Greibach Normal Form Transformation, Revisited

Norbert Blum
Robert Koch
Informatik IV, Universität Bonn
Römerstr. 164, D-53117 Bonn, Germany
email: blum@cs.uni-bonn.de

August 25, 1998

Abstract

We develop a new method for placing a given context-free grammar into Greibach normal form with only polynomial increase of its size. Starting with an arbitrary ε-free context-free grammar G, we transform G into an equivalent context-free grammar H in extended Greibach normal form; i.e., in addition to rules, fulfilling the Greibach normal form properties, the grammar can have chain rules. The size of H will be $O(|G|^3)$, where $|G|$ is the size of G. Moreover, in the case that G is chain rule free, H will be already in Greibach normal form. If H is not chain rule free then we use the standard method for chain rule elimination for the transformation of H into Greibach normal form. The size of the constructed grammar is $O(|G|^4)$.

1 Introduction and definitions

We assume that the reader is familiar with the elementary theory of finite automata and context-free grammars as written in standard text books, e.g. [1, 4, 5, 11]. First, we will review the notations used in the subsequence.

A *context-free grammar* G is a 4-tuple (V, Σ, P, S) where V is a finite, nonempty set of symbols called the *total vocabulary*, $\Sigma \subset V$ a finite set of *terminal symbols*, $N = V \setminus \Sigma$ the set of *nonterminal symbols* (or *variables*), P a finite set of *rules* (or *productions*), and $S \in N$ is the *start symbol*. The productions are of the form $A \to \alpha$, where $A \in N$ and $\alpha \in V^*$. α is called *alternative* of A. $L(G)$ denotes the *context-free language* generated by G. The *size* $|G|$ of the context-free grammar G is defined by

$$|G| = \sum_{A \to \alpha \in P} lg(A\alpha),$$

where $lg(A\alpha)$ is the length of the string $A\alpha$. Two context-free grammars G and G' are *equivalent* if both grammars generate the same language; i.e., $L(G) = L(G')$. Let ε denote the empty word. A production $A \to \varepsilon$ is called *ε-rule*. A production $A \to B$ with $B \in N$ is called *chain rule*.

A *leftmost (rightmost) derivation* is a derivation where, at every step, the variable replaced has no variable to its left (right) in the sentential form from which the replacement is made.

A context-free grammar $G = (V, \Sigma, P, S)$ is *ε-free* if each production is of the form

i) $A \to \alpha$ with $\alpha \in (V \setminus \{S\})^+$, or

ii) $S \to \varepsilon$.

A context-free grammar G is in *Chomsky normal form* if each production is of the form

i) $A \to BC$ with $B, C \in N \setminus \{S\}$,

ii) $A \to a$ with $a \in \Sigma$, or

iii) $S \to \varepsilon$.

A context-free grammar G is in *extended Chomsky normal form* if each production is of the form

i) $A \to BC$ with $B, C \in N \setminus \{S\}$,

ii) $A \to B$ with $B \in N \setminus \{S\}$,

iii) $A \to a$ with $a \in \Sigma$, or

iv) $S \to \varepsilon$.

A context-free grammar $G = (V, \Sigma, P, S)$ is in *Greibach normal form* if each production is of the form

i) $A \to a\alpha$ with $a \in \Sigma$, $\alpha \in (V \setminus \{S\})^*$, or

ii) $S \to \varepsilon$.

A context-free grammar $G = (V, \Sigma, P, S)$ is in *extended Greibach normal form* if each production is of the form

i) $A \to a\alpha$ with $a \in \Sigma$, $\alpha \in (V \setminus \{S\})^*$,

ii) $A \to B$ with $B \in N \setminus \{S\}$, or

iii) $S \to \varepsilon$.

A context-text free grammar in (extended) Greibach normal form is in *2 (extended) Greibach normal form* if for all productions of type i) $lg(\alpha) \leq 2$.

Given an arbitrary context-free grammar $G = (V, \Sigma, P, S)$, it is well known that G can be transformed into an equivalent context-free grammar G' which is in Greibach normal form [3, 4, 5, 11]. But the usual algorithms possibly construct a context-free grammar G', where the size of G' is exponential in the size of G (see [4], pp. 113–115 for an example). Given a context-free grammar G without ε-rules and without chain rules, Rosenkrantz [9] has given an algorithm which produces an equivalent context-free grammar G' in Greibach normal form such that $|G'| = O(|G|^3)$. Rosenkrantz gave no analysis of the size of G'. For an analysis, see [4], pp. 129–130 or [7]. Given an arbitrary context-free grammar $G = (V, \Sigma, P, S)$, the usual algorithm for the elimination of the chain rules can square the size of the grammar (see [4], p. 102 for an example). No better algorithm is known. Hence, given an arbitrary context-free grammar G, the elimination of the chain rules in a first step and applying Rosenkrantz's algorithm in a second step can produce an equivalent context-free grammar G' in Greibach normal form of size $O(|G|^6)$.

