Context-Free Grammars

Context-Free Grammars

Example: We would like to describe a language of arithmetic expressions, containing expressions such as:

`175` `(9+15)` `(((10-4)*((1+34)+2))/(3+(-37)))`

For simplicity we assume that:

- Expressions are fully parenthesized.
- The only arithmetic operations are "+", "-", "*", "/" and unary "-".
- Values of operands are natural numbers written in decimal — a number is represented as a non-empty sequence of digits.

Alphabet: $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, (,)\}$

Context-Free Grammars

Example (cont.): A description by an inductive definition:

- **Digit** is any of characters 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- **Number** is a non-empty sequence of digits, i.e.:
  - If α is a digit then α is a number.
  - If α is a digit and β is a number then also $\alpha\beta$ is a number.
- **Expression** is a sequence of symbols constructed according to the following rules:
  - If α is a number then α is an expression.
  - If α is an expression then also $(-\alpha)$ is an expression.
  - If α and β are expressions then also $(\alpha+\beta)$ is an expression.
  - If α and β are expressions then also $(\alpha-\beta)$ is an expression.
  - If α and β are expressions then also $(\alpha*\beta)$ is an expression.
  - If α and β are expressions then also (α/β) is an expression.

Context-Free Grammars

Example (cont.): The same information that was described by the previous inductive definition can be represented by a **context-free grammar**:

New auxiliary symbols, called **nonterminals**, are introduced:

- D — stands for an arbitrary digit
- C — stands for an arbitrary number
- E — stands for an arbitrary expression

$$
\begin{array}{llll}
D \to 0 & D \to 5 & C \to D & E \to C \\
D \to 1 & D \to 6 & C \to DC & E \to (\texttt{-}E) \\
D \to 2 & D \to 7 & & E \to (E\texttt{+}E) \\
D \to 3 & D \to 8 & & E \to (E\texttt{-}E) \\
D \to 4 & D \to 9 & & E \to (E\texttt{*}E) \\
 & & & E \to (E\texttt{/}E)
\end{array}
$$

Example (cont.): Written in a more succinct way:

$$
\begin{aligned}
D &\rightarrow \mathtt{0} \mid \mathtt{1} \mid \mathtt{2} \mid \mathtt{3} \mid \mathtt{4} \mid \mathtt{5} \mid \mathtt{6} \mid \mathtt{7} \mid \mathtt{8} \mid \mathtt{9} \\
C &\rightarrow D \mid DC \\
E &\rightarrow C \mid \mathtt{(-}E\mathtt{)} \mid \mathtt{(}E\mathtt{+}E\mathtt{)} \mid \mathtt{(}E\mathtt{-}E\mathtt{)} \mid \mathtt{(}E\mathtt{*}E\mathtt{)} \mid \mathtt{(}E\mathtt{/}E\mathtt{)}
\end{aligned}
$$

Example: A language where words are (possibly empty) sequences of expressions described in the previous example, where individual expressions are separated by commas (the alphabet must be extended with symbol "$,$"):

$$
\begin{aligned}
S &\rightarrow T \mid \varepsilon \\
T &\rightarrow E \mid E\,,T \\
D &\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\
C &\rightarrow D \mid DC \\
E &\rightarrow C \mid (-E) \mid (E+E) \mid (E-E) \mid (E*E) \mid (E/E)
\end{aligned}
$$

Context-Free Grammars

Example: Statements of some programming language (a fragment of a grammar):

$$
\begin{aligned}
S \;&\rightarrow E\texttt{;} \mid T \mid \texttt{if (}E\texttt{)}\ S \mid \texttt{if (}E\texttt{)}\ S\ \texttt{else}\ S \\
&\mid \texttt{while (}E\texttt{)}\ S \mid \texttt{do}\ S\ \texttt{while (}E\texttt{);} \mid \texttt{for (}F\texttt{;}F\texttt{;}F\texttt{)}\ S \\
&\mid \texttt{return}\ F\texttt{;} \\
T \;&\rightarrow \{\ U\ \} \\
U \;&\rightarrow \varepsilon \mid SU \\
F \;&\rightarrow \varepsilon \mid E \\
E \;&\rightarrow \qquad \ldots
\end{aligned}
$$

Remark:

- S — statement
- T — block of statements
- U — sequence of statements
- E — expression
- F — optional expression that can be omitted

Formally, a **context-free grammar** is a tuple

$$\mathcal{G} = (\Pi, \Sigma, S, P)$$

where:

- Π is a finite set of **nonterminal symbols** (**nonterminals**)
- Σ is a finite set of **terminal symbols** (**terminals**), where $\Pi \cap \Sigma = \emptyset$
- $S \in \Pi$ is an **initial nonterminal**
- $P \subseteq \Pi \times (\Pi \cup \Sigma)^*$ is a finite set of **rewrite rules**

Context-Free Grammars

Remarks:

- We will use uppercase letters A, B, C, ... to denote nonterminal symbols.
- We will use lowercase letters a, b, c, ... or digits 0, 1, 2, ... to denote terminal symbols.
- We will use lowercase Greek letters α, β, γ, ... do denote strings from $(\Pi \cup \Sigma)^*$.
- We will use the following notation for rules instead of (A, α)

$$A \to \alpha$$

A – left-hand side of the rule
α – right-hand side of the rule

Context-Free Grammars

Example: Grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ where

- $\Pi = \{A, B, C\}$
- $\Sigma = \{a, b\}$
- $S = A$
- P contains rules

$$
\begin{aligned}
A &\rightarrow aBBb \\
A &\rightarrow AaA \\
B &\rightarrow \varepsilon \\
B &\rightarrow bCA \\
C &\rightarrow AB \\
C &\rightarrow a \\
C &\rightarrow b
\end{aligned}
$$

Context-Free Grammars

Remark: If we have more rules with the same left-hand side, as for example

$$A \rightarrow \alpha_1 \qquad A \rightarrow \alpha_2 \qquad A \rightarrow \alpha_3$$

we can write them in a more succinct way as

$$A \rightarrow \alpha_1 \mid \alpha_2 \mid \alpha_3$$

For example, the rules of the grammar from the previous slide can be written as

$$\begin{aligned} A &\rightarrow aBBb \mid AaA \\ B &\rightarrow \varepsilon \mid bCA \\ C &\rightarrow AB \mid a \mid b \end{aligned}$$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\begin{aligned} A &\rightarrow aBBb \mid AaA \\ B &\rightarrow \varepsilon \mid bCA \\ C &\rightarrow AB \mid a \mid b \end{aligned}$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

A

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\underline{A} \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

For example, the word *abbabb* can be in grammar $\mathcal{G}$ generated as follows:

$\underline{A}$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\begin{aligned} \underline{A} &\rightarrow \underline{aBBb} \mid AaA \\ B &\rightarrow \varepsilon \mid bCA \\ C &\rightarrow AB \mid a \mid b \end{aligned}$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$\underline{A} \Rightarrow \underline{aBBb}$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \rightarrow aBBb \mid AaA$$
$$\underline{B} \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow a\underline{B}Bb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \rightarrow aBBb \mid AaA$$
$$\underline{B} \rightarrow \varepsilon \mid \underline{bCA}$$
$$C \rightarrow AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow a\underline{B}Bb \Rightarrow a\underline{bCA}Bb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\begin{aligned} A &\rightarrow aBBb \mid AaA \\ B &\rightarrow \varepsilon \mid bCA \\ C &\rightarrow AB \mid a \mid b \end{aligned}$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\underline{A} \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\underline{A} \rightarrow \underline{aBBb} \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb \Rightarrow abC\underline{aBBb}Bb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \to aBBb \mid AaA$$
$$\underline{B} \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\begin{aligned} A &\rightarrow aBBb \mid AaA \\ \underline{B} &\rightarrow \underline{\varepsilon} \mid bCA \\ C &\rightarrow AB \mid a \mid b \end{aligned}$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb \Rightarrow abCaBbBb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\begin{aligned} A &\rightarrow aBBb \mid AaA \\ B &\rightarrow \varepsilon \mid bCA \\ \underline{C} &\rightarrow AB \mid a \mid b \end{aligned}$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:
$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$\underline{C} \to AB \mid a \mid \underline{b}$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb \Rightarrow ab\underline{b}aBbBb$$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\begin{aligned} A &\rightarrow aBBb \mid AaA \\ \underline{B} &\rightarrow \varepsilon \mid bCA \\ C &\rightarrow AB \mid a \mid b \end{aligned}$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \rightarrow aBBb \mid AaA$$
$$\underline{B} \rightarrow \underline{\varepsilon} \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b \Rightarrow abbaBbb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$
\begin{aligned}
A &\to aBBb \mid AaA \\
B &\to \varepsilon \mid bCA \\
C &\to AB \mid a \mid b
\end{aligned}
$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$
\begin{aligned}
A &\rightarrow aBBb \mid AaA \\
\underline{B} &\rightarrow \varepsilon \mid bCA \\
C &\rightarrow AB \mid a \mid b
\end{aligned}
$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abba\underline{B}bb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$\begin{aligned} A &\rightarrow aBBb \mid AaA \\ \underline{B} &\rightarrow \underline{\varepsilon} \mid bCA \\ C &\rightarrow AB \mid a \mid b \end{aligned}$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abba\underline{B}bb \Rightarrow abbabb$

Context-Free Grammars

Grammars are used for generating words.

Example: $\mathcal{G} = (\Pi, \Sigma, A, P)$ where $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$, and P contains rules

$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

For example, the word $abbabb$ can be in grammar $\mathcal{G}$ generated as follows:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb \Rightarrow abbabb$

Context-Free Grammars

On strings from $(\Pi \cup \Sigma)^*$ we define relation $\Rightarrow \subseteq (\Pi \cup \Sigma)^* \times (\Pi \cup \Sigma)^*$ such that

$$\alpha \Rightarrow \alpha'$$

iff $\alpha = \beta_1 A \beta_2$ and $\alpha' = \beta_1 \gamma \beta_2$ for some $\beta_1, \beta_2, \gamma \in (\Pi \cup \Sigma)^*$ and $A \in \Pi$ where $(A \rightarrow \gamma) \in P$.

