

Two Power-Decreasing Derivation Restrictions in Grammar Systems

Martin Čermák

University of Technology in Brno
Faculty of Information Technology

TID presentation 2/2

1 Preliminary Definitions

2 Two Derivation Restrictions

3 Language Families

Phrase structure grammar

Phrase structure grammar is a quadruple $G = (N, T, P, S)$, where:

- N and T are alphabets,
- $S \in N$ is the start symbol and
- P is a finite set of productions of the form $\alpha \rightarrow \beta$, where:
 - $\alpha \in N^+$
 - $\beta \in (N \cup T)^*$

Derivation

If $\alpha \rightarrow \beta \in P$, $u = x\alpha y$, $v = x\beta y$ and $x, y \in (T \cup N)^*$, then
 $u \Rightarrow v[\alpha \rightarrow \beta]$ or simply $u \Rightarrow v$.

$N_{left}(P)$

$$N_{left}(P) = \{\alpha \mid \alpha \in N^+, \beta \in (N \cup T)^*, \alpha \rightarrow \beta \in P\}$$

Example

Let $P = \{S \rightarrow BA, BA \rightarrow AAc, AA \rightarrow aAC, AC \rightarrow ac, AC \rightarrow acBA\}$
then $N_{left}(P) = \{S, BA, AA, AC\}$

 $occur(\omega, W)$

If V is total alphabet, $W \subseteq V$ and $\omega \in V^*$, then $occur(\omega, W)$ denotes
the number of occurrences of symbols of W in ω

Example

Let $W = \{a, B, C, S\}$ and $\omega = aacaaBAcc$, then $occur(\omega, W) = 5$

Derivation Restriction

Let:

- $G = (N, T, P, S)$ be a phrase structure grammar,
- $V = N \cup T$ be the total alphabet of G ,
- $I \geq 1$.

If there is $\alpha \rightarrow \beta \in P$, $u = x_0\alpha x_1$, and $v = x_0\beta x_1$, where:

- $x_0 \in T^*N^*$,
- $x_1 \in V^*$,
- $\text{occur}(x_0\alpha, N) \leq I$,

then $u \not\Rightarrow v[\alpha \rightarrow \beta]$ in G or simply $u \not\Rightarrow v$

$\not\Rightarrow^*$ and $\not\Rightarrow^+$ denote transitive-reflexive and transitive closure of $\not\Rightarrow$, respectively.

$W(m)$

Let $m \geq 1$. $W(m)$ denotes the set of all strings $x \in V^*$, where

- $x \in (T^*N^*)^m T^*$

Example

- $aaBadCcb \in W(3)$,
- $BBCD \in W(3)$,
- $BBaBAASaaCaC \notin W(3)$

Product

Let $u, v \in V^*$ and $u \Rightarrow v$. If $u, v \in W(m)$, then $u \underset{m}{\circ} \Rightarrow v$.

$W(m, h)$

Let $m, h \geq 1$. $W(m, h)$ denotes the set of all strings $x \in V^*$, where

- $x \in (T^*N^*)^m T^*$
- $(y \in \text{sub}(x) \wedge |y| > h) \text{ implies } \text{alph}(y) \cap T \neq \emptyset$

Example

- $aaBadCcb \in W(3, 1)$,
- $BBCD \notin W(3, 1)$,
- $BBaBAASaaCaC \notin W(3, 1)$
- $BaAaaCaC \notin W(3, 1)$

Product

Let $u, v \in V^*$ and $u \Rightarrow v$. If $u, v \in W(m, h)$, then $u \stackrel{h}{m} \Rightarrow v$.

CD Grammar System

CD grammar system is $(n + 3)$ -tuple

$$\Gamma = (N, T, S, P_1, \dots, P_n),$$

where:

- N, T are alphabets, where $T \cap N \neq \emptyset$,
- $V = N \cup T$,
- $S \in N$ is start nonterminal, and
- for every $1 \leq i \leq n$, where $n \geq 1$, $G_i = (N, T, S, P_i)$ is a **phrase-structure** grammar.

Derivation modes

Let $\Gamma = (N, T, S, P_1, \dots, P_n)$ is CD grammar system and $\Rightarrow \in R$, where $R = \{{}_I\!\diamond\Rightarrow, {}_m\!\diamond\Rightarrow, {}^h_m\!\diamond\Rightarrow\}$.

- For all $i = 1, 2, \dots, n$, *terminating derivation* by the i -th component, denoted $\Rightarrow_{P_i}^t$, defined by

$x \Rightarrow_{P_i}^t y$ iff $x \Rightarrow_{P_i}^* y$ there is no $z \in \Sigma^*$ with $y \Rightarrow_{P_i} z$.

