3.6 The Greibach Normal Form

Every CFG G can also be converted to an equivalent grammar in *Greibach Normal Form (for short, GNF)*. A context-free grammar $G = (V, \Sigma, P, S)$ is in Greibach Normal Form iff its productions are of the form

$$
\begin{aligned}
A &\to aBC, \\
A &\to aB, \\
A &\to a, \quad \text{or} \\
S &\to \epsilon,
\end{aligned}
$$

where $A, B, C \in N$, $a \in \Sigma$, $S \to \epsilon$ is in P iff $\epsilon \in L(G)$, and S does not occur on the right-hand side of any production.

Note that a grammar in Greibach Normal Form does not have ϵ-rules other than possibly $S \rightarrow \epsilon$. More importantly, except for the special rule $S \rightarrow \epsilon$, every rule produces some terminal symbol.

An important consequence of the Greibach Normal Form is that every nonterminal is not left recursive. A nonterminal A is *left recursive* iff $A \stackrel{+}{\Longrightarrow} A\alpha$ for some $\alpha \in V^*$. Left recursive nonterminals cause top-down determinitic parsers to loop. The Greibach Normal Form provides a way of avoiding this problem.

There are no easy proofs that every CFG can be converted to a Greibach Normal Form. We will give an elegant method due to Rosenkrantz (using matrices).

Lemma 3.6. *Given any context-free grammar* $G = (V, \Sigma, P, S)$*, one can construct a context-free grammar* $G' = (V', \Sigma, P', S')$ *such that* $L(G') = L(G)$ *and* G' *is in Greibach Normal Form, that is, a grammar whose productions are of the form*

$$\begin{aligned} A &\rightarrow aBC, \\ A &\rightarrow aB, \\ A &\rightarrow a, \quad \textit{or} \\ S' &\rightarrow \epsilon, \end{aligned}$$

where $A, B, C \in N'$*,* $a \in \Sigma$*,* $S' \rightarrow \epsilon$ *is in* P' *iff* $\epsilon \in L(G)$*, and* S' *does not occur on the right-hand side of any production in* P'*.*

3.7 Least Fixed-Points

Context-free languages can also be characterized as least fixed-points of certain functions induced by grammars.

This characterization yields a rather quick proof that every context-free grammar can be converted to Greibach Normal Form.

This characterization also reveals very clearly the recursive nature of the context-free languages.

We begin by reviewing what we need from the theory of partially ordered sets.

Definition 3.7. *Given a partially ordered set* $\langle A, \leq \rangle$*, an* *ω-chain* $(a_n)_{n \geq 0}$ *is a sequence such that* $a_n \leq a_{n+1}$ *for all* $n \geq 0$*. The* *least-upper bound* *of an* ω*-chain* (a_n) *is an element* $a \in A$ *such that:*

(1) $a_n \leq a$*, for all* $n \geq 0$*;*

(2) For any $b \in A$*, if* $a_n \leq b$*, for all* $n \geq 0$*, then* $a \leq b$*.*

A partially ordered set $\langle A, \leq \rangle$ *is an* *ω-chain complete poset* *iff it has a least element* $\perp$*, and iff every* ω*-chain has a least upper bound denoted as* $\bigsqcup a_n$*.*

Remark: The ω in ω-chain means that we are considering countable chains (ω is the ordinal associated with the order-type of the set of natural numbers).

For example, given any set X, the power set 2^X ordered by inclusion is an ω-chain complete poset with least element $\emptyset$.

The Cartesian product $\underbrace{2^X \times \cdots \times 2^X}_{n}$ ordered such that

$$(A_1, \ldots, A_n) \leq (B_1, \ldots, B_n)$$

iff $A_i \subseteq B_i$ (where $A_i, B_i \in 2^X$) is an ω-chain complete poset with least element $(\emptyset, \ldots, \emptyset)$.

We are interested in functions between partially ordered sets.

