Simplifications for Context-Free Grammars

- Let $G = (V, \Sigma, R, S)$ be a context-free grammar.
- A string $\alpha \in \Sigma^*$ is referred to as a *terminal string*.
- A rule of the form $A \leftarrow B$, where $A, B \in V$ is called a *unit rule*.
- Each of the four simplifications below converts the context-free grammar into an equivalent context-free grammar G'. These simplifications have to be done in the order presented:
  1. Removing all *useless variables*:
     (a) Removing all variables that do not derive a terminal string.
     (b) Removing all variables that are not reachable from the start symbol S.
  2. Removing all ϵ rules such that:
     - if $\epsilon \notin L(G)$ then G' has no epsilon rules, and
     - if $\epsilon \in L(G)$ then G' has only the following *epsilon* rule: $S' \rightarrow \epsilon$.
  3. Remove all unit rules.
- Remove from the resulting grammar all duplicate rules.
- All the simplifications in this notes involve inductive constructions.

Remove all variables that do not derive a terminal string

Goal: Given a contex-free grammar $G = (V, \Sigma, R, S)$, construct an equivalent grammar $G' = (V', \Sigma, R', S')$ such that all variables in G' derive some terminal string.

The first step is to construct the set T of all variabls that derive a terminal string:

$$\mathcal{T} = \{A \mid A \Longrightarrow^* \alpha \in \Sigma^*\}.$$

Construction of the set $\mathcal{T}$:

Base Case:

$$\mathcal{T}_0 = \{A \mid A \leftarrow \alpha \in R \text{ and } \alpha \in \Sigma^*\}.$$

Inductive Step:

Given $\mathcal{T}_i$, construct $\mathcal{T}_{i+1}$ as follows:

$$\mathcal{T}_{i+1} = \mathcal{T}_i \cup \{A \mid A \rightarrow \alpha \in R \text{ and } \alpha \; in \; (\mathcal{T}_i \cup \Sigma^*)\}.$$

Stopping condition:

Stop when $\mathcal{T}_i = \mathcal{T}_{i+1}$ for some i. Let $\mathcal{T}$ be the final set of variables.

Algorithm to decide of $L(G) = \emptyset$: Check if $S \notin \mathcal{T}$.

Normalization step: Remove from V all variables in $\mathcal{T}$ and remove from R all rules in which variables in $\mathcal{T}$ appear to get the new grammar G'.

Remove all variables that are not reachable from the start variable

Goal: Given a contex-free grammar $G = (V, \Sigma, R, S)$, construct an equivalent grammar $G' = (V', \Sigma, R', S')$ such that all variables in G' are reachable from the start symbol S'.

The first step is to construct the set $\mathcal{R}$ of all variables reachable from the start symbol S:

$$\mathcal{R} = \{A \in V \mid S \Longrightarrow^* \alpha A\, \beta \text{ for } \alpha, \beta\ in\ (V \cup \Sigma^*)\}.$$

Construction of the set $\mathcal{R}$:

Base case:

$$\mathcal{R}_0 = \{S\}.$$

Inductive Step:

Given $\mathcal{R}_i$, construct R_{i+1} as follows:

$$\mathcal{R}_{i+1} = \mathcal{R}_i \cup \{A \mid B \to \alpha A\, \beta \in R \text{ for some } B \in \mathcal{R}_i \text{ and } \alpha, \beta\ in\ (V \cup \Sigma^*)\}.$$

Stopping condition:

Stop when $\mathcal{R}_i = \mathcal{R}_{i+1}$ for some i. Let $\mathcal{R}$ be the final set of variables constructed.

Normalization step: Remove from V all variables in $\mathcal{R}$ and remove from R all rules in which variables in $\mathcal{R}$ appear to get the new grammar G'.

Remove *epsilon* rules

Goal: Given a contex-free grammar $G = (V, \Sigma, R, S)$, construct an equivalent grammar $G' = (V', \Sigma, R', S')$ such that:

- If $\epsilon \notin L(G)$ then G' has no epsilon rules.
- If $\epsilon \in L(G)$ then G' has only the following *epsilon* rule: $S' \rightarrow \epsilon$.

The first step is to construct the set $\mathcal{E}$ of all variables that derive the empty string ϵ:

$$\mathcal{E} = \{A \in V \mid A \Longrightarrow^* \epsilon\}.$$

Construction of the set $\mathcal{E}$:

Base case:

$$\mathcal{E}_1 = \{A \in V \mid A \rightarrow \epsilon\}.$$

Inductive Step:
Given $\mathcal{E}_i$, construct $\mathcal{E}_{i+1}$ as follows:

$$\mathcal{E}_{i+1} = \mathcal{E}_i \cup \{A \mid A \rightarrow \alpha \in R \text{ and } \alpha \in (V \cap \mathcal{E}_i)^+\}.$$

Stopping condition:
Stop when $\mathcal{E}_i = \mathcal{E}_{i+1}$ for some i. Let $\mathcal{E}$ be the final set of variables constructed.

Algorithm to decide if $\epsilon \in L(G)$: Check if $S \in \mathcal{E}$.

Normalization step:

Construct new rules as follows:

- Include in R' any rule of the form $A \rightarrow \alpha$ from R where α is not ϵ and does not have any variable from $\mathcal{E}$.
- For every rule of the form:

  $$A \rightarrow \alpha_0 B_1 \alpha_1 B_2 \cdots \alpha_{k-1} B_k,$$

  where for $0 \leq i \leq k$, $\alpha_i \cap \mathcal{E} = \emptyset$ and for $1 \leq i \leq k$, each $B_i \in \mathcal{E}$, include in R' all rules of the form:

  $$A \rightarrow \alpha_0 X_1 \alpha_1 X_2 \cdots \alpha_{k-1} X_k,$$

  where X_i is either B_i or ϵ.
- Do not include in R' any rule from R of the form $A \rightarrow \epsilon$.
- If $S \in \mathcal{E}$, then include in R' a rule of the form $S' \rightarrow \epsilon$.

Remove all unit rules

Goal: Given a contex-free grammar $G = (V, \Sigma, R, S)$, construct an equivalent grammar $G' = (V', \Sigma, R', S')$ such that G' has no unit rules.

The first step is to construct, for each variable $A \in V$, the set $\mathcal{U}(A)$ of all variables that are derivable using only unit rules:

$$\mathcal{U}(A) = \{B \in V \mid A \Longrightarrow^* B \text{ using only unit rules}\}.$$

Construction of the set $\mathcal{U}(A)$:

Base case:

$$\mathcal{U}_1(A) = \{B \in V \mid A \to B\}.$$

Inductive Step:
Given $\mathcal{U}_i(A)$, construct $\mathcal{U}_{i+1}(A)$ as follows:

$$\mathcal{U}_{i+1}(A) = \mathcal{U}_i(A) \cup \{D \in V \mid B \to D \text{ for some } B \in \mathcal{U}_i(A)\}.$$

Stopping condition:
Stop when $\mathcal{U}_i(A) = \mathcal{U}_{i+1}(A)$ for some i. Let $\mathcal{U}(A)$ be the final set of variables constructed.
Normalization step:

- Include in R' all non-unit rules from R.
- For all $A \in V$

  For all $B \in \mathcal{U}(A)$ include in R' the rule $A \to \alpha$ for each non-unit rule $B \to \alpha$ in R.