Example: If $(B \rightarrow bCA) \in P$ then

$$aCBbA \Rightarrow aC\,bCA\,bA$$

Remark: Informally, $\alpha \Rightarrow \alpha'$ means that it is possible to derive α' from α by one step where an occurrence of some nonterminal A in α is replaced with the right-hand side of some rule $A \rightarrow \gamma$ with A on the left-hand side.

Context-Free Grammars

On strings from $(\Pi \cup \Sigma)^*$ we define relation $\Rightarrow \subseteq (\Pi \cup \Sigma)^* \times (\Pi \cup \Sigma)^*$ such that

$$\alpha \Rightarrow \alpha'$$

iff $\alpha = \beta_1 A \beta_2$ and $\alpha' = \beta_1 \gamma \beta_2$ for some $\beta_1, \beta_2, \gamma \in (\Pi \cup \Sigma)^*$ and $A \in \Pi$ where $(A \rightarrow \gamma) \in P$.

Example: If $(B \rightarrow bCA) \in P$ then

$$aC\underline{B}bA \Rightarrow aC\underline{bCA}bA$$

Remark: Informally, $\alpha \Rightarrow \alpha'$ means that it is possible to derive α' from α by one step where an occurrence of some nonterminal A in α is replaced with the right-hand side of some rule $A \rightarrow \gamma$ with A on the left-hand side.

A **derivation** of length n is a sequence $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$, where $\beta_i \in (\Pi \cup \Sigma)^*$, and where $\beta_{i-1} \Rightarrow \beta_i$ for all $1 \leq i \leq n$, which can be written more succinctly as

$$\beta_0 \Rightarrow \beta_1 \Rightarrow \beta_2 \Rightarrow \ldots \Rightarrow \beta_{n-1} \Rightarrow \beta_n$$

The fact that for given $\alpha, \alpha' \in (\Pi \cup \Sigma)^*$ and $n \in \mathbb{N}$ there exists some derivation $\beta_0 \Rightarrow \beta_1 \Rightarrow \beta_2 \Rightarrow \ldots \Rightarrow \beta_{n-1} \Rightarrow \beta_n$, where $\alpha = \beta_0$ and $\alpha' = \beta_n$, is denoted

$$\alpha \Rightarrow^n \alpha'$$

The fact that $\alpha \Rightarrow^n \alpha'$ for some $n \geq 0$, is denoted

$$\alpha \Rightarrow^* \alpha'$$

Remark: Relation $\Rightarrow^*$ is the reflexive and transitive closure of relation $\Rightarrow$ (i.e., the smallest reflexive and transitive relation containing relation $\Rightarrow$).

Sentential forms are those $\alpha \in (\Pi \cup \Sigma)^*$, for which

$$S \Rightarrow^* \alpha$$

where S is the initial nonterminal.

Context-Free Grammars

A **language** $\mathcal{L}(\mathcal{G})$ generated by a grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ is the set of all words over alphabet Σ that can be derived by some derivation from the initial nonterminal S using rules from P, i.e.,

$$\mathcal{L}(\mathcal{G}) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$$

Definition

A language L is **context-free** if there exists some context-free grammar $\mathcal{G}$ such that $L = \mathcal{L}(\mathcal{G})$.

Example: We want to construct a grammar generating the language

$$L = \{a^n b^n \mid n \geq 0\}$$

Context-Free Grammars

Example: We want to construct a grammar generating the language

$$L = \{a^n b^n \mid n \geq 0\}$$

Grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ where $\Pi = \{S\}$, $\Sigma = \{a, b\}$, and P contains

$$S \to \varepsilon \mid aSb$$

Context-Free Grammars

Example: We want to construct a grammar generating the language

$$L = \{a^n b^n \mid n \geq 0\}$$

Grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ where $\Pi = \{S\}$, $\Sigma = \{a, b\}$, and P contains

$$S \rightarrow \varepsilon \mid aSb$$

$$
\begin{aligned}
&S \Rightarrow \varepsilon \\
&S \Rightarrow aSb \Rightarrow ab \\
&S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb \\
&S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb \\
&S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaaSbbbb \Rightarrow aaaabbbb \\
&\qquad \ldots
\end{aligned}
$$

Context-Free Grammars

Example: We want to construct a grammar generating the language consisting of all palindroms over the alphabet $\{a, b\}$, i.e.,

$$L = \{w \in \{a, b\}^* \mid w = w^R\}$$

Remark: w^R denotes the **reverse** of a word w, i.e., the word w written backwards.

Example: We want to construct a grammar generating the language consisting of all palindroms over the alphabet $\{a, b\}$, i.e.,

$$L = \{w \in \{a, b\}^* \mid w = w^R\}$$

Remark: w^R denotes the **reverse** of a word w, i.e., the word w written backwards.

Solution:

$$S \to \varepsilon \mid a \mid b \mid aSa \mid bSb$$

Example: We want to construct a grammar generating the language consisting of all palindroms over the alphabet $\{a, b\}$, i.e.,

$$L = \{w \in \{a, b\}^* \mid w = w^R\}$$

Remark: w^R denotes the **reverse** of a word w, i.e., the word w written backwards.

Solution:

$$S \to \varepsilon \mid a \mid b \mid aSa \mid bSb$$

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abaSaba \Rightarrow abaaaba$$

Context-Free Grammars

Example: We want to construct a grammar generating the language L consisting of all correctly parenthesised sequences of symbols '(' and ')'.

For example $(()())(()) \in L$ but $)()) \notin L$.

Example: We want to construct a grammar generating the language L consisting of all correctly parenthesised sequences of symbols '$($' and '$)$'.

For example $(()())(()) \in L$ but $)()) \notin L$.

Solution:

$$S \to \varepsilon \mid (S) \mid SS$$

Context-Free Grammars

Example: We want to construct a grammar generating the language L consisting of all correctly parenthesised sequences of symbols '$($' and '$)$'.

For example $(()())(()) \in L$ but $)()) \notin L$.

Solution:

$$S \rightarrow \varepsilon \mid (S) \mid SS$$

$S \Rightarrow SS \Rightarrow (S)S \Rightarrow (S)(S) \Rightarrow (SS)(S) \Rightarrow ((S)S)(S) \Rightarrow$
$(()S)(S) \Rightarrow (()(S))(S) \Rightarrow (()())(S) \Rightarrow (()())((S)) \Rightarrow$
$(()())(())$

Context-Free Grammars

Example: We want to construct a grammar generating the language L consisting of all correctly constructed arithmetic experessions where operands are always of the form 'a' and where symbols $+$ and $*$ can be used as operators.

For example $(a + a) * a + (a * a) \in L$.

Context-Free Grammars

Example: We want to construct a grammar generating the language L consisting of all correctly constructed arithmetic experessions where operands are always of the form 'a' and where symbols $+$ and $*$ can be used as operators.

For example $(a + a) * a + (a * a) \in L$.

Solution:

$$E \rightarrow a \mid E + E \mid E * E \mid (E)$$

Example: We want to construct a grammar generating the language L consisting of all correctly constructed arithmetic experessions where operands are always of the form 'a' and where symbols $+$ and $*$ can be used as operators.

For example $(a+a)*a+(a*a) \in L$.

Solution:

$$E \rightarrow a \mid E+E \mid E*E \mid (E)$$

$E \Rightarrow E+E \Rightarrow E*E+E \Rightarrow (E)*E+E \Rightarrow (E+E)*E+E \Rightarrow (a+E)*E+E \Rightarrow (a+a)*E+E \Rightarrow (a+a)*a+E \Rightarrow (a+a)*a+(E) \Rightarrow (a+a)*a+(E*E) \Rightarrow (a+a)*a+(a*E) \Rightarrow (a+a)*a+(a*a)$

Derivation Tree

$$
\begin{aligned}
A &\rightarrow aBBb \mid AaA \\
B &\rightarrow \varepsilon \mid bCA \\
C &\rightarrow AB \mid a \mid b
\end{aligned}
$$

Derivation Tree

A

$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

A

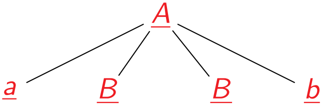
Derivation Tree

$\underline{A}$

$$\underline{A} \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

$\underline{A}$

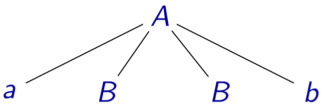
Derivation Tree



$A \rightarrow aBBb \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$

$A \Rightarrow aBBb$

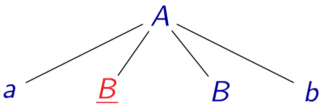
Derivation Tree



$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

$$A \Rightarrow aBBb$$

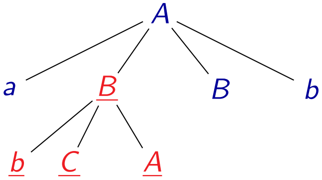
Derivation Tree



$A \rightarrow aBBb \mid AaA$
$\underline{B} \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$

$A \Rightarrow a\underline{B}Bb$

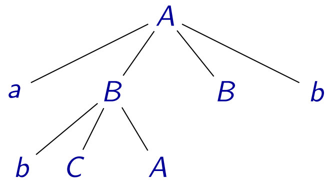
Derivation Tree



$$A \rightarrow aBBb \mid AaA$$
$$\underline{B} \rightarrow \varepsilon \mid \underline{bCA}$$
$$C \rightarrow AB \mid a \mid b$$

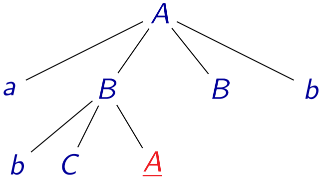
$$A \Rightarrow a\underline{B}Bb \Rightarrow a\underline{bCA}Bb$$