Definition 3.8. *Given any two partially ordered sets $\langle A_1, \leq_1\rangle$ and $\langle A_2, \leq_2\rangle$, a function $f\colon A_1 \to A_2$ is monotonic iff for all $x, y \in A_1$,*

$$x \leq_1 y \quad \textit{implies that} \quad f(x) \leq_2 f(y).$$

If $\langle A_1, \leq_1\rangle$ and $\langle A_2, \leq_2\rangle$ are ω-chain complete posets, a function $f\colon A_1 \to A_2$ is ω-continuous iff it is monotonic, and for every ω-chain (a_n),

$$f(\bigsqcup a_n) = \bigsqcup f(a_n).$$

Remark: Note that we are not requiring that an ω-continuous function $f\colon A_1 \to A_2$ preserve least elements, i.e., it is possible that $f(\perp_1) \neq \perp_2$.

We now define the crucial concept of a least fixed-point.

Definition 3.9. *Let $\langle A, \leq\rangle$ be a partially ordered set, and let $f\colon A \to A$ be a function. A fixed-point of f is an element $a \in A$ such that $f(a) = a$. The least fixed-point of f is an element $a \in A$ such that $f(a) = a$, and for every $b \in A$ such that $f(b) = b$, then $a \leq b$.*

The following lemma gives sufficient conditions for the existence of least fixed-points. It is one of the key lemmas in denotational semantics.

Lemma 3.7. *Let $\langle A, \leq \rangle$ be an ω-chain complete poset with least element $\perp$. Every ω-continuous function $f \colon A \to A$ has a unique least fixed-point x_0 given by*

$$x_0 = \bigsqcup f^n(\perp).$$

Furthermore, for any $b \in A$ such that $f(b) \leq b$, then $x_0 \leq b$.

The second part of lemma 3.7 is very useful to prove that functions have the same least fixed-point.

For example, under the conditions of lemma 3.7, if $g \colon A \to A$ is another ω-chain continuous function, letting x_0 be the least fixed-point of f and y_0 be the least fixed-point of g, if $f(y_0) \leq y_0$ and $g(x_0) \leq x_0$, we can deduce that $x_0 = y_0$.

Lemma 3.7 also shows that the least fixed-point x_0 of f can be approximated as much as desired, using the sequence $(f^n(\perp))$.

We will now apply this fact to context-free grammars. For this, we need to show how a context-free grammar $G = (V, \Sigma, P, S)$ with m nonterminals induces an ω-continuous map

$$\Phi_G \colon \underbrace{2^{\Sigma^*} \times \cdots \times 2^{\Sigma^*}}_{m} \rightarrow \underbrace{2^{\Sigma^*} \times \cdots \times 2^{\Sigma^*}}_{m}.$$

3.8 Context-Free Languages as Least Fixed-Points

Given a context-free grammar $G = (V, \Sigma, P, S)$ with m nonterminals $A_1, \ldots A_m$, grouping all the productions having the same left-hand side, the grammar G can be concisely written as

$$\begin{aligned} A_1 &\to \alpha_{1,1} + \cdots + \alpha_{1,n_1}, \\ \cdots &\to \cdots \\ A_i &\to \alpha_{i,1} + \cdots + \alpha_{i,n_i}, \\ \cdots &\to \cdots \\ A_m &\to \alpha_{m,1} + \cdots + \alpha_{m,n_n}. \end{aligned}$$

Given any set A, let $\mathcal{P}_{fin}(A)$ be the set of finite subsets of A.

Definition 3.10. *Let $G = (V, \Sigma, P, S)$ be a context-free grammar with m nonterminals A_1, ..., A_m. For any m-tuple $\Lambda = (L_1, \ldots, L_m)$ of languages $L_i \subseteq \Sigma^*$, we define the function*

$$\Phi[\Lambda] \colon \mathcal{P}_{fin}(V^*) \to 2^{\Sigma^*}$$

inductively as follows:

$$\begin{aligned}
\Phi[\Lambda](\emptyset) &= \emptyset, \\
\Phi[\Lambda](\{\epsilon\}) &= \{\epsilon\}, \\
\Phi[\Lambda](\{a\}) &= \{a\}, \quad \textit{if } a \in \Sigma, \\
\Phi[\Lambda](\{A_i\}) &= L_i, \quad \textit{if } A_i \in N, \\
\Phi[\Lambda](\{\alpha X\}) &= \Phi[\Lambda](\{\alpha\})\Phi[\Lambda](\{X\}), \\
&\quad \textit{if } \alpha \in V^+,\ X \in V, \\
\Phi[\Lambda](Q \cup \{\alpha\}) &= \Phi[\Lambda](Q) \cup \Phi[\Lambda](\{\alpha\}), \\
&\quad \textit{if } Q \in \mathcal{P}_{fin}(V^*), Q \neq \emptyset, \alpha \in V^*, \alpha \notin Q.
\end{aligned}$$

