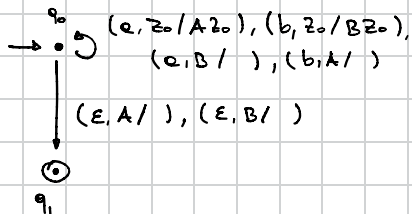


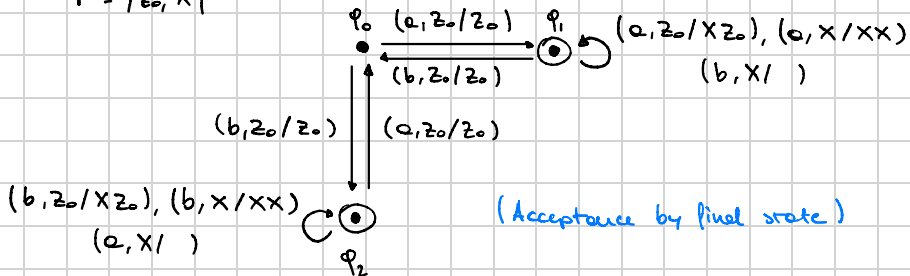
$$\Sigma = \{a, b\}$$

$$\Gamma = \{z_0, A, B\}$$

$$L = \{w \mid w \in \Sigma^*, \#_a(w) \neq \#_b(w)\}$$



$$\Gamma = \{z_0, x\}$$



Elimination of ϵ -productions

Observation : If $\epsilon \in L$ we **cannot** eliminate ϵ -productions preserving the generated language

We prove that if L is a context-free language, then there is a CFG without ϵ -productions that generates $L \setminus \{\epsilon\}$

Elimination of ϵ -productions

Variable A is **nullable** if $A \xRightarrow{*} \epsilon$

Idea : If A is nullable and there exists a production $B \rightarrow CAD$, then

- we remove productions with right-hand side ϵ
- we construct two alternative versions of the above production

$B \rightarrow CD$

A generates ϵ

$B \rightarrow CAD$

A generates other strings

Algorithm for nullable variables

Let $G = (V, T, P, S)$. We can compute the set $n(G)$ of all nullable variables of G by means of the following inductive algorithm

Base $n(G) \leftarrow \{A \mid (A \rightarrow \epsilon) \in P\}$

Induction If there exists in G a production $A \rightarrow B_1 B_2 \cdots B_k$ such that $B_i \in n(G)$ for each i , $1 \leq i \leq k$, then

$$n(G) \leftarrow n(G) \cup \{A\}$$

Elimination of ϵ -productions

Let $G = (V, T, P, S)$ be some CFG. We can compute $n(G)$ and build a new CFG $G_1 = (V, T, P_1, S)$ where P_1 is computed from P as follows

- each production $(A \rightarrow \epsilon) \in P$ is excluded from P_1
- let $p : (A \rightarrow X_1 X_2 \cdots X_k) \in P$ with $k \geq 1$; define $\mathcal{N} = \{i_1, i_2, \dots, i_m\}$ as the set of all indices of nullable variables X_{i_j} , $m \leq k$
- for every possible choice of set $\mathcal{N}' \subseteq \mathcal{N}$, we add to P_1 a production constructed from p by deleting each X_{i_j} with $i_j \in \mathcal{N}'$

Exception : In case $m = k$, we do not add to P_1 the null production $A \rightarrow \epsilon$

Example

Elimination of ϵ -production from CFG G with productions

$$S \rightarrow AB$$

$$A \rightarrow aAA \mid \epsilon$$

$$B \rightarrow bBB \mid \epsilon$$

We first compute set $n(G)$

- $A, B \in n(G)$ since $A \rightarrow \epsilon$ and $B \rightarrow \epsilon$
- $S \in n(G)$ since $S \rightarrow AB$, with $A, B \in n(G)$

Example

From $S \rightarrow AB$ we construct the new productions $S \rightarrow AB \mid A \mid B$

From $A \rightarrow aAA$ we construct the new productions

$A \rightarrow aAA \mid aA \mid a$

From $B \rightarrow bBB$ we construct the new productions

$B \rightarrow bBB \mid bB \mid b$

The resulting CFG G_1 has productions

$$S \rightarrow AB \mid A \mid B$$
$$A \rightarrow aAA \mid aA \mid a$$
$$B \rightarrow bBB \mid bB \mid b$$

and we have $L(G_1) = L(G) - \{\epsilon\}$

Elimination of unary productions

Let $G = (V, T, P, S)$ be some CFG. A **unary** production has the form $A \rightarrow B$, where both A and B are variables in V

Observation : $A \rightarrow a$ and $A \rightarrow \epsilon$ are not unary productions

We can eliminate unary productions by expanding the variables in the right-hand side