Derivation Tree



$A \rightarrow aBBb \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$

$A \Rightarrow aBBb \Rightarrow abCABb$

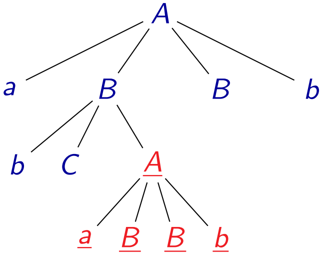


$\underline{A} \rightarrow aBBb \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$

$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb$

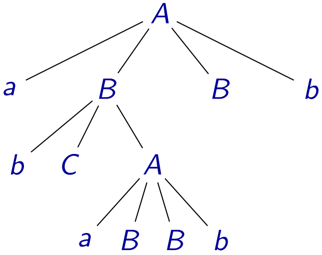
Derivation Tree

$\underline{A} \rightarrow \underline{aBBb} \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb \Rightarrow abC\underline{aBBb}Bb$

Derivation Tree

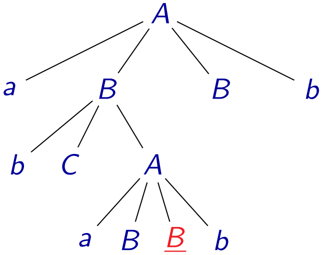


$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$

$$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb$$

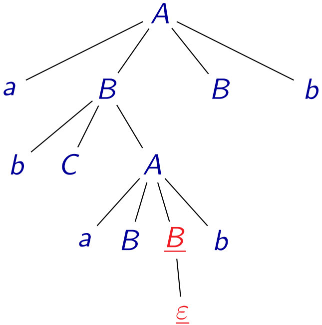
Derivation Tree

$A \rightarrow aBBb \mid AaA$
$\underline{B} \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$



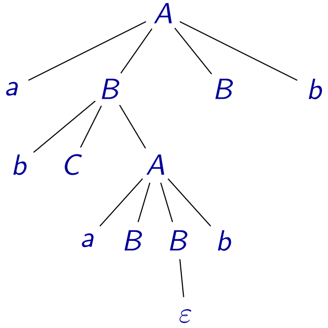
$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb$

$A \to aBBb \mid AaA$
$\underline{B} \to \underline{\varepsilon} \mid bCA$
$C \to AB \mid a \mid b$



$$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb \Rightarrow abCaBbBb$$

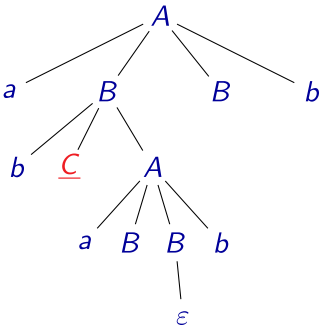
$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$



$$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb$$

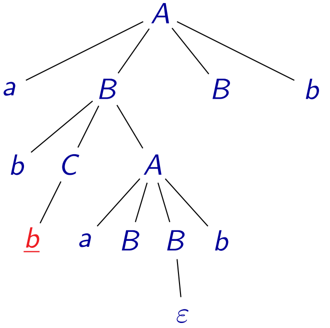
Derivation Tree

$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$\underline{C} \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb$

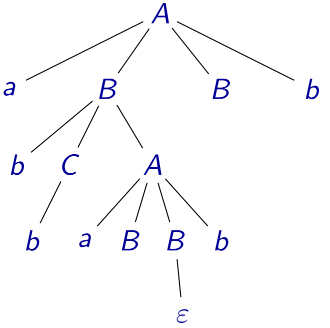
$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$\underline{C} \to AB \mid a \mid \underline{b}$$



$$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb \Rightarrow ab\underline{b}aBbBb$$

Derivation Tree

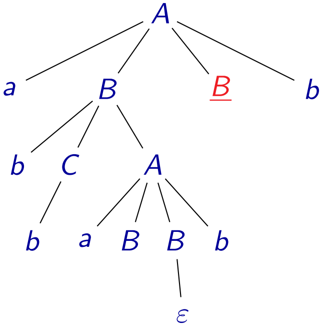
$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$



$$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb$$

Derivation Tree

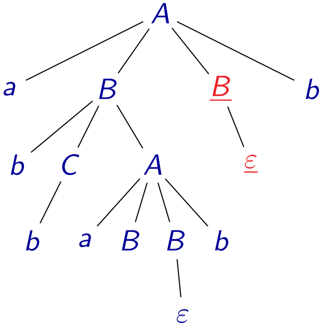
$$A \rightarrow aBBb \mid AaA$$
$$\underline{B} \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$



$$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b$$

Derivation Tree

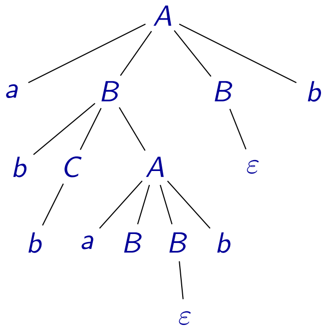
$A \rightarrow aBBb \mid AaA$
$\underline{B} \rightarrow \underline{\varepsilon} \mid bCA$
$C \rightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b \Rightarrow abbaBbb$

Derivation Tree

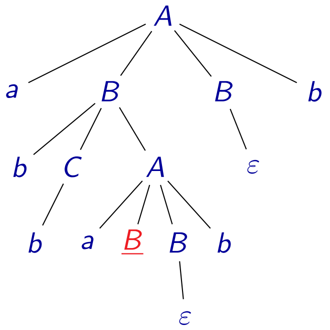
$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$



$$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb$$

Derivation Tree

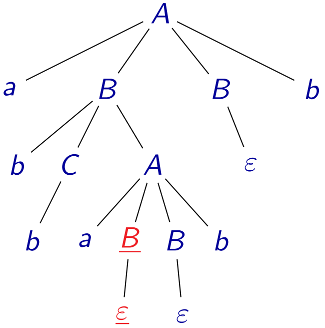
$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abba\underline{B}bb$

Derivation Tree

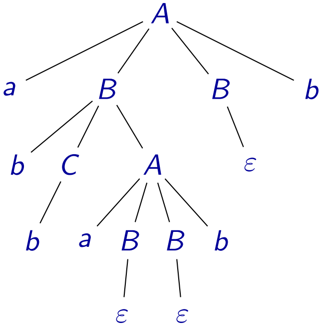
$A \rightarrow aBBb \mid AaA$
$\underline{B} \rightarrow \underline{\varepsilon} \mid bCA$
$C \rightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abba\underline{B}bb \Rightarrow abbabb$

Derivation Tree

$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb \Rightarrow abbabb$

Derivation Tree

For each derivation there is some **derivation tree**:

- Nodes of the tree are labelled with terminals and nonterminals.
- The root of the tree is labelled with the initial nonterminal.
- The leafs of the tree are labelled with terminals or with symbols ε.
- The remaining nodes of the tree are labelled with nonterminals.
- If a node is labelled with some nonterminal A then its children are labelled with the symbols from the right-hand side of some rewriting rule $A \rightarrow \alpha$.

Left and Right Derivation

$$E \to a \mid E + E \mid E * E \mid (E)$$

A **left derivation** is a derivation where in every step we always replace the leftmost nonterminal.

$$\underline{E} \Rightarrow \underline{E} + E \Rightarrow \underline{E} * E + E \Rightarrow a * \underline{E} + E \Rightarrow a * a + \underline{E} \Rightarrow a * a + a$$

A **right derivation** is a derivation where in every step we always replace the rightmost nonterminal.

$$\underline{E} \Rightarrow E + \underline{E} \Rightarrow \underline{E} + a \Rightarrow E * \underline{E} + a \Rightarrow \underline{E} * a + a \Rightarrow a * a + a$$

A derivation need not be left or right:

$$\underline{E} \Rightarrow \underline{E} + E \Rightarrow E * \underline{E} + E \Rightarrow E * a + \underline{E} \Rightarrow \underline{E} * a + a \Rightarrow a * a + a$$

Left and Right Derivation

- There can be several different derivations corresponding to one derivation tree.
- For every derivation tree, there is exactly one left and exactly one right derivation corresponding to the tree.

Equvalence of Grammars

Grammars $\mathcal{G}_1$ and $\mathcal{G}_2$ are **equivalent** if they generate the same language, i.e., if $\mathcal{L}(\mathcal{G}_1) = \mathcal{L}(\mathcal{G}_2)$.

Remark: The problem of equivalence of context-free grammars is algorithmically undecidable. It can be shown that it is not possible to construct an algorithm that would decide for any pair of context-free grammars if they are equivalent or not.

Even the problem to decide if a grammar generates the language Σ^* is algorithmically undecidable.

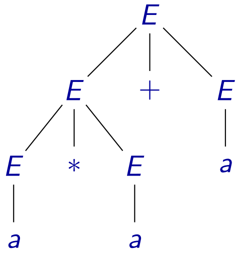
Ambiguous Grammars

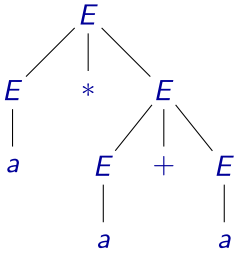
A grammar $\mathcal{G}$ is **ambiguous** if there is a word $w \in \mathcal{L}(\mathcal{G})$ that has two different derivation trees, resp. two different left or two different right derivations.

Example:

$E \Rightarrow E + E \Rightarrow E * E + E \Rightarrow a * E + E \Rightarrow a * a + E \Rightarrow a * a + a$

$E \Rightarrow E * E \Rightarrow E * E + E \Rightarrow a * E + E \Rightarrow a * a + E \Rightarrow a * a + a$





Ambiguous Grammars

Sometimes it is possible to replace an ambiguous grammar with a grammar generating the same language but which is not ambiguous.

Example: A grammar

$$E \to E + E \mid E * E \mid (E) \mid a$$

can be replaced with the equivalent grammar

$$\begin{aligned} E &\to T \mid T + E \\ T &\to F \mid F * T \\ F &\to a \mid (E) \end{aligned}$$

Remark: If there is no unambiguous grammar equivalent to a given ambiguous grammar, we say it is **inherently ambiguous**.