Then, writing the grammar G as

$$
\begin{aligned}
A_1 &\to \alpha_{1,1} + \cdots + \alpha_{1,n_1},\\
\cdots &\to \cdots\\
A_i &\to \alpha_{i,1} + \cdots + \alpha_{i,n_i},\\
\cdots &\to \cdots\\
A_m &\to \alpha_{m,1} + \cdots + \alpha_{m,n_n},
\end{aligned}
$$

we define the map

$$\Phi_G \colon \underbrace{2^{\Sigma^*} \times \cdots \times 2^{\Sigma^*}}_{m} \to \underbrace{2^{\Sigma^*} \times \cdots \times 2^{\Sigma^*}}_{m}$$

such that

$$
\begin{aligned}
&\Phi_G(L_1, \ldots L_m) =\\
&\quad (\Phi[\Lambda](\{\alpha_{1,1}, \ldots, \alpha_{1,n_1}\}), \ldots, \Phi[\Lambda](\{\alpha_{m,1}, \ldots, \alpha_{m,n_m}\}))
\end{aligned}
$$

for all $\Lambda = (L_1, \ldots, L_m) \in \underbrace{2^{\Sigma^*} \times \cdots \times 2^{\Sigma^*}}_{m}$.

One should verify that the map $\Phi[\Lambda]$ is well defined, but this is easy.

The following lemma is easily shown:

Lemma 3.8. *Given a context-free grammar $G = (V, \Sigma, P, S)$ with m nonterminals A_1, ..., A_m, the map*

$$\Phi_G\colon \underbrace{2^{\Sigma^*} \times \cdots \times 2^{\Sigma^*}}_{m} \to \underbrace{2^{\Sigma^*} \times \cdots \times 2^{\Sigma^*}}_{m}$$

is ω-continuous.

Now, $\underbrace{2^{\Sigma^*} \times \cdots \times 2^{\Sigma^*}}_{m}$ is an ω-chain complete poset, and the map Φ_G is ω-continous.

Thus, by lemma 3.7, the map Φ_G has a least-fixed point.

It turns out that the components of this least fixed-point are precisely the languages generated by the grammars (V, Σ, P, A_i).

Example. Consider the grammar
$G = (\{A, B, a, b\}, \{a, b\}, P, A)$ defined by the rules

$$\begin{aligned} A &\to BB + ab, \\ B &\to aBb + ab. \end{aligned}$$

The least fixed-point of Φ_G is the least upper bound of the chain

$$(\Phi_G^n(\emptyset, \emptyset)) = ((\Phi_{G,A}^n(\emptyset, \emptyset), \Phi_{G,B}^n(\emptyset, \emptyset)),$$

where

$$\Phi_{G,A}^0(\emptyset, \emptyset) = \Phi_{G,B}^0(\emptyset, \emptyset) = \emptyset,$$

and

$$\begin{aligned} \Phi_{G,A}^{n+1}(\emptyset, \emptyset) &= \Phi_{G,B}^n(\emptyset, \emptyset)\Phi_{G,B}^n(\emptyset, \emptyset) \cup \{ab\}, \\ \Phi_{G,B}^{n+1}(\emptyset, \emptyset) &= a\Phi_{G,B}^n(\emptyset, \emptyset)b \cup \{ab\}. \end{aligned}$$

It is easy to verify that

$$
\begin{aligned}
&\Phi^1_{G,A}(\emptyset,\emptyset) = \{ab\},\\
&\Phi^1_{G,B}(\emptyset,\emptyset) = \{ab\},\\
&\Phi^2_{G,A}(\emptyset,\emptyset) = \{ab, abab\},\\
&\Phi^2_{G,B}(\emptyset,\emptyset) = \{ab, aabb\},\\
&\Phi^3_{G,A}(\emptyset,\emptyset) = \{ab, abab, abaabb, aabbab, aabbaabb\},\\
&\Phi^3_{G,B}(\emptyset,\emptyset) = \{ab, aabb, aaabbb\}.
\end{aligned}
$$

By induction, we can easily prove that the two components of the least fixed-point are the languages

$$L_A = \{a^m b^m a^n b^n \mid m, n \geq 1\} \cup \{ab\}$$

and

$$L_B = \{a^n b^n \mid n \geq 1\}.$$

Letting $G_A = (\{A, B, a, b\}, \{a, b\}, P, A)$ and $G_B = (\{A, B, a, b\}, \{a, b\}, P, B)$, it is indeed true that $L_A = L(G_A)$ and $L_B = L(G_B)$.