Example

Our grammar for arithmetic expressions with productions

$$\begin{aligned}
 I &\rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \\
 F &\rightarrow I \mid (E) \\
 T &\rightarrow F \mid T * F \\
 E &\rightarrow T \mid E + T
 \end{aligned}$$

has unary productions $E \rightarrow T$, $T \rightarrow F$ and $F \rightarrow I$

Expanding the right-hand side of production $E \rightarrow T$ results in

$$E \rightarrow F \mid T * F$$

which introduces a new unary production

Example

If we in turn expand the right-hand side of $E \rightarrow F$ we get

$$E \rightarrow I \mid (E) \mid T * F$$

Finally, if we expand $E \rightarrow I$ we get

$$E \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \mid (E) \mid T * F$$

The method of successive expansions works if there is **no cycle** between unary rules, such as in

$$A \rightarrow B, \quad B \rightarrow C, \quad C \rightarrow A$$

Elimination of unary productions

We now present a method based on the notion of unary pairs which eliminates the unary productions in the **general case**

Let $G = (V, T, P, S)$ be some CFG. (A, B) is a **unary pair** if

$A \xRightarrow{*} B$ using only unary productions

Observation : For productions $A \rightarrow BC$ and $C \rightarrow \epsilon$ we have $A \xRightarrow{*} B$; however, we have not used unary productions only

Algorithm for unary pairs

Let $G = (V, T, P, S)$. We can compute the set $u(G)$ of all unary pairs G by means of the following inductive algorithm

Base $u(G) \leftarrow \{(A, A) \mid A \in V\}$ ↪ $A \Rightarrow A$ in 0 steps always (0 steps = all unary)

Induction If $(A, B) \in u(G)$ and $(B \rightarrow C) \in P$, then

$$u(G) \leftarrow u(G) \cup \{(A, C)\}$$

Example

Consider the CFG

$$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$F \rightarrow I \mid (E)$$

$$T \rightarrow F \mid T * F$$

$$E \rightarrow T \mid E + T$$

In the base step we derive the unary pairs (E, E) , (T, T) , (F, F) e (I, I)

Example

In the inductive step

- from (E, E) and $E \rightarrow T$ we add pair (E, T)
- from (E, T) and $T \rightarrow F$ we add pair (E, F)
- from (E, F) and $F \rightarrow I$ we add pair (E, I)
- from (T, T) and $T \rightarrow F$ we add pair (T, F)
- from (T, F) and $F \rightarrow I$ we add pair (T, I)
- from (F, F) and $F \rightarrow I$ we add pair (F, I)

Eliminating unary productions

Let $G = (V, T, P, S)$ be some CFG. We produce a new CFG $G_1 = (V, T, P_1, S)$, where P_1 is constructed from P as follows

- compute $u(G)$
- for each $(A, B) \in u(G)$ add to P_1 a production $A \rightarrow \alpha$ for each $(B \rightarrow \alpha) \in P$ which is not a unary production

Observation : In the second step, we might have $A = B$; in this way non-unary productions in P are all transferred to P_1

Example

We eliminate unary productions from CFG

$$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$F \rightarrow I \mid (E)$$

$$T \rightarrow F \mid T * F$$

$$E \rightarrow T \mid E + T$$

We have already computed set $u(G)$ in a previous example

Example

The second step of the algorithm results in the following productions

Pair	Productions
(E, E)	$E \rightarrow E + T$
(E, T)	$E \rightarrow T * F$
(E, F)	$E \rightarrow (E)$
(E, I)	$E \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$
(T, T)	$T \rightarrow T * F$
(T, F)	$T \rightarrow (E)$
(T, I)	$T \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$
(F, F)	$F \rightarrow (E)$
(F, I)	$F \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$
(I, I)	$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

Example

Summing up, after eliminating unary productions from the grammar G with productions

$$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$F \rightarrow I \mid (E)$$

$$T \rightarrow F \mid T * F$$

$$E \rightarrow T \mid E + T$$

we have the CFG G_1 with productions

$$E \rightarrow E + T \mid T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$T \rightarrow T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$F \rightarrow (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

CFG simplification

When simplifying a CFG we need to pay special attention to the **order** in which we apply the previous transformations

The correct ordering is

- elimination of ϵ -productions
- elimination of unary productions
- elimination of useless symbols

Chomsky normal form

A CFG is in **Chomsky normal form**, or CNF for short, if its productions have one of the two forms

- $A \rightarrow BC$, with $A, B, C \in V$
- $A \rightarrow a$, with $A \in V$ and $a \in T$

and the grammar does not have useless symbols

We show that every CFL without the empty string ϵ can be generated by CNF grammar

Chomsky normal form

In order to transform a CFG in CNF, we first need to eliminate in the **specified order**

- ϵ -productions
- unary productions
- useless symbols

The resulting grammar has productions of the form

- $A \rightarrow a$
- $A \rightarrow \alpha$, where $\alpha \in (V \cup T)^*$ and $|\alpha| \geq 2$

Chomsky normal form

To transform the previous CFG in CNF, we need to perform two further transformations

- right-hand sides of length larger than 2 must only have variables
- right-hand sides of length larger than 2 must be decomposed into chains of productions with only two variables in their right-hand side

First transformation

For each production with right-hand side α such that $|\alpha| \geq 2$ and for each **occurrence** in α of $a \in T$

- construct a new production $A \rightarrow a$ (A a fresh variable)
- use A in place of a in α

