in rules(P) such that

(1) $|\mathrm{pos}(\xi_i)| > n_1 \cdot n_2 \cdot n_3$,
(since Δ and N are finite sets, there is such a tree ξ_i)

(2) s is an outside-in derivation and there is a ξ_{j_B} with $A(\overline{x}) \Rightarrow_s B(\xi_{1..\ell})$,
(cf. Claim 1)

(3) s_1 is a subsequence of s such that s_1 top-induces a generating cycle $p_1\colon (B, m) \to^* (B, m)$ in a SCC of pg(H) different from P,
(since ξ_i is sufficiently large, there must be a generating cycle)

(4) s_2 is an outside-in derivation and it top-induces a generating cycle $p_2\colon (B, j_B) \to^* (B, j_B)$ in P.
(since P is a top-recursive SCC, there must be such a generating cycle)

We show the following statement by induction.

Statement (†): For each $n \in \mathbb{N}$, there is an $m_n \in [\mathrm{rk}_N(B)]$ such that

(i) for each $i \in [n]$, the sequence $s_1 s_2$ top-induces a path $(B, m_{i-1}) \to^* (B, m_i)$, and

(ii) $m_n \notin \{j_B, m_0, \ldots, m_{n-1}\}$.

For the induction base ($n = 0$), we let $m_0 = m$ and recall that $m \neq j_B$. For the induction step, we assume that (†) holds for $n \in \mathbb{N}$. A consequence of Observation 6.2 is that there is a unique m' such that $s_1 s_2$ top-induces the path $(B, m_n) \to^* (B, m')$. We let $m_{n+1} = m'$. Then, (i) holds for m_{n+1}. Now, we show that m_{n+1} satisfies (ii). If $m_{n+1} = j_B$, then $(B, m) \to^* (B, j_B)$, but because (B, m) is in a generating SCC, this would contradict the assumption that P is not reachable from any other generating SCC. It remains to prove $m_{n+1} \notin \{m_0, \ldots, m_n\}$.

Assume that $m_{n+1} = m_j$ for some $j \in \{0, 1, \ldots, n\}$. The sequence $(s_1 s_2)^{n-j+1}$ top-induces

$$
\begin{aligned}
p\colon (B, m_j) &\to_{(s_1 s_2)^{n-j}} (B, m_n) \to_{s_1 s_2} (B, m_j)\\
p'\colon (B, j_B) &\to_{(s_1 s_2)^{n-j}} (B, j_B) \;\; \to_{s_1 s_2} (B, j_B) \ .
\end{aligned}
$$

We show that p and p' are generating. If $j = 0$, then the cycle p is generating, because it contains p_1. If $j \neq 0$, then $s_1 s_2$ top-induces $p''\colon (B, m_n) \to_{s_1 s_2} (B, m_j)$ and $p'''\colon (B, m_{j-1}) \to_{s_1 s_2} (B, m_j)$, and, by Observation 6.3, p'' is generating and therefore p is generating. Thus p is generating regardless of the choice of j. The path p' is generating, because it contains p_2.

We will now combine p and p' into one cycle in ppg(H). For this, we consider each step simultaneously in both paths. We let $k \in [|s_1 s_2| \cdot (n - j + 1)]$ and consider the k-th step. We let $((B_1, i_1), U_1, (B_2, i_2))$ be the k-th edge in p and $((B_1, j_1), U_2, (B_2, j_2))$ be the k-th edge in p'. Both edges are top-induced by the same rule r. Hence, we have that x_{i_1} and x_{j_1} occur in the subtrees $\mathrm{rhs}(r)|_{i_2}$ and $\mathrm{rhs}(r)|_{j_2}$, respectively. By

Observation 6.2 and since $m_j \neq j_B$, we have that $i_1 \neq j_1$ and $i_2 \neq j_2$. By Definition 4.1, there is an edge $((B_1, i_1, j_1), U', (B_2, i_2, j_2))$ in $\mathrm{ppg}(H)$. Furthermore, we have that $1 \in U'$ if $U_1 = \{g\}$ and $2 \in U'$ if $U_2 = \{g\}$.

Hence, from p and p', we obtain the following path $\tilde{p}$ in $\mathrm{ppg}(H)$:

$$\tilde{p}\colon (B, m_j, j_B) \to_{(s_1 s_2)^{n-j}} (B, m_n, j_B) \to_{s_1 s_2} (B, m_j, j_B) \ .$$

Since p and p' are both generating, $\tilde{p}$ is a cycle such that the union of all its path labels contains 1 and 2. By Theorem 4.3, this contradicts H being non-self-embedding and thus, (ii) holds for m_{n+1}. This proves (†).

However, (†) conflicts with the finiteness of $\mathrm{rk}_N(B)$ and thus, K is a finite set. ◇

We modify H with the help of K to construct a lnCFTG H'. We let H' contain all original rules, except the ones of $\mathrm{rules}(P)$. We further add copies of the rules from $\mathrm{rules}(P)$ after transforming them into bottom-recursive rules. This is done by reversing the rules, i.e., if H applies r_1 and afterwards r_2 ($r_1, r_2 \in \mathrm{rules}(P)$), then H' applies first r_2 and then r_1.

Reversing a rule is achieved by considering the rule in the context of a derivation. This context is represented by using the elements from K as nonterminals for H'. As an example, consider the rule $r\colon A(x_1, x_2, x_3) \to B(\sigma(x_1, x_2), x_3)$ from H_1 (cf. the running example). We have $j_A = 2$ and $j_B = 1$ and we consider the context $k_1 = \langle A, A, \beta, \mathbf{d}, x_3\rangle$ in K. If we consider r in the context of k_1 we obtain

$$A(\beta, x_2, x_3) \to B(\sigma(\beta, x_2), x_3) \ .$$

In the RHS, we obtain the context $k_2 = \langle A, B, \mathbf{d}, x_3\rangle$. We reverse the rule and construct a new rule with LHS $k_2(x_1, x_2)$, where x_1 and x_2 are those variables of $X_{\mathrm{rk}_N(A)}$ not present in k_2. The RHS of the new rule is obtained from the subtree of $\mathrm{rhs}(r)$ at position j_B as follows. We replace x_{j_A} by $k_1(x_1, x_2)$ where again, x_1 and x_2 are the variables of $X_{\mathrm{rk}_N(A)}$ not present in k_1. Hence, r is turned into the rule

$$\langle A, B, \mathbf{d}, x_3\rangle(x_1, x_2) \to \sigma(\beta, \langle A, A, \beta, \mathbf{d}, x_3\rangle(x_1, x_2), x_3) \ .$$

We add some rules which handle the connection to rules outside of $\mathrm{rules}(P)$. For each $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K$ and each rule $r\colon B(\overline{x}) \to \zeta$ in $R|_{M_P} \setminus \mathrm{rules}(P)$, we create the rule

$$A(\overline{x}) \to \zeta[\xi_{1..(j_B-1)}, \langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \ldots, x_{\ell_d}), \xi_{(j_B+1)..\ell}] \ .$$

Intuitively, the nonterminal from K generates all symbols that would have been generated by an iteration of rules in $\mathrm{rules}(P)$ below the dynamic position of B. By the substitution into ζ, we ensure that the result of the recursion is placed outside of the nonterminal and argument position participating in P.

Formally, we construct $H' = (N', \Delta, A_0, R')$ as follows. We let $N' = N \cup K$ where, for each $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K$, we define its rank to be the number of variables from $X_{\mathrm{rk}_N(A)}$ not present in $\xi_{1..(j_B-1)}, \xi_{(j_B+1)..\ell}$. We define R' using the following rules.

