Automata accepting coupled-context-free languages

Peeter Laud
Institute of Computer Science
Tartu University
Liivi 2, 50409 TARTU, ESTONIA
e-mail peeter_l@ut.ee

Abstract

Coupled-context-free grammars are a natural generalization of context-free grammars by combining non-terminals to parenthesizes which can only be substituted together. They are a class of *mildly context-sensitive* languages, being semilinear and polynomial time parseable, but also providing multiple agreements, crossed agreements and duplication, thus they may be useful in analyzing natural languages.

We define a class of automata and prove that the class of languages accepted by them is exactly the class of coupled-context-free languages.

1 Introduction

The coupled-context-free grammars, a generalization of context-free grammars, have been introduced by G. Hotz and investigated by Y. Guan [Gua92]. This class has many good properties of context-free grammars, for example being closed under union, concatenation, homomorphisms, semilinearity, etc. G. Pitsch has given polynomial-time parsing algorithms for this class of grammars [Pit93, HP96], but we are not aware of any automata-theoretic results similar to ours. In [Gua92] the construction of a class of automata accepting coupled-context-free languages was stated as an open question.

The tree-generating automata, defined in this paper, are basically pushdown automata ([Cho62], see e.g. [Har78]) with some additional details. Similarly to coupled-context-free grammars, which place restrictions on context-free derivations, tree-generating automata place restrictions on the computations of underlying pushdown automaton. We show that these restrictions are "equivalent", i.e. the generating power of coupled-context-free grammars and the accepting power of tree-generating automata are the same.

2 Preliminaries

Let $\mathbb{N}$ denote the set of positive integers, let $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Let $[i..j]$, where $i \leq j$ denote the set $\{i, i+1, \ldots, j\}$. We denote the set inclusion by $\subseteq$ and strict inclusion by $\subset$. All trees considered in this article are rooted and the descendants of its vertices are ordered. Let V be a finite set called *alphabet*. *A word over the alphabet* V is a finite sequence of the elements — letters — of V. The set of all words over V is denoted by V^*. The set V^* equipped with concatenation is a monoid which unit element is the empty word ϵ. We denote the i-th letter of a word w by $w(i)$. For any expression $\mathcal{A}(i)$, where $i \in [1..n]$, the notation $(\mathcal{A}(i))_{i=1}^n$ means the sequence $(\mathcal{A}(1), \ldots, \mathcal{A}(n))$.

Definition 1. Let K be a finite set and let $g : K \longrightarrow \mathbb{N}$ be a mapping. The set

$$[K, g] := \{(x, n) \in K \times \mathbb{N} \,|\, 1 \leq n \leq g(x)\}$$

is called *bracket alphabet* and g its *arity function*.

We denote the elements (x, n) of $[K, g]$ also (mainly) by x_n. If the arity of some $x \in K$ is 1, then we do not distinguish x and its only component $(x, 1)$.

Definition 2. The *bracket language* $\mathrm{D}(K, g)$ over bracket alphabet $[K, g]$ is the smallest subset of $[K, g]^*$ satisfying the following conditions:

1. $\epsilon \in \mathrm{D}(K, g)$;
2. $u, v \in \mathrm{D}(K, g)$ implies $uv \in \mathrm{D}(K, g)$;
3. $x \in K$, $g(x) = k$, $w_1, \ldots, w_{k-1} \in \mathrm{D}(K, g)$ implies $x_1 w_1 x_2 w_2 \cdots x_{k-1} w_{k-1} x_k \in \mathrm{D}(K, g)$.

Example 1. Let $K = \{x, y, z\}$ and $g(x) = 1, g(y) = 2, g(z) = 3$, then the following words are elements of $\mathrm{D}(K, g)$:

$$\epsilon, \quad y_1 y_1 y_2 y_2, \quad y_1 z_1 x z_2 y_1 y_2 x z_3 y_2.$$

Example 2. Let $K = \{x_1, \ldots, x_n\}$. If for all $i \in [1..n]$ holds $g(x_i) = 2$ then the set $\mathrm{D}(K, g)$ is just the semi-Dyck language over bracket pairs

$$\{(x_1, 1), (x_1, 2), (x_2, 1), (x_2, 2), \ldots, (x_n, 1), (x_n, 2)\}.$$

Let $X \subseteq \mathbb{N}_0$, $k \in \mathbb{N}_0$. Denote $X + k := \{x + k \mid x \in X\}$.

Definition 3. Let $[K, g]$ be a bracket alphabet and let $w \in \mathrm{D}(K, g)$. The *w-partitioning* is a partitioning $\mathcal{S}(w)$ of the set $[1..|w|]$ defined inductively by

1. $\mathcal{S}(\epsilon) := \emptyset$;
2. $\mathcal{S}(uv) := \mathcal{S}(u) \cup \{X + |u| \mid X \in \mathcal{S}(v)\}$ for $u, v \in \mathrm{D}(K, g)$;
3. Let $x \in K, g(x) = k, w_1, \ldots, w_{k-1} \in \mathrm{D}(K, g)$. Then

$$\mathcal{S}(x_1 w_1 x_2 w_2 \cdots x_{g(x)-1} w_{g(x)-1} x_{g(x)}) :=$$
$$\{\{i + \sum_{j=1}^{i-1} |w_j| \mid i \in [1..g(x)]\}\} \cup \bigcup_{i=1}^{k-1} \{X + (i + \sum_{j=1}^{i-1} |w_j|) \mid X \in \mathcal{S}(w_i)\}.$$

The w-partitioning describes, which bracket components in the word w are corresponding components.

Example 3. For $[K, g]$ and words from example 1 the partitionings defined by those words are respectively

$$\emptyset, \quad \{\{1, 4\}, \{2, 3\}\}, \quad \{\{1, 9\}, \{2, 4, 8\}, \{3\}, \{5, 6\}, \{7\}\}.$$

Definition 4. *Coupled-context-free grammar* (CCFG) is quadruple $G = ([V, g], \Sigma, P, s)$ where

- $[V, g]$ is a bracket alphabet with $V = N \dot{\cup} \Sigma$, $N, \Sigma \neq \emptyset$ and the arity of all elements of Σ equal to 1.
- P is a finite set of *coupled-context-free productions* $(x_1, \ldots, x_k) \to (y_1, \ldots, y_k)$, where $x \in N$, $g(x) = k$, $y_1, \ldots, y_k \in [V, g]^*$,$y_1 \cdots y_k \in \mathrm{D}(V, g)$.
- $s \in N$, $g(s) = 1$, is the *axiom* of the grammar.

Definition 5. Let $G = ([V, g], \Sigma, P, s)$ be a CCFG and $w_1, w_2 \in \mathrm{D}(V, g)$. w_2 is *directly derivable* from w_1 by G (denoted $w_1 \Rightarrow w_2$) if $w_1 = u_1 x_1 v_1 x_2 v_2 \cdots x_{k-1} v_{k-1} x_k u_2$ and $w_2 = u_1 y_1 v_1 y_2 v_2 \cdots y_{k-1} v_{k-1} y_k u_2$, where $u_1, u_2 \in [V, g]^*$, $v_1, \ldots, v_{k-1} \in \mathrm{D}(V, g)$ and $(x_1, \ldots, x_k) \to (y_1, \ldots, y_k) \in P$.

I.e. the nonterminals $x_1, \ldots, x_{g(x)}$ in w_1 can be substituted together iff they are the corresponding bracket components.

