Context-free languages I

- In Chapter 1 we consider two ways to describe languages

  automata & regular expressions
- Context-free grammars (CFG)

  More powerful than automata
- CFG is used in compilers and interpreters for parsers to read programs

Context-free grammars I

- A grammar G_1:

$$A \to 0A1$$
$$A \to B$$
$$B \to \#$$

  Each one is called a substitution rule
- Variables: A, B (capital letters)
- Terminals: $0, 1, \#$ (lowercase letters, number, special symbols)
- Start variable: A

Context-free grammars II

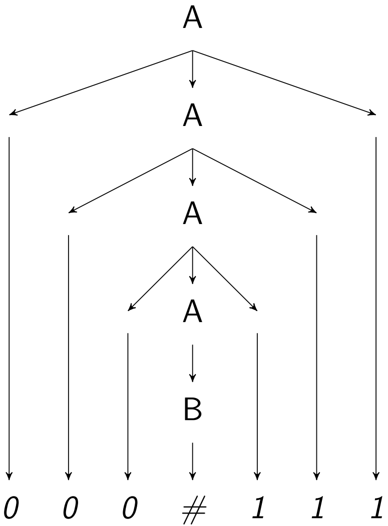
- A grammar: a collection of substitution rules
- Derivation: G_1 generates $000\#111$

$$
\begin{aligned}
& A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \\
& \Rightarrow 000B111 \Rightarrow 000\#111
\end{aligned}
$$

- Parse tree

  Fig 2.1

Context-free grammars III



Context-free grammars IV

- $L(G)$: language of grammar
- For the CFG example we just discussed,

$$L(G_1) = \{0^n\#1^n \mid n \geq 0\}$$

- CFG is more powerful than regular expressions because we showed earlier that this language is not regular
- Representation of rules:

$$A \rightarrow 0A1 \text{ and } A \rightarrow B$$

is often simplified to

Context-free grammars V

$$A \to 0A1 \mid B$$

- Example

$$\begin{aligned}
\langle \text{S} \rangle &\Rightarrow \langle \text{Noun-Phrase} \rangle \langle \text{Verb-Phrase} \rangle \\
&\Rightarrow \langle \text{Complex-Noun} \rangle \langle \text{Verb-Phrase} \rangle \\
&\Rightarrow \langle \text{Article} \rangle \langle \text{Noun} \rangle \langle \text{Verb-Phrase} \rangle \\
&\Rightarrow \text{a} \langle \text{Noun} \rangle \langle \text{Verb-Phrase} \rangle \\
&\Rightarrow \text{a boy} \langle \text{Verb-Phrase} \rangle \\
&\Rightarrow \text{a boy} \langle \text{Complex-Verb} \rangle \\
&\Rightarrow \text{a boy} \langle \text{Verb} \rangle \\
&\Rightarrow \text{a boy sees}
\end{aligned}$$

Context-free grammars VI

- Why called “context-free” ?
  Rules independent of context

Formal definition of a context-free grammar I

- (V, Σ, R, S)

  V: variables, finite set

  Σ: terminals, finite set

  R: rules

  variable $\rightarrow$ strings of variables and terminals (including ϵ)

- $S \in V$, start variable

Formal definition of a context-free grammar II

- For the example G_1:

$$
\begin{aligned}
A &\rightarrow 0A1 \\
A &\rightarrow B \\
B &\rightarrow \#
\end{aligned}
$$

$V = \{A, B\}, \Sigma = \{0, 1, \#\}, S = A, R$: the above three rules

Derivation of strings I

- If u, v, w are strings and a rule $A \to w$ is applied, then we say

  $$uAv \text{ yields } uwv$$

  and this is denoted as

  $$uAv \Rightarrow uwv$$

- if

  $$u = v \text{ or } u \Rightarrow u_1 \Rightarrow \cdots \Rightarrow u_k \Rightarrow v$$

  then we say

  $$u \stackrel{*}{\Rightarrow} v$$

Derivation of strings II

- Language of a CFG

$$\{w \in \Sigma^* \mid S \stackrel{*}{\Rightarrow} w\}$$

Example 2.3 I

- $G_3 = (\{S\}, \{a, b\}, R, S)$
  R:
  $$S \rightarrow aSb \mid SS \mid \epsilon$$
- What is the language?
- If we treat a, b respectively as (and), then we have
  all valid nested parentheses