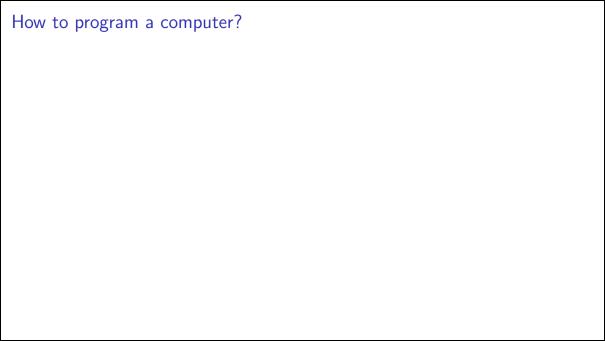
01204211 Discrete Mathematics Lecture 11a: Context-free languages and grammars (1)¹

Jittat Fakcharoenphol

August 29, 2024

¹Based on lecture notes of *Models of Computation* course by Jeff Erickson.



High-level programming languages

- ► 1954 FORTRAN
- ▶ 1958 LISP
- ▶ 1958 COBOL
- ▶ 1960 ALGO60





Building up languages

Regular languages:

- Contenation
- Union
- ► Kleene star

Context-free grammar:

- Contenation
- ► Union
- Recursion

Example

$$\begin{array}{ccc} S & \rightarrow & 0S1 \\ S & \rightarrow & \varepsilon \end{array}$$

You can use "|" to write production rules more succinctly.

$$S \rightarrow 0S1 \mid \varepsilon$$

Definition

A context-free grammer consists of the following components:

- ightharpoonup a finite set Σ , a set of *symbols* (or *terminals*),
- ▶ a finite set Γ disjoint from Σ , a set of *non-terminals* (you can think of them as variables).
- ▶ a finite set R of production rules of the form $A \to w$ where $A \in \Gamma$ and $w \in (\Sigma \cup \Gamma)^*$ is a string of symbols and variable, and
- ▶ a starting non-terminal (usually the non-terminal of the first production rule).

Another example

$$S \rightarrow A \mid B$$

$$A \rightarrow 0A \mid 0C$$

$$B \rightarrow B1 \mid C1$$

$$C \rightarrow \varepsilon \mid 0C1$$

Here $\Sigma = \{\mathbf{0}, \mathbf{1}\}$ and $\Gamma = \{S, A, B, C\}$.

Applying the rules

If you have strings $x,y,z\in (\Sigma\cup\Gamma)^*$ and the production rule

$$A \to y$$
,

You can apply the rule to the string xAz. This yields the string

We use the notation

$$xAz \leadsto xyz$$

to describe this application.

Derivation

We say that z derives from x if we can obtain z from x by production rule applications, denoted by $x \leadsto^* z$.

Formally, for any string $x,z\in (\Sigma\cup\Gamma)^*$, we say that $x\leadsto^*z$ if either

- ightharpoonup x=z, or
- $ightharpoonup x\leadsto y$ and $y\leadsto^*z$ for some string $y\in (\Sigma\cup\Gamma)^*.$

L(w)

The language L(w) of string $w \in (\Sigma \cup \Gamma)^*$ is the set of all strings in Σ^* that derive from w, i.e.,

$$L(w) = \{ x \in \Sigma^* \mid w \leadsto^* x \}.$$

The language **generated by** a context-free grammar G, denoted by L(G) is the language of its starting non-terminal.

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

Grammar G_1

$$S \rightarrow NPVP$$
 $NP \rightarrow CN|CNPP$
 $VP \rightarrow CV|CVPP$
 $PP \rightarrow PREPCN$
 $CN \rightarrow ARTN$
 $CV \rightarrow V|VNP$
 $ART \rightarrow a|the$
 $N \rightarrow boy|girl|flower$
 $V \rightarrow touches|likes|sees$
 $PREP \rightarrow with$

Small English grammar

$$S \rightarrow NPVP$$
 $NP \rightarrow CN|CNPP$
 $VP \rightarrow CV|CVPP$
 $PP \rightarrow PREPCN$
 $CN \rightarrow ARTN$
 $CV \rightarrow V|VNP$
 $ART \rightarrow a|the$
 $N \rightarrow boy|girl|flower$
 $V \rightarrow touches|likes|sees$
 $PREP \rightarrow with$

- ightharpoonup Examples of strings in $L(G_2)$ are:
 - a boy sees
 - the boy sees a flower
 - ▶ a girl with a flower likes the boy

Parse tree

- ▶ 00011
- **01111**
- **111110**

$$\begin{array}{ccc} S & \rightarrow & A \mid B \\ A & \rightarrow & 0A \mid 0C \\ B & \rightarrow & B1 \mid C1 \\ C & \rightarrow & \varepsilon \mid 0C1 \end{array}$$

Parse tree

a girl with a flower likes the boy

```
S \rightarrow NPVP
    NP \rightarrow CN|CN|PP
    VP \rightarrow CV|CV|PP
    PP \rightarrow PREP CN
    CN \rightarrow ART N
    CV \rightarrow V|V|NP
  ART \rightarrow \text{a}|\text{the}
      N \rightarrow \text{boy|girl|flower}
      V \rightarrow \text{touches}|\text{likes}|\text{sees}|
PREP \rightarrow \text{with}
```

Ambiguity

- ▶ 1 + 1 * 1
- 1+1+1+1+1

$$S \rightarrow 1 \mid S + S \mid S * S$$

- lacktriangle A string w is **ambiguous** with respect to a grammar G if more than one parse tree for w exists.
- ightharpoonup A grammar G is **ambiguous** if some string is ambiguous with respect to G.