We have the following theorem due to Ginsburg and Rice:

Theorem 3.9. *Given a context-free grammar* $G = (V, \Sigma, P, S)$ *with* m *nonterminals* $A_1, \ldots, A_m$, *the least fixed-point of the map* Φ_G *is the* m*-tuple of languages*

$$(L(G_{A_1}), \ldots, L(G_{A_m})),$$

where $G_{A_i} = (V, \Sigma, P, A_i)$.

Proof. Writing G as

$$\begin{aligned} A_1 &\to \alpha_{1,1} + \cdots + \alpha_{1,n_1}, \\ \cdots &\to \cdots \\ A_i &\to \alpha_{i,1} + \cdots + \alpha_{i,n_i}, \\ \cdots &\to \cdots \\ A_m &\to \alpha_{m,1} + \cdots + \alpha_{m,n_n}, \end{aligned}$$

let $M = \max\{|\alpha_{i,j}|\}$ be the maximum length of right-hand sides of rules in P.

Let

$$\Phi_G^n(\emptyset, \ldots, \emptyset) = (\Phi_{G,1}^n(\emptyset, \ldots, \emptyset), \ldots, \Phi_{G,m}^n(\emptyset, \ldots, \emptyset)).$$

Then, for any $w \in \Sigma^*$, observe that

$$w \in \Phi_{G,i}^1(\emptyset, \ldots, \emptyset)$$

iff there is some rule $A_i \to \alpha_{i,j}$ with $w = \alpha_{i,j}$, and that

$$w \in \Phi_{G,i}^n(\emptyset, \ldots, \emptyset)$$

for some $n \geq 2$ iff there is some rule $A_i \to \alpha_{i,j}$ with $\alpha_{i,j}$ of the form

$$\alpha_{i,j} = u_1 A_{j_1} u_2 \cdots u_k A_{j_k} u_{k+1},$$

where $u_1, \ldots, u_{k+1} \in \Sigma^*$, $k \geq 1$, and some $w_1, \ldots, w_k \in \Sigma^*$ such that

$$w_h \in \Phi_{G,j_h}^{n-1}(\emptyset, \ldots, \emptyset),$$

and

$$w = u_1 w_1 u_2 \cdots u_k w_k u_{k+1}.$$

We prove the following two claims:

Claim 1: For every $w \in \Sigma^*$, if $A_i \stackrel{n}{\Longrightarrow} w$, then $w \in \Phi^p_{G,i}(\emptyset, \ldots, \emptyset)$, for some $p \geq 1$.

Claim 2: For every $w \in \Sigma^*$, if $w \in \Phi^n_{G,i}(\emptyset, \ldots, \emptyset)$, with $n \geq 1$, then $A_i \stackrel{p}{\Longrightarrow} w$ for some $p \leq (M+1)^{n-1}$.

Combining Claim 1 and Claim 2, we have

$$L(G_{A_i}) = \bigcup_n \Phi^n_{G,i}(\emptyset, \ldots, \emptyset),$$

which proves that the least fixed-point of the map Φ_G is the m-tuple of languages

$$(L(G_{A_1}), \ldots, L(G_{A_m})).$$

$\square$

We now show how theorem 3.9 can be used to give a short proof that every context-free grammar can be converted to Greibach Normal Form.

3.9 Least Fixed-Points and the Greibach Normal Form

The hard part in converting a grammar $G = (V, \Sigma, P, S)$ to Greibach Normal Form is to convert it to a grammar in so-called *weak Greibach Normal Form*, where the productions are of the form

$$\begin{aligned} A &\to a\alpha, \quad \text{or} \\ S &\to \epsilon, \end{aligned}$$

where $a \in \Sigma$, $\alpha \in V^*$, and if $S \to \epsilon$ is a rule, then S does not occur on the right-hand side of any rule.

