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

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

Context-Free Grammars

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

$$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$
$$C \rightarrow D \mid DC$$
$$E \rightarrow C \mid (-E) \mid (E+E) \mid (E-E) \mid (E*E) \mid (E/E)$$

Context-Free Grammars

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 "$,$"):

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

Context-Free Grammars

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

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

Remark:

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

Context-Free Grammars

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

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

Context-Free Grammars

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

$$A \to \alpha_1 \qquad A \to \alpha_2 \qquad A \to \alpha_3$$

we can write them in a more succinct way as

$$A \to \alpha_1 \mid \alpha_2 \mid \alpha_3$$

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

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

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

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

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

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

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

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

$\underline{A}$

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

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

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

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

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

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

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

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

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

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

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

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

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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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:

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

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

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

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

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

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.

Context-Free Grammars

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 $\Rightarrow$).

Context-Free Grammars

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 $L = \mathcal{L}(\mathcal{G})$.

Derivation Tree

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

Derivation Tree

A

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

A

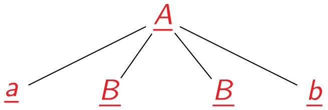
Derivation Tree

$\underline{A}$

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

$\underline{A}$

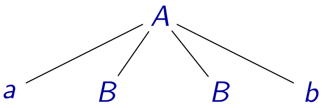
Derivation Tree



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

$A \Rightarrow aBBb$

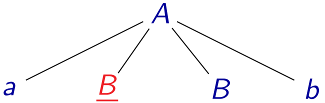
Derivation Tree



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

$A \Rightarrow aBBb$

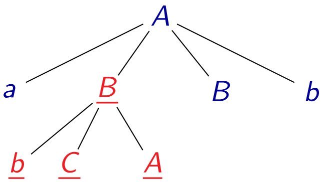
Derivation Tree



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

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

Derivation Tree

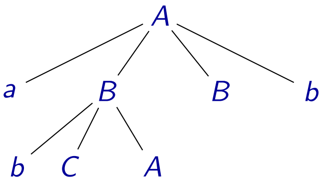


$A \rightarrow aBBb \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$

$A \Rightarrow aBBb \Rightarrow abCABb$

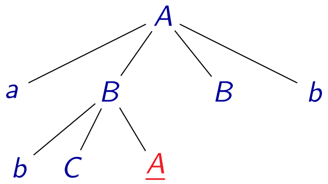
Derivation Tree

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



$$A \Rightarrow aBBb \Rightarrow abCABb$$

Derivation Tree

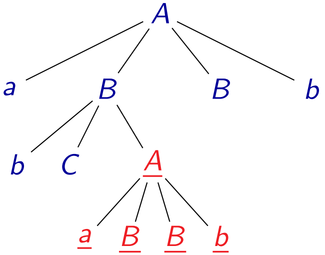


$$\underline{A} \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$

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

Derivation Tree

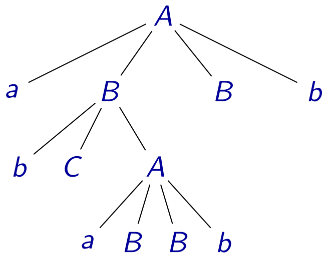
$\underline{A} \rightarrow \underline{aBBb} \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$



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

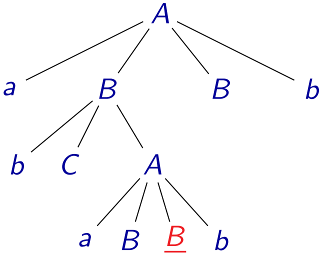
Derivation Tree

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



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

Derivation Tree

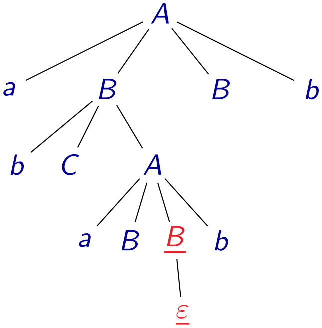


$$A \to aBBb \mid AaA$$
$$\underline{B} \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

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

Derivation Tree

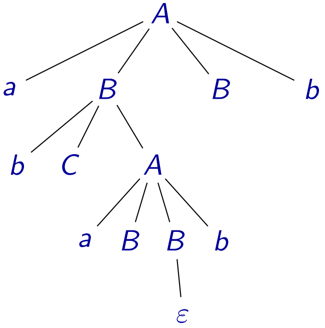
$A \rightarrow aBBb \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$