Let $\overset{*}{\Rightarrow}$ be the reflexive transitive closure of $\Rightarrow$. The language generated by G is

$$\mathcal{L}(G) := \{w \in \Sigma^* \mid s \overset{*}{\Rightarrow} w\}.$$

Thus CCFGs are a generalization of context-free grammars (context-free grammars are just CCFGs where the maximum arity of nonterminals is 1). In [GHR92, Pit93, HP96] the *context-free base grammar* of a CCFG is defined, by considering $[V,g]$ as an ordinary alphabet and replacing each production $(x_1,\dots,x_k)\to(y_1,\dots,y_k)$ by (context-free) productions $x_1\to y_1,\dots,x_k\to y_k$. Clearly, every derivation in some CCFG is a derivation in its context-free base grammar, but the opposite generally doesn't hold.

Example 4. The coupled-context-free grammar

$$([\{s,A,a,b,c,d\},g],\{a,b,c,d\},\\ \{s\to A_1A_2,(A_1,A_2)\to(aA_1b,cA_2d),(A_1,A_2)\to(\epsilon,\epsilon)\},s)$$

where $g(A)=2$, generates the language $\{a^nb^nc^nd^n\mid n\in\mathbb{N}_0\}$. The non-context-freeness of this language is a well-known fact (e.g. [Har78]).

3 Definition of tree-generating automata

Definition 6. Let $[K,g]$ be bracket alphabet. *Bracketed tree over* $[K,g]$ is a tree whose vertices are labeled with the elements of $[K,g]$, together with a partitioning on the set of its vertices, defined inductively as follows:

1. One vertex, labeled with $x\in K$ where $g(x)=1$, is bracketed tree over $[K,g]$. The partitioning of vertices contains one class, containing that vertex.

2. Let T be a bracketed tree over $[K,g]$, let Z be one of the classes of the corresponding partitioning, such that the elements of Z lie on the frontier of T. Let $w\in\mathrm{D}(K,g)$. Let T' be a tree constructed from T by adding $|w|$ vertices $v_1,\dots,v_{|w|}$ to T, such that each new vertex is a descendant of some vertex in Z and $1\le i<j\le|w|$ implies that v_i lies to the left from v_j. Moreover, for each $i\in[1..|w|]$ label v_i by $w(i)$ and define the partitioning of vertices of T' as

$$\mathcal{T}\cup\{\{v_n\mid n\in X\}\mid X\in\mathcal{S}(w)\},$$

where $\mathcal{T}$ is the partitioning of vertices of T.

Actually, the partitioning of vertices is uniquely defined by the labeling of vertices. The partitioning was introduced only to make the description of valid labelings simpler.

An important class of bracketed trees are the *derivation trees* of CCFGs. Let $G=([V,g],\Sigma,P,s)$ be a CCFG. Derivation tree of G is a bracketed tree over $[V\dot\cup P,g]$ where for each $x\to y\in P$ is defined $g(x\to y):=g(x)$ with the following restrictions:

- The leaves of the tree are labeled with elements of $[V,g]$, other vertices by the elements of $[P,g]$.

- For every $x_i\in[V,g]$ denote

$$\mathrm{LS}(x_i)=\{x_i\}\cup\{p_i\in[P,g]\mid \text{ left side of production } p \text{ is } x\}.$$

  If some vertex of the tree is labeled with some $p_i\in[P,g]$ where $p=(x_1,\dots,x_{g(x)})\to(y_1,\dots,y_{g(x)})$, then this vertex has exactly $|y_i|$ descendants and for each $j\in[1..|y_i|]$ the j-th descendant is labeled by an element of $\mathrm{LS}(y_i(j))$.

A *parse tree* of some CCFG is a derivation tree whose leaves are labeled by terminal symbols.

For a derivation tree T of CCFG $([V,g],\Sigma,P,s)$ let $\mathrm{Root}(T)$ denote the bracket component $x_i \in [V,g]$, such that the label of the root vertex of T is an element of $\mathrm{LS}(x_i)$. If T is a parse tree, then denote its frontier by $\mathrm{Fr}(T)$.

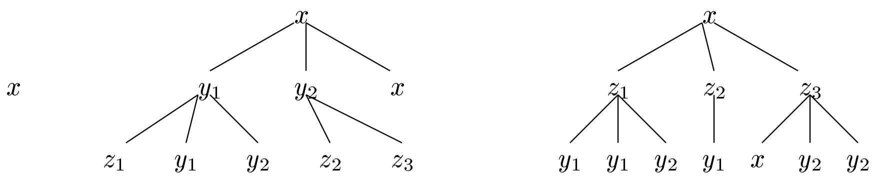
Every tree T whose vertices are labeled with the elements of some alphabet Σ can be expressed as a word $\mathbf{wrd}(T) \in (\Sigma \dot\cup \{\llparenthesis\})^*$ as follows:

1. If T consists of one vertex labeled with $x \in \Sigma$, then $\mathbf{wrd}(T) := x$.

2. If the root of T is labeled with $x \in \Sigma$ and it has n descendants which are the roots of trees $T_1, \dots, T_n$, then $\mathbf{wrd}(T) := x\llparenthesis^n \mathbf{wrd}(T_1) \cdots \mathbf{wrd}(T_n)$.

Let $\mathrm{Tr}(K,g) := \{\mathbf{wrd}(T) \,|\, T \text{ is bracketed tree over } [K,g]\}$. According to the remarks after def. 6, this is a well-formed definition.

Example 5. Let $K = \{x,y,z\}$ and $g(x)=1, g(y)=2, g(z)=3$. Following trees are bracketed trees over $[K,g]$.



They can be written down as x, $x\llparenthesis^3 y_1 \llparenthesis^3 z_1 y_1 y_2 y_2 \llparenthesis^2 z_2 z_3 x$ and $x\llparenthesis^3 z_1 \llparenthesis^3 y_1 y_1 y_2 z_2 \llparenthesis y_1 z_3 \llparenthesis^3 x y_2 y_2$, respectively.

In the rest of this section we define a new class of automata and how they accept their input.

Definition 7. *Tree-generating automaton* (TGA) is octuple $(Q,\Sigma,\Gamma,[K,g],\delta,q_0,Z_0,F)$, where

- Q is the set of *states*;
- Σ is the *terminal alphabet*;
- Γ is the *stack alphabet*;
- $[K,g]$ is the *tree alphabet*;
- $q_0 \in Q$ is the *initial state*;
- $Z_0 \in \Gamma$ is the *bottom symbol of stack*;
- $F \subseteq Q$ is the set of *final states*;
- $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \longrightarrow \mathcal{P}(Q \times \Gamma^* \times [K,g])$ is the *transition function.*

Definition 8. The *configuration* of TGA $(Q,\Sigma,\Gamma,[K,g],\delta,q_0,Z_0,F)$ is quadruple (q,w,z,t) where $q \in Q$, $w \in \Sigma^*$, $z \in \Gamma^*$ and $t \in ([K,g]\dot\cup\{\llparenthesis\})^*$.

Definition 9. Let $C_1 = (q_1, w_1, z_1, t_1)$ and $C_2 = (q_2, w_2, z_2, t_2)$ be two configurations of TGA $(Q, \Sigma, \Gamma, [K, g], \delta, q_0, Z_0, F)$. The state C_2 is *directly derivable* from the state C_1 (denoted $C_1 \vdash C_2$) if there exist such $a \in \Sigma \cup \{\epsilon\}$, $x \in [K, g]$, $u, v \in \Gamma^*$ and $z' \in \Gamma$, that

$$\begin{aligned} w_1 &= a w_2, & z_1 &= u z', \\ t_2 &= t_1 x \llparenthesis^{|v|}, & z_2 &= uv, \\ &(q_2, v, x) \in \delta(q_1, a, z'). \end{aligned}$$

Let $\overset{*}{\vdash}$ be the reflexive transitive closure of $\vdash$.