Rosenkrantz's algorithm uses formal power series. In [10] Urbanek has given an algorithm for the transformation of a given context-free grammar

in Chomsky normal form into Greibach normal form which produces in a pure derivation-oriented way without using systems of equations the same grammar as Rosenkrantz's algorithm. Ehrenfeucht and Rozenberg [2] have given another algorithm which constructs for a given arbitrary ε-free context-free grammar G an equivalent grammar in 2 Greibach normal form of size $O(|G|^6)$. They also use the language L_B of sentential forms of terminal leftmost derivations introduced in Section 2. But during the construction they use a chain rule free right linear scheme H, where the absence of chain rules seems to be essential. In [8], we have given a similiar construction. But since we do not need the chain rule freedom in between, we get an equivalent context-free grammar in 2 Greibach normal form of size $O(|G|^4)$.

We will develop a more direct method for placing a given context-free grammar into Greibach normal form with only polynomial increase of its size. Starting with an arbitrary ε-free context-free grammar G, we transform G into an equivalent context-free grammar H in extended Greibach normal form. The size of H will be $O(|G|^3)$. Moreover, in the case that G is chain rule free, H will be already in Greibach normal form. If H is not chain rule free, then we use the standard method for chain rule elimination for the transformation of H into Greibach normal form. The size of the constructed grammar is $O(|G|^4)$.

In [6], Pirická-Kelemenová has shown for a specific infinite family of context-free grammars G that any equivalent context-free grammar in Greibach normal form has size $\Omega(|G|^2)$ (see also [4], p. 131). This is the best lower bound known so far such that a gap of $O(|G|^2)$ between to the best lower bound and the new upper bound still exists.

2 The method

Let $G = (V, \Sigma, P, S)$ be an arbitrary ε-free context-free grammar. Note that by the definition of ε-freedom, the start symbol does not appear at the right side of any production. Productions of type $A \to a\alpha$ with $a \in \Sigma$ already fulfill the Greibach normal form properties. Our goal is now to replace the productions of type $A \to B\alpha$, $B \in N \setminus \{S\}$ by productions which fulfill the Greibach normal form properties.

The idea is the following. For all $B \in N \setminus \{S\}$, we want to construct a

context-free grammar $G_B = (V_B, V, P_B, S_B)$ such that

a) G_B is in extended Greibach normal form; i.e., for each rule $A \to \alpha$ there holds $\alpha = a\gamma$ with $a \in V$ or $\alpha \in N_B = V_B \setminus V$,

b) $S_B \to \alpha \in P_B$ implies that $\alpha = a\gamma$ with $a \in \Sigma$ and $\gamma \in (V_B \setminus \{S_B\})^*$, and

c) H is obtained from G by replacing each production $A \to B\alpha$, $B \in N \setminus \{S\}$ by the set $\{A \to a\gamma\alpha \mid S_B \to a\gamma \in P_B\}$ of productions and adding $P_B \setminus \{S_B \to \alpha \mid \alpha \in (V_B \setminus \{S_B\})^*\}$, $B \in N \setminus \{S\}$ to the set of productions.

For the construction of G_B, we are interested in leftmost derivations of the form

$$B \Rightarrow a\gamma \text{ or } B \Rightarrow^*_{lm} C\alpha \Rightarrow a\gamma\alpha,$$

where $a \in \Sigma$, $C \in N \setminus \{S\}$ and $\alpha, \gamma \in (V \setminus \{S\})^*$. Up to the last replacement, only alternatives from $N(V \setminus \{S\})^*$ are chosen. The last replacement chooses for C an alternative in $\Sigma(V \setminus \{S\})^*$. Such a leftmost derivation is called *terminal leftmost derivation* and is denoted by

$$B \Rightarrow_{tlm} a\gamma \text{ and } B \Rightarrow^*_{tlm} a\gamma\alpha, \text{ respectively.}$$

Let $L_B = \{a\delta \in \Sigma(V \setminus \{S\})^* \mid B \Rightarrow^*_{tlm} a\delta\}$. Our goal is to construct a context-free grammar $G_B = (V_B, V, P_B, S_B)$ such that

a) $L(G_B) = L_B$, and

b) each alternative of a variable begins with a symbol in V or is itself a variable.