Context-Free Languages

The class of context-free languages is closed with respect to:

- concatenation
- union
- iteration

The class of context-free languages is not closed with respect to:

- complement
- intersection

Context-Free Languages

We have two grammars $\mathcal{G}_1 = (\Pi_1, \Sigma, S_1, P_1)$ and $\mathcal{G}_2 = (\Pi_2, \Sigma, S_2, P_2)$, and can assume that $\Pi_1 \cap \Pi_2 = \emptyset$ and $S \notin \Pi_1 \cup \Pi_2$.

- Grammar $\mathcal{G}$ such that $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{G}_1) \cdot \mathcal{L}(\mathcal{G}_2)$:

$$\mathcal{G} = (\Pi_1 \cup \Pi_2 \cup \{S\},\ \Sigma,\ S,\ P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\})$$

- Grammar $\mathcal{G}$ such that $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{G}_1) \cup \mathcal{L}(\mathcal{G}_2)$:

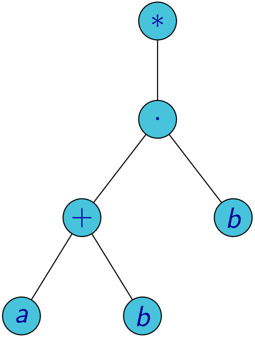
$$\mathcal{G} = (\Pi_1 \cup \Pi_2 \cup \{S\},\ \Sigma,\ S,\ P_1 \cup P_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\})$$

- Grammar $\mathcal{G}$ such that $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{G}_1)^*$:

$$\mathcal{G} = (\Pi_1 \cup \{S\},\ \Sigma,\ S,\ P_1 \cup \{S \rightarrow \varepsilon, S \rightarrow S_1 S\})$$

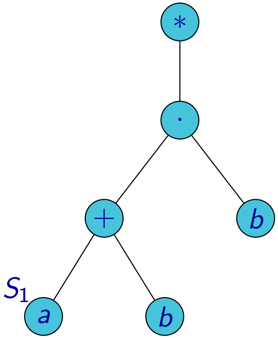
A Context-Free Grammar for a Regular Expression

Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



A Context-Free Grammar for a Regular Expression

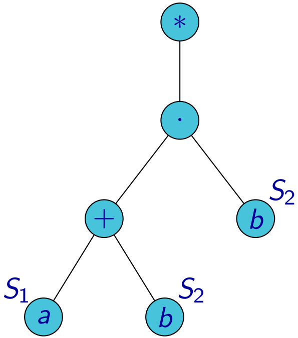
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



$S_1 \to a$

A Context-Free Grammar for a Regular Expression

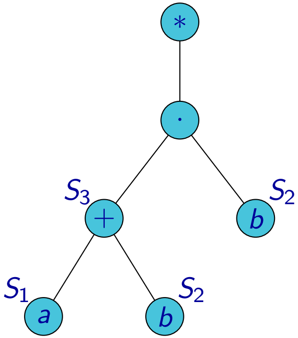
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



$S_2 \rightarrow b$
$S_1 \rightarrow a$

A Context-Free Grammar for a Regular Expression

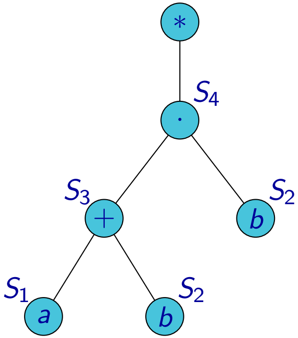
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



$$
\begin{aligned}
S_3 &\rightarrow S_1 \mid S_2 \\
S_2 &\rightarrow b \\
S_1 &\rightarrow a
\end{aligned}
$$

A Context-Free Grammar for a Regular Expression

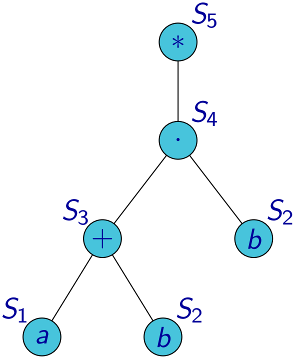
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



$$S_4 \to S_3 S_2$$
$$S_3 \to S_1 \mid S_2$$
$$S_2 \to b$$
$$S_1 \to a$$

A Context-Free Grammar for a Regular Expression

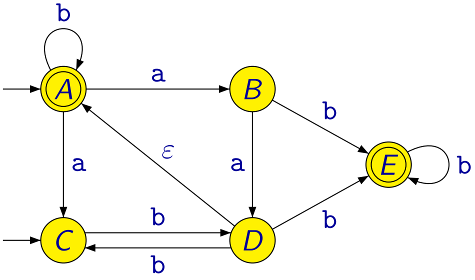
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



$$
\begin{aligned}
S_5 &\rightarrow \varepsilon \mid S_4 S_5 \\
S_4 &\rightarrow S_3 S_2 \\
S_3 &\rightarrow S_1 \mid S_2 \\
S_2 &\rightarrow b \\
S_1 &\rightarrow a
\end{aligned}
$$

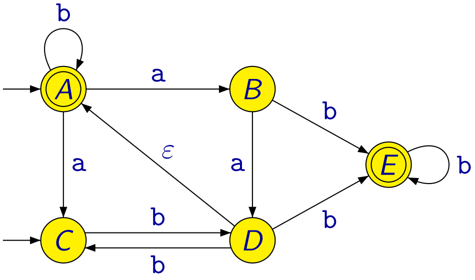
A Context-Free Grammar for a Finite Automaton

Example:



A Context-Free Grammar for a Finite Automaton

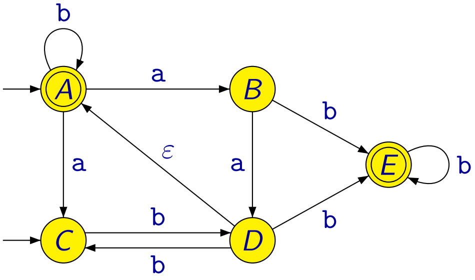
Example:



$S \to A \mid C$

A Context-Free Grammar for a Finite Automaton

Example:

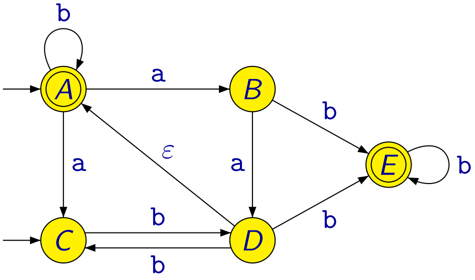


$$S \to A \mid C$$

$$
\begin{aligned}
A &\to aB \mid aC \mid bA \\
B &\to aD \mid bE \\
C &\to bD \\
D &\to bC \mid bE \mid A \\
E &\to bE
\end{aligned}
$$

A Context-Free Grammar for a Finite Automaton

Example:



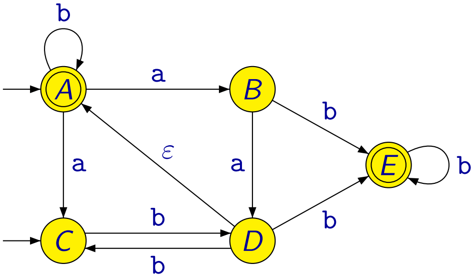
$$S \to A \mid C$$

$$A \to aB \mid aC \mid bA$$
$$B \to aD \mid bE$$
$$C \to bD$$
$$D \to bC \mid bE \mid A$$
$$E \to bE$$

$$A \to \varepsilon$$
$$E \to \varepsilon$$

A Context-Free Grammar for a Finite Automaton

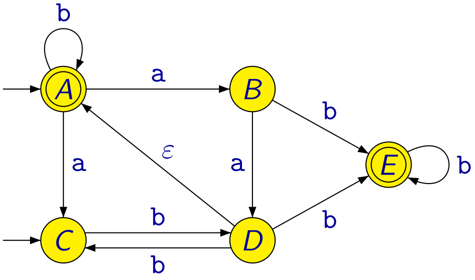
Example:



Alternative construction:

A Context-Free Grammar for a Finite Automaton

Example:

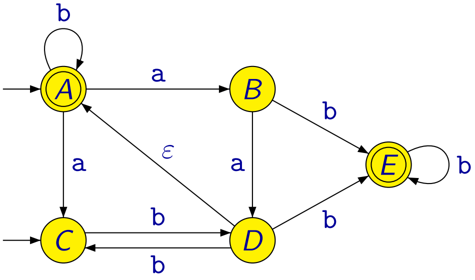


Alternative construction:

$$S \rightarrow A \mid E$$

A Context-Free Grammar for a Finite Automaton

Example:

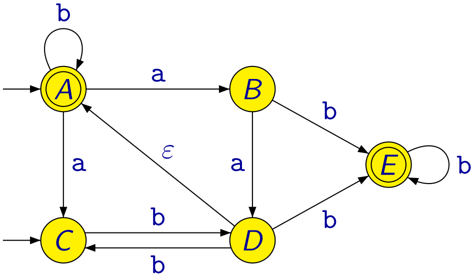


Alternative construction:

$$S \rightarrow A \mid E$$

$$A \rightarrow Ab \mid D$$
$$B \rightarrow Aa$$
$$C \rightarrow Aa \mid Db$$
$$D \rightarrow Ba \mid Cb$$
$$E \rightarrow Bb \mid Db \mid Eb$$

Example:



Alternative construction:

$$S \rightarrow A \mid E$$

$$A \rightarrow Ab \mid D$$
$$B \rightarrow Aa$$
$$C \rightarrow Aa \mid Db$$
$$D \rightarrow Ba \mid Cb$$
$$E \rightarrow Bb \mid Db \mid Eb$$

$$A \rightarrow \varepsilon$$
$$C \rightarrow \varepsilon$$

Regular grammars

Definition

A grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ is **right regular** if all rules in P are of the following forms (where $A, B \in \Pi$, $a \in \Sigma$):

- $A \rightarrow B$
- $A \rightarrow aB$
- $A \rightarrow \varepsilon$

Definition

A grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ is **left regular** if all rules in P are of the following forms (kde $A, B \in \Pi$, $a \in \Sigma$):

- $A \rightarrow B$
- $A \rightarrow Ba$
- $A \rightarrow \varepsilon$

Regular grammars

Definition

A grammar $\mathcal{G}$ is **regular** if it right regular or left regular.