Definition 10. Tree-generating automaton $(Q, \Sigma, \Gamma, [K, g], \delta, q_0, Z_0, F)$ *accepts the word* $w \in \Sigma$

- *... by empty stack* if there exist such $q \in Q$ and $t \in \mathrm{Tr}(K, g)$, that $(q_0, w, Z_0, \epsilon) \overset{*}{\vdash} (q, \epsilon, \epsilon, t)$;
- *... by final state* if there exist such $q \in F$, $z \in \Gamma^*$, $t \in \mathrm{Tr}(K, g)$ and $u, v \in ([K, g] \cup \{\llparenthesis\})^*$, that $t = uv$ and $(q_0, w, Z_0, \epsilon) \overset{*}{\vdash} (q, \epsilon, z, u)$;
- *... by empty stack and final state* if there exist such $q \in F$ and $t \in \mathrm{Tr}(K, g)$, that $(q_0, w, Z_0, \epsilon) \overset{*}{\vdash} (q, \epsilon, \epsilon, t)$.

Note that if the arity of each element in tree alphabet of TGA is 1, then this TGA is basically just a pushdown automaton.

Let $A = (Q, \Sigma, \Gamma, [K, g], \delta, q_0, Z_0, F)$ be a TGA. Denote

$$\begin{aligned} \mathcal{L}_{\mathcal{E}}(A) &:= \{w \in \Sigma^* \mid A \text{ accepts } w \text{ by empty stack}\}; \\ \mathcal{L}_{\mathcal{F}}(A) &:= \{w \in \Sigma^* \mid A \text{ accepts } w \text{ by final state}\}; \\ \mathcal{L}(A) &:= \{w \in \Sigma^* \mid A \text{ accepts } w \text{ by empty stack and final state}\}. \end{aligned}$$

Example 6. The tree-generating automaton

$$(\{q_0, q_1, q_2, q_3, f\}, \{a, b, c, d\}, \{S, X\}, [\{p, r\}, g], \delta, q_0, S, \{f\}),$$

where $g(p) = 1$, $g(r) = 2$ and

$$\begin{aligned} \delta(q_0, a, S) &= \{(q_0, SX, p)\} & \delta(q_0, a, X) &= \{(q_0, XX, r_1)\} \\ \delta(q_0, b, X) &= \{(q_1, \epsilon, r_1)\} & \delta(q_1, b, X) &= \{(q_1, \epsilon, p)\} \\ \delta(q_1, c, S) &= \{(q_2, SX, r_2)\} & \delta(q_2, c, X) &= \{(q_2, XX, r_2)\} \\ \delta(q_2, d, X) &= \{(q_3, \epsilon, p)\} & \delta(q_3, d, X) &= \{(q_3, \epsilon, p)\} \\ \delta(q_3, \epsilon, S) &= \{(f, \epsilon, p)\}, \end{aligned}$$

accepts the language $\{a^n b^n c^n d^n \mid n \in \mathbb{N}\}$ by empty stack and final state.

4 Equivalence of different modes of accepting

Proposition 1. *Let $A_1 = (Q, \Sigma, \Gamma, [K, g], \delta, q_0, Z_0, F)$ be a TGA. There exists a TGA A_2, such that $\mathcal{L}_{\mathcal{E}}(A_1) = \mathcal{L}_{\mathcal{F}}(A_2) = \mathcal{L}(A_2)$.*

Proof. Construct A_2 as follows:

$$A_2 := (Q\dot{\cup}\{q_0', f\}, \Sigma, \Gamma\dot{\cup}\{Y_0\}, [K\dot{\cup}\{k_0\}, g], \delta', q_0', Y_0, \{f\}),$$

where $g(k_0) := 1$ and δ' is defined as follows:

1. $\delta'(q_0', \epsilon, Y_0) := \{(q_0, Y_0Z_0, k_0)\}$;
2. $\delta'(q, a, z) := \delta(q, a, z)$ for $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$, $z \in \Gamma$;
3. $\delta'(q, \epsilon, Y_0) := \{(f, \epsilon, k_0)\}$ for $q \in Q$;
4. the value of δ' is equal to $\emptyset$ in other cases.

Every possible computation of A_2 ending in the state f has the following form:

1. A_2 goes to the initial configuration of A_1 with an extra symbol Y_0 at the bottom of the stack;
2. A_2 simulates A_1 until the stack of A_1 becomes empty;
3. A_2 goes to the state f and deletes the last symbol (which is Y_0) from the stack.

Thus $\mathcal{L}_{\mathcal{E}}(A_1) = \mathcal{L}_{\mathcal{F}}(A_2)$ and, as A_2 can go to the state f only while deleting the last symbol from its stack, also $\mathcal{L}_{\mathcal{F}}(A_2) = \mathcal{L}(A_2)$. □

Proposition 2. *Let $A_1 = (Q, \Sigma, \Gamma, [K, g], \delta, q_0, Z_0, F)$ be a TGA. There exists a TGA A_2, such that $\mathcal{L}_{\mathcal{F}}(A_1) = \mathcal{L}_{\mathcal{E}}(A_2)$.*

Proof. Let $\bar{g} = \max_{x\in K} g(x)$ and $\bar{z} = \max\{|v| \mid (q_2, v, t) \in \delta(q_1, a, z)\}$ for $q_1, q_2 \in Q$, $t \in [K, g]$, $a \in \Sigma \cup \{\epsilon\}$ and $z \in \Gamma$. Construct A_2 as follows:

$$A_2 := (Q\dot{\cup}\{q_0', f\}, \Sigma, \Gamma\dot{\cup}\{Y_0, Z_f\}, [K\dot{\cup}\{k_0\}, g], \delta', q_0', Y_0, \emptyset),$$

where $g(k_0) := 1$ and δ' is defined as follows:

1. $\delta'(q_0', \epsilon, Y_0) := \{(q_0, Y_0Z_0, k_0)\}$;
2. $\delta'(q, a, z) := \delta(q, a, z)$ for $q \in Q\backslash F$, $a \in \Sigma \cup \{\epsilon\}$ and $z \in \Gamma$;
3. $\delta'(q, a, z) := \delta(q, a, z)$ for $q \in F$, $a \in \Sigma$ and $z \in \Gamma$;
4. $\delta'(q, \epsilon, z) := \delta(q, \epsilon, z) \cup (\{f\} \times \{Z_f^k \mid 0 \leq k \leq (\bar{g}-1)^2\bar{z}\} \times [K, g])$ for $q \in F$ and $z \in \Gamma$;
5. $\delta'(f, \epsilon, z) := \{f\} \times \{Z_f^k \mid 0 \leq k \leq (\bar{g}-1)^2\bar{z}\} \times [K, g]$ for $z \in \Gamma \cup \{Z_f\}$;
6. $\delta'(f, \epsilon, Y_0) := \{(f, \epsilon, k_0)\}$;
7. the value of δ' is equal to $\emptyset$ in other cases.

Every possible computation of A_2 ending with empty stack has the following form:

1. A_2 goes to the initial configuration of A_1 with an extra symbol Y_0 at the bottom of the stack;

2. A_2 simulates A_1 until A_1 reaches some final state. From that point A_2 can either continue simulating A_1 or

3. A_2 cleans up the stack and constructs a bracketed tree whose prefix is constructed by A_1.

We claim that if the fourth component (the "tree") of automaton's configuration is prefix of some bracketed tree in the beginning of third phase of computation, then, among the possible continuations of the computation, there exists one which finishes that tree, so that it becomes a bracketed tree. Indeed, let T be a labeled tree over $[K, g]$. The prefix of $\mathbf{wrd}(T)$ can be interpreted as a labeled tree T' with labels from the set $[K, g] \cup \{\overline{unknown}\}$ with following properties:

- All vertices labeled with $\overline{unknown}$ are leaves and if some vertex is labeled with $\overline{unknown}$ then all its right siblings are, too.
- The ancestors of vertices labeled with $\overline{unknown}$ are "rightmost" vertices not labeled with $\overline{unknown}$ (i.e. the root vertex, its rightmost descendant, the rightmost descendant of this vertex etc.) only.