(1) $R \setminus R|_{M_P} \subseteq R'$;

(2) for each $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K$ and each $B(\overline{x}) \to \zeta$ in $R|_{M_P} \setminus \mathrm{rules}(P)$, we let

$$A(\overline{x}) \to \zeta[\xi_{1..(j_B-1)}, \langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \ldots, x_{\ell_d}), \xi_{(j_B+1)..\ell}]$$

be in R' where $x_{\ell_1}, \ldots, x_{\ell_d}$ are those variables of $X_{\mathrm{rk}_N(A)}$ that do not occur in $\xi_{1..(j_B-1)}, \xi_{(j_B+1)..\ell}$ in ascending order;

(3) for each $A \in M_P$, we let

$$\langle A, A, x_{1..(j_A-1)}, \mathbf{d}, x_{(j_A+1)..\mathrm{rk}_N(A)}\rangle(x_{j_A}) \to x_{j_A}$$

be a rule in R';

(4) for each $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K$ and $r \in \mathrm{rules}(P)|_{\{B\}}$ of the form $B(\overline{x}) \to C(\zeta_{1..m})$ such that $B(\xi_{1..(j_B-1)}, x_{j_B}, \xi_{(j_B+1)..\ell}) \Rightarrow_r C(\xi'_{1..m})$ is an outside-in derivation, we let

$$\begin{aligned}&\langle A, C, \xi'_{1..(j_C-1)}, \mathbf{d}, \xi'_{(j_C+1)..m}\rangle(x_{m_1}, \ldots, x_{m_{d'}})\\ &\quad \to \zeta_{j_C}[\xi_{1..(j_B-1)}, \langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \ldots, x_{\ell_d}), \xi_{(j_B+1)..\ell}]\end{aligned}$$

be in R' where $x_{m_1}, \ldots, x_{m_{d'}}$ and $x_{\ell_1}, \ldots, x_{\ell_d}$ are those variables of $X_{\mathrm{rk}_N(A)}$ which are not present in $\xi'_{1..(j_C-1)}, \xi_{(j_C+1)..m}$ and $\xi_{1..(j_B-1)}, \xi_{(j_B+1)..\ell}$, respectively, in ascending order of their indices;

(5) no other rules are in R'.

By inspecting all newly introduced rules, it can be verified that H' is non-self-embedding.

Claim 3a: Let $A, B \in M_P$, $\ell = \mathrm{rk}_N(B)$, and $\xi_{1..\ell} \in \mathrm{T}_{N\cup\Delta}(X_{\mathrm{rk}_N(A)})$. Then the following are equivalent.

(i) There is an outside-in derivation s consisting exclusively of rules in rules(P) such that $A(\overline{x}) \Rightarrow_s B(\xi_{1..\ell})$.

(ii) There is a derivation s' of rules in H' created due to (4) such that

$$\begin{aligned}&\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \ldots, x_{\ell_d})\\ &\quad \Rightarrow_{s'} \xi_{j_B}[x_{j_A}/\langle A, A, x_{1..(j_A-1)}, \mathbf{d}, x_{(j_A+1)..\mathrm{rk}_N(A)}\rangle(x_{j_A})]\end{aligned}$$

where $x_{\ell_1}, \ldots, x_{\ell_d}$ are those variables of $X_{\mathrm{rk}_N(A)}$ that do not occur in the sequence of trees $\xi_{1..(j_B-1)}, \xi_{(j_B+1)..\ell}$.

Proof of Claim 3a: We abbreviate $x_{1..(j_A-1)}, \mathbf{d}, x_{(j_A+1)..\mathrm{rk}_N(A)}$ by $\overline{x_{\mathbf{d}}}$.

(i)⇒(ii): We prove the claim by induction on the length of s. For $|s| = 0$, we trivially get $|s'| = 0$. Now we assume that $s = s_1 r$ for some sequence s_1 of rules from rules(P) and $r\colon B(\overline{x}) \to C(\zeta_{1..m})$ in rules$(P)|_{\{B\}}$ (cf. Lemma 5.5) such that $A(\overline{x}) \Rightarrow_{s_1} B(\xi_{1..\ell}) \Rightarrow_r C(\xi'_{1..m})$. Note that $\xi'_{j_C} = \zeta_{j_C}[\xi_{1..\ell}]$. By the induction hypothesis, there is a derivation s'_1 such that

$$\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \ldots, x_{\ell_d}) \Rightarrow_{s'_1} \xi_{j_B}[x_{j_A}/\langle A, A, \overline{x_{\mathbf{d}}}\rangle(x_{j_A})] \ .$$

Furthermore, by Claim 1, we have $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K$. Then, by (4), there is a rule

$$\begin{aligned} r'\colon\ &\langle A, C, \xi'_{1..(j_C-1)}, \mathbf{d}, \xi'_{(j_C+1)..m}\rangle(x_{m_1}, \dots, x_{m_{d'}})\\ &\quad \to \zeta_{j_C}[\xi_{1..(j_B-1)}, \langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \dots, x_{\ell_d}), \xi_{(j_B+1)..\ell}]\ . \end{aligned}$$

It can be seen that

$$\begin{aligned} &\langle A, C, \xi'_{1..(j_C-1)}, \mathbf{d}, \xi'_{(j_C+1)..m}\rangle(x_{m_1}, \dots, x_{m_{d'}})\\ &\quad \Rightarrow_{r's'_1}\ \zeta_{j_C}[\xi_{1..(j_B-1)}, \xi_{j_B}[x_{j_A}/\langle A, A, \overline{x_{\mathbf{d}}}\rangle(x_{j_A})], \xi_{(j_B+1)..\ell}]\\ &\quad = \qquad \zeta_{j_C}[\xi_{1..\ell}][x_{j_A}/\langle A, A, \overline{x_{\mathbf{d}}}\rangle(x_{j_A})]\ . \end{aligned}$$

Hence, $s' = r's'_1$ is the desired derivation of rules created due to (4).

(ii)⇒(i): We prove this by induction on the length of s'. For the base case $|s'| = 0$, we trivially get $|s| = 0$. For $|s'| \geq 1$, we let $s' = r's'_1$ and

$$\langle A, C, \xi'_{1..(j_C-1)}, \mathbf{d}, \xi'_{(j_C+1)..m}\rangle(x_{m_1}, \dots, x_{m_{d'}}) \Rightarrow_{r's'_1} \xi'_{j_C}[x_{j_A}/\langle A, A, \overline{x_{\mathbf{d}}}\rangle(x_{j_A})]$$

where r' is a rule due to (4) of the form

$$\begin{aligned} r'\colon\ &\langle A, C, \xi'_{1..(j_C-1)}, \mathbf{d}, \xi'_{(j_C+1)..m}\rangle(x_{m_1}, \dots, x_{m_{d'}})\\ &\quad \to \zeta[z/\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \dots, x_{\ell_d})] \end{aligned}$$

for some $\zeta \in \mathrm{C}_{N\cup\Delta\cup X}(\{z\})$, $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K$, and $r\colon B(\overline{x}) \to C(\zeta_{1..m})$ in R such that $B(\xi_{1..(j_B-1)}, x_{j_B}, \xi_{(j_B+1)..\ell}) \Rightarrow_r C(\xi'_{1..(j_C-1)}, \xi', \xi'_{(j_C+1)..m})$ is an outside-in derivation. We note that $\xi' = \zeta[z/x_{j_B}]$ and $\xi' = \zeta_{j_C}[\xi_{1..(j_B-1)}, x_{j_B}, \xi_{(j_B+1)..\ell}]$.

Furthermore, there is some $\xi_{j_B} \in \mathrm{T}_{N\cup\Delta}(X_{\mathrm{rk}_N(A)})$ such that

$$\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \dots, x_{\ell_d}) \Rightarrow_{s'_1} \xi_{j_B}[x_{j_A}/\langle A, A, \overline{x_{\mathbf{d}}}\rangle(x_{j_A})]\ .$$

By the induction hypothesis, there is an outside-in derivation s_1 such that $A(\overline{x}) \Rightarrow_{s_1} B(\xi_{1..\ell})$. Consider the outside-in derivation $A(\overline{x}) \Rightarrow_{s_1} B(\xi_{1..\ell}) \Rightarrow_r C(\xi''_{1..m})$. It can be seen that, for each $i \in [m] \setminus \{j_C\}$, we have $\xi''_i = \xi'_i$. We furthermore have $\xi''_{j_C} = \zeta_{j_C}[\xi_{1..\ell}] = \zeta[z/\xi_{j_B}] = \xi'_{j_C}$. Hence, $s_1 r$ is the desired outside-in derivation. ◇

Claim 3b: Let $A, B \in M_P$, $\ell = \mathrm{rk}_N(B)$, and $\xi_{1..\ell} \in \mathrm{T}_{N\cup\Delta}(X_{\mathrm{rk}_N(A)})$. Then the following are equivalent.

(i) There is an outside-in derivation s consisting exclusively of rules in $\mathrm{rules}(P)$ such that $A(\overline{x}) \Rightarrow_s B(\xi_{1..\ell})$.

(ii) There is a derivation s' of rules in H' created due to (3) and (4) such that $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \dots, x_{\ell_d}) \Rightarrow_{s'} \xi_{j_B}$ where $x_{\ell_1}, \dots, x_{\ell_d}$ are those variables of $X_{\mathrm{rk}_N(A)}$ that do not occur in the sequence of trees $\xi_{1..(j_B-1)}, \xi_{(j_B+1)..\ell}$.