Remark: Sometimes a slightly more general definition of right (resp. left) regular grammars is given, allowing all rules of the following forms:

- $A \rightarrow wB$ (resp. $A \rightarrow Bw$)
- $A \rightarrow w$

where $A, B \in \Pi$, $w \in \Sigma^*$.

Such rules can be easily “decomposed” into rules of the form in the previous definition.

Example: Rule $A \rightarrow abbB$ can be replaced with rules

$$A \rightarrow aZ_1 \qquad Z_1 \rightarrow bZ_2 \qquad Z_2 \rightarrow bB$$

where Z_1, Z_2 are new nonterminals, not used anywhere else in the grammar.

Regular grammars

Proposition

For every regular language L there is a left regular grammar $\mathcal{G}$ such that $\mathcal{L}(\mathcal{G}) = L$ and a right regular grammar $\mathcal{G}'$ such that $\mathcal{L}(\mathcal{G}') = L$.

Proposition

For every regular grammar $\mathcal{G}$ there is a finite automaton $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{G})$.

Reduction of a Context-Free Grammar

Definition

A context-free grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ is **reduced** if for every $A \in \Pi$:

- there are some $u, v \in \Sigma^*$ such that $S \Rightarrow^* uAv$, and
- there is some $w \in \Sigma^*$ such that $A \Rightarrow^* w$.

Remark: Obviously, if $S \Rightarrow^* uAv$ and $A \Rightarrow^* w$ where $u, v, w \in \Sigma^*$, then $S \Rightarrow^* uwv$, and so A is used in some derivation of a word from Σ^*.

On the other hand, if A is used in some derivation $S \Rightarrow^* z$ of a word $z \in \Sigma^*$, then z can be divided into parts u, v, w such that $z = uwv$ and $S \Rightarrow^* uAv$ and $A \Rightarrow^* w$.

Reduction of a Context-Free Grammar

Obviously, every $A \in \Pi$ with the property that

- there are no $u, v \in \Sigma^*$ such that $S \Rightarrow^* uAv$, or
- there is no $w \in \Sigma^*$ such that $A \Rightarrow^* w$,

can be safely removed from the grammar (together with all rules where it occurs) without affecting the generated language.

Reduction of a Context-Free Grammar

An algorithm that for a given CFG $\mathcal{G}$ contructs an equivalent reduced grammar:

1. Construct the set $\mathcal{T}$ of all nonterminals that can generate a terminal word:
$$\mathcal{T} = \{ A \in \Pi \mid (\exists w \in \Sigma^*)(A \Rightarrow^* w) \}$$
2. Remove from $\mathcal{G}$ all nonterminals from the set $\Pi - \mathcal{T}$ together with all rules where they occur.
Denote the rusulting grammar $\mathcal{G}' = (\Pi', \Sigma, S, P')$.
3. Construct the set $\mathcal{D}$ of all nonterminals that can be "reached" from the initial nonterminal S:
$$\mathcal{D} = \{ A \in \Pi' \mid (\exists \alpha, \beta \in (\Pi' \cup \Sigma)^*)(S \Rightarrow^* \alpha A \beta) \}$$
4. Remove from $\mathcal{G}'$ all nonterminals from the set $\Pi' - \mathcal{D}$ together with all rules where they occur.
The rusulting grammar $\mathcal{G}''$ is the result of the whole algorithm.

Example:

$$
\begin{aligned}
S &\rightarrow AC \mid B \\
A &\rightarrow aC \mid AbA \\
B &\rightarrow Ba \mid BbA \mid DB \\
C &\rightarrow aa \mid aBC \\
D &\rightarrow aA \mid \varepsilon
\end{aligned}
$$

Reduction of a Context-Free Grammar

Example:

$$\mathcal{T}_0 = \{C, D\}$$

$$\begin{aligned}
S &\to AC \mid B \\
A &\to aC \mid AbA \\
B &\to Ba \mid BbA \mid DB \\
C &\to aa \mid aBC \\
D &\to aA \mid \varepsilon
\end{aligned}$$

Reduction of a Context-Free Grammar

Example:

$$\mathcal{T}_0 = \{C, D\}$$
$$\mathcal{T}_1 = \{C, D, A\}$$

$$S \rightarrow AC \mid B$$
$$A \rightarrow aC \mid AbA$$
$$B \rightarrow Ba \mid BbA \mid DB$$
$$C \rightarrow aa \mid aBC$$
$$D \rightarrow aA \mid \varepsilon$$

Reduction of a Context-Free Grammar

Example:

$$\mathcal{T}_0 = \{C, D\}$$
$$\mathcal{T}_1 = \{C, D, A\}$$
$$\mathcal{T}_2 = \{C, D, A, S\}$$

$$S \to AC \mid B$$
$$A \to aC \mid AbA$$
$$B \to Ba \mid BbA \mid DB$$
$$C \to aa \mid aBC$$
$$D \to aA \mid \varepsilon$$

Reduction of a Context-Free Grammar

Example:

$$\mathcal{T}_0 = \{C, D\}$$
$$\mathcal{T}_1 = \{C, D, A\}$$
$$\mathcal{T}_2 = \{C, D, A, S\}$$

$$S \rightarrow AC \mid B$$
$$A \rightarrow aC \mid AbA$$
$$B \rightarrow Ba \mid BbA \mid DB$$
$$C \rightarrow aa \mid aBC$$
$$D \rightarrow aA \mid \varepsilon$$

$$\mathcal{T} = \{C, D, A, S\}$$

Example:

$$\mathcal{T}_0 = \{C, D\}$$
$$\mathcal{T}_1 = \{C, D, A\}$$
$$\mathcal{T}_2 = \{C, D, A, S\}$$

$$\begin{aligned} S &\to AC \mid B \\ A &\to aC \mid AbA \\ B &\to Ba \mid BbA \mid DB \\ C &\to aa \mid aBC \\ D &\to aA \mid \varepsilon \end{aligned}$$

$$\mathcal{T} = \{C, D, A, S\}$$

$$\begin{aligned} S &\to AC \\ A &\to aC \mid AbA \\ C &\to aa \\ D &\to aA \mid \varepsilon \end{aligned}$$

Reduction of a Context-Free Grammar

Example:

$$\mathcal{T}_0 = \{C, D\}$$
$$\mathcal{T}_1 = \{C, D, A\}$$
$$\mathcal{T}_2 = \{C, D, A, S\}$$

$$\mathcal{D}_0 = \{S\}$$

$$\begin{aligned} S &\to AC \mid B \\ A &\to aC \mid AbA \\ B &\to Ba \mid BbA \mid DB \\ C &\to aa \mid aBC \\ D &\to aA \mid \varepsilon \end{aligned}$$

$$\mathcal{T} = \{C, D, A, S\}$$

$$\begin{aligned} S &\to AC \\ A &\to aC \mid AbA \\ C &\to aa \\ D &\to aA \mid \varepsilon \end{aligned}$$

Reduction of a Context-Free Grammar

Example:

$$\mathcal{T}_0 = \{C, D\}$$
$$\mathcal{T}_1 = \{C, D, A\}$$
$$\mathcal{T}_2 = \{C, D, A, S\}$$

$$\mathcal{D}_0 = \{S\}$$
$$\mathcal{D}_1 = \{S, A, C\}$$

$$\begin{aligned} S &\to AC \mid B \\ A &\to aC \mid AbA \\ B &\to Ba \mid BbA \mid DB \\ C &\to aa \mid aBC \\ D &\to aA \mid \varepsilon \end{aligned}$$

$$\mathcal{T} = \{C, D, A, S\}$$

$$\begin{aligned} S &\to AC \\ A &\to aC \mid AbA \\ C &\to aa \\ D &\to aA \mid \varepsilon \end{aligned}$$

Example:

$$\mathcal{T}_0 = \{C, D\}$$
$$\mathcal{T}_1 = \{C, D, A\}$$
$$\mathcal{T}_2 = \{C, D, A, S\}$$

$$\mathcal{D}_0 = \{S\}$$
$$\mathcal{D}_1 = \{S, A, C\}$$

$$\begin{aligned} S &\to AC \mid B \\ A &\to aC \mid AbA \\ B &\to Ba \mid BbA \mid DB \\ C &\to aa \mid aBC \\ D &\to aA \mid \varepsilon \end{aligned}$$

$$\mathcal{T} = \{C, D, A, S\} \qquad \mathcal{D} = \{S, A, C\}$$

$$\begin{aligned} S &\to AC \\ A &\to aC \mid AbA \\ C &\to aa \\ D &\to aA \mid \varepsilon \end{aligned}$$

Reduction of a Context-Free Grammar

Example:

$$
\begin{aligned}
S &\rightarrow AC \mid B \\
A &\rightarrow aC \mid AbA \\
B &\rightarrow Ba \mid BbA \mid DB \\
C &\rightarrow aa \mid aBC \\
D &\rightarrow aA \mid \varepsilon
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{T}_0 &= \{C, D\} \\
\mathcal{T}_1 &= \{C, D, A\} \\
\mathcal{T}_2 &= \{C, D, A, S\}
\end{aligned}
$$

$$\mathcal{T} = \{C, D, A, S\}$$

$$
\begin{aligned}
S &\rightarrow AC \\
A &\rightarrow aC \mid AbA \\
C &\rightarrow aa \\
D &\rightarrow aA \mid \varepsilon
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{D}_0 &= \{S\} \\
\mathcal{D}_1 &= \{S, A, C\}
\end{aligned}
$$

$$\mathcal{D} = \{S, A, C\}$$

$$
\begin{aligned}
S &\rightarrow AC \\
A &\rightarrow aC \mid AbA \\
C &\rightarrow aa
\end{aligned}
$$

Some Properties of Context-free Grammars

Let us assume we have a context-free grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$.