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

Derivation Tree

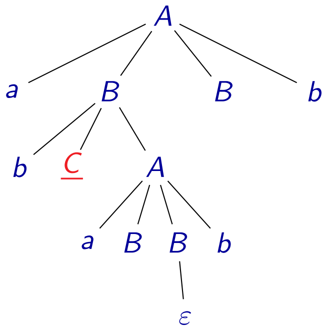
$A \rightarrow aBBb \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$



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

Derivation Tree

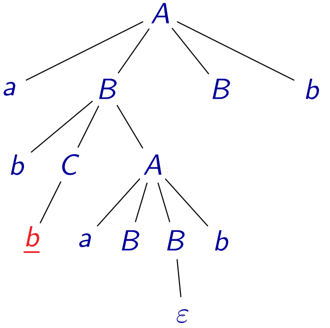
$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$



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

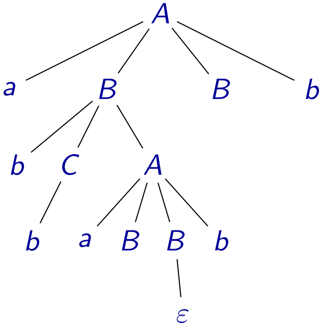
Derivation Tree

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



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

Derivation Tree

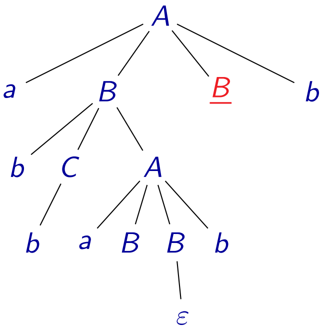


$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

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

Derivation Tree

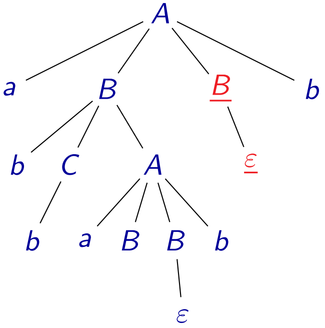
$A \rightarrow aBBb \mid AaA$
$\underline{B} \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$



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

Derivation Tree

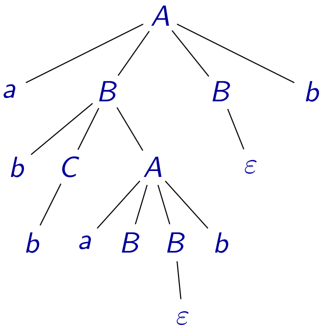
$A \rightarrow aBBb \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$



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

Derivation Tree

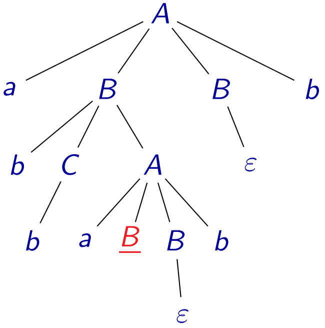
$$A \rightarrow aBBb \mid AaA$$
$$B \rightarrow \varepsilon \mid bCA$$
$$C \rightarrow AB \mid a \mid b$$



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

Derivation Tree

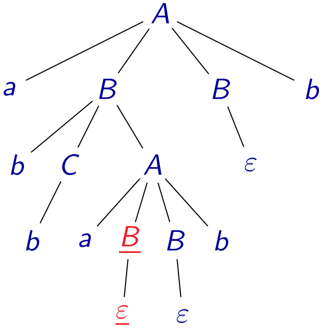
$A \rightarrow aBBb \mid AaA$
$\underline{B} \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$



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

Derivation Tree

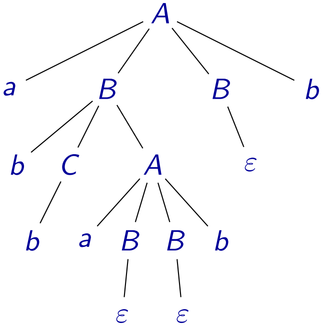
$A \rightarrow aBBb \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$



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

Derivation Tree

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



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

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 $A \rightarrow \alpha$.

Context-Free Grammars

Example: A grammar generating the language

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

Context-Free Grammars

Example: 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

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

Example: 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

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

$$S \Rightarrow \varepsilon$$
$$S \Rightarrow aSb \Rightarrow ab$$
$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$
$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$
$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaaSbbbb \Rightarrow aaaabbbb$$
$$\dots$$

Context-Free Grammars

Example: A grammar generating the language L 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.

Context-Free Grammars

Example: A grammar generating the language L 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: A grammar generating the language L 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: A grammar generating the language L consisting of all correctly parenthesised sequences of symbols '$($' and '$)$'.

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

Context-Free Grammars

Example: A grammar generating the language L consisting of all correctly parenthesised sequences of symbols '(' and ')'.