From this tree, the tree T can be constructed from it by adding some subtrees to it, with roots in the vertices labeled with $\overline{unknown}$, and relabeling those vertices by elements of $[K, g]$. If the prefix of $\mathbf{wrd}(T)$ is constructed by some TGA then no new descendants have to be added to vertices not labeled with $\overline{unknown}$. If T is a bracketed tree, then there exists a partitioning of the vertices of T', except the vertices labeled by $\overline{unknown}$, such that the ancestors of two vertices from the same class also belong to same class (this allows us to speak about ancestors and descendants of classes), and the "labels" of classes (i.e. concatenation of labels of vertices belonging to that class, from left to right) are prefixes of words from $\mathrm{D}(K, g)$.

No label is longer than $\bar{z}\bar{g}$ and if the label of some class is not an element of $\mathrm{D}(K, g)$ then either it will be an element of $\mathrm{D}(K, g)$ after relabeling the vertices labeled with $\overline{unknown}$ or the ancestor of that class has no more than $(g-1)$ elements. It is easy to prove that if u is a prefix of some element of $\mathrm{D}(K, g)$ then it is a prefix of some element of $\mathrm{D}(K, g)$ whose length is not greater than $(g-1)|u|$. Now the claim follows from the construction of A_2 (items 4–5). $\square$

Thus all three modes of acceptance are equivalent.

5 Equivalence of CCFGs and TGAs

Proposition 3. *Let $G = ([V, g], \Sigma, P, s)$ be a CCFG. There exists a TGA A such that $\mathcal{L}(G) = \mathcal{L}_{\mathcal{E}}(A)$.*

Proof. Let $A := (\{q\}, \Sigma, [V, g], [P \dot{\cup} \Sigma, g], \delta, q, s, \emptyset)$, where

1. for all productions $p = x \to y$ is defined $g(p) := g(x)$;
2. $\delta(q, \epsilon, x_i) := \{(q, (y_i)^{\mathrm{R}}, p_i) \,|\, p = (x_1, \ldots, x_{g(x)}) \to (y_1, \ldots, y_{g(x)}) \in P\}$ for $x \in V \backslash \Sigma$ and $i \in [1..g(x)]$ ($(w)^{\mathrm{R}}$ denotes the mirror image of w);
3. $\delta(q, a, a) := \{(q, \epsilon, a)\}$, for $a \in \Sigma$;

4. the value of δ is equal to $\emptyset$ in other cases.

Analogously to the context-free case we can prove that for every $w \in \Sigma^*$ and $B \in [V, g]$ holds $B \overset{*}{\Rightarrow} w$ in the context-free base grammar of G iff there exists a labeled tree T such that $(q, w, B, \epsilon) \models^{*} (q, \epsilon, \epsilon, \mathbf{wrd}(T))$. Moreover, the right side holds iff T is a derivation tree in the context-free base grammar with $\mathrm{Root}(T) = B$ and $\mathrm{Fr}(T) = w$. The proof of the "if"-part is by induction over the height of T (or, equivalently, over the length of derivation). The proof of the "only if"-part is by induction over the computation length.

Denote the context-free base grammar of G by G'.

Proof of the "if"-part.

Base: The height of the tree T is 1. In this case there exists a production $p = B \to w$ in the context-free base grammar of G. Thus $(q, (w)^{\mathrm{R}}, B \to w) \in \delta(q, \epsilon, B)$ and there exists a computation of A

$$\begin{aligned}(q, w, B, \epsilon) &\vdash (q, w, (w)^{\mathrm{R}}, p\llparenthesis^{|w|}) \\ &\vdash (q, w(2) \cdots w(|w|), w(|w|) \cdots w(2), p\llparenthesis^{|w|} w(1)) \\ &\models^{*} (q, w(|w|), w(|w|), p\llparenthesis^{|w|} w(1) \cdots w(|w|-1)) \vdash (q, \epsilon, \epsilon, \mathbf{wrd}(T)).\end{aligned}$$

Step: The height of the tree is greater than 1. In this case there exist such $k \in \mathbb{N}$, $w_1, \ldots w_k \in \Sigma^*$, $C^{(1)}, \ldots, C^{(k)} \in [V, g]$ and a production $p = B \to C^{(1)} \cdots C^{(k)}$ in G', such that $w = w_1 \cdots w_k$ and $C^{(i)} \overset{*}{\Rightarrow} w_i$ in G' for every $i \in [1..k]$. Let $T_1, \ldots, T_k$ be derivation trees of G' with $\mathrm{Root}(T_i) = C^{(i)}$ and $\mathrm{Fr}(T_i) = w_i$ for every $i \in [1..k]$. We may assume that $\mathbf{wrd}(T) = p\llparenthesis^{k} \mathbf{wrd}(T_1) \cdots \mathbf{wrd}(T_k)$ By induction assumption for every $i \in [1..k]$ there exists a computation of A

$$(q, w_i, C^{(i)}, \epsilon) \models^{*} (q, \epsilon, \epsilon, \mathbf{wrd}(T_i)).$$

According to the construction of A, $(q, C^{(n)} \cdots C^{(1)}, p) \in \delta(q, \epsilon, B)$. Thus there exists the following computation of A:

$$\begin{aligned}(q, w_1 \cdots w_k, B, \epsilon) &\vdash (q, w_1 \cdots w_k, C^{(k)} \cdots C^{(1)}, p\llparenthesis^{k}) \\ &\models^{*} (q, w_2 \cdots w_k, C^{(k)} \cdots C^{(2)}, p\llparenthesis^{k} \mathbf{wrd}(T_1)) \\ &\models^{*} (q, w_2 \cdots w_k, C^{(k)} \cdots C^{(2)}, p\llparenthesis^{k} \mathbf{wrd}(T_1)\mathbf{wrd}(T_2)) \\ &\models^{*} \cdots \models^{*} (q, \epsilon, \epsilon, \mathbf{wrd}(T)).\end{aligned}$$

Thus we have proven the "if"-part.

Proof of the "only if"-part.

Base: The length of the computation is 1, i.e. $(q, w, B, \epsilon) \vdash (q, \epsilon, \epsilon, \mathbf{wrd}(T))$. In this case $\mathbf{wrd}(T) = B$ and $B = w \in \Sigma$. Thus $B \overset{*}{\Rightarrow} w$, derivation's length is 0.

Step: The length of the computation is greater than 1. In this case there exist such $w' \in \Sigma^*$, $k \in \mathbb{N}$, $C^{(1)}, \ldots, C^{(k)} \in [V, g]$ and $p \in [P \dot{\cup} \Sigma, g]$, that

$$(q, w, B, \epsilon) \vdash (q, w', C^{(k)} \cdots C^{(1)}, p\llparenthesis^{k}) \models^{*} (q, \epsilon, \epsilon, PST).$$

The length of that computation, denoted by n, is greater than one, therefore $p \in [P, g]$, $w = w'$ and hence the grammar G' contains production $p = B \to C^{(1)} \cdots C^{(k)}$.

