7. Normal Forms for Context-Free Grammars

Goal: Transform a given context-free grammar G
into another grammar G'
that
- has the same language but
- a simpler structure.

Motivation:
- Algorithms can then rely on the simpler structure
which will make them faster
(unless the blow-up in the conversion is prohibitive).
- It is easier for us to understand properties of context-free languages
if the grammars have a simpler structure.

7.1 Getting Rid of ε-Productions, Unit Productions, and Useless Non-Terminals

Lemma:
For every CFG G, we can construct a CFG G'
without ε-productions $(A \to \varepsilon)$ and
without unit productions $(A \to B)$
satisfying $L(G') = L(G) \setminus \{\varepsilon\}$.

Note: The result is a statement of existence of G'.
But it says more: We also know how to obtain G'.
If we not only know an algorithm exists
but we can actually give the method,
we say the algorithm is *effective*.

Proof: Let $G = (N, \Sigma, P, S)$.
We define $\hat{G} := (N, \Sigma, \hat{P}, S)$.

Here, $\hat{P}$ is the smallest set containing P and closed under the following rules:

(a) If $A \to \alpha B \beta$ and $B \to \varepsilon$ are in $\hat{P}$, then $A \to \alpha\beta$ is in $\hat{P}$.

(b) If $A \to B$ and $B \to \gamma$ are in $\hat{P}$, then $A \to \gamma$ is in $\hat{P}$.

- Note that only finitely many productions are added. The reason is that the length of the right-hand sides is bounded by the length of the longest right-hand side in P.
- Moreover, note that $L(\hat{G}) = L(G)$.

"$\supseteq$" Holds because $P \subseteq \hat{P}$.

"$\subseteq$" Holds because each new production can be simulated by two old productions that caused it to be added (formally, this needs an induction).

Claim:
For every $w \in \Sigma^+$, every derivation $S \Rightarrow_{\hat{G}}^* w$ of minimal length neither uses ε-productions nor unit productions.

Proof:
Consider $w \neq \varepsilon$ and a shortest derivation $S \Rightarrow_{\hat{G}}^* w$.
- Towards a contradiction, assume that $B \to \varepsilon$ is used at some point, say

$$S \Rightarrow^* \alpha_1 B \alpha_2 \Rightarrow \alpha_1 \alpha_2 \Rightarrow^* w.$$

- One of α_1 and α_2 is $\neq \varepsilon$, otherwise $w = \varepsilon$.

Thus, the B is not S but has been introduced at some point, say with a production $A \to \beta_1 B \beta_2$:

$$S \Rightarrow^m \gamma_1 A \gamma_2 \Rightarrow \gamma_1 \beta_1 B \beta_2 \gamma_2 \Rightarrow^n \alpha_1 B \alpha_2 \Rightarrow \alpha_1 \alpha_2 \Rightarrow^k w.$$

- But by Rule (a), also $A \to \beta_1 \beta_2$ is in $\hat{P}$.

If we apply that production instead, we obtain

$$S \Rightarrow^m \gamma_1 A \gamma_2 \Rightarrow \gamma_1 \beta_1 \beta_2 \gamma_2 \Rightarrow^n \alpha_1 \alpha_2 \Rightarrow^k w.$$

This contradicts the assumption that the given derivation was of minimal length. □

- For unit productions, the reasoning is similar.
- Since we neither need ε-productions nor unit productions to generate $w \neq \varepsilon$, we remove them from $\hat{G}$ and obtain G' with $L(G') = L(G) \setminus \{\varepsilon\}$. □

Definition:
Let $G = (N, \Sigma, P, S)$. A non-terminal $X \in N$ is useful, if

$$S \Rightarrow^* \alpha X \beta \Rightarrow^* w \in \Sigma^* \text{ for some } \alpha, \beta \in (N \cup \Sigma)^*.$$

Otherwise, X is called useless.

Theorem:
For every CFG G, we can construct a CFG G' without ε-productions, without unit productions, and without useless non-terminals so that $L(G') = L(G) \setminus \{\varepsilon\}$.

Proof:
Consider the grammar G'' without ε-productions and without unit productions obtained with the previous lemma.