We can easily construct algorithms for the following problems dealing with some properties of context-free grammar $\mathcal{G}$:

- To find out for given $\alpha \in (\Pi \cup \Sigma)^*$ whether $\alpha \Rightarrow^* \varepsilon$.
- To find, for given $\alpha \in (\Pi \cup \Sigma)^*$, the set $\mathit{first}(\alpha)$, where
$$\mathit{first}(\alpha) = \{\, a \in \Sigma \mid \alpha \Rightarrow^* a\beta \text{ for some } \beta \in (\Pi \cup \Sigma)^* \,\}$$
- To find, for given $\alpha \in (\Pi \cup \Sigma)^*$, the set $\mathit{last}(\alpha)$, where
$$\mathit{last}(\alpha) = \{\, a \in \Sigma \mid \alpha \Rightarrow^* \beta a \text{ for some } \beta \in (\Pi \cup \Sigma)^* \,\}$$

Some Properties of Context-free Grammars

- To find, for given nonterminal $A \in \Pi$, the set $follow(A)$, where
  $follow(A) = \{\, a \in \Sigma \mid S \Rightarrow^* \beta_1 A\, a\, \beta_2 \text{ for some } \beta_1, \beta_2 \in (\Pi \cup \Sigma)^* \,\}$

- To find all nonterminals $A \in \Pi$, for which grammar $\mathcal{G}$ contains the **left recursion**, i.e., those for which
  $$A \Rightarrow^+ A\alpha \qquad \text{for some } \alpha \in (\Pi \cup \Sigma)^*$$

- To find all nonterminals $A \in \Pi$, for which grammar $\mathcal{G}$ contains the **right recursion**, i.e., those for which
  $$A \Rightarrow^+ \alpha A \qquad \text{for some } \alpha \in (\Pi \cup \Sigma)^*$$

Remark: Notation $\alpha \Rightarrow^+ \beta$, where $\alpha, \beta \in (\Pi \cup \Sigma)^*$, denotes that α can be rewritten to β (i.e., $\alpha \Rightarrow^* \beta$) by a derivation with a nonzero number of steps.

Some Properties of Context-free Grammars

To be able to use a given context-free grammar $\mathcal{G}$ for a straightforward implementation of **recursive descent**, it must have some particular properties:

- It must not contain left recursion.
- For each nonterminal $A \in \Pi$ and all rules with A on the left-hand side, i.e.,

  $$A \to \alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_n$$

  the sets $\mathit{first}(\alpha_1)$, $\mathit{first}(\alpha_2)$, ..., $\mathit{first}(\alpha_n)$ must be pairwise disjoint.
- For every nonterminal $A \in \Pi$ and all rules $A \to \alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_n$ there can be at most one right-hand side α_i such that $\alpha_i \Rightarrow^* \varepsilon$.

  If there is such right-hand side (and so $A \Rightarrow^* \varepsilon$), the sets $\mathit{first}(\alpha_1)$, $\mathit{first}(\alpha_2)$, ..., $\mathit{first}(\alpha_n)$ must be disjoint with the set $\mathit{follow}(A)$.

Removing Epsilon-rules

Rules of the form $A \to \varepsilon$ are called **epsilon-rules** (ε-rules).

Proposition

For every context-free grammar $\mathcal{G}$ there is a context-free grammar $\mathcal{G}'$ without ε-rules such that $\mathcal{L}(\mathcal{G}') = \mathcal{L}(\mathcal{G}) - \{\varepsilon\}$.

Proof: Construct the set $\mathcal{E}$ of all nonterminals that can be rewritten to ε, i.e.,

$$\mathcal{E} = \{ A \in \Pi \mid A \Rightarrow^* \varepsilon \}$$

Remove all ε-rules and replace every other rule $A \to \alpha$ with a set of rules obtained by all possible rules of the form $A \to \alpha'$ where α' is obtained from α by possible ommitting of (some) occurrences of nonterminals from $\mathcal{E}$.

Removing Epsilon-rules

Example:

$$
\begin{aligned}
S &\rightarrow ASA \mid aBC \mid b \\
A &\rightarrow BD \mid aAB \\
B &\rightarrow bB \mid \varepsilon \\
C &\rightarrow AaA \mid b \\
D &\rightarrow AD \mid BBB \mid a
\end{aligned}
$$

Removing Epsilon-rules

Example:

$$\mathcal{E}_0 = \{B\}$$

$$\begin{aligned}
S &\rightarrow ASA \mid aBC \mid b \\
A &\rightarrow BD \mid aAB \\
B &\rightarrow bB \mid \varepsilon \\
C &\rightarrow AaA \mid b \\
D &\rightarrow AD \mid BBB \mid a
\end{aligned}$$

Example:

$$\mathcal{E}_0 = \{B\}$$
$$\mathcal{E}_1 = \{B, D\}$$

$$\begin{aligned}
S &\rightarrow ASA \mid aBC \mid b \\
A &\rightarrow BD \mid aAB \\
B &\rightarrow bB \mid \varepsilon \\
C &\rightarrow AaA \mid b \\
D &\rightarrow AD \mid BBB \mid a
\end{aligned}$$

Removing Epsilon-rules

Example:

$$\mathcal{E}_0 = \{B\}$$
$$\mathcal{E}_1 = \{B, D\}$$
$$\mathcal{E}_2 = \{B, D, A\}$$

$$\begin{aligned}
S &\rightarrow ASA \mid aBC \mid b \\
A &\rightarrow BD \mid aAB \\
B &\rightarrow bB \mid \varepsilon \\
C &\rightarrow AaA \mid b \\
D &\rightarrow AD \mid BBB \mid a
\end{aligned}$$

Example:

$$\mathcal{E}_0 = \{B\}$$
$$\mathcal{E}_1 = \{B, D\}$$
$$\mathcal{E}_2 = \{B, D, A\}$$

$$S \rightarrow ASA \mid aBC \mid b$$
$$A \rightarrow BD \mid aAB$$
$$B \rightarrow bB \mid \varepsilon$$
$$C \rightarrow AaA \mid b$$
$$D \rightarrow AD \mid BBB \mid a$$

$$\mathcal{E} = \{B, D, A\}$$

Removing Epsilon-rules

Example:

$$\mathcal{E}_0 = \{B\}$$
$$\mathcal{E}_1 = \{B, D\}$$
$$\mathcal{E}_2 = \{B, D, A\}$$

$$\begin{aligned} S &\to ASA \mid aBC \mid b \\ A &\to BD \mid aAB \\ B &\to bB \mid \varepsilon \\ C &\to AaA \mid b \\ D &\to AD \mid BBB \mid a \end{aligned}$$

$$\mathcal{E} = \{B, D, A\}$$

$$\begin{aligned} S &\to ASA \mid SA \mid AS \mid S \mid aBC \mid aC \mid b \\ A &\to BD \mid B \mid D \mid aAB \mid aB \mid aA \mid a \\ B &\to bB \mid b \\ C &\to AaA \mid aA \mid Aa \mid a \mid b \\ D &\to AD \mid D \mid A \mid BBB \mid BB \mid B \mid a \end{aligned}$$

Removing Epsilon-rules

For every context-free grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ there is a context-free grammar $\mathcal{G}' = (\Pi', \Sigma, S', P')$ such that $\mathcal{L}(\mathcal{G}') = \mathcal{L}(\mathcal{G})$ and either:

- $\mathcal{G}'$ does not contain ε-rules, or
- the only ε-rule in $\mathcal{G}'$ is the rule $S' \to \varepsilon$ and S' does not occur on the right-hand side of any rule in $\mathcal{G}'$.

Removing Unit-rules

Rules of the form $A \to B$ where $A, B \in \Pi$ are called **unit rules**.

Proposition

For every context-free grammar $\mathcal{G}$ there is a context-free grammar $\mathcal{G}'$ without ε-rules and without unit rules such that $\mathcal{L}(\mathcal{G}') = \mathcal{L}(\mathcal{G}) - \{\varepsilon\}$.

Proof: Assume $\mathcal{G} = (\Pi, \Sigma, S, P)$ does not contain ε-rules.

For each $A \in \Pi$ compute the set $\mathcal{N}_A$ of all nonterminals that can be obtained from A by using only unit rules, i.e.,

$$\mathcal{N}_A = \{\, B \in \Pi \mid A \Rightarrow^* B \,\}$$

Construct CFG $\mathcal{G}' = (\Pi, \Sigma, S, P')$ where P' consist of rules of the form $A \to \beta$ where $A \in \Pi$, β is not a single nonterminal, and $(B \to \beta) \in P$ for some $B \in \mathcal{N}_A$.