According to the construction of the automaton A, there exist such words $w_1, \ldots, w_k \in \Sigma^*$ and trees $T_1, \ldots, T_k$, whose vertices are labeled by the elements of the set $[P \dot{\cup} \Sigma, g]$, that the following properties hold:

1. $w = w_1 \cdots w_k$ and $\mathbf{wrd}(T) = p\langle\!\!\langle^k \mathbf{wrd}(T_1) \cdots \mathbf{wrd}(T_k)$.
2. There exists a computation

$$\begin{aligned}(q, w_1 \cdots w_k, C^{(k)} \cdots C^{(1)}, p\langle\!\!\langle^k) &\models^* (q, w_2 \cdots w_k, C^{(k)} \cdots C^{(2)}, p\langle\!\!\langle^k \mathbf{wrd}(T_1)) \\ &\models^* (q, w_3 \cdots w_k, C^{(k)} \cdots C^{(3)}, p\langle\!\!\langle^k \mathbf{wrd}(T_1)\mathbf{wrd}(T_2)) \\ \models^* \cdots &\models^* (q, w_k, C^{(k)}, p\langle\!\!\langle^k \mathbf{wrd}(T_1) \cdots \mathbf{wrd}(T_{k-1})) \\ &\models^* (q, \epsilon, \epsilon, \mathbf{wrd}(T)),\end{aligned}$$

such that for every $i \in [1..k]$ the configuration

$$(q, w_{i+1} \cdots w_k, C^{(k)} \cdots C^{(i+1)}, p\langle\!\!\langle^k \mathbf{wrd}(T_1) \cdots \mathbf{wrd}(T_i))$$

is the first configuration of that computation where the "stack" (i.e. the third component of the configuration) contains no more than $(k-i)$ symbols. Thus for every $i \in [1..k]$ there exists a computation

$$(q, w_i, C^{(i)}, \epsilon) \models^* (q, \epsilon, \epsilon, \mathbf{wrd}(T_i)).$$

The lengths of these computations are less than n, hence by induction assumption T_i is a derivation tree of the grammar G' with $\mathrm{Root}(T_i) = C^{(i)}$ and $\mathrm{Fr}(T_i) = w_i$. Thus T is a derivation tree of the grammar G' with $\mathrm{Root}(T) = B$ and $\mathrm{Fr}(T) = w$; we have proven the "only if"-part.

Thus A accepts w by empty stack iff $s \overset{*}{\Rightarrow} w$ in context-free base grammar of G and w has a parse tree which is bracketed tree. But the latter holds iff $s \overset{*}{\Rightarrow} w$ in G. □

Lemma 1. *Every language accepted by some TGA is accepted by empty stack and final state by some TGA having exactly one final state.*

Proof. Let $A = (Q, \Sigma, \Gamma, [K, g], \delta, q_0, Z_0, F)$ be a TGA. Define the automaton $A' := (Q \dot\cup \{f\}, \Sigma, \Gamma, [K, g], \delta', q_0, Z_0, \{f\})$, where

$$\begin{aligned}\delta'(q, a, z) := \delta(q, a, z) \cup \{(f, v, t) \in \{f\} \times \Gamma^* \times [K, g] \mid \\ \exists f' \in F : (f', v, t) \in \delta(q, a, z)\}\end{aligned}$$

for $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ and $z \in \Gamma$; the value of δ is equal to $\emptyset$ in other cases. It holds $\mathcal{L}(A) = \mathcal{L}(A')$. Indeed, the automaton A' simulates A until A is about to reach some final state. Then A' either continues simulating A or goes to the final state and stops. □

Proposition 4. *Let $A = (Q, \Sigma, \Gamma, [K, g], \delta, q_0, Z_0, \{f\})$ be a TGA. There exists a CCFG G such that $\mathcal{L}(A) = \mathcal{L}(G)$.*

Proof. In the following steps we will construct a suitable grammar G.

1. Let $\mathcal{Q} := Q \times \Gamma \times Q$. Let $\pi_1, \pi_3 : \mathcal{Q} \longrightarrow Q$ and $\pi_2 : \mathcal{Q} \longrightarrow \Gamma$ be the corresponding projections.

2. Let $N := \{\langle k,\tilde{q}\rangle \mid k \in K, \tilde{q} \in \mathcal{Q}^{g(k)}\}$. For every $x = \langle k,\tilde{q}\rangle \in N$ define $g(x) := g(k)$. Also define "projections" $\bar{K} : N \longrightarrow K$ and $\bar{Q} : [N,g] \longrightarrow \mathcal{Q}$ as follows:

$$\bar{K}(\langle k, \big((q_1^{(1)}, z_1, q_1^{(2)}), \dots, (q_{g(k)}^{(1)}, z_{g(k)}, q_{g(k)}^{(2)})\big)\rangle) := k$$
$$\bar{Q}\big((\langle k, \big((q_1^{(1)}, z_1, q_1^{(2)}), \dots, (q_{g(k)}^{(1)}, z_{g(k)}, q_{g(k)}^{(2)})\big)\rangle, i)\big) := (q_i^{(1)}, z_i, q_i^{(2)}).$$

3. Let $V := N \dot\cup \Sigma \dot\cup \{s\}$, where s is a new variable. Let $g(s) := 1$ and for every $a \in \Sigma$ let $g(a) := 1$.

4. Define $\mathcal{K} : Q \times \Gamma \longrightarrow \mathcal{P}([K,g])$ as follows:
$$\mathcal{K}(q,z) := \{k_i \in [K,g] \mid \exists a \in \Sigma \cup \{\epsilon\}, v \in \Gamma^*, q' \in Q : (q', v, k_i) \in \delta(q,a,z)\}.$$

5. Define $\mathcal{T} : \mathcal{Q} \times [K,g] \longrightarrow \mathcal{P}([V,g]^*)$ as follows:
$$\begin{aligned}\mathcal{T}((q_1,z,q_2),k_i) := \{& a(x_n,j_n)(x_{n-1},j_{n-1})\cdots(x_1,j_1) \mid \\ & n \in \mathbb{N}, x_1,\dots,x_n \in N, \forall l \in [1..n] : j_l \in [1..g(x_l)], a \in \Sigma \cup \{\epsilon\}, \\ & \exists z_1,\dots,z_n \in \Gamma, q' \in Q : \big[(q', z_1\cdots z_n, k_i) \in \delta(q_1,a,z), \\ & \pi_3(\bar{Q}((x_1,j_1))) = q_2, \pi_1(\bar{Q}((x_n,j_n))) = q', \\ & \forall l \in [1..n-1] : \pi_3(\bar{Q}((x_{l+1},j_{l+1}))) = \pi_1(\bar{Q}((x_l,j_l))), \\ & \forall l \in [1..n] : [\pi_2(\bar{Q}((x_l,j_l))) = z_l \wedge (\bar{K}(x_l), j_l) \in \mathcal{K}(\pi_1(\bar{Q}((x_l,j_l))), z_l)]\big]\} \\ & \cup \{a \in \Sigma \cup \{\epsilon\} \mid (q_2, \epsilon, k_i) \in \delta(q_1,a,z)\}.\end{aligned}$$

6. For every $x \in N$ define $\mathcal{R}(x) \subseteq ([V,g]^*)^{g(x)}$ as follows:
$$\begin{aligned}\mathcal{R}(x) := \{& (w_1,\dots,w_{g(x)}) \in \\ & \mathcal{T}(\bar{Q}((x,1)), (\bar{K}(x),1)) \times \cdots \times \mathcal{T}(\bar{Q}((x,g(x))), (\bar{K}(x),g(x))) \mid \\ & w_1\cdots w_{g(x)} \in \mathrm{D}(V,g)\}\end{aligned}$$

7. Let $G = ([V,g],\Sigma,P,s)$ be a CCFG where the set of productions P is defined as follows:
$$P := \{s \to \langle k, \big((q_0,Z_0,f)\big)\rangle \mid k \in K, g(k) = 1\} \cup \{x \to y \mid x \in N, y \in \mathcal{R}(x)\}.$$

Here a nonterminal $x = \langle k, \big((q_1^{(1)}, z_1, q_1^{(2)}), \dots, (q_{g(k)}^{(1)}, z_{g(k)}, q_{g(k)}^{(2)})\big)\rangle \in N$ has the following meaning:
$(x_1,\dots,x_{g(k)}) \overset{*}{\Rightarrow} (w_1,\dots,w_{g(k)})$, where $w_1,\dots,w_{g(k)} \in \Sigma^*$, iff

1. For every $i \in [1..g(k)]$ holds
$$(q_i^{(1)}, w_i, z_i, \epsilon) \mathrel{|\!\!\!\frac{*}{\;\;}} (q_i^{(2)}, \epsilon, \epsilon, \mathbf{wrd}(T_i)), \tag{1}$$
where T_i is some labeled tree over $[K,g]$.