- With a fixed point computed backwards, we determine the non-terminals that can derive a terminal word.
- With a fixed point computed forwards, we determine the non-terminals that are reachable from S.
- Intersecting the sets and removing the remaining non-terminals together with the productions that use them yields the desired grammar. □

7.2 Chomsky Normal Form

Definition:
A CFG is in Chomsky normal form (CNF) if all productions take the form
$A \to BC$ or $A \to a$.

Theorem:
For every CFG G, we can construct a CFG G' in CNF and with $L(G') = L(G) \setminus \{\varepsilon\}$.

We base our construction on the previous theorem.
As a result, all non-terminals will be useful,
there are no ε-productions and no unit productions.

Proof:
- We first apply the previous theorem to turn G into G'' without useless non-terminals, ε-productions, and unit productions.

- We introduce a new non-terminal A_a for every $a \in \Sigma$ that occurs in the right-hand side of a production. We replace every occurrence of a in a right-hand side by A_a. We add the production $A_a \to a$. (Note that this step does not introduce useless non-terminals.)

- Now all productions are of the form

  $A \to a$ or $A \to B_1 \dots B_k$ with $k \geq 2$.

- For any production $A \to B_1 \dots B_k$ with $k \geq 3$, we introduce a new non-terminal C and replace the production by

  $A \to B_1 C$ and $C \to B_2 \dots B_k$.

We repeat this until all productions have length 2. □

Example:

We derive a grammar in CNF for

$\{a^n b^n \mid n \in \mathbb{N}\} \setminus \{\varepsilon\} = \{a^n b^n \mid n \geq 1\}$.

Starting from

$S \to aSb \mid \varepsilon$ for $\{a^n b^n \mid n \in \mathbb{N}\}$,

we remove ε-productions as described in the first lemma.

This yields

$S \to aSb \mid ab$ generating $\{a^n b^n \mid n \geq 1\}$.

We add non-terminals A, B and replace the productions by

$S \to ASB \mid AB$ $A \to a$ $B \to b$.

Finally, we add a non-terminal C and replace $S \to ASB$ by

$S \to AC$ $C \to SB$.

7.3 Greibach Normal Form

Definition:

A CFG G is in <u>Greibach normal form (GNF)</u>.
if every production is of the form

$$A \to a B_1 \dots B_k, \quad k \geq 0.$$

Theorem:

For every CFG G, we can construct a CFG G' in GNF
with $L(G') = L(G) \setminus \{\varepsilon\}$.

We will base our construction on the Chomsky normal form.
Therefore, we will obtain $k \leq 2$, no useless non-terminals,
no ε-productions, no unit productions.

Proof:

- Let $G = (N, \Sigma, P, S)$ be in CNF, $A \in N$, $a \in \Sigma$.
  We define

$$R_{A,a} := \{ \beta \in N^* \mid A \Rightarrow_{SL}^* a\beta \}.$$

The <u>strong left derivation relation</u> $\Rightarrow_{SL}^* \subseteq (N \cup \Sigma)^* \times (N \cup \Sigma)^*$ is defined by

$\alpha_1 \Rightarrow_L^* \alpha_2$ if α_2 can be obtained from α_1
by a sequence of derivation steps
in which productions are only applied to the <u>leftmost symbol</u>
in which productions are only applied to the leftmost symbol

- The set $R_{A,a}$ is a <u>regular language</u> over the alphabet N.
  Indeed, take the left-linear grammar

$$(\{B' \mid B \in N\}, N, \{B' \to C'D \mid B \to CD \in P\}, A')$$
$$\cup \{B' \to \varepsilon \mid B' \to a\}$$

to generate it.

It contains a non-terminal B' for every non-terminal B in G.
The former non-terminals N now form the terminal symbols.
The productions always replace the leftmost non-terminal,
thus simulating a sequence of left derivations.
The process stops when the terminal symbol of interest a
has been found.

// Note that it is a substantial conceptual twist
to think about words over non-terminals
as a regular language.
You should really think this through.