- For all $i = 1, 2, \dots, n$, *k -step derivation* by the i -th component, denoted $\Rightarrow_{P_i}^{=k}$, defined by

$x \Rightarrow_{P_i}^{=k} y$ iff there are $x_1, \dots, x_{k+1} \in (N \cup T)^*$ such that
 $x = x_1, y = x_{k+1}$, and $\forall j = 1, \dots, k, x_j \Rightarrow_{P_i} x_{j+1}$.

Derivation modes

- For all $i = 1, 2, \dots, n$, the at most k -step derivation by the i -th component, denoted $\Rightarrow_{P_i}^{\leq k}$, defined by

$$x \Rightarrow_{P_i}^{\leq k} y \text{ iff } x \Rightarrow_{P_i}^{= k'} y \text{ for some } k' \leq k.$$

Derivation modes D

Let $D = \{*, t\} \cup \{\leq k, = k, \geq k | k \in \mathbb{N}^+\}$

CD grammar system language

$$L_f(\Gamma) = \{\omega \in T^* | S \Rightarrow_{P_{i_1}}^f \omega_1 \Rightarrow_{P_{i_2}}^f \dots \Rightarrow_{P_{i_m}}^f \omega_m = \omega, m \geq 1, 1 \leq j \leq m, \\ 1 \leq i_j \leq n\}$$

CD grammar system with control language

For a CD grammar system $\Gamma = (N, T, S, P_1, \dots, P_n)$ and a control language L , we set

$$\text{l-left } \mathcal{L}_t^L(\Gamma) = \{w \in T^* | S \xrightarrow{\text{l}}_{P_{i_1}}^t w_1 \xrightarrow{\text{l}}_{P_{i_2}}^t \dots \xrightarrow{\text{l}}_{P_{i_p}}^t w_p = w, \\ p \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq p, i_1 i_2 \dots i_p \in L\}$$

$$\text{nonter } \mathcal{L}_t^L(\Gamma, m, h) = \{w \in T^* | S \xrightarrow{h}_{m}^t w_1 \xrightarrow{h}_{m}^t \dots \xrightarrow{h}_{m}^t w_p = w, \\ p \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq p, i_1 i_2 \dots i_p \in L\}$$

$$\text{nonter } \mathcal{L}_t^L(\Gamma, m) = \{w \in T^* | S \xrightarrow{m}_{P_{i_1}}^t w_1 \xrightarrow{m}_{P_{i_2}}^t \dots \xrightarrow{m}_{P_{i_p}}^t w_p = w, \\ p \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq p, i_1 i_2 \dots i_p \in L\}$$

$$\text{left-most nonter } \mathcal{L}_t^L(\Gamma, m, h) = \{w \in T^* | S \xrightarrow{h}_{m}^t w_1 \xrightarrow{h}_{m}^t \dots \xrightarrow{h}_{m}^t w_p = w, \\ p \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq p, i_1 i_2 \dots i_p \in L, \\ \text{and all } \xrightarrow{h}_{m}^t \text{ are left-most wrt. } P_{i_k}\}$$

Language Families

Let GSs denote the family of all CD grammar systems. Let $l, m, h \geq 1$. Define the following language families:

$${}_{\text{1-left}} GS_t^{REG} = \{{}_{\text{1-left}} \mathcal{L}_t^L(\Gamma) : \Gamma \in GSs, L \in REG\}$$

$${}_{\text{nonter}} GS_t^{REG}(m, h) = \{{}_{\text{nonter}} \mathcal{L}_t^L(\Gamma, m, h) : \Gamma \in GSs, L \in REG\}$$

$${}_{\text{nonter}} GS_t^{REG}(m) = \{{}_{\text{nonter}} \mathcal{L}_t^L(\Gamma, m) : \Gamma \in GSs, L \in REG\}$$

Results

1. $CF = {}_{\text{1-left}} GS_t^{REG}$,
2. $RE = {}_{\text{nonter}} GS_t^{REG}(1)$,
3. $SG_m \supseteq {}_{\text{nonter}} GS_t^{REG}(m, 1)$.

$$CF = \text{l-left } GS_t^{REG}$$

Lemma

For every CD grammar system Γ , every finite automaton \overline{M} and every $l \geq 1$, there is a pushdown automaton M , such that

$$\mathcal{L}(M) = \text{l-left } \mathcal{L}_t^{\mathcal{L}(\overline{M})}(\Gamma).$$

Proof idea

- M simulates t -mode derivations of Γ regulated by \overline{M} ,
- States of M are composed of 4 elements:
 1. contains first o symbols from the first continuous block of nonterminal for $o \leq l$,
 2. stage of simulated derivation step,
 3. state of \overline{M} ,
 4. index of an active component of Γ .