For the construction of P_B, let us consider a terminal leftmost derivation

$$B \Rightarrow D_1\alpha_1 \Rightarrow D_2\alpha_2\alpha_1 \Rightarrow \ldots \Rightarrow D_t\alpha_t \ldots \alpha_1 \Rightarrow a\gamma\alpha_t \ldots \alpha_1$$

in more detail. Then $a \in \Sigma$, $D_i \in N \setminus \{S\}$ and $\gamma, \alpha_i \in (V \setminus \{S\})^*$, $1 \leq i \leq t$. $a\gamma\alpha = a\gamma\alpha_t \ldots \alpha_1$ is the corresponding terminal string in L_B.

For $A \in N$, the set $W(A)$ contains exactly the variables which can be reached from A using only chain rules; i.e.,

$$W(A) = \{C \in N \mid A \Rightarrow^* C\}.$$

Our goal is now to define the productions in P_B in a way such that a terminal leftmost derivation is simulated by a rightmost derivation backwards. For doing this, we introduce for all $C \in N$ the new variable C_B. The rightmost derivation with respect to the terminal leftmost derivation above is the following

$$S_B \Rightarrow a\gamma D_{B,t} \Rightarrow a\gamma\alpha_t D_{B,t-1} \Rightarrow \ldots \Rightarrow a\gamma\alpha_t \ldots \alpha_2 D_{B,1} \Rightarrow a\gamma\alpha_t \ldots \alpha_1.$$

There are three types of productions:

1. Productions with the start symbol S_B on the left side, the so-called *start productions*. These productions correspond to productions of G where the first symbol of the right side is in Σ.

2. Productions which are no start productions with a variable in $N_B \backslash \{S_B\}$ on the right side, the so-called *inner productions*. These productions correspond to productions of G where the first symbol of the right side is in $N \setminus \{S\}$.

3. Productions which are no start productions with no variable in $N_B \setminus \{S_B\}$ on the right side, the so-called *final productions*. These productions correspond to productions of G where the left side is in $W(B)$ and the first symbol of the right side is in $N \setminus \{S\}$.

Altogether, we obtain the context-free grammar $G_B = (V_B, V, P_B, S_B)$ defined by

$$\begin{array}{lll} V_B = & \{A_B & \mid A \in N\} \cup V, \text{ and} \\ P_B = & \{S_B \to a\gamma & \mid C \to a\gamma \in P \text{ for } C \in W(B), a \in \Sigma, \gamma \in V^*\} \\ & \cup\{S_B \to a\gamma C_B & \mid C \to a\gamma \in P, a \in \Sigma, \gamma \in V^*\} \\ & \cup\{C_B \to \alpha D_B & \mid D \to C\alpha \in P, D \in N \setminus \{S\}, C \in N, \alpha \in V^*\} \\ & \cup\{C_B \to \alpha & \mid D \to C\alpha \in P, D \in W(B), C \in N, \alpha \in V^+\}. \end{array}$$

The grammar G_B has the following properties:

1. $L(G_B) = L_B$

2. $|G_B| \leq 3|G|$

3. $S_B \to \alpha \in P_B$ implies that $\alpha = a\delta$, $a \in \Sigma$.

4. G_B is in extended Greibach normal form with respect to the terminal alphabet V.

5. $B \neq C$ implies $N_B \cap N_C = \emptyset$.

Starting with an arbitrary derivation in G_B and G, respectively, Property 1 can easily be proven by construction of the corresponding derivation with respect to the other context-free grammar G and G_B, respectively. Property 2 follows from the observation that for every production of G with first symbol in Σ there correspond at most two start productions of G_B and the fact that each other production of G corresponds to at most one inner production and to at most one final production. Note that the length of a start production is at most equal the length of the corresponding production in G plus 1 and the length of any other production is at most equal the length of the corresponding production in G. Properties 3 – 5 follow directly from the construction.

Now, we obtain H from G by performing the following algorithm:

(1) For all $B \in N \setminus \{S\}$ add P_B to P.

(2) For all $B, E \in N \setminus \{S\}$ replace

- each production $A \to B\alpha$ by $A \to S_B\alpha$ and
- each production $A_E \to B\alpha$ by $A_E \to S_B\alpha$.

(3) For all $B, E \in N \setminus \{S\}$ replace

- each production $A \to S_B\alpha$ by $\{A \to a\gamma\alpha \mid S_B \to a\gamma \in P_B\}$ and
- each production $A_E \to S_B\alpha$ by $\{A_E \to a\gamma\alpha \mid S_B \to a\gamma \in P_B\}$.

(4) For all $B \in N \setminus \{S\}$ remove $\{S_B \to \alpha \mid S_B \to \alpha \in P_B\}$.