2. The tuple $(T_1,\dots,T_{g(k)})$ is a "bracketed forest" over $[K,g]$. Bracketed forest is a generalization of bracketed tree and is defined similarly to it, only the first item of definition should be changed to "For every $k \in K$, an ordered tuple of $g(k)$ vertices, labeled respectively by $k_1,\dots,k_{g(k)}$ is bracketed forest over $[K,g]$. The partitioning of vertices contains one class, which contains all vertices."

3. The tuple $(T_1, \ldots, T_{g(k)})$ can be further characterized. Let $\dot{T} = (\dot{T}_1, \ldots, \dot{T}_{g(k)})$ be a "parse forest" of G with $\mathrm{Root}(\dot{T}) = (x_1, \ldots, x_{g(k)})$ and $\mathrm{Fr}(\dot{T}) = (w_1, \ldots, w_{g(k)})$. Computation (1) exists if for every $i \in [1..g(k)]$ the tree T_i can be constructed from $\dot{T}_i$ as follows:

   (a) Remove these vertices of $\dot{T}_i$ which are labeled by the elements of Σ (these vertices can only be leaves).

   (b) Relabel the remaining vertices as follows: the vertex labeled by $((y_1, \ldots, y_{g(y)}) \to (z_1, \ldots, z_{g(y)}), j) \in [P, g]$ is relabeled by $(\bar{K}(y), j) \in [K, g]$.

   Denote the result of this operation by $\mathbf{pr}(\dot{T}_i)$.

   Also, if computation (1) exists then there exists a parse forest of G with root $(x_1, \ldots, x_{g(k)})$ and frontier $(w_1, \ldots, w_{g(k)})$, such that $(T_1, \ldots, T_{g(k)})$ can be constructed from it in the way just described.

The proof of the "if"-part is by induction over the height of the forest $\dot{T}$. The proof of the "only if"-part is by induction over the sum of computation lengths.

Proof of the 'if"-part.

Base: Height of the forest $\dot{T}$ is 1. In this case the grammar G contains production $(x_1, \ldots, x_{g(k)}) \to (y_1, \ldots, y_{g(k)})$. According to the 6th and 7th items of the construction of G, for every $i \in [1..g(k)]$ it holds $w_i \in \mathcal{T}(\bar{Q}(x_i), k_i)$. According to the item 5 of the construction of G, for every $i \in [1..g(k)]$,

$$(\pi_3(\bar{Q}(x_i)), \epsilon, k_i) \in \delta(\pi_1(\bar{Q}(x_i)), w_i, \pi_2(\bar{Q}(x_i))).$$

Hence for every $i \in [1..g(k)]$,

$$(\pi_1(\bar{Q}(x_i)), w_i, \pi_2(\bar{Q}(x_i)), \epsilon) \vdash (\pi_3(\bar{Q}(x_i)), \epsilon, \epsilon, \mathbf{wrd}(\mathbf{pr}(\dot{T}_i))),$$

because $\mathbf{pr}(\dot{T}_i)$ contains one vertex, this vertex is labeled by k_i.

Step: Height of the forest $\dot{T}$, denoted by n, is greater than 1. Let the root vertices of $\dot{T}_1, \ldots, \dot{T}_k$ be labeled by $p_1, \ldots, p_k$ respectively, where

$$p = (x_1, \ldots, x_{g(k)}) \to (y_1, \ldots, y_{g(k)}) \in P. \tag{2}$$

Consider the partitioning $\mathcal{S}(y_1 \cdots y_{g(k)})$, suppose that it contains l classes, denoted by $Y_1, \ldots, Y_l$. For every $r \in [1..l]$ let $C^{(r)} \in V$ be such, that $(y_1 \cdots y_{g(k)})(\mathcal{E}l_{Y_r}(j)) = C^{(r)}_j$ for every $j \in [1..g(C^{(r)})]$. For every $i \in [1..g(k)]$ and $j \in [1..|y_i|]$ let $\dot{T}_{i,j}$ be the subtree of $\dot{T}_i$, whose root is the j-th direct descendant of the root vertex of $\dot{T}_i$; let $w_{i,j} = \mathrm{Fr}(\dot{T}_{i,j})$. For every $r \in [1..l]$ and $m \in [1..g(C^{(r)})]$ let $i'_{r,m}$ and $j'_{r,m}$ be such that the letter $C^{(r)}_m$, corresponding to the m-th element of Y_r, stands on the $j'_{r,m}$-th position in the word $y_{i'_{r,m}}$. For every $r \in [1..l]$ where $C^{(r)}$ is a nonterminal, $\dot{T}^{(r)} = (\dot{T}_{i'_{r,1}, j'_{r,1}}, \ldots, \dot{T}_{i'_{r,g(C^{(r)})}, j'_{r,g(C^{(r)})}})$ is a parse forest of G with $\mathrm{Root}(\dot{T}^{(r)}) = (C^{(r)}_1, \ldots, C^{(r)}_{g(C^{(r)})})$ and $\mathrm{Fr}(\dot{T}^{(r)}) = (w_{i'_{r,1}, j'_{r,1}}, \ldots, w_{i'_{r,g(C^{(r)})}, j'_{r,g(C^{(r)})}})$. The height of $\dot{T}^{(r)}$ is less than n, hence by induction assumption for every $m \in [1..g(C^{(r)})]$ the following holds:

$$(\pi_1(\bar{Q}(C^{(r)}_m)), w_{i'_{r,m}, j'_{r,m}}, \pi_2(\bar{Q}(C^{(r)}_m)), \epsilon) \overset{*}{\vdash} (\pi_3(\bar{Q}(C^{(r)}_m)), \epsilon, \epsilon, \mathbf{wrd}(\mathbf{pr}(\dot{T}_{i'_{r,m}, j'_{r,m}}))).$$

Grammar G contains the production (2), hence it follows from the 6th and 7th item of the construction of G, that for every $i \in [1..g(k)]$ it holds $y_i \in \mathcal{T}(\bar{Q}(x_i), k_i)$. Observe the following two cases:

1. $y_i \in \Sigma \cup \{\epsilon\}$. In this case $w_i = y_i$, $\mathbf{wrd}(\mathbf{pr}(\dot{T}_i)) = k_i$ and according to the 5th item of the construction of G,

$$(\pi_3(\bar{Q}(x_i)), \epsilon, k_i) \in \delta(\pi_1(\bar{Q}(x_i)), w_i, \pi_2(\bar{Q}(x_i))),$$

which implies

$$(\pi_1(\bar{Q}(x_i)), w_i, \pi_2(\bar{Q}(x_i)), \epsilon) \vdash (\pi_3(\bar{Q}(x_i)), \epsilon, \epsilon, \mathbf{wrd}(\mathbf{pr}(\dot{T}_i))).$$

2. y_i contains a symbol from $[N, g]$, thus $y_i = a y_i(x) y_i(x+1) \cdots y_i(|y_i|)$, where

   - $a \in \Sigma$ and $x = 2$ (also $a = w_{i,1}$), or $a = \epsilon$ and $x = 1$;
   - $y_i(x), y_i(x+1), \ldots, y_i(|y_i|) \in [N, g]$.