Proof of Claim 3b: This claim follows directly from Claim 3a by using exactly one rule due to (3) at the end of s'. ◇

Claim 4: $\mathcal{L}(H) = \mathcal{L}(H')$.
Proof of Claim 4: $\mathcal{L}(H) \subseteq \mathcal{L}(H')$: We let $\widetilde{\xi} \in \mathrm{T}_\Delta$ and s be a derivation such that $A_0 \Rightarrow_s \widetilde{\xi}$ holds in H. If s contains no rules from $R|_{M_P}$, then $A_0 \Rightarrow_s \widetilde{\xi}$ holds in H' as well, since all rules are in R' due to (1). Now we assume that a rule from $R|_{M_P}$ occurs in s. Then we may reorder s such that rules of $R|_{M_P}$ are executed in sequence as follows. There are $A, B \in M_P$ and some $\zeta \in \mathrm{C}_\Delta(\{z\})$ such that

$$A_0 \Rightarrow_{s_0} \zeta[A(\xi_{1..k})] \Rightarrow_{s_1} \zeta[B(\xi'_{1..\ell})] \Rightarrow_r \zeta[\zeta'[\xi'_{1..\ell}]] \Rightarrow_{s_2} \widetilde{\xi}$$

where s_0 is a sequence of rules such that $A_0 \Rightarrow_{s_0} \zeta[A(\xi_{1..k})]$ holds in both H and H', s_1 is an outside-in derivation consisting of rules from $\mathrm{rules}(P)$, $r\colon B(\overline{x}) \to \zeta'$ is a rule in $R|_{\{B\}} \setminus \mathrm{rules}(P)$, and s_2 is the remaining sequence of rules in R. Note that rules in s_0 can be due to (1) or they can be chosen recursively by the following argument.

We consider the outside-in derivation $A(\overline{x}) \Rightarrow_{s_1} B(\zeta_{1..\ell})$. We note that $\xi'_i = \zeta_i[\xi_{1..k}]$ for each $i \in [\ell]$, i.e., the derivation is independent from its context. By Claim 1, we have $\langle A, B, \zeta_{1..(j_B-1)}, \mathbf{d}, \zeta_{(j_B+1)..\ell}\rangle \in K$. Note that r has LHS-nonterminal B and thus there is a rule due to (2) using $\langle A, B, \zeta_{1..(j_B-1)}, \mathbf{d}, \zeta_{(j_B+1)..\ell}\rangle$ and r.

By Claim 3, there is a derivation $\langle A, B, \zeta_{1..(j_B-1)}, \mathbf{d}, \zeta_{(j_B+1)..\ell}\rangle(\overline{x}) \Rightarrow_{s'_1} \zeta_B$. Hence, we replace $s_0 s_1 r$ by

$$\begin{aligned}
A_0 &\Rightarrow_{s_0} \zeta[A(\xi_{1..k})] = \zeta[A(x_{1..k})][\xi_{1..k}] \\
&\Rightarrow \zeta[\zeta'[\zeta_{1..(j_B-1)}, \langle A, B, \zeta_{1..(j_B-1)}, \mathbf{d}, \zeta_{(j_B+1)..\ell}\rangle(x_{\ell_1}, \dots, x_{\ell_d}), \zeta_{(j_B+1)..\ell}]][\xi_{1..k}] && (2) \\
&\Rightarrow_{s'_1} \zeta[\zeta'[\zeta_{1..\ell}]][\xi_{1..k}] && \text{(Claim 3b)} \\
&= \zeta[\zeta'[\xi'_{1..\ell}]] \ .
\end{aligned}$$

Note that the above derivation does not use any rules from $\mathrm{rules}(P)$. By applying the above method repeatedly, we can replace the remaining rules from $\mathrm{rules}(P)$ in s_2 and obtain a derivation in H'.

$\mathcal{L}(H) \supseteq \mathcal{L}(H')$: A derivation in H' consists either of rules also present in H or can be reordered to contain rule sequences described by the following regular expression

$$(1)^* \Big(\underbrace{(2)\ (4)^*\ (3)}_{\text{use Claim 3b}}\ (1)^*\Big)^*$$

where (i) stands for a rule due to the item (i) in the construction of R'. Using Claim 3b, it can be seen that we can replace each underbraced sequence by a sequence of rules in H with the same effect. ◇

Next we prove that the top-recursive rank of H' is smaller than the top-recursive rank of H and show that H' is unique in argument positions. For this, we observe two properties of $\mathrm{pg}(H')$, based on the construction of the rules of H'.

(P1) For each top-recursive $P' \in \mathrm{scc}(\mathrm{pg}(H'))$, we have $M_{P'} \cap K = \emptyset$.
(Intuition: A generating cycle $(A, i) \to^* (A, i)$ in $\mathrm{pg}(H')$ where $A \in N$ can be translated into a generating cycle in $\mathrm{pg}(H)$. In addition, an edge on such a path from a nonterminal $\langle A, B, \xi_{1..(j_B-1)}, \mathbf{d}, \xi_{(j_B+1)..\ell}\rangle \in K$ to a nonterminal $C \in N$ implies that symbols are generated *above* a corresponding nonterminal

occurrence of A (cf. rules due to (4)). In a top-recursive SCC $P' \in \mathrm{scc}(\mathrm{pg}(H'))$, this contradicts H being non-self-embedding and hence $M_{P'} \cap K = \emptyset$.)

(P2) We let $(A,i),(B,j) \in V_{\mathrm{pg}(H)}$. If $(A,i) \not\to^* (B,j)$ in $\mathrm{pg}(H)$, then $(A,i) \not\to^* (B,j)$ in $\mathrm{pg}(H')$.

Claim 5a: For each top-recursive SCC $P' \in \mathrm{scc}(\mathrm{pg}(H'))$ and each $(A,i) \in V_{P'}$, there is a top-recursive SCC $\widetilde{P} \in \mathrm{scc}(H)$ such that $(A,i) \in V_{\widetilde{P}}$.
Proof of Claim 5a: Let $P' \in \mathrm{scc}(\mathrm{pg}(H'))$ be top-recursive and $(A,i) \in V_{P'}$. By (P1), $A \in N$ and thus, there is a uniquely determined $\widetilde{P} \in \mathrm{scc}(\mathrm{pg}(H))$ such that $(A,i) \in \widetilde{P}$. We show that $\widetilde{P}$ is top-recursive.

There is a rule sequence s' of rules from R' such that s' induces a generating path $p' \colon (A,i) \to_{s'} (A,i)$ in P'. We transform p' into a path $p \colon (A,i) \to^* (A,i)$ in $\mathrm{pg}(H)$ with the following case distinction on rules in s'. Let r' be a rule in s'.

If $r' \in R$, then the induced edge is not changed. Trivially, if r' induces a generating edge in p', then it induces a generating edge in p. Now assume that $r' \in R' \setminus R$ and r' induces the edge $(B_1,i_1) \to_{r'} (B_2,i_2)$ in p where $B_1, B_2 \in N$, $i_1 \in [\mathrm{rk}_N(B_1)]$, and $i_2 \in [\mathrm{rk}_N(B_2)]$. In this case, r' is due to (2). By Claims 1 and 3b, there is a sequence s of rules in R such that $p_1 \colon (B_1,i_1) \to_s (B_2,i_2)$. Furthermore, it can be seen that s is generating if r' is. We replace $(B_1,i_1) \to_{r'} (B_2,i_2)$ by p_1.

We transform s' rule by rule as described above and obtain the path $p \colon (A,i) \to_p (A,i)$ in $\mathrm{pg}(H)$. We have that p is generating, because p' is. Thus, $\widetilde{P}$ is top-recursive. ◇

Claim 5b: For each $(A,i) \in V_P$ and $P' \in \mathrm{scc}(\mathrm{pg}(H'))$ such that $(A,i) \in V_{P'}$, we have that P' is not top-recursive.
Proof of Claim 5b: Let $(A,i) \in V_P$ and $P' \in \mathrm{scc}(\mathrm{pg}(H'))$ such that $(A,i) \in V_{P'}$. We assume that P' is top-recursive. Then there is a generating path $p' \colon (A,i) \to^* (A,i)$ in P'. Furthermore, there are $(B,j) \in V_{\mathrm{pg}(H')}$ and $r' \in R'$ such that $(A,i) \to_{r'} (B,j)$ is the first edge of p'. By (P1), we have $B \in N$. Since $A \in M_P$, we have that r' is due to (2) and, by the construction of H', we have $(B,j) \notin V_P$. Thus, we have $(B,j) \not\to^* (A,i)$ in $\mathrm{pg}(H)$ and thus, by (P2), $(B,j) \not\to^* (A,i)$ in $\mathrm{pg}(H')$. This contradicts the existence of p'. Since there is no generating path $(A,i) \to (A,i)$ in $\mathrm{pg}(H)$, we obtain a contradiction to P' being top-recursive. ◇

Claim 6: $\mathrm{topRank}(H') < \mathrm{topRank}(H)$.
Proof of Claim 6: This is a consequence of Claims 5a and b. ◇

Claim 7: The lnCFTG H' is unique in argument positions.
Proof of Claim 7: We analyze newly introduced generating SCCs. Let $P' \in \mathrm{scc}(\mathrm{pg}(H'))$ be generating. If P' is bottom-recursive, then by Observation 5.2, P' is trivially unique in argument positions.