Second transformation

For each production of the form

$$A \rightarrow B_1 B_2 \cdots B_k, \quad k \geq 3$$

- introduce new variables C_1, C_2, \dots, C_{k-2}
- replace the production with the chain of new productions (all B_i 's fresh symbols)

$$A \rightarrow B_1 C_1$$

$$C_1 \rightarrow B_2 C_2$$

$$\vdots$$

$$C_{k-3} \rightarrow B_{k-2} C_{k-2}$$

$$C_{k-2} \rightarrow B_{k-1} B_k$$

Example

Consider the CFG from the previous example

$$E \rightarrow E + T \mid T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$T \rightarrow T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$F \rightarrow (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

$$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$$

The first transformation adds productions for the terminal symbols

$$A \rightarrow a \quad B \rightarrow b \quad Z \rightarrow 0 \quad O \rightarrow 1$$

$$P \rightarrow + \quad M \rightarrow * \quad L \rightarrow (\quad R \rightarrow)$$

Example

The first transformation results in the CFG

$$E \rightarrow EPT \mid TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$$

$$T \rightarrow TMF \mid LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$$

$$F \rightarrow LER \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$$

$$I \rightarrow a \mid b \mid IA \mid IB \mid IZ \mid IO$$

$$A \rightarrow a, \quad B \rightarrow b, \quad Z \rightarrow 0, \quad O \rightarrow 1$$

$$P \rightarrow +, \quad M \rightarrow *, \quad L \rightarrow (, \quad R \rightarrow)$$

Example

The second transformations performs the following replacements

- $E \rightarrow EPT$ replaced by
 $E \rightarrow EC_1, C_1 \rightarrow PT$
- $E \rightarrow TMF, T \rightarrow TMF$ replaced by
 $E \rightarrow TC_2, T \rightarrow TC_2, C_2 \rightarrow MF$
- $E \rightarrow LER, T \rightarrow LER, F \rightarrow LER$ replaced by
 $E \rightarrow LC_3, T \rightarrow LC_3, F \rightarrow LC_3, C_3 \rightarrow ER$

Example

The second transformation results in the final CFG

$$E \rightarrow EC_1 \mid TC_2 \mid LC_3 \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$$

$$T \rightarrow TC_2 \mid LC_3 \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$$

$$F \rightarrow LC_3 \mid a \mid b \mid IA \mid IB \mid IZ \mid IO$$

$$I \rightarrow a \mid b \mid IA \mid IB \mid IZ \mid IO$$

$$C_1 \rightarrow PT, \quad C_2 \rightarrow MF, \quad C_3 \rightarrow ER$$

$$A \rightarrow a, \quad B \rightarrow b, \quad Z \rightarrow 0, \quad O \rightarrow 1$$

$$P \rightarrow +, \quad M \rightarrow *, \quad L \rightarrow (, \quad R \rightarrow)$$

Exercise

Cast into CNF the CFG $G = (\{S, A, B\}, \{a, b\}, P, S)$ with production set P

$$S \rightarrow bA \mid aB$$

$$A \rightarrow bAA \mid aS \mid a$$

$$B \rightarrow aBB \mid bS \mid b$$

There are no ϵ -productions, unary productions, or useless symbols. Therefore we apply the two transformations for the construction of the CNF

Exercise

The first transformation performs the following replacements

- $S \rightarrow bA$ replaced by $C_b \rightarrow b$ and $S \rightarrow C_bA$
- $S \rightarrow aB$ replaced by $C_a \rightarrow a$ and $S \rightarrow C_aB$
- $A \rightarrow bAA$ replaced by $A \rightarrow C_bAA$
- $A \rightarrow aS$ replaced by $A \rightarrow C_aS$
- $B \rightarrow aBB$ replaced by $B \rightarrow C_aBB$
- $B \rightarrow bS$ replaced by $B \rightarrow C_bS$

Exercise

The second transformation performs the following replacements

- $A \rightarrow C_b AA$ replaced by $A \rightarrow C_b D_1$ and $D_1 \rightarrow AA$
- $B \rightarrow C_a BB$ replaced by $B \rightarrow C_a D_2$ and $D_2 \rightarrow BB$

Exercise

The resulting CFG is

$$G_1 = (\{S, A, B, C_a, C_b, D_1, D_2\}, \{a, b\}, P', S)$$

where P' consists of the following productions

$$S \rightarrow C_b A \mid C_a B$$

$$A \rightarrow C_a S \mid C_b D_1 \mid a$$

$$B \rightarrow C_b S \mid C_a D_2 \mid b$$

$$D_1 \rightarrow AA$$

$$D_2 \rightarrow BB$$

$$C_a \rightarrow a$$

$$C_b \rightarrow b$$

Greibach normal form

A CFG is in **Greibach normal form** (GNF) if every production has the form

$$A \rightarrow a\alpha$$

with $a \in T$ and $\alpha \in V^*$

Important properties of GNF :

- every nonempty CFL with non-empty strings only has a GNF grammar
- a grammar in GNF generates a string of length n in exactly n steps
- if we turn a GNF grammar into a PDA, we get an automaton without ϵ -transitions