Indeed, if we first convert G to Chomsky Normal Form, it turns out that we will get rules of the form $A \to aBC$, $A \to aB$ or $A \to a$.

Using the algorithm for eliminating ϵ-rules and chain rules, we can first convert the original grammar to a grammar with no chain rules and no ϵ-rules except possibly $S \to \epsilon$, in which case, S does not appear on the right-hand side of rules.

Thus, for the purpose of converting to weak Greibach Normal Form, we can assume that we are dealing with grammars without chain rules and without ϵ-rules.

Let us also assume that we computed the set $T(G)$ of nonterminals that actually derive some terminal string, and that useless productions involving symbols not in $T(G)$ have been deleted.

Let us explain the idea of the conversion using the following grammar:

$$\begin{aligned} A &\to AaB + BB + b. \\ B &\to Bd + BAa + aA + c. \end{aligned}$$

The first step is to group the right-hand sides α into two categories: those whose leftmost symbol is a terminal ($\alpha \in \Sigma V^*$) and those whose leftmost symbol is a nonterminal ($\alpha \in NV^*$).

It is also convenient to adopt a matrix notation, and we can write the above grammar as

$$(A, B) = (A, B) \begin{pmatrix} aB & \emptyset \\ B & \{d, Aa\} \end{pmatrix} + (b, \{aA, c\})$$

Thus, we are dealing with matrices (and row vectors) whose entries are finite subsets of V^*.

For notational simplicity, braces around singleton sets are omitted.

The finite subsets of V^* form a semiring, where addition is union, and multiplication is concatenation.

Addition and multiplication of matrices are as usual, except that the semiring operations are used.

We will also consider matrices whose entries are languages over Σ.

Again, the languages over Σ form a semiring, where addition is union, and multiplication is concatenation. The identity element for addition is $\emptyset$, and the identity element for multiplication is $\{\epsilon\}$.

As above, addition and multiplication of matrices are as usual, except that the semiring operations are used.

For example, given any languages $A_{i,j}$ and $B_{i,j}$ over Σ, where $i, j \in \{1, 2\}$, we have

$$\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix} = \begin{pmatrix} A_{1,1}B_{1,1} \cup A_{1,2}B_{2,1} & A_{1,1}B_{1,2} \cup A_{1,2}B_{2,2} \\ A_{2,1}B_{1,1} \cup A_{2,2}B_{2,1} & A_{2,1}B_{1,2} \cup A_{2,2}B_{2,2} \end{pmatrix}$$

Letting $X = (A, B)$, $K = (b, \{aA, c\})$, and

$$H = \begin{pmatrix} aB & \emptyset \\ B & \{d, Aa\} \end{pmatrix}$$

the above grammar can be concisely written as

$$X = XH + K.$$

More generally, given any context-free grammar $G = (V, \Sigma, P, S)$ with m nonterminals A_1, ..., A_m, assuming that there are no chain rules, no ϵ-rules, and that every nonterminal belongs to $T(G)$, letting

$$X = (A_1, \ldots, A_m),$$

we can write G as

$$X = XH + K,$$

for some appropriate $m \times m$ matrix H in which every entry contains a set (possibly empty) of strings in V^+, and some row vector K in which every entry contains a set (possibly empty) of strings α each beginning with a terminal ($\alpha \in \Sigma V^*$).

Given an $m \times m$ square matrix $A = (A_{i,j})$ of languages over Σ, we can define the matrix A^* whose entry $A^*_{i,j}$ is given by

$$A^*_{i,j} = \bigcup_{n \geq 0} A^n_{i,j},$$

where $A^0 = Id_m$, the identity matrix, and A^n is the n-th power of A. Similarly, we define A^+, where

$$A^+_{i,j} = \bigcup_{n \geq 1} A^n_{i,j}.$$

Given a matrix A where the entries are finite subset of V^*, where $N = \{A_1, \ldots, A_m\}$, for any m-tuple $\Lambda = (L_1, \ldots, L_m)$ of languages over Σ, we let

$$\Phi[\Lambda](A) = (\Phi[\Lambda](A_{i,j})).$$

Given a system $X = XH + K$ where H is an $m \times m$ matrix and X, K are row matrices, if H and K do not contain any nonterminals, we claim that the least fixed-point of the grammar G associated with $X = XH + K$ is KH^*.