If P' is top-recursive, then we analyze the rules from $\mathrm{rules}(P')$. We note that, by (P1), there are no rules due to (3) or (4). Hence, we consider rules due to (1) and (2). We let $(A,i),(A,j) \in V_{P'}$ and $p' \colon (A,i) \to^* (A,j) \to^* (A,i)$ be a generating path in $\mathrm{pg}(H')$. Then, we construct the path p in $\mathrm{pg}(H)$ by modifying p' as follows. Edges induced by rules due to (1) are taken over without modification. Each edge induced by a rule due to (2) is replaced by the path induced by the corresponding sequence of rules in rules(P) (cf. Claim 1) followed by the single rule outside of rules(P). Hence, $p\colon (A,i) \to^* (A,j) \to^* (A,i)$ is a path in pg(H). Since H is unique in argument positions, we have $i = j$ and thus, H' is unique in argument positions. ◇

□

We illustrate the constructions from the proof of Lemma 6.4 using the example lnCFTG H_1 from the beginning of Section 6. It is clear that P (cf. Figure 11) is the selected SCC since it is the only top-recursive SCC. We note that P is reachable from one other SCC, which is not top-recursive.

The set K contains the contexts of trees that can be generated in the generating argument positions using rules from rules(P). We note that $j_A = 2$ and $j_B = 1$. The set K contains the following items:

$$K = \{ \quad \begin{array}{lll} \langle A, A, x_1, \mathbf{d}, x_3 \rangle \ , & \langle A, B, \mathbf{d}, x_3 \rangle \ , & \langle A, A, \beta, \mathbf{d}, x_3 \rangle \ , \\ \langle B, B, \mathbf{d}, x_2 \rangle \ , & \langle B, A, \beta, \mathbf{d}, x_2 \rangle & \end{array} \quad \}$$

We now present the rules constructed by (3) and (4).

$$(3)\ \langle A, A, x_1, \mathbf{d}, x_3 \rangle(x_2) \to x_2 \qquad (3)\ \langle B, B, \mathbf{d}, x_2 \rangle(x_1) \to x_1$$

$$(4)\ \langle B, A, \beta, \mathbf{d}, x_2 \rangle(x_1) \to \begin{array}{c} \langle B, B, \mathbf{d}, x_2 \rangle \\ | \\ x_1 \end{array}$$

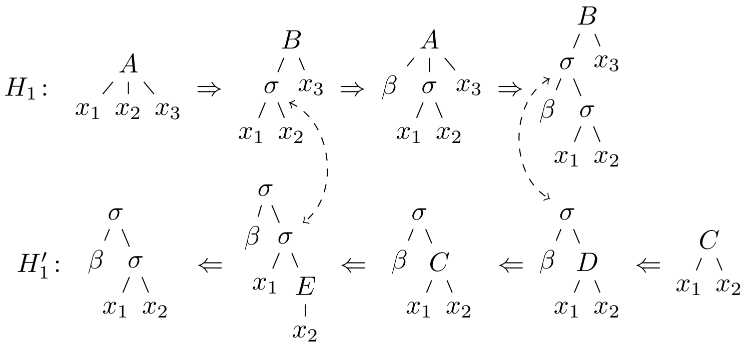
$$(4)\ \langle A, B, \mathbf{d}, x_3 \rangle(x_1, x_2) \to \begin{array}{ccc} & \sigma & \\ / & & \backslash \\ x_1 & & \begin{array}{c} \langle A, A, x_1, \mathbf{d}, x_3 \rangle \\ | \\ x_2 \end{array} \end{array}$$

$$(4)\ \langle A, A, \beta, \mathbf{d}, x_3 \rangle(x_1, x_2) \to \begin{array}{c} \langle A, B, \mathbf{d}, x_3 \rangle \\ / \ \ \backslash \\ x_1 \ \ x_2 \end{array}$$

$$(4)\ \langle A, B, \mathbf{d}, x_3 \rangle(x_1, x_2) \to \begin{array}{ccc} & \sigma & \\ / & & \backslash \\ \beta & & \begin{array}{c} \langle A, A, \beta, \mathbf{d}, x_3 \rangle \\ / \ \ \backslash \\ x_1 \ \ x_2 \end{array} \end{array}$$

$$(4)\ \langle B, B, \mathbf{d}, x_2 \rangle(x_1) \to \begin{array}{ccc} & \sigma & \\ / & & \backslash \\ \beta & & \begin{array}{c} \langle B, A, \beta, \mathbf{d}, x_2 \rangle \\ | \\ x_1 \end{array} \end{array}$$

With the help of these rules we can illustrate Claim 3a and b. Consider the following two derivations from H_1 and H_1'. For easier comparison we denote the derivation of H_1 from left to right and the derivation of H_1' from right to left. The two occurrences of the terminal symbol σ are created at different stages of the derivations. We link the corresponding creations by dashed curves.



We use the abbreviations $C = \langle A, B, \mathbf{d}, x_3\rangle$, $D = \langle A, A, \beta, \mathbf{d}, x_3\rangle$, and $E = \langle A, A, x_1, \mathbf{d}, x_3\rangle$.

We will now present the rules of H_1' created due to (1) and (2).

(1) There is only one rule not in $R \setminus R|_{M_P}$: $A_0 \to A(\alpha, \alpha, \alpha)$.

(2) There is only one rule in $R|_{M_P} \setminus \text{rules}(P)$, viz. r: $A(x_1, x_2, x_3) \to \kappa(x_1, x_2, x_3)$. Hence, we create rules for items in K where the second component is A, viz., $\langle A, A, x_1, \mathbf{d}, x_3\rangle$, $\langle A, A, \beta, \mathbf{d}, x_3\rangle$, and $\langle B, A, \beta, \mathbf{d}, x_2\rangle$. The context information from these elements is incorporated into r.

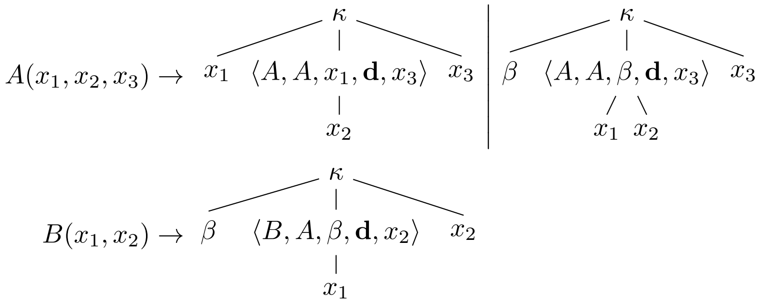

$A(x_1, x_2, x_3) \to$
κ
x_1 $\langle A, A, x_1, \mathbf{d}, x_3\rangle$ x_3
x_2
κ
β $\langle A, A, \beta, \mathbf{d}, x_3\rangle$ x_3
x_1 x_2
$B(x_1, x_2) \to$
κ
β $\langle B, A, \beta, \mathbf{d}, x_2\rangle$ x_2
x_1

Figure 12 shows part of $\text{pg}(H_1')$ using the abbreviations C, D, and E as before. The SCC formed by nonterminals from K is bottom-recursive. We observe that the resulting lnCFTG H_1' does not have any top-recursive SCC and thus $\text{topRank}(H_1') = 0$. Claims 5a and b are trivial for this example and Claim 6 is illustrated. It can be seen that $\text{pg}(H_1')$ is unique in argument positions and this is an example for Claim 7.

Theorem 6.5 *Let H be a non-self-embedding lnCFTG which is unique in argument positions. Then there is a non-self-embedding lnCFTG H' which is unique in argument positions, $\mathcal{L}(H') = \mathcal{L}(H)$, and $\text{pg}(H')$ does not contain top-recursive SCCs, i.e.,* $\text{topRank}(H') = 0$.