   According to the 5th item of the construction of G, we have:

$$(\pi_1(\bar{Q}(y_i(x))), \pi_2(\bar{Q}(y_i(|y_i|)))\pi_2(\bar{Q}(y_i(|y_i|-1))) \cdots \pi_2(\bar{Q}(y_i(x))), k_i) \\ \in \delta(\pi_1(\bar{Q}(x_i)), a, \pi_2(\bar{Q}(x_i))),$$

$$\pi_3(\bar{Q}(y_i(|y_i|))) = \pi_3(\bar{Q}(x_i))$$

and for all $j \in [x..|y_i|-1]$:

$$\pi_3(\bar{Q}(y_i(j))) = \pi_1(\bar{Q}(y_i(j+1))).$$

Hence the automaton A has the following computation:

$$\begin{aligned}
&(\pi_1(\bar{Q}(x_i)), w_{i,1} \cdots w_{i,|y_i|}, \pi_2(\bar{Q}(x_i)), \epsilon) \\
&\vdash (\pi_1(\bar{Q}(y_i(x))), w_{i,x} \cdots w_{i,|y_i|}, \pi_2(\bar{Q}(y_i(|y_i|))) \cdots \pi_2(\bar{Q}(y_i(x))), k_i \langle\!\langle^{|y_i|-x+1}) \\
&\vdash^{*} (\pi_1(\bar{Q}(y_i(x+1))), w_{i,x+1} \cdots w_{i,|y_i|}, \pi_2(\bar{Q}(y_i(|y_i|))) \cdots \pi_2(\bar{Q}(y_i(x+1))), \\
&\qquad\qquad k_i \langle\!\langle^{|y_i|-x+1} \mathbf{wrd}(\mathbf{pr}(\dot{T}_{i,x}))) \\
\vdash^{*} \cdots &\vdash^{*} (\pi_1(\bar{Q}(y_i(|y_i|))), w_{i,|y_i|}, \pi_2(\bar{Q}(y_i(|y_i|))), \\
&\qquad k_i \langle\!\langle^{|y_i|-x+1} \mathbf{wrd}(\mathbf{pr}(\dot{T}_{i,x})) \cdots \mathbf{wrd}(\mathbf{pr}(\dot{T}_{i,|y_i|-1}))) \\
&\vdash^{*} (\pi_3(\bar{Q}(y_i(|y_i|))), \epsilon, \epsilon, k_i \langle\!\langle^{|y_i|-x+1} \mathbf{wrd}(\mathbf{pr}(\dot{T}_{i,x})) \cdots \mathbf{wrd}(\mathbf{pr}(\dot{T}_{i,|y_i|}))) \\
&= (\pi_3(\bar{Q}(x_i)), \epsilon, \epsilon, \mathbf{wrd}(\mathbf{pr}(\dot{T}_i))),
\end{aligned}$$

which finishes the proof of the "if"-part.

Proof of the "only if"-part
We have to construct a parse forest $\dot{T} = (\dot{T}_1, \ldots, \dot{T}_{g(k)})$ of G with $\mathrm{Root}(\dot{T}) = (x_1, \ldots, x_{g(k)})$ and $\mathrm{Fr}(\dot{T}) = (w_1, \ldots, w_{g(k)})$, such that $\mathbf{pr}(\dot{T}) = T$.
Base: The sum of the lengths of computations is $g(k)$. In this case every computation (1) is of length 1, i.e. for every $i \in [1..g(k)]$ it holds

$$(\pi_1(\bar{Q}(x_i)), w_i, \pi_2(\bar{Q}(x_i)), \epsilon) \vdash (\pi_3(\bar{Q}(x_i)), \epsilon, \epsilon, k_i),$$

therefore
$$(\pi_3(\bar{Q}(x_i)), \epsilon, k_i) \in \delta(\pi_1(\bar{Q}(x_i)), w_i, \pi_2(\bar{Q}(x_i))).$$
According to the 5th item of the construction of G, for every $i \in [1..g(k)]$ it holds $w_i \in \mathcal{T}(\bar{Q}(x_i), k_i)$, hence by the 6th and 7th items of the construction of G,
$$p = (x_1, \ldots, x_{g(k)}) \to (w_1, \ldots, w_{g(k)}) \in P.$$
For every $i \in [1..g(k)]$ let the tree $\dot{T}_i$ be as follows: the root of the tree is labeled by (p, i), it has $|w_i|$ descendants (i.e. 0 or 1), and if the number of descendants is 1, then it is labeled by the symbol w_i. In this case $\dot{T}$ is a parse forest of G and $\mathbf{pr}(\dot{T}) = T$.

Step: The sum of the lengths of computations, denoted by n, is greater than $g(k)$. Suppose that for each $i \in [1..g(k)]$ the first step of computation (1) is
$$(\pi_1(\bar{Q}(x_i)), w_i, \pi_2(\bar{Q}(x_i)), \epsilon) \vdash (q_{i,0}, w'_i, z_{i,l_i} z_{i,l_i-1} \cdots z_{i,1}, k_i \langle\!\langle^{l_i}) \tag{3}$$
(the fourth component of the right side begins with k_i due to the 5th and 6th items of the construction of G), where $q_{i,0} \in Q$, $w_i = a_i w'_i$ with $a_i \in \Sigma \cup \{\epsilon\}$, $l_i \in \mathbb{N}_0$, and $z_{i,1}, \ldots, z_{i,l_i} \in \Gamma$. Let $q_{i,1}, \ldots, q_{i,l_i-1} \in Q$, $w'_{i,1}, \ldots, w'_{i,l_i} \in \Sigma^*$ and $T_{i,1}, \ldots, T_{i,l_i}$ labeled trees with labels from the set $[K, g]$, such that the following properties hold:

1. $w'_i = w'_{i,1} \cdots w'_{i,l_i}$ and $\mathbf{wrd}(T_i) = k_i \langle\!\langle^{l_i} \mathbf{wrd}(T_{i,1}) \cdots \mathbf{wrd}(T_{i,l_i})$.
2. The computation (1) has the form
$$\begin{aligned}(\pi_1(\bar{Q}(x_i)), w_i, &\pi_2(\bar{Q}(x_i)), \epsilon)\\ &\vdash (q_{i,0}, w'_{i,1} \cdots w'_{i,l_i}, z_{i,l_i} \cdots z_{i,1}, k_i \langle\!\langle^{l_i})\\ &\vdash^* (q_{i,1}, w'_{i,2} \cdots w'_{i,l_i}, z_{i,l_i} \cdots z_{i,2}, k_i \langle\!\langle^{l_i} \mathbf{wrd}(T_{i,1}))\\ \vdash^* \cdots &\vdash^* (q_{i,l_i-1}, w'_{i,l_i}, z_{i,l_i}, k_i \langle\!\langle^{l_i} \mathbf{wrd}(T_{i,1}) \cdots \mathbf{wrd}(T_{i,l_i-1}))\\ &\vdash^* (\pi_3(\bar{Q}(x_i)), \epsilon, \epsilon, \mathbf{wrd}(T_i)),\end{aligned} \tag{4}$$
and for each $j \in [1..l_i - 1]$ the configuration
$$(q_{i,j}, w'_{i,j+1} \cdots w'_{i,l_i}, z_{i,l_i} \cdots z_{i,j+1}, k_i \langle\!\langle^{l_i} \mathbf{wrd}(T_{i,1}) \cdots \mathbf{wrd}(T_{i,j}))$$
is the first configuration in this computation (except the initial configuration) where the stack contains no more than $(l_i - j)$ symbols.