Removing Unit-rules

Example:

$$
\begin{aligned}
S &\rightarrow AB \mid C \\
A &\rightarrow a \mid bA \\
B &\rightarrow C \mid b \\
C &\rightarrow D \mid AA \mid AaA \\
D &\rightarrow B \mid ABb
\end{aligned}
$$

Removing Unit-rules

Example:

$$\mathcal{N}_S^0 = \{S\}$$

$$\begin{aligned} S &\to AB \mid C \\ A &\to a \mid bA \\ B &\to C \mid b \\ C &\to D \mid AA \mid AaA \\ D &\to B \mid ABb \end{aligned}$$

Removing Unit-rules

Example:

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$

$S \to AB \mid C$
$A \to a \mid bA$
$B \to C \mid b$
$C \to D \mid AA \mid AaA$
$D \to B \mid ABb$

Removing Unit-rules

Example:

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$

$S \to AB \mid C$
$A \to a \mid bA$
$B \to C \mid b$
$C \to D \mid AA \mid AaA$
$D \to B \mid ABb$

Removing Unit-rules

Example:

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$S \rightarrow AB \mid C$
$A \rightarrow a \mid bA$
$B \rightarrow C \mid b$
$C \rightarrow D \mid AA \mid AaA$
$D \rightarrow B \mid ABb$

Removing Unit-rules

Example:

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$S \rightarrow AB \mid C$
$A \rightarrow a \mid bA$
$B \rightarrow C \mid b$
$C \rightarrow D \mid AA \mid AaA$
$D \rightarrow B \mid ABb$

Removing Unit-rules

Example:

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$\mathcal{N}_B^0 = \{B\}$$

$S \rightarrow AB \mid C$
$A \rightarrow a \mid bA$
$B \rightarrow C \mid b$
$C \rightarrow D \mid AA \mid AaA$
$D \rightarrow B \mid ABb$

Removing Unit-rules

Example:

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$\mathcal{N}_B^0 = \{B\}$$
$$\mathcal{N}_B^1 = \{B, C\}$$

$$S \rightarrow AB \mid C$$
$$A \rightarrow a \mid bA$$
$$B \rightarrow C \mid b$$
$$C \rightarrow D \mid AA \mid AaA$$
$$D \rightarrow B \mid ABb$$

Removing Unit-rules

Example:

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$\mathcal{N}_B^0 = \{B\}$$
$$\mathcal{N}_B^1 = \{B, C\}$$
$$\mathcal{N}_B^2 = \{B, C, D\}$$

$$S \rightarrow AB \mid C$$
$$A \rightarrow a \mid bA$$
$$B \rightarrow C \mid b$$
$$C \rightarrow D \mid AA \mid AaA$$
$$D \rightarrow B \mid ABb$$

Removing Unit-rules

Example:

$$S \to AB \mid C$$
$$A \to a \mid bA$$
$$B \to C \mid b$$
$$C \to D \mid AA \mid AaA$$
$$D \to B \mid ABb$$

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$\mathcal{N}_B^0 = \{B\}$$
$$\mathcal{N}_B^1 = \{B, C\}$$
$$\mathcal{N}_B^2 = \{B, C, D\}$$

$$\mathcal{N}_C^0 = \{C\}$$

Removing Unit-rules

Example:

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$S \rightarrow AB \mid C$$
$$A \rightarrow a \mid bA$$
$$B \rightarrow C \mid b$$
$$C \rightarrow D \mid AA \mid AaA$$
$$D \rightarrow B \mid ABb$$

$$\mathcal{N}_B^0 = \{B\}$$
$$\mathcal{N}_B^1 = \{B, C\}$$
$$\mathcal{N}_B^2 = \{B, C, D\}$$

$$\mathcal{N}_C^0 = \{C\}$$
$$\mathcal{N}_C^1 = \{C, D\}$$

Removing Unit-rules

Example:

$$S \to AB \mid C$$
$$A \to a \mid bA$$
$$B \to C \mid b$$
$$C \to D \mid AA \mid AaA$$
$$D \to B \mid ABb$$

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$\mathcal{N}_B^0 = \{B\}$$
$$\mathcal{N}_B^1 = \{B, C\}$$
$$\mathcal{N}_B^2 = \{B, C, D\}$$

$$\mathcal{N}_C^0 = \{C\}$$
$$\mathcal{N}_C^1 = \{C, D\}$$
$$\mathcal{N}_C^2 = \{C, D, B\}$$

Removing Unit-rules

Example:

$$S \rightarrow AB \mid C$$
$$A \rightarrow a \mid bA$$
$$B \rightarrow C \mid b$$
$$C \rightarrow D \mid AA \mid AaA$$
$$D \rightarrow B \mid ABb$$

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$\mathcal{N}_B^0 = \{B\}$$
$$\mathcal{N}_B^1 = \{B, C\}$$
$$\mathcal{N}_B^2 = \{B, C, D\}$$

$$\mathcal{N}_C^0 = \{C\}$$
$$\mathcal{N}_C^1 = \{C, D\}$$
$$\mathcal{N}_C^2 = \{C, D, B\}$$

$$\mathcal{N}_D^0 = \{D\}$$

Removing Unit-rules

Example:

$$S \rightarrow AB \mid C$$
$$A \rightarrow a \mid bA$$
$$B \rightarrow C \mid b$$
$$C \rightarrow D \mid AA \mid AaA$$
$$D \rightarrow B \mid ABb$$

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$\mathcal{N}_B^0 = \{B\}$$
$$\mathcal{N}_B^1 = \{B, C\}$$
$$\mathcal{N}_B^2 = \{B, C, D\}$$

$$\mathcal{N}_C^0 = \{C\}$$
$$\mathcal{N}_C^1 = \{C, D\}$$
$$\mathcal{N}_C^2 = \{C, D, B\}$$

$$\mathcal{N}_D^0 = \{D\}$$
$$\mathcal{N}_D^1 = \{D, B\}$$

Removing Unit-rules

Example:

$$S \rightarrow AB \mid C$$
$$A \rightarrow a \mid bA$$
$$B \rightarrow C \mid b$$
$$C \rightarrow D \mid AA \mid AaA$$
$$D \rightarrow B \mid ABb$$

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$\mathcal{N}_B^0 = \{B\}$$
$$\mathcal{N}_B^1 = \{B, C\}$$
$$\mathcal{N}_B^2 = \{B, C, D\}$$

$$\mathcal{N}_C^0 = \{C\}$$
$$\mathcal{N}_C^1 = \{C, D\}$$
$$\mathcal{N}_C^2 = \{C, D, B\}$$

$$\mathcal{N}_D^0 = \{D\}$$
$$\mathcal{N}_D^1 = \{D, B\}$$
$$\mathcal{N}_D^2 = \{D, B, C\}$$

Removing Unit-rules

Example:

$$S \rightarrow AB \mid C$$
$$A \rightarrow a \mid bA$$
$$B \rightarrow C \mid b$$
$$C \rightarrow D \mid AA \mid AaA$$
$$D \rightarrow B \mid ABb$$

$\mathcal{N}_S^0 = \{S\}$
$\mathcal{N}_S^1 = \{S, C\}$
$\mathcal{N}_S^2 = \{S, C, D\}$
$\mathcal{N}_S^3 = \{S, C, D, B\}$

$\mathcal{N}_A^0 = \{A\}$

$\mathcal{N}_B^0 = \{B\}$
$\mathcal{N}_B^1 = \{B, C\}$
$\mathcal{N}_B^2 = \{B, C, D\}$

$\mathcal{N}_C^0 = \{C\}$
$\mathcal{N}_C^1 = \{C, D\}$
$\mathcal{N}_C^2 = \{C, D, B\}$

$\mathcal{N}_D^0 = \{D\}$
$\mathcal{N}_D^1 = \{D, B\}$
$\mathcal{N}_D^2 = \{D, B, C\}$

$\mathcal{N}_S = \{S, C, D, B\}$
$\mathcal{N}_A = \{A\}$
$\mathcal{N}_B = \{B, C, D\}$
$\mathcal{N}_C = \{C, D, B\}$
$\mathcal{N}_D = \{D, B, C\}$

Removing Unit-rules

Example:

$$S \to AB \mid C$$
$$A \to a \mid bA$$
$$B \to C \mid b$$
$$C \to D \mid AA \mid AaA$$
$$D \to B \mid ABb$$

$$\mathcal{N}_S^0 = \{S\}$$
$$\mathcal{N}_S^1 = \{S, C\}$$
$$\mathcal{N}_S^2 = \{S, C, D\}$$
$$\mathcal{N}_S^3 = \{S, C, D, B\}$$

$$\mathcal{N}_A^0 = \{A\}$$

$$\mathcal{N}_B^0 = \{B\}$$
$$\mathcal{N}_B^1 = \{B, C\}$$
$$\mathcal{N}_B^2 = \{B, C, D\}$$

$$\mathcal{N}_C^0 = \{C\}$$
$$\mathcal{N}_C^1 = \{C, D\}$$
$$\mathcal{N}_C^2 = \{C, D, B\}$$

$$\mathcal{N}_D^0 = \{D\}$$
$$\mathcal{N}_D^1 = \{D, B\}$$
$$\mathcal{N}_D^2 = \{D, B, C\}$$

$$\mathcal{N}_S = \{S, C, D, B\}$$
$$\mathcal{N}_A = \{A\}$$
$$\mathcal{N}_B = \{B, C, D\}$$
$$\mathcal{N}_C = \{C, D, B\}$$
$$\mathcal{N}_D = \{D, B, C\}$$

$$S \to AB \mid AA \mid AaA \mid ABb \mid b$$
$$A \to a \mid bA$$
$$B \to b \mid AA \mid AaA \mid ABb$$
$$C \to AA \mid AaA \mid ABb \mid b$$
$$D \to ABb \mid b \mid AA \mid AaA$$

Chomsky Normal Form

Definition

A context-free grammar is in **Chomsky normal form** if every rule is of on of the following forms:

- $A \to BC$
- $A \to a$

where a is any terminal and A, B,and C are any nonterminals.

In addition we permit the rule $S \to \varepsilon$, where S the initial nonterminal. In that case, nonterminal S cannot occur on the right-hand side of any rule.

Chomsky Normal Form

Proposition

For every context-free grammar $\mathcal{G}$ there is an equivalent context-free grammar $\mathcal{G}'$ in Chomsky normal form.

Proof: Perform the following transformations on $\mathcal{G}$:

1. Decompose each rule $A \rightarrow \alpha$ where $|\alpha| \geq 3$ into a sequence of rules where each right-hand size has length 2.
2. Remove ε-rules.
3. Remove unit rules.
4. For each terminal a occurring on the right-hand size of some rule $A \rightarrow \alpha$ where $|\alpha| = 2$ introduce a new nonterminal N_a, replace occurrences of a on such right-hand sides with N_a, and add $N_a \rightarrow a$ as a new rule.