Proof. This theorem follows immediately from the repeated application of Lemma 6.4 to H. Since $\text{topRank}(H)$ is finite and in every application the top-recursive rank strictly decreases, the construction terminates. □

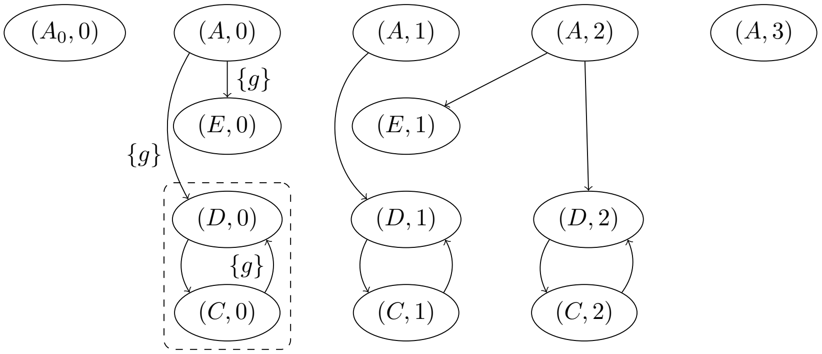


Figure 12:: Part of the position graph of H_1'.

6.2. Transforming Top-Recursion-Free lnCFTG into RTG

In this section, we consider a non-self-embedding lnCFTG H such that $\mathrm{pg}(H)$ does not contain a top-recursive SCC. Hence, for each rule r in H, exactly one of the following three cases holds:

- $r \in \mathrm{rules}(P)$ for some bottom-recursive SCC P,
- $r \in \mathrm{rules}(P)$ for some SCC P which is not generating, and r is of the form $A(x_{1..k}) \to B(x_{j_1}, \ldots, x_{j_k})$ where $j_1 \ldots j_k$ is a permutation of $[k]$,
- $r \in R \setminus (\bigcup_{P \in \mathrm{scc}(\mathrm{pg}(H))} \mathrm{rules}(P))$, i.e., r is not in the rules of any SCC and thus, r is not involved in any recursion.

We note that each non-self-embedding lnCFTG H such that $\mathrm{pg}(H)$ does not contain a top-recursive SCC, is unique in argument positions.

Lemma 6.6 *Let H be a non-self-embedding lnCFTG and* $\mathrm{topRank}(H) = 0$. *We can construct a RTG G' such that $\mathcal{L}(H) = \mathcal{L}(G')$.*

Proof. Let $H = (N, \Delta, A_0, R)$. We will show that, since H contains no top-recursion, in any outside-in derivation, no unboundedly large trees will occur in any argument position of any nonterminal. In other words, the set

$$K = \{\langle \xi|_w \rangle \mid A_0 \Rightarrow_d \xi,\ w \in \mathrm{pos}_N(\xi),\ d \text{ is outside-in}\}$$

is finite.

We will construct K and use its elements as nonterminals of the RTG G'. The subtrees from the nonterminals in K can then be used to construct rules that derive the same language as H.

As an auxiliary tool we define, for every $\xi \in \mathrm{T}_{N \cup \Delta}(X)$,

$$\mathrm{cut}_N(\xi) = \{\langle \xi|_v \rangle \mid v \in \mathrm{pos}_N(\xi), v \text{ outermost in } \mathrm{pos}_N(\xi)\} \ .$$

We let $P_0 = \{\langle A_0 \rangle\}$ and define inductively, for each $i \in \mathbb{N}$,

$$P_{i+1} = P_i \cup \bigcup_{\substack{\langle \xi \rangle \in P_i \\ r \in R|_{\xi(\varepsilon)}}} \operatorname{cut}_N(\operatorname{rhs}(r)[\xi|_1, \ldots, \xi|_\ell])$$

where in each case $\ell = \operatorname{rk}_N(\xi(\varepsilon))$. Note that P_i is finite for every $i \in \mathbb{N}$.

Claim 1: There is an $n \in N$ such that $K = P_n = P_{n+1}$.
Proof of Claim 1: Since H is not top-recursive, each SCC P involving (B, j) with $j \neq 0$ is non-generating, i.e., considering all outside-in derivations using rules from rules(P), the set of trees generated below a nonterminal is finite. Hence, an item in K is a nonterminal plus a choice of argument values drawn from a finite set. We use a saturation process to find all relevant argument values.

Furthermore, since $P_0 = \{\langle A_0 \rangle\}$, it can be seen that $P_n = K$. ◇

Now we let $n \in \mathbb{N}$ be such that $P_n = P_{n+1}$. We construct the desired RTG $G' = (P_n, \Delta, \langle A_0 \rangle, R')$ where R' is defined as follows. For each $\langle \xi \rangle \in P_n$ with $\ell = \operatorname{rk}_N(\xi(\varepsilon))$ and $r \in R|_{\xi(\varepsilon)}$, let $\langle \xi \rangle \to \zeta$ be in R' where ζ is obtained from $\operatorname{rhs}(r)[\xi|_1, \ldots, \xi|_\ell]$ by replacing the subtree at each outermost position $v \in \operatorname{pos}_N(\operatorname{rhs}(r))$ by $\langle \operatorname{rhs}(r)[\xi|_1, \ldots, \xi|_\ell]|_v \rangle$. We denote the rule constructed in this way by $[r, \langle \xi \rangle]$.

Claim 2: For each $m \in \mathbb{N}$ and $\xi \in \mathrm{T}_{N \cup \Delta}$, the following are equivalent.

(i) There is an outside-in derivation d such that $|d| = m$ and $A_0 \Rightarrow_d \xi$.

(ii) There is a $\xi' \in \mathrm{T}_\Delta(K)$ and a derivation d' such that $|d'| = m$, $\langle A_0 \rangle \Rightarrow_{d'} \xi'$, and $\xi = \operatorname{removeBrackets}(\xi')$ where $\operatorname{removeBrackets}(\xi')$ is obtained from ξ' by replacing each position labeled $\langle B(\xi_{1..\ell}) \rangle$ by the tree $B(\xi_{1..\ell})$.

Proof of Claim 2: We assume that d and d' are of the form

$$\begin{aligned} d\colon\ & A_0 = \zeta_0 \Rightarrow_{r_1} \zeta_1 \Rightarrow_{r_2} \ldots \Rightarrow_{r_{m-1}} \zeta_{m-1} \Rightarrow_{r_m} \xi \quad \text{and} \\ d'\colon\ & \langle A_0 \rangle = \zeta'_0 \Rightarrow_{[\tilde{r}_1, \langle \xi_1 \rangle]} \zeta'_1 \Rightarrow_{[\tilde{r}_2, \langle \xi_2 \rangle]} \ldots \Rightarrow_{[\tilde{r}_{m-1}, \langle \xi_{m-1} \rangle]} \zeta'_{m-1} \Rightarrow_{[\tilde{r}_m, \langle \xi_m \rangle]} \xi' \ . \end{aligned}$$

(i)⇒(ii): Assume that, for each $i \in [m]$, the rule r_i is applied at position w_i in ζ_{i-1}. We obtain d' by defining, for each $i \in [m]$, that $[\tilde{r}_i, \langle \xi_i \rangle] = [r_i, \zeta'_{i-1}(w_i)]$. Then, we can show by induction that, for each $i \in [m]$, we have $\zeta_{i-1} = \operatorname{removeBrackets}(\zeta'_{i-1})$. Furthermore, since $A_0 \Rightarrow_{r_{1..(i-1)}} \zeta_{i-1}$ holds we have by Claim 1 that $\langle \zeta_{i-1}|_{w_i} \rangle \in P_n$ and it can be seen that $\langle \zeta_{i-1}|_{w_i} \rangle = \zeta'_{i-1}(w_i)$.