The grammar $H = (V', \Sigma, P', S)$ has the following properties:

1. $L(H) = L(G)$

2. $|H| = O(|G|^3)$

3. H is in extended Greibach normal form.

4. For all $B \in N \setminus \{S\}$, G_B is replaced by an equivalent grammar G'_B of size $O(|G|^2)$.

5. If G has no chain rules then H is already in Greibach normal form.

6. If H is not chain rule free then all chain rules of H are of the form $D_E \to C_E$.

Property 1 follows directly from the construction. By the definition of L_B, $B \in N \setminus \{S\}$ it is clear that Steps 1 – 2 do not change the generated language. Step 3 replaces only some variables by all possible alternatives. Property 2 holds since for all $B \in N \setminus \{S\}$ the size of G_B is $O(|G|)$. Note that after performing Step 2, the size of the grammar is $O(|G|^2)$ and hence, after Step 3 $O(|G|^3)$. Moreover, Step 3 produces for all $B \in N \setminus \{S\}$ for G_B an equivalent context-free grammar G'_B of size $O(|G|^2)$. By construction, it is clear that the grammar is in extended Greibach normal form. The only possibility to construct chain rules is during the construction of inner productions in the case that $\alpha = \varepsilon$. But then, P contains the chain rule $D \to C$. Hence, Properties 5 and 6 are fulfilled.

In the case that H is not chain rule free, we use the standard method for chain rule elimination, getting an equivalent context-free grammar G' in Greibach normal form. This is done by performing for all $B \in N \setminus \{S\}$ the following algorithm:

(1) Compute $W(D_B)$ for all $D_B \in N_B$.

(2) Replace for all $D_B \in N_B$ $\{D_B \to E_B \mid D_B \to E_B \in P'\}$ by $\{D_B \to \alpha \mid \alpha \notin N_B \text{ and } \exists C_B \in W(D_B) : C_B \to \alpha \in P'\}$.

The grammar G' has the following properties:

1. $L(G') = L(G)$.

2. G' is in Greibach normal form.

3. $|G'| = O(|G|^4)$.

Properties 1 and 2 follow directly from the construction. Since for all $B \in N \setminus \{S\}$, the size of N_B is bounded by $|G|$ and the size of G'_B is bounded by

$O(|G|^2)$, Step 2 replaces each grammar G'_B by an equivalent grammar G''_B of size $O(|G|^3)$. This implies Property 3.

Altogether, we have proven the following theorem:

Theorem 1 *Let $G = (V, \Sigma, P, S)$ be an arbitrary ε-free context-free grammar. Then there exists an equivalent context-free grammar $G' = (V', \Sigma, P', S)$ in Greibach normal form such that $|G'| = O(|G|^3)$ if G is chain rule free and $|G'| = O(|G|^4)$ otherwise.*

If we want to construct for an arbitrary ε-free context-free grammar an equivalent context-free grammar in 2 Greibach normal form, then we transform G in a first step into an equivalent context-free grammar in extended Chomsky normal form and apply in a second step the algorithm above to the resulting grammar. It is easy to see that we get a context-free grammar G' in 2 Greibach normal form. Since the first step increases the size of the grammar only by a small constant factor (see e.g. [4]), $|G'| = O(|G|^4)$.

Acknowledgment: We thank Claus Rick for helpful comments, Werner Kuich for pointing out the work of Urbanek and of Ehrenfeucht and Rozenberg to the first author at the 14th STACS in Lübeck, and the referees for helpful suggestions and pointing out some minor errors.

References

[1] A. V. Aho, and J. D. Ullman, *The Theory of Parsing, Translation, and Compiling*, Vol. I: Parsing, Prentice-Hall (1972).

[2] A. Ehrenfeucht, and G. Rozenberg, An easy proof of Greibach normal form, *Inform. and Control* **63** (1984), 190–199.

[3] S. A. Greibach, A new normal-form theorem for context-free, phrase-structure grammars, *JACM* **12** (1965), 42–52.

[4] M. A. Harrison, *Introduction to Formal Language Theory*, Addison-Wesley (1978).

[5] J. E. Hopcroft, and J. D. Ullman, *Introduction to Autmata Theory, Languages, and Computation*, Addison-Wesley (1979).

[6] A. Pirická-Kelemenová, Greibach normal form complexity, *4th MFCS* (1975), 344–350.

[7] A. Kelemenová, Complexity of normal form grammars, *TCS* **28** (1984), 299–314.

[8] R. Koch, and N. Blum, Greibach normal form transformation, revisited, *14th STACS* (1997), 47–54.

[9] D. J. Rosenkrantz, Matrix equations and normal forms for context-free grammers, *JACM* **14** (1967), 501–507.

[10] F. J. Urbanek, On Greibach normal form construction, *TCS* **40** (1985), 315–317.

[11] D. Wood, *Theory of Computation*, Harper & Row (1987).