- It is not difficult to show that for every left-linear grammar $G_{\Pi,a}$
there is a strongly right-linear grammar $G_{\Pi,a}$
with the same language.
Strong here means the productions are of the form

$$X \to B.Y \quad \text{or} \quad X \to \varepsilon.$$

where X, Y are non-terminals and B is a terminal symbol.
Let $T_{\Pi,a}$ be the start symbol of $G_{\Pi,a}$.

- We can assume that the non-terminals in all $G_{\Pi,a}$ and in G
are pairwise disjoint.
We form the grammar G_1 by adding to G
all non-terminals and productions of all $G_{\Pi,a}$.
The start symbol of G_1 is S.
The productions of G_1 are of the form

$$X \to \varepsilon \qquad X \to b \qquad X \to B.Y, \qquad X, B, Y \text{ non-terminals.}$$

Moreover, $L(G_1) = L(G)$ since none of the new non-terminals is reachable.

- We define the grammar G_2 from G_1 by replacing $X \to B.Y$ with the rules $X \to a.T_{B,a}.Y$ for all $a \in \Sigma$.

- What is going on here?
The idea is to guess the leading terminal symbol a that will be generated in a derivation sequence from B.
The possible sequences of non-terminals following B is a (that are generated from B) are captured by $G_{B,a}$ with start symbol $T_{B,a}$.
It may seem restrictive to only consider sequences of non-terminals following a.
But we do the same replacement also for the rules in $G_{B,a}$.
Therefore, we capture all sentential forms derivable from B.
The productions in G_2 are of the form

$X \to \varepsilon$, $X \to b$, and $X \to a.T_{B,a}.Y$.

- We get rid of ε-productions using the construction in the first lemma.
This construction does not introduce unit productions.
To see this, note that every non-ε-production has a terminal on the right-hand side.

- The resulting grammar G_3
  - is in GNF and
  - satisfies $L(G_3) = L(G) \setminus \{\varepsilon\}$.

- It has no ε-productions and no unit productions.
- It has no useless non-terminals if we make sure to only consider $R_{A,a} \neq \emptyset$ (and modify $G_{A,a}$ accordingly). □

Example:

Consider the grammar in CNF for balanced parantheses:

G: $S \to AB \mid AC \mid SS$ $\quad A \to [$ $\quad B \to]$

$C \to SB$

We compute the regular languages

(1) $R_{S,[} = (B \cup C).S^*$

(2) $R_{C,[} = (B+C).S^*.B$

(3) $R_{A,[} = \{\varepsilon\} = R_{B,]}$.

All other languages are $\emptyset$.

Corresponding strongly right-linear grammars are

(1) $T_{S,[} \to B.X \mid C.X$ $\quad X \to SX \mid \varepsilon$

(2) $T_{C,[} \to B.Y \mid CY$ $\quad Y \to S.Y \mid B.Z$ $\quad Z \to \varepsilon$

(3) $T_{A,[} \to \varepsilon$

$T_{B,]} \to \varepsilon$.

Combining these grammars with G, we obtain:

$S \to [. T_{A,[} . B \mid [. T_{A,[} . C \mid [. T_{S,[} . S$

$T_{S,[} \to] . T_{B,]} . X \mid [. T_{C,[} . X$

$T_{C,[} \to] . T_{B,]} . Y \mid [. T_{C,[} . Y$

$T_{A,[} \to \varepsilon$

$T_{B,]} \to \varepsilon$

$C \to [. T_{S,[} . B$

$X \to [. T_{S,[} . X \mid \varepsilon$

$Y \to [. T_{S,[} . Y \mid] . T_{B,]} . Z$

$A \to [$

$B \to]$

$Z \to \varepsilon .$

Removing ε-productions yields:

$S \to [. B \mid [. C \mid [. T_{S,[} . S$

$T_{S,[} \to] \mid] . X \mid [. T_{C,[} \mid [. T_{C,[} . X$

$T_{C,[} \to] . Y \mid [. T_{C,[} . Y$

$C \to [. T_{S,[} . B$

$X \to [. T_{S,[} . X \mid [. T_{S,[}$

$Y \to [. T_{S,[} . Y \mid]$

$A \to [$

$B \to] .$