(ii)⇒(i): For each $i \in [m]$, we can define $r_i = \tilde{r}_i$. It can be shown by induction on m that, for each $i \in [m]$, we have $\zeta_{i-1} = \operatorname{removeBrackets}(\zeta'_{i-1})$. ◇

As discussed in Section 3, the rule applications in a lnCFTG can be reordered without changing the language. Hence, any derivation d can be turned into an outside-in derivation. Thus, by Claim 2, we have $\mathcal{L}(H) = \mathcal{L}(G')$. □

We illustrate the construction of this section by considering the following lnCFTG

$$H_2 = (\{A_0^{(0)}, A^{(2)}\}, \{\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)}, \delta^{(2)}, \kappa^{(2)}\}, A_0, R)$$

where R contains the rules

$$A_0 \to A(\alpha,\beta) \qquad \text{and} \qquad A(x_1,x_2) \to \begin{array}{c} \delta \\ / \ \backslash \\ x_1 \quad A \\ \quad / \ \backslash \\ \quad x_2 \quad \gamma \end{array} \left| \begin{array}{c} \kappa \\ / \ \backslash \\ x_1 \ x_2 \end{array} \right. .$$

Applying the construction from the proof of Lemma 6.6 to H_2 yields the RTG G_2' with the rules

$$\langle A_0 \rangle \to \left\langle \begin{array}{c} A \\ / \ \backslash \\ \alpha \quad \beta \end{array} \right\rangle , \qquad \left\langle \begin{array}{c} A \\ / \ \backslash \\ \alpha \quad \beta \end{array} \right\rangle \to \begin{array}{c} \delta \\ / \quad \backslash \\ \alpha \quad \left\langle \begin{array}{c} A \\ / \ \backslash \\ \beta \quad \gamma \end{array} \right\rangle \end{array} \left| \begin{array}{c} \kappa \\ / \ \backslash \\ \alpha \quad \beta \end{array} \right. ,$$

$$\left\langle \begin{array}{c} A \\ / \ \backslash \\ \beta \quad \gamma \end{array} \right\rangle \to \begin{array}{c} \delta \\ / \quad \backslash \\ \beta \quad \left\langle \begin{array}{c} A \\ / \ \backslash \\ \gamma \quad \gamma \end{array} \right\rangle \end{array} \left| \begin{array}{c} \kappa \\ / \ \backslash \\ \beta \quad \gamma \end{array} \right. , \text{ and } \quad \left\langle \begin{array}{c} A \\ / \ \backslash \\ \gamma \quad \gamma \end{array} \right\rangle \to \begin{array}{c} \delta \\ / \quad \backslash \\ \gamma \quad \left\langle \begin{array}{c} A \\ / \ \backslash \\ \gamma \quad \gamma \end{array} \right\rangle \end{array} \left| \begin{array}{c} \kappa \\ / \ \backslash \\ \gamma \quad \gamma \end{array} \right. .$$

It can be seen that H_2 and G_2' are equivalent.

6.3. Main Theorem

Theorem 6.7 *For each non-self-embedding lnCFTG G, we can construct a RTG G' such that $\mathcal{L}(G) = \mathcal{L}(G')$, i.e., the language $\mathcal{L}(G)$ is regular.*

Proof. By Lemma 5.7 we may assume that G is non-self-embedding and unique in argument positions. Furthermore, according to Theorem 6.5 we can assume that $\operatorname{topRank}(G) = 0$. The application of Lemma 6.6 yields an equivalent RTG G'. Hence, $\mathcal{L}(G)$ is regular. □

7. Relationship to Non-Self-Embedding CFG

On an informal level we relate our result to the corresponding result for the string case. In [2, 3] it was proved that each non-self-embedding context-free (string) grammar (CFG) generates a regular language. In [22] self-embedding was expressed as a syntactic criterion, accompanied by a direct construction of a regular (string) grammar starting from a non-self-embedding CFG.

For the sake of comparing CFG and lnCFTG, we relax the condition that the initial nonterminal of a lnCFTG must be of rank 0. Then, informally, we can view a CFG as a lnCFTG (N, Δ, A_0, R) in which each nonterminal and each terminal has rank 1. Let us call such a lnCFTG *monadic*. Clearly, there is a bijection, say, φ, between Δ^* (where Δ is viewed as a usual alphabet) and $\mathrm{T}_\Delta(\{x_1\})$. Moreover, for each CFG G there is a monadic lnCFTG G' such that $\mathcal{L}(G') = \varphi(\mathcal{L}(G))$, and vice versa, for each monadic lnCFTG G' there is a CFG G such that $\mathcal{L}(G) = \varphi^{-1}(\mathcal{L}(G'))$ (cf. [5, Thm. 7.13] for $n = 1$).

For monadic lnCFTGs, Property (2) of the definition of self-embedding is false. Moreover, Property (1) of that definition corresponds to the definition of self-embedding of CFG given in [3, 22]. Thus, there is a one-to-one correspondence between non-self-embedding CFGs and non-self-embedding monadic lnCFTGs.

A regular (string) grammar (REG) can be viewed as a monadic lnCFTG (N, Δ, A_0, R) in which the RHS of each rule satisfies the property that the subtree below a nonterminal only consists of x_1. Let us call such grammars *monadic RTGs*. There is an obvious one-to-one correspondence between REGs and monadic RTGs: it is the restriction of the above mentioned one-to-one correspondence between CFGs and monadic lnCFTGs (to REGs and monadic RTGs, respectively).

We will now analyze the proof of our Theorem 6.7 when G is a non-self-embedding monadic lnCFTG. Note that in a monadic lnCFTG, every position in the RHS is variable dominating. Hence, each occurrence of a nonterminal in the RHS of a rule induces an edge in the position graph. We compare our construction to the function *make_fa* (cf. [22, Figure 1.3]) of the string case.

The first part of the proof is concerned with the property of being unique in argument positions. In case of a monadic lnCFTG, there are only two argument positions, viz., 0 and 1. By Observation 5.2, we have that those two argument positions are never in the same SCC. Hence, every monadic lnCFTG is unique in argument positions.

The second part of the proof removes top-recursive SCCs. We discuss it for an example rule of a top-recursive SCC P. Let r be a rule $A(x_1) \to B(\gamma(C(x_1)))$ in rules(P), where A and B are both nonterminals in M_P. Removing top-recursion in a monadic lnCFTG is similar to handling left-recursion in the string case. According to (4) from the proof of Lemma 6.4, we reverse r to the rule $r'\colon \langle A, B, \mathbf{d}\rangle(x_1) \to \gamma(C(\langle A, A, \mathbf{d}\rangle(x_1)))$. We compare r' to the output of *make_fa* applied to the rule $A \to B\gamma C$. This yields a rule $q_B \to \gamma C q_A$, which corresponds to r'.

Lastly, we consider the transformation of a monadic non-self-embedding lnCFTG which does not contain top-recursive SCCs into a RTG. This construction corresponds to the application of *make_fa* to A_0 and uses the case for right-recursion.

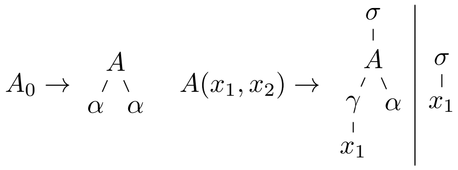
8. Deleting rules

The theory developed in this paper concerns linear nondeleting CFTGs. We now consider linear CFTGs. A *linear CFTG* (lCFTG) is defined exactly as a lnCFTG except that each rule has the form $A(x_{1..k}) \to \xi$ where $\xi \in \mathrm{T}_{N\cup\Delta}(X_k)$ and each variable of X_k occurs at most once in ξ. The derivation relation and the generated tree language are defined exactly as for lnCFTG.

It is known that for each lCFTG we can find an equivalent lnCFTG. The construction can be traced back to [7, Thm. 3.1.10]; see also e.g. [28, Lm. 3.1]. The idea is as follows. Let G be a given lCFTG, to be transformed into a lnCFTG G'. The nonterminals of G' are of the form A_α, where A is a nonterminal of G of rank k, and α is a subset of $[k]$. The argument positions in subscript α are those that will be removed in the transformation from G to G'. For a rule $A(x_{1..k}) \to \xi$ from G, we construct a rule $A_\alpha(x_{k_1}, \ldots, x_{k_d}) \to \xi'$ from G' as follows. Each occurrence of a subtree $B(\xi_{1..m})$ in the RHS ξ is replaced by a subtree $B_\beta(\xi'_{1..m'})$, where β is some subset of $[m]$ and $\xi'_{1..m'}$ are obtained from $\xi_{1..m}$ by omitting the argument positions in β. This gives us the RHS ξ' of the transformed rule. For the LHS, we let α consist of all i such that x_i does not occur in ξ', and we let $x_{k_1}, \ldots, x_{k_d}$ be the sequence of variables in X_k consisting of all those variables that do occur in ξ'. The new rule is clearly nondeleting. This construction is done exhaustively, considering all possible choices of β for each occurrence of a nonterminal B in the RHS.

An obvious question is now whether the results in our paper carry over to all lCFTGs. For this, we first need to extend our definition of self-embedding to lCFTGs that may include deleting rules. We define *self-embedding lCFTG* in the same way as we have defined self-embedding lnCFTG (at the beginning of Section 4). Furthermore, we also define the *position pair graph* for lCFTG exactly as in Definitions 4.1. It can be seen that Theorem 4.3 also holds for lCFTG without any change.

As an example consider the following rules of a self-embedding lCFTG that delete the second argument position of A. The grammar is self-embedding since σ's and γ's are synchronously generated above and below (resp.) the nonterminal A.