$$CF = \text{1-left } GS_t^{REG}$$

Algorithm 1/2

In: $\Gamma = (N, T, S, P_1, \dots, P_n)$, $\bar{M} = (\bar{Q}, \bar{\Sigma}, \bar{\delta}, \bar{s}_0, \bar{F})$ and $l \geq 1$

Out: $M = (Q, \Sigma, Z, \delta, s_0, z_0, F)$, where $\mathcal{L}(M) = \text{1-left } \mathcal{L}_t^{\mathcal{L}(\bar{M})}(\Gamma)$

Algorithm:

$$Q = \{s_0, f\} \cup \{[\gamma, s, \bar{s}, i] : \gamma \in N^*, |\gamma| \leq l, s \in \{q, r, e\}, \bar{s} \in \bar{Q}, i \in \{1 \dots n\}\}$$

$$\Sigma = T$$

$$Z = T \cup N \cup \{z_0\}$$

$$F = \{f\}$$

δ: next slide

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s_0}, i]$, s.t. $\gamma \not\Rightarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $sub(\gamma) \cap N_{left}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $sub(\gamma) \cap N_{left}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s}_0, i]$, s.t. $\gamma \not\rightsquigarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s}_0, i]$, s.t. $\gamma \not\rightsquigarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s_0}, i]$, s.t. $\gamma \not\Rightarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $sub(\gamma) \cap N_{left}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $sub(\gamma) \cap N_{left}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s_0}, i]$, s.t. $\gamma \not\Rightarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $sub(\gamma) \cap N_{left}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $sub(\gamma) \cap N_{left}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s_0}, i]$, s.t. $\gamma \not\Rightarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $sub(\gamma) \cap N_{left}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $sub(\gamma) \cap N_{left}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s_0}, i]$, s.t. $\gamma \not\Rightarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $sub(\gamma) \cap N_{left}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $sub(\gamma) \cap N_{left}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s}_0, i]$, s.t. $\gamma \not\Rightarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s}_0, i]$, s.t. $\gamma \not\Rightarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$CF = \text{I-left } GS_t^{REG}$$

Algorithm 2/2

δ contains rules of the following forms:

1. $s_0 \rightarrow [S, q, \bar{s}_0, i]$, s.t. $\gamma \not\Rightarrow_{P_i}^1 \gamma'$
2. if $\gamma \in N^*$ and $|\gamma| \leq l$ then $[\gamma, q, s, i] \rightarrow (\gamma')^R[\varepsilon, r, s, i]$
3. $a[\varepsilon, r, s, i]a \rightarrow [\varepsilon, r, s, i]$
4. if $si \rightarrow s' \in \bar{\delta}$ for some $s' \in F$ then $z_0[\varepsilon, r, s, i] \rightarrow f$
5. if $A \in N$ and $o < l$ then $A[A_1 \dots A_o, r, s, i] \rightarrow [A_1 \dots A_o A, r, s, i]$
6. $[A_1 \dots A_l, r, s, i] \rightarrow [A_1 \dots A_l, e, s, i]$
7. if $o < l$ and $a \in T$ then $a[A_1 \dots A_l, r, s, i] \rightarrow a[A_1 \dots A_l, e, s, i]$
8. if $o < l$ then $z_0[A_1 \dots A_l, r, s, i] \rightarrow z_0[A_1 \dots A_l, e, s, i]$
9. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) = \emptyset$ and $si \rightarrow s' \in \bar{\delta}$ then
 $[\gamma, e, s, i] \rightarrow [\gamma, q, s', i']$
10. if $\text{sub}(\gamma) \cap N_{\text{left}}(P_i) \neq \emptyset$ then $[\gamma, e, s, i] \rightarrow [\gamma, q, s, i]$

$$RE = \text{nonter } GS_t^{REG}(1)$$

Proof idea

- Any recursively enumerable language L is generated by a grammar G in the Geffert normal form:
 - $G = (\{S, A, B, C\}, T, P \cup \{ABC \rightarrow \varepsilon\})$, where
 - P contains context-free productions of the form:
 - $S \rightarrow uSa$
 - $S \rightarrow uSv$
 - $S \rightarrow uv$
 where $u \in \{A, AB\}^*$, $v \in \{BC, C\}^*$, and $a \in T$
 - any terminal derivation is of the form $S \Rightarrow^* w_1 w_2 w$ by P where
 - $w_1 \in \{A, AB\}^*$
 - $w_2 \in \{BC, C\}^*$
 - $w \in T^*$
 - $w_1 w_2 w \Rightarrow^* w$ by $ABC \rightarrow \varepsilon$
 - G is a CD grammar system with one component. Set the control language to be $\{1\}^*$.
 - then, the theorem holds.

Thank you.