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

Solution:

$$A \rightarrow \varepsilon \mid (A) \mid AA$$

Context-Free Grammars

Example: A grammar generating the language L consisting of all correctly parenthesised sequences of symbols '$($' and '$)$'.

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

Solution:

$$A \rightarrow \varepsilon \mid (A) \mid AA$$

$A \Rightarrow AA \Rightarrow (A)A \Rightarrow (A)(A) \Rightarrow (AA)(A) \Rightarrow ((A)A)(A) \Rightarrow (()A)(A) \Rightarrow (()(A))(A) \Rightarrow (()())(A) \Rightarrow (()())((A)) \Rightarrow (()())(())$

Context-Free Grammars

Example: 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: 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:

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

Context-Free Grammars

Example: 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:

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

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

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.

Equivalence 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.

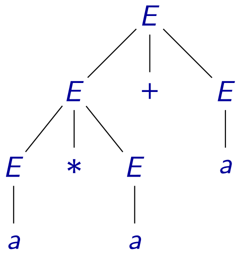
Ambiguous Grammars

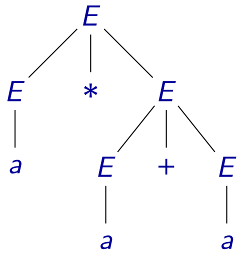
A grammar $\mathcal{G}$ is **ambiguous** if there is a word $w \in \mathcal{L}(\mathcal{G})$ that has two different derivation trees, resp. two different left or two different right derivations.

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

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

can be replaced with the equivalent grammar

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

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 $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{G}_1) \cup \mathcal{L}(\mathcal{G}_2)$:

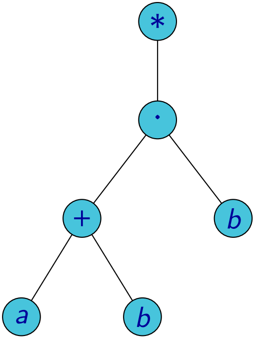
$$\mathcal{G} = (\Pi_1 \cup \Pi_2 \cup \{S\},\ \Sigma,\ S,\ P_1 \cup P_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\})$$

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

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

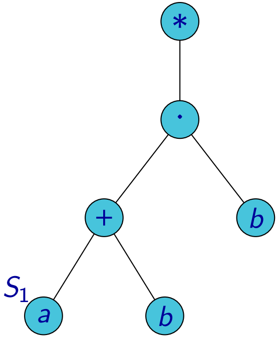
A Context-Free Grammar for a Regular Expression

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



A Context-Free Grammar for a Regular Expression

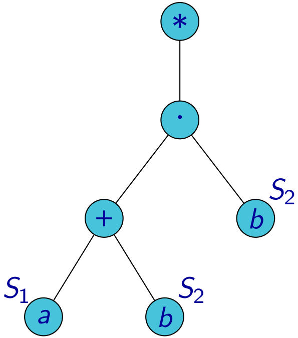
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



$S_1 \rightarrow a$

A Context-Free Grammar for a Regular Expression

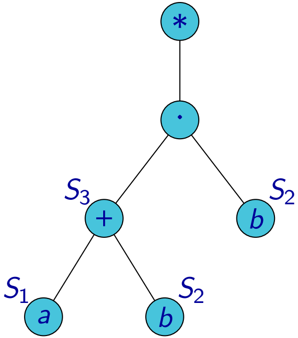
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



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

A Context-Free Grammar for a Regular Expression

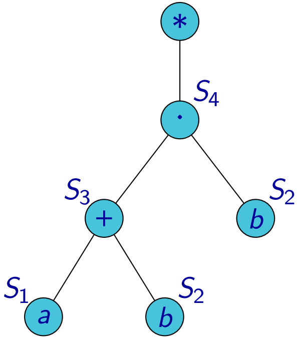
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



$$S_3 \rightarrow S_1 \mid S_2$$
$$S_2 \rightarrow b$$
$$S_1 \rightarrow a$$

A Context-Free Grammar for a Regular Expression

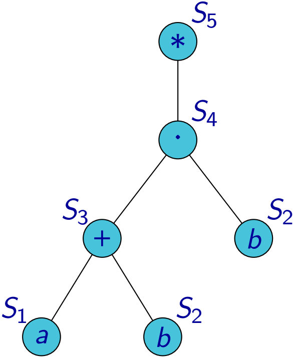
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