Chomsky Normal Form

Example:

$$S \rightarrow ASA \mid aB$$
$$A \rightarrow B \mid S$$
$$B \rightarrow b \mid \varepsilon$$

Chomsky Normal Form

Example:

$$
\begin{aligned}
S &\rightarrow ASA \mid aB \\
A &\rightarrow B \mid S \\
B &\rightarrow b \mid \varepsilon
\end{aligned}
$$

Step 1:

$$
\begin{aligned}
S &\rightarrow AZ \mid aB \\
Z &\rightarrow SA \\
A &\rightarrow B \mid S \\
B &\rightarrow b \mid \varepsilon
\end{aligned}
$$

Chomsky Normal Form

Example:

$$
\begin{aligned}
S &\to ASA \mid aB \\
A &\to B \mid S \\
B &\to b \mid \varepsilon
\end{aligned}
$$

Step 1:

$$
\begin{aligned}
S &\to AZ \mid aB \\
Z &\to SA \\
A &\to B \mid S \\
B &\to b \mid \varepsilon
\end{aligned}
$$

Step 2:

$\mathcal{E} = \{B, A\}$

Chomsky Normal Form

Example:

$$
\begin{aligned}
S &\to ASA \mid aB \\
A &\to B \mid S \\
B &\to b \mid \varepsilon
\end{aligned}
$$

Step 1:

$$
\begin{aligned}
S &\to AZ \mid aB \\
Z &\to SA \\
A &\to B \mid S \\
B &\to b \mid \varepsilon
\end{aligned}
$$

Step 2:

$\mathcal{E} = \{B, A\}$

$$
\begin{aligned}
S_0 &\to S \\
S &\to AZ \mid Z \mid aB \mid a \\
Z &\to SA \mid S \\
A &\to B \mid S \\
B &\to b
\end{aligned}
$$

Chomsky Normal Form

Example:

$$S \to ASA \mid aB$$
$$A \to B \mid S$$
$$B \to b \mid \varepsilon$$

Step 1:

$$S \to AZ \mid aB$$
$$Z \to SA$$
$$A \to B \mid S$$
$$B \to b \mid \varepsilon$$

Step 2:

$\mathcal{E} = \{B, A\}$

$$S_0 \to S$$
$$S \to AZ \mid Z \mid aB \mid a$$
$$Z \to SA \mid S$$
$$A \to B \mid S$$
$$B \to b$$

Step 3:

$$\mathcal{N}_{S_0} = \{S_0, S, Z\}$$
$$\mathcal{N}_S = \{S, Z\}$$
$$\mathcal{N}_Z = \{Z, S\}$$
$$\mathcal{N}_A = \{A, B, S, Z\}$$
$$\mathcal{N}_B = \{B\}$$

Chomsky Normal Form

Example:

$$S \to ASA \mid aB$$
$$A \to B \mid S$$
$$B \to b \mid \varepsilon$$

Step 1:

$$S \to AZ \mid aB$$
$$Z \to SA$$
$$A \to B \mid S$$
$$B \to b \mid \varepsilon$$

Step 2:

$\mathcal{E} = \{B, A\}$

$$S_0 \to S$$
$$S \to AZ \mid Z \mid aB \mid a$$
$$Z \to SA \mid S$$
$$A \to B \mid S$$
$$B \to b$$

Step 3:

$\mathcal{N}_{S_0} = \{S_0, S, Z\}$
$\mathcal{N}_S = \{S, Z\}$
$\mathcal{N}_Z = \{Z, S\}$
$\mathcal{N}_A = \{A, B, S, Z\}$
$\mathcal{N}_B = \{B\}$

$$S_0 \to AZ \mid aB \mid a \mid SA$$
$$S \to AZ \mid aB \mid a \mid SA$$
$$Z \to SA \mid AZ \mid aB \mid a$$
$$A \to b \mid AZ \mid aB \mid a \mid SA$$
$$B \to b$$

Chomsky Normal Form

Example:

$$
\begin{aligned}
S &\to ASA \mid aB \\
A &\to B \mid S \\
B &\to b \mid \varepsilon
\end{aligned}
$$

Step 1:

$$
\begin{aligned}
S &\to AZ \mid aB \\
Z &\to SA \\
A &\to B \mid S \\
B &\to b \mid \varepsilon
\end{aligned}
$$

Step 2:

$\mathcal{E} = \{B, A\}$

$$
\begin{aligned}
S_0 &\to S \\
S &\to AZ \mid Z \mid aB \mid a \\
Z &\to SA \mid S \\
A &\to B \mid S \\
B &\to b
\end{aligned}
$$

Step 3:

$$
\begin{aligned}
\mathcal{N}_{S_0} &= \{S_0, S, Z\} \\
\mathcal{N}_S &= \{S, Z\} \\
\mathcal{N}_Z &= \{Z, S\} \\
\mathcal{N}_A &= \{A, B, S, Z\} \\
\mathcal{N}_B &= \{B\}
\end{aligned}
$$

$$
\begin{aligned}
S_0 &\to AZ \mid aB \mid a \mid SA \\
S &\to AZ \mid aB \mid a \mid SA \\
Z &\to SA \mid AZ \mid aB \mid a \\
A &\to b \mid AZ \mid aB \mid a \mid SA \\
B &\to b
\end{aligned}
$$

Step 4:

$$
\begin{aligned}
S_0 &\to AZ \mid YB \mid a \mid SA \\
S &\to AZ \mid YB \mid a \mid SA \\
Z &\to SA \mid AZ \mid YB \mid a \\
A &\to b \mid AZ \mid YB \mid a \mid SA \\
B &\to b \\
Y &\to a
\end{aligned}
$$

Chomsky Normal Form

Grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ in Chomsky normal form has some properties that allow to determine whether $w \in \Sigma^*$ belongs to the language generated by grammar $\mathcal{G}$ (i.e., if $w \in \mathcal{L}(\mathcal{G})$):

- Let us assume that $w \in \mathcal{L}(\mathcal{G})$ (and so $S \Rightarrow^* w$)and that $|w| = n$, where $n \geq 1$. Then for (every) derivation $S \Rightarrow^* w$ holds:
  - The rules of the form $A \rightarrow a$ (i.e., a nonterminal is rewritten to exactly one terminal) are used in exactly n steps of the derivation.
  - The rules of the form $A \rightarrow BC$ (i.e., a nonterminal is rewritten to a pair of nonterminals) are used in exactly $n - 1$ steps of the derivation.

So every derivation $S \Rightarrow^* w$, where $|w| = n$, has $2n - 1$ steps, where n of these steps are of the form $A \rightarrow a$ and $n - 1$ of the form $A \rightarrow BC$.

Chomsky Normal Form

To find out whether $S \Rightarrow^* w$, it is sufficient to try by brute force all possible derivations of length $2n - 1$.

Such algorithm has exponential time complexity with respect to the length of w.

Such systematic trying of all possibilities can be implemented by using so called **dynamic programming** in a way that is much more efficient than a straightforward algorithm that generates all derivations of the given length.

Cocke-Younger-Kasami algorithm, with time complexity $O(n^3)$, is based on this idea. (Assuming a fixed grammar $\mathcal{G}$.)

Cocke-Younger-Kasami Algorithm

The question if $S \Rightarrow^* w$ is a special case of the question if

$$A \Rightarrow^* w\,,$$

where $A \in \Pi$ is an arbitrary nonterminal and $w \in \Sigma^*$ is an arbitrary word consisting of terminals.

It is obvious that:

- If $|w| = 1$: Then $A \Rightarrow^* w$ iff there is a rule $A \rightarrow b$ in P where $w = b$.
- If $|w| > 1$: Then $A \Rightarrow^* w$ iff there is a rule $A \rightarrow BC$ in P where for some words u and v such that $w = uv$, $|u| \geq 1$ and $|v| \geq 1$, it holds that $B \Rightarrow^* u$ and $C \Rightarrow^* v$.

Cocke-Younger-Kasami Algorithm

Let us assume that a word $w \in \Sigma^*$ with $|w| = n$ where $n \geq 1$ and

$$w = a_1 a_2 \cdots a_n .$$

Instead of solving the original question whether $S \Rightarrow^* w$, we will solve the following more general problem for all nonempty subwords v of the word w:

- To find the set of all nonterminals A from the set Π such that $A \Rightarrow^* v$.

Let us denote the set of all nonterminals generating subword v of length i and starting on position j as $\mathcal{F}[i][j]$, i.e., for each $A \in \Pi$ it holds that

$$A \in \mathcal{F}[i][j] \qquad \Longleftrightarrow \qquad A \Rightarrow^* a_j a_{j+1} \ldots a_{j+(i-1)}$$

To find out whether $S \Rightarrow^* w$, is therefore the same problem as to find out whether $S \in \mathcal{F}[n][1]$.

Cocke-Younger-Kasami Algorithm

- The algorithm computes values $\mathcal{F}[i][j]$ at first for subwords of length 1 (i.e., $i = 1$), then for subwords of length 2 (i.e., $i = 2$), then for subwords of length 3, length 4, etc.
- Values $\mathcal{F}[i][j]$ are stored in a twodimensional array $\mathcal{F}$, where $1 \leq i \leq n$ a $1 \leq j \leq n - i + 1$, where the elements of this array are subsets of nonterminals from the set Π.
- In the computation of the value $\mathcal{F}[i][j]$ the previously computed values $\mathcal{F}[i'][j']$, where $i' < i$, are used.
- Let us assume that at the beginning all elements of array $\mathcal{F}$ are initialized to $\emptyset$.

Cocke-Younger-Kasami Algorithm

```
for j := 1 to n do
    for each (A → b) ∈ P do
        if b = a_j then
            add A to F[1][j]
for i := 2 to n do
    for j := 1 to n − i + 1 do
        for k := 1 to i − 1 do
            for each (A → BC) ∈ P do
                if B ∈ F[k][j] and C ∈ F[i − k][j + k] then
                    add A to F[i][j]
```