All that remains is to show that the above transformation that removes deleting rules preserves the property of being non-self-embedding.

Lemma 8.1 *For each non-self-embedding lCFTG G, we can construct a non-self-embedding lnCFTG G' such that $\mathcal{L}(G) = \mathcal{L}(G')$.*

Proof. We prove this lemma by contraposition and thus assume that the transformation applied on a lCFTG G results in a lnCFTG G' that is self-embedding. Then we have a cycle in ppg(G') from a vertex (A_α, i, j) to (A_α, i, j) such that the union of all labels in the cycle contains 1 and 2. Due to the nature of the transformation, which does no more than systematically remove argument positions, we must then have a cycle in ppg(G) from some vertex (A, i', j') to (A, i', j') such that the union of all labels in the cycle once more contains 1 and 2. The argument positions $i' \geq i$ and $j' \geq j$ are straightforwardly obtained from i and j by accounting for the removed positions as recorded in α. Thereby G must be self-embedding as well. □

We note that the removal of deleting rules (including the removal of useless rules) may turn a self-embedding lCFTG into a non-self-embedding lnCFTG. For instance, if we remove the deleting rule from the lCFTG with the rules $A(x) \to \delta(A(G(x))) \mid x$, and $G(x) \to \alpha$, then we obtain the non-self-embedding lnCFTG with the rules $A_{\{1\}} \to \delta(A_{\{1\}}) \mid \delta(A_\emptyset(G_{\{1\}}))$, $G_{\{1\}} \to \alpha$, and $A_\emptyset(x) \to x$.

As a consequence of Lemma 8.1 and Theorem 6.7 we obtain the following corollary.

Corollary 8.2 *For each non-self-embedding lCFTG G, we can construct a RTG G' such that $\mathcal{L}(G) = \mathcal{L}(G')$, in particular, the language $\mathcal{L}(G)$ is regular.*

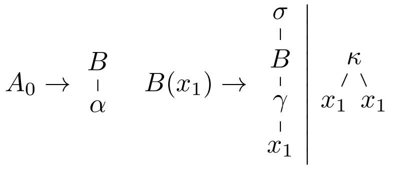
9. Non-Weakly-Self-Embedding CFTG

In the literature, there is another instance of non-self-embedding string grammars. Parchmann and Duske [25] define self-embedding indexed grammars and they show that indexed grammars which are not self-embedding induce context-free languages.

Indexed grammars are related to (arbitrary) CFTG in the following way. The languages induced by indexed grammars are exactly those generated by macro grammars with outside-in derivation mode [7, Thm. 4.2.8] and the latter generate exactly the yields of the languages generated by CFTG with outside-in derivation mode (cf. [26, p. 113] and [5, Thm. 7.17]). Thus, it seems worthwhile to lift the definition of self-embedding from indexed grammars to CFTG. In order not to mix up the resulting property with our property of self-embedding lnCFTG, we call the former *weakly-self-embedding.* We compare self-embedding lCFTG and weakly-self-embedding lCFTG and we prove that each non-weakly-self-embedding CFTG (with outside-in derivation mode) generates a regular tree language. We will not investigate the formal relationship between self-embedding indexed grammars and weakly-self-embedding CFTG.

Formally, a *context-free tree grammar* (CFTG) is defined in exactly the same way as lnCFTG except that each rule has the form $A(x_{1..k}) \to \xi$ where $\xi \in \mathrm{T}_{N\cup\Delta}(X_k)$. In particular, a variable of X_k may occur in ξ more than once (copying) or not at all (deleting). As derivation mode we only consider outside-in. The *tree language* generated by the CFTG $G = (N, \Delta, A_0, R)$ is defined as $\mathcal{L}(G) = \{\xi \in \mathrm{T}_\Delta \mid A_0 \Rightarrow_d \xi,\ d \text{ is outside-in}\}$.

As an example, consider the following rules of a CFTG. The variable x_1 appears twice in the last rule and thus, the tree in the argument position of B is copied.

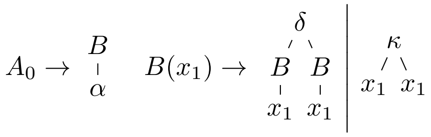


A CFTG $G = (N, \Delta, A_0, R)$ is *weakly-self-embedding* if there are $k \in \mathbb{N}$, $A \in N^{(k)}$, $i \in [k]$, $\xi \in \mathrm{C}_{N\cup\Delta\cup X_k}(\{z\})$, and $\xi_{1..k} \in \mathrm{T}_{N\cup\Delta}(X_k)$ such that

- $A(\overline{x}) \Rightarrow^* \xi[A(\xi_{1..k})]$,
- ξ_i contains x_i, and
- $\xi_i \neq x_i$.

Intuitively, the ith argument position of A corresponds to the string of indices attached to nonterminal A of an indexed grammar.

We give an example of a CFTG that is non-weakly-self-embedding and has two rules that copy the first argument of B.



It can be seen that in every derivation starting from A_0, the variable x_1 always denotes the subtree α. Hence, in such derivations there is a unique tree that may appear below any occurrence of B. This will help us later to determine regularity of the induced tree language.

Similar to Theorem 4.3 it is decidable whether a CFTG G is weakly-self-embedding. Since weakly-self-embedding is a property of individual argument positions, independent of other positions of the same nonterminal, it suffices to consider the position graph of G. For an arbitrary CFTG this is defined in exactly the same way as in Definition 5.1.

Lemma 9.1 *A CFTG G is weakly-self-embedding iff* $\mathrm{pg}(G)$ *contains a top-recursive SCC.*

Proof. The proof is very similar to the proof of Theorem 4.3, but only considers one argument position per nonterminal. □

Next we compare weakly-self-embedding lCFTG and self-embedding lCFTG. By analyzing the definitions we obtain the following inclusion.

Observation 9.2 *Each self-embedding lCFTG is also weakly-self-embedding.*

Conversely, there are non-self-embedding lCFTG which are weakly-self-embedding. As an example consider the following rules of a non-self-embedding lnCFTG.

$$A_0 \to \begin{array}{c} B \\ \mid \\ \alpha \end{array} \qquad B(x_1) \to \begin{array}{c} B \\ \mid \\ \gamma \\ \mid \\ x_1 \end{array} \left| \begin{array}{c} \sigma \\ \mid \\ x_1 \end{array} \right.$$

This grammar is weakly-self-embedding, since $B(x_1) \Rightarrow B(\gamma(x_1))$.

Now we prove that each non-weakly-self-embedding CFTG generates a regular tree language.

Theorem 9.3 *For each non-weakly-self-embedding CFTG G, we can construct a RTG G' such that $\mathcal{L}(G) = \mathcal{L}(G')$, i.e., the language $\mathcal{L}(G)$ is regular.*

Proof. We let $G = (N, \Delta, A_0, R)$ be a non-weakly-self-embedding CFTG. It is easy to see that $\mathrm{topRank}(G) = 0$. Thus there cannot be unbounded generation of symbols in argument positions of nonterminals. It can now be seen that the set

$$K = \{\langle \xi|_w \rangle \mid A_0 \Rightarrow_d \xi,\ w \in \mathrm{pos}_N(\xi),\ d \text{ is outside-in}\}$$

is finite.

Using this fact, we can apply the construction in the proof of Lemma 6.6 to obtain a RTG G' that is equivalent to G. Note that, for Claim 2 of the proof of Lemma 6.6, a reordering of derivations of G is not required because we only consider outside-in derivations. □

