Algorithm for Eliminating Left Recursion

Input: A grammar G with no cycles and no ε-productions.
Output: An equivalent grammar with no left recursion.

```
arrange the nonterminals in some order A_1, A_2, ..., A_n.
for ( each i from 1 to n ) {
        for ( each j from 1 to i − 1 ) {
                replace each production of the form A_i → A_j γ by the
                    productions A_i → δ_1 γ | δ_2 γ | ··· | δ_k γ, where
                    A_j → δ_1 | δ_2 | ··· | δ_k are all current A_j-productions
        }
        eliminate the immediate left recursion among the A_i-productions
}
```

Figure 4.11: Algorithm to eliminate left recursion from a grammar

Eliminating ϵ-productions

- ϵ-production: $A \to \epsilon$
- nullable: $A \Rightarrow^+ \epsilon$

The set of nullable symbols $\mathcal{N}(G)$ can be calculated as follows:

$$\begin{array}{lll} 1. \ \mathcal{N}_0(G) & = & \{A \in N \mid A \to \epsilon \in P\}; \\ 2. \ \mathcal{N}_{i+1}(G) & = & \mathcal{N}_i(G) \cup \\ & & \{B \in N \mid B \to C_1 \cdots C_k \in P \\ & & \ \text{e} \ \ C_1, \ldots, C_k \in \mathcal{N}_i(G)\}. \end{array}$$

- $\mathcal{N}_i(G) \subseteq \mathcal{N}_{i+1}(G)$;
- there exists i_c such that $\mathcal{N}_{i_c}(G) = \mathcal{N}_{i_c+1}(G)$.

Eliminating ϵ-productions

An Algorithm:

1. Pick a production $X \rightarrow \alpha$ with $\alpha \neq \epsilon$.
2. Calculate $\mathcal{N}(G)$.
3. Let $Z_1, \ldots, Z_k$ be all the occurrences of nullables in α. Add to P all productions obtained from $X \rightarrow \alpha$ by eliminating all possible subsets of $Z_1, \ldots, Z_k$ (including the empty set.
4. If α is composed by nullable non-terminals only, then **no** production is added to P.

Example

$$\begin{array}{rcl} A & \rightarrow & aXbXcXd \\ X & \rightarrow & \epsilon \end{array}$$

becomes

$$A \rightarrow aXbXcXd \mid aXbXcd \mid aXbcXd \mid abXcXd \mid abXcd \mid abcXd \mid abcd$$

Exercise

Apply the algorithm to the grammar

$$S \quad \rightarrow \quad aSbS \mid bSaS \mid \epsilon$$

Single Productions

- A single production is $A \to B$, with A and B non-terminals.
- A single pair is a pair of non-terminals A and B such that $A \Rightarrow^* B$ with only single productions.

Computing the set $\mathcal{U}(G)$ of single pairs:

$$\begin{array}{lrcl} 1. & \mathcal{U}_0(G) & = & \{(A, A) \mid A \in N\}; \\ 2. & \mathcal{U}_{i+1}(G) & = & \mathcal{U}_i(G) \cup \\ & & & \{(A, C) \mid B \to C \in P \\ & & & \text{with } C \in N \text{ and } (A, B) \in \mathcal{U}_i(G)\}. \end{array}$$

Eliminating all single pairs from a grammar corresponds to eliminating cycles.

Eliminating Cycles

Given a grammar $G = (T, N, P, S)$, construct $G' = (T, N, P', S)$ where P' contains all non-single productions of P, $B \rightarrow \alpha$, for all $(A, B) \in \mathcal{U}(G)$.

Exercise: apply the above method to the following grammar:

$$
\begin{array}{rcl}
S & \rightarrow & A \mid SaB \\
A & \rightarrow & B \mid AbB \\
B & \rightarrow & C \mid BcC \\
C & \rightarrow & d \mid e \mid f
\end{array}
$$