This is easily seen by computing the approximations $X^n = \Phi_G^n(\emptyset, \ldots, \emptyset)$. Indeed, $X^0 = K$, and

$$\begin{aligned} X^n &= KH^n + KH^{n-1} + \cdots + KH + K \\ &= K(H^n + H^{n-1} + \cdots + H + I_m). \end{aligned}$$

Similarly, if Y is an $m \times m$ matrix of nonterminals, the least fixed-point of the grammar associated with $Y = HY + H$ is H^+ (provided that H does not contain any nonterminals).

Given any context-free grammar $G = (V, \Sigma, P, S)$ with m nonterminals A_1, ..., A_m, writing G as $X = XH + K$ as explained earlier, we can form another grammar GH by creating m^2 new nonterminals $Y_{i,j}$, where the rules of this new grammar are defined by the system of two matrix equations

$$\begin{aligned} X &= KY + K, \\ Y &= HY + H, \end{aligned}$$

where $Y = (Y_{i,j})$.

The following lemma is the key to the Greibach Normal Form:

Lemma 3.10. *Given any context-free grammar* $G = (V, \Sigma, P, S)$ *with* m *nonterminals* $A_1, \ldots, A_m$, *writing* G *as*

$$X = XH + K$$

as explained earlier, if GH *is the grammar defined by the system of two matrix equations*

$$\begin{aligned} X &= KY + K, \\ Y &= HY + H, \end{aligned}$$

as explained above, then the components in X *of the least-fixed points of the maps* Φ_G *and* Φ_{GH} *are equal.*

Note that the above lemma actually applies to any grammar.

Applying lemma 3.10 to our example grammar, we get the following new grammar:

$$(A, B) = (b, \{aA, c\}) \begin{pmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{pmatrix} + (b, \{aA, c\}),$$

$$\begin{pmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{pmatrix} = \begin{pmatrix} aB & \emptyset \\ B & \{d, Aa\} \end{pmatrix} \begin{pmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{pmatrix} + \begin{pmatrix} aB & \emptyset \\ B & \{d, Aa\} \end{pmatrix}$$

There are still some nonterminals appearing as leftmost symbols, but using the equations defining A and B, we can replace A with

$$\{bY_1, aAY_3, cY_3, b\}$$

and B with

$$\{bY_2, aAY_4, cY_4, aA, c\},$$

obtaining a system in weak Greibach Normal Form.

This amounts to converting the matrix

$$H = \begin{pmatrix} aB & \emptyset \\ B & \{d, Aa\} \end{pmatrix}$$

to the matrix L shown below

$$\begin{pmatrix} aB & \emptyset \\ \{bY_2, aAY_4, cY_4, aA, c\} & \{d, bY_1a, aAY_3a, cY_3a, ba\} \end{pmatrix}$$

The weak Greibach Normal Form corresponds to the new system

$$\begin{aligned} X &= KY + K, \\ Y &= LY + L. \end{aligned}$$

This method works in general for any input grammar with no ϵ-rules, no chain rules, and such that every nonterminal belongs to $T(G)$.

Under these conditions, the row vector K contains some nonempty entry, all strings in K are in ΣV^*, and all strings in H are in V^+.

After obtaining the grammar GH defined by the system

$$\begin{aligned} X &= KY + K, \\ Y &= HY + H, \end{aligned}$$

we use the system $X = KY + K$ to express every nonterminal A_i in terms of expressions containing strings $\alpha_{i,j}$ involving a terminal as the leftmost symbol ($\alpha_{i,j} \in \Sigma V^*$), and we replace all leftmost occurrences of nonterminals in H (occurrences A_i in strings of the form $A_i\beta$, where $\beta \in V^*$) using the above expressions.

In this fashion, we obtain a matrix L, and it is immediately shown that the system

$$\begin{aligned} X &= KY + K, \\ Y &= LY + L, \end{aligned}$$

generates the same tuple of languages. Furthermore, this last system corresponds to a weak Greibach Normal Form.

It we start with a grammar in Chomsky Normal Form (with no production $S \to \epsilon$) such that every nonterminal belongs to $T(G)$, we actually get a Greibach Normal Form (the entries in K are terminals, and the entries in H are nonterminals).

The method is also quite economical, since it introduces only m^2 new nonterminals. However, the resulting grammar may contain some useless nonterminals.