10. Summary and Overview

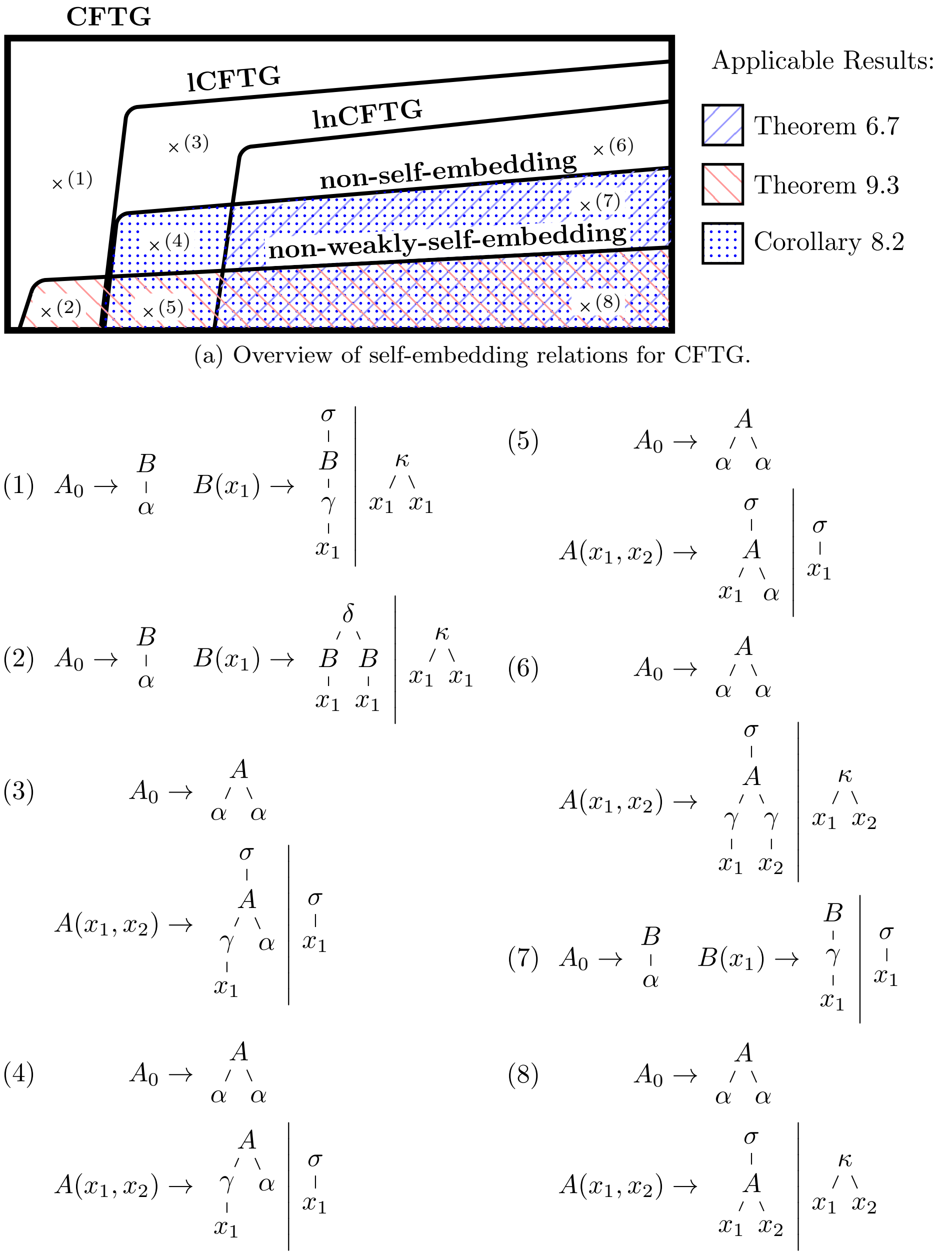
We have defined a subclass of lCFTG called self-embedding lCFTG and proved that it is decidable whether a given lCFTG is self-embedding. Each non-self-embedding lCFTG induces a regular tree language. This is a generalization of the original result for CFG from [2] to the realm of trees.

Moreover, we have defined a subclass of (the full class) CFTG called weakly-self-embedding CFTG; again this is a decidable property. The subclass is inspired by self-embedding indexed grammars [25]. We have proved that each non-weakly-self-embedding CFTG induces a regular tree language.

All mentioned syntactic subclasses of CFTG can be found in Figure 13(a); additionally we have indicated those subclasses for which we could prove that the grammars induce regular tree languages (shaded areas). Moreover, for each class we show an example grammar in Figure 13(b).

Acknowledgment

The authors wish to thank an anonymous referee of a previous submission for suggesting a more succinct representation of the proof of Lemma 6.4. We further wish to thank both referees of the current submission for helpful suggestions which improved the presentation.



(a) Overview of self-embedding relations for CFTG.

(b) Examples for each subclass of Figure 13(a).

Figure 13:: An overview over the classes of CFTG.

References

[1] W. S. Brainerd, Tree generating regular systems. *Information and Control* **14** (1969) 2, 217–231.

[2] N. Chomsky, A note on phrase structure grammars. *Information and Control* **2** (1959) 4, 393–395.

[3] N. Chomsky, On Certain Formal Properties of Grammars. *Information and Control* **2** (1959) 2, 137–167.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*. 3rd edition, The MIT Press, 2009.

[5] W. Damm, The IO- and OI-hierarchies. *Theoretical Computer Science* **20** (1982) 2, 95–207.

[6] J. Engelfriet, E. Schmidt, IO and OI. I. *Journal of Computer and System Sciences* **15** (1977) 3, 328–353.

[7] M. Fischer, *Grammars with macro-like productions*. Ph.D. thesis, Harvard University, Massachusetts, 1968.

[8] A. Fujiyoshi, Restrictions on Monadic Context-Free Tree Grammars. In: *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*. 2004, 78–84.
`http://aclweb.org/anthology/C04-1012`

[9] A. Fujiyoshi, Linearity and nondeletion on monadic context-free tree grammars. *Information Processing Letters* **93** (2005) 3, 103–107.

[10] A. Fujiyoshi, T. Kasai, Spinal-formed context-free tree grammars. *Theory of Computing Systems* **33** (2000) 1, 59–83.

[11] K. Gebhardt, J. Osterholzer, A Direct Link between Tree-Adjoining and Context-Free Tree Grammars. In: T. Hanneforth, C. Wurm (eds.), *Proceedings of the 12th International Conference on Finite-State Methods and Natural Language Processing (FSMNLP)*. 2015.
`http://aclweb.org/anthology/W15-4805`

[12] F. Gécseg, M. Steinby, Tree languages. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages*. 3, Springer-Verlag, 1997, 1–68.

[13] J. Hopcroft, J. Ullman, *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Cambridge, 1979.

[14] A. 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.), *Natural Language Parsing*. Cambridge University Press, 1985, 206–250.

[15] A. Joshi, Y. Schabes, Tree-adjoining grammars. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages*. 3, Springer-Verlag, 1997, 69–123.

[16] D. Jurafsky, J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, 2000.

[17] M. Kanazawa, A generalization of linear indexed grammars equivalent to simple context-free tree grammars. In: G. Morrill, R. Muskens, R. Osswald, F. Richter (eds.), *Formal Grammar*. Lecture Notes in Computer Science 8612, Springer, 2014, 86–103.

[18] M. Kanazawa, Multidimensional trees and a Chomsky-Schützenberger-Weir representation theorem for simple context-free tree grammars. *Journal of Logic and Computation* (2014).

[19] S. Kepser, U. Mönnich, Closure properties of linear context-free tree languages with an application to optimality theory. *Theoretical Computer Science* **354** (2006) 1, 82–97. Algebraic Methods in Language Processing Third International AMAST Workshop on Algebraic Methods in Language Processing 2003.

[20] S. Kepser, J. Rogers, The equivalence of tree adjoining grammars and monadic linear context-free tree grammars. *Journal of Logic, Language and Information* **20** (2011) 3, 361–384.

[21] A. Maletti, J. Engelfriet, Strong Lexicalization of Tree Adjoining Grammars. In: H. Li, C.-Y. Lin, M. Osborne, G. G. Lee, J. C. Park (eds.), *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012, 506–515.

[22] M.-J. Nederhof, Regular approximation of CFLs: A grammatical view. In: H. Bunt, A. Nijholt (eds.), *Advances in Probabilistic and Other Parsing Technologies*. Text, Speech and Language Technology 16, Springer Netherlands, 2000, 221–241.

[23] M.-J. Nederhof, H. Vogler, Synchronous Context-Free Tree Grammars. In: *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*. 2012, 55–63.

[24] J. Osterholzer, Pushdown Machines for Weighted Context-Free Tree Translation. In: M. Holzer, M. Kutrib (eds.), *Proceedings of 19th International Conference on Implementation and Application of Automata (CIAA 2014)*. Lecture Notes in Computer Science 8587, 2014, 290–303.

[25] R. Parchmann, J. Duske, Self-embedding indexed grammars. *Theoretical Computer Science* **47** (1986), 219–223.

[26] W. C. Rounds, Tree-oriented Proofs of Some Theorems on Context-free and Indexed Languages. In: *Proceedings of the Second Annual ACM Symposium on Theory of Computing*. STOC '70, 1970, 109–116.

[27] S. Shieber, Evidence against the context-freeness of natural language. *Linguistics and Philosophy* **8** (1985) 3, 333–343.

[28] H. Stamer, *Restarting Tree Automata. Formal Properties and Possible Variations*. kassel university press GmbH, 2008.

[29] K. Vijay-Shanker, D. Weir, A. Joshi, Characterizing Structural Descriptions Produced by Various Grammatical Formalisms. In: *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1987, 104–111.