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

A Context-Free Grammar for a Regular Expression

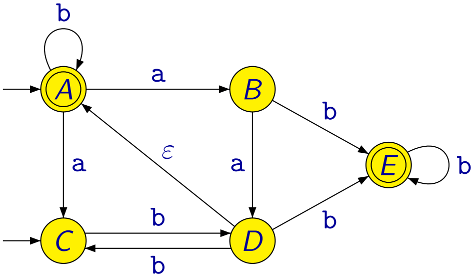
Example: The construction of a context-free grammar for regular expression $((a + b) \cdot b)^*$:



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

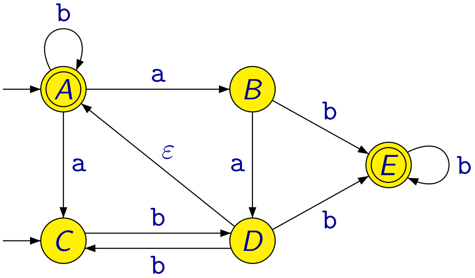
A Context-Free Grammar for a Finite Automaton

Example:



A Context-Free Grammar for a Finite Automaton

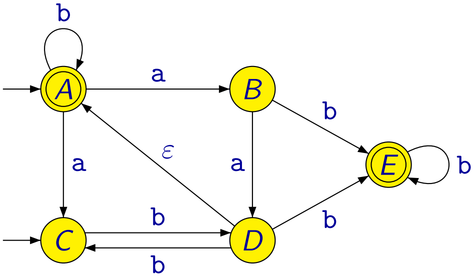
Example:



$$S \rightarrow A \mid C$$

A Context-Free Grammar for a Finite Automaton

Example:

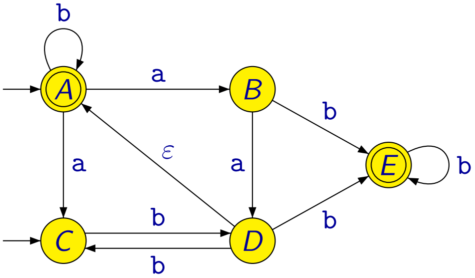


$$S \rightarrow A \mid C$$

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

A Context-Free Grammar for a Finite Automaton

Example:



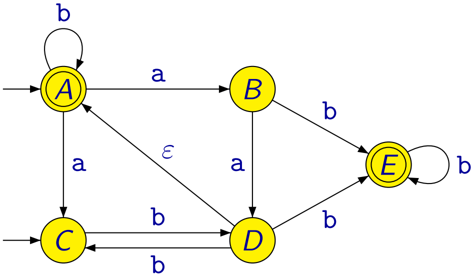
$$S \rightarrow A \mid C$$

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

$$A \rightarrow \varepsilon$$
$$E \rightarrow \varepsilon$$

A Context-Free Grammar for a Finite Automaton

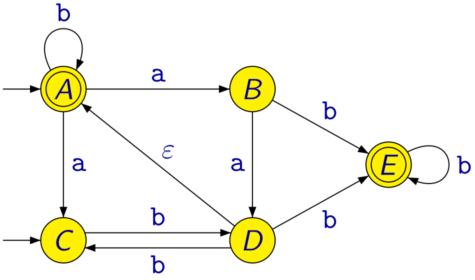
Example:



Alternative construction:

A Context-Free Grammar for a Finite Automaton

Example:

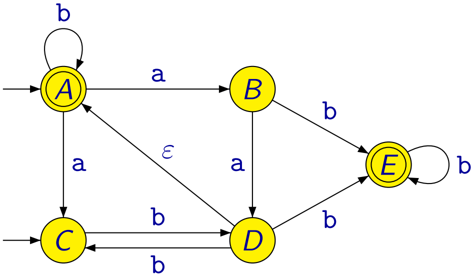


Alternative construction:

$S \rightarrow A \mid E$

A Context-Free Grammar for a Finite Automaton

Example:

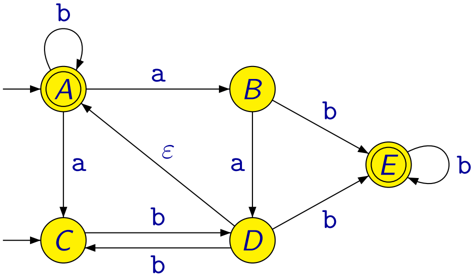


Alternative construction:

$$S \to A \mid E$$

$$\begin{aligned} A &\to Ab \mid D \\ B &\to Aa \\ C &\to Aa \mid Db \\ D &\to Ba \mid Cb \\ E &\to Bb \mid Db \mid Eb \end{aligned}$$

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

$$A \rightarrow aX_1 \qquad X_1 \rightarrow bX_2 \qquad X_2 \rightarrow bB$$

where X_1, X_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 $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{G})$.