For every $i \in [1..g(k)]$ denote $q_{i,l_i} := \pi_3(\bar{Q}(x_i))$; for every $j \in [1..l_i]$ there exists a computation
$$(q_{i,j-1}, w'_{i,j}, z_{i,j}, \epsilon) \vdash^* (q_{i,j}, \epsilon, \epsilon, \mathbf{wrd}(T_{i,j})). \tag{5}$$

Let $U_1, \ldots, U_l$ be the classes of the partitioning of the bracketed forest T, which contain the direct descendants of the roots of $T_1, \ldots, T_{g(k)}$ (i.e. the elements of $U_1, \ldots, U_l$ are exactly the roots of the trees $T_{i,j}$). For every $r \in [1..l]$ let $k^{(r)} \in K$ be such, that the vertices in U_r are labeled by $k^{(r)}_1, \ldots, k^{(r)}_{g(k^{(r)})}$, respectively. For every $r \in [1..l]$ and $m \in [1..g(k^{(r)})]$ let $i'_{r,m} \in \mathbb{N}$ and $j'_{r,m} \in \mathbb{N}$ be such, that the vertex in U_r labeled with $k^{(r)}_m$ is the root of the tree $T_{i'_{r,m}, j'_{r,m}}$

(i.e. the tree $T_{i'_{r,m},j'_{r,m}}$ is just one of the trees $T_{i,j}$, where $i \in [1..g(k)]$ and $j \in [1..g(l_i)]$). Thus for every $r \in [1..l]$ and $m \in [1..g(k^{(r)})]$

$$(q_{i'_{r,m},j'_{r,m}-1}, w'_{i'_{r,m},j'_{r,m}}, z_{i'_{r,m},j'_{r,m}}, \epsilon) \models^* (q_{i'_{r,m},j'_{r,m}}, \epsilon, \epsilon, \mathbf{wrd}(T_{i'_{r,m},j'_{r,m}}))$$

(these are just the computations (5) with different notation). For every $r \in [1..l]$ denote

$$C^{(r)} := \langle k^{(r)}, \big((q_{i'_{r,m},j'_{r,m}-1}, z_{i'_{r,m},j'_{r,m}}, q_{i'_{r,m},j'_{r,m}})\big)_{m=1}^{g(k^{(r)})}\rangle.$$

The induction assumption gives that for every $r \in [1..l]$ there exists a parse forest $\dot{T}^{(r)} = (\dot{T}_1^{(r)}, \ldots, \dot{T}_{g(k^{(r)})}^{(r)})$, such that

1. The root of $\dot{T}^{(r)}$ is $(C_1^{(r)}, \ldots, C_{g(k^{(r)})}^{(r)})$ and frontier is $(w_{i'_{r,1},j'_{r,1}}, \ldots, w_{i'_{r,g(k^{(r)})},j'_{r,g(k^{(r)})}})$.

2. For every $m \in [1..g(k^{(r)})]$ it holds $\mathbf{pr}(\dot{T}_m^{(r)}) = T_{i'_{r,m},j'_{r,m}}$.

On the other hand, (3) gives that for every $i \in [1..g(k)]$

$$(q_{i,0}, z_{i,l_i} \cdots z_{i,1}, k_i) \in \delta(\pi_1(\bar{Q}(x_i)), a_i, \pi_2(\bar{Q}(x_i))).$$

Also, for every $j \in [1..l_i]$ it holds $\bar{K}(\mathrm{Root}(T_{i,j})) \in \mathcal{K}(q_{i,j-1}, z_{i,j})$.

Let $(u_1, \ldots, u_{g(k)}) \in ([V,g]^*)^{g(k)}$ be as described in the following:

1. For every $i \in [1..g(k)]$ it holds $|u_i| = l_i$.

2. For every $i \in [1..g(k)]$ and $j \in [1..l_i]$ it holds $u_i(j) = C_m^{(r)}$, where $r \in [1..l]$ and $m \in [1..g(k^{(r)}]$ are such, that $i = i'_{r,m}$ and $j = j'_{r,m}$.

From the 5th item of the construction of G follows that for every $i \in [1..g(k)]$ it holds $a_i u_i \in \mathcal{T}(\bar{Q}(x_i), k_i)$. Also, from the fact that T is a bracketed forest follows that $a_1 u_1 \cdots a_{g(k)} u_{g(k)} \in \mathrm{D}(V,g)$, hence by the 6th and 7th item of the construction of G, the grammar G contains the production $p = (x_1, \ldots, x_{g(k)}) \to (a_1 u_1, \ldots, a_{g(k)} u_{g(k)})$.

It is now easy to construct the parse forest $\dot{T} = (\dot{T}_1, \ldots, \dot{T}_{g(k)})$ of G with necessary properties.

1. For every $i \in [1..g(k)]$ the root vertex of $\dot{T}_i$ is labeled by p_i. The root vertex of $\dot{T}_i$ has $|a_i u_i|$ direct descendants, if $a_i \neq \epsilon$, then the leftmost direct descendant is labeled by a_i.

2. For every $i \in [1..g(k)]$ and $j \in [1..l_i]$ the $(j + |a_i|)$-th direct descendant of the root vertex of $\dot{T}_i$ is itself the root vertex of the tree $\dot{T}_m^{(r)}$, where $r \in [1..l]$ and $m \in [1..g(k^{(r)})]$ are such, that $i = i'_{r,m}$ and $j = j'_{r,m}$.

According to this construction we have $\mathrm{Root}(\dot{T}) = (x_1, \ldots, x_{g(k)})$, $\mathrm{Fr}(\dot{T}) = (w_1, \ldots, w_{g(k)})$ and $\bar{K}(\dot{T}) = T$; we have proven the “only if”-part.

From this claim easily follows that for every $k \in K$, where $g(k) = 1$, and $w \in \Sigma^*$ holds $\langle k, \big((q_0, Z_0, f)\big)\rangle \stackrel{*}{\Rightarrow} w$ iff there exists a bracketed tree T over $[K,g]$ whose root is labeled with k, such that $(q_0, w, Z_0, \epsilon) \models^* (f, \epsilon, \epsilon, \mathbf{wrd}(T))$. $\square$

Thus we have shown that the following theorem is true.

Theorem 1. *For a language $L \subseteq \Sigma^*$ there exists a CCFG which generates L iff there exists a TGA which accepts L.*

References

[Cho62] N. Chomsky. Context-free grammars and pushdown storage. M.I.T. Res. Lab. Electron. Quart. Prog. Rept. 65, 1962.

[Gua92] Y. Guan. Klammergrammatiken, Netzgrammatiken und Interpretationen von Netzen. Dissertation, Universität des Saarlandes, 1992.

[GHR92] Y. Guan, G. Hotz, A. Reichert. Tree Grammars with Multilinear Interpretation. Technischer Bericht, Universität des Saarlandes, 1992.

[Har78] M. Harrison. Introduction to Formal Language Theory. Addison-Wesley, 1978.

[HP96] G. Hotz, G. Pitsch. On parsing coupled-context-free languages, Theor. Comp. Sci. **161** (1996), pp205-233.

[Pit93] G. Pitsch. Analyse von Klammergrammatiken. Dissertation, Universität des Saarlandes, 1993.