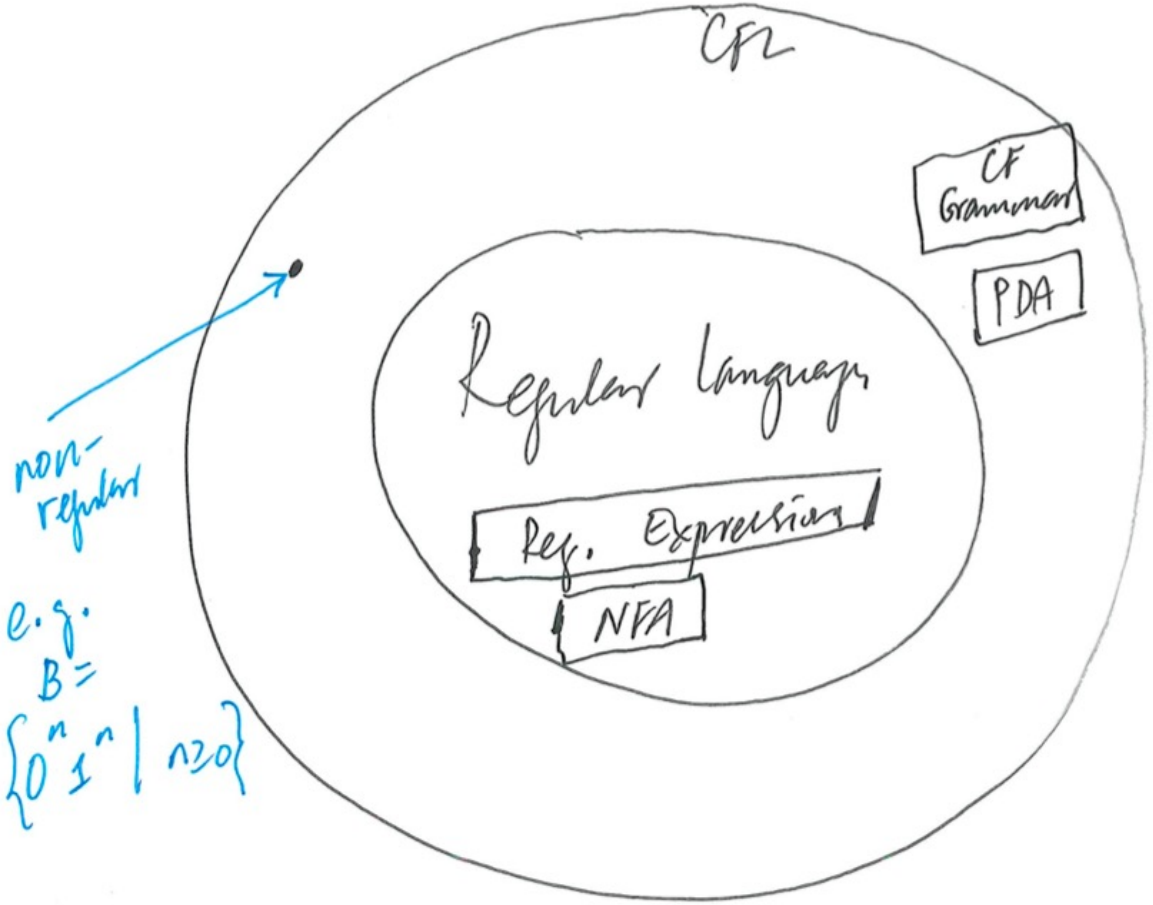
Context-free Languages

Preview



CFL
CF Grammar
PDA
Regular language
Reg. Expression
NFA
non-regular
e.g.
B = $\{0^n 1^n \mid n \geq 0\}$

Context-free grammars

- More powerful method (than NFAs and regular expressions) for describing languages
- Describe certain features that have recursive structure
- Major applications:
  - **Study of human languages**
  - **Compilers**
    - **parsers** – extract the meaning of a program prior to generating a compiled code or performing the interpreted execution
    - a number of methodologies facilitate the construction of a parser once a context-free grammar is available
- The collection of languages associated with CFGs are called ***context-free languages***

Context-free grammars

An example of CFG – G_1

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

A grammar consists of:

- **substitution rules** (**productions**) – each line above is a rule
- **variables –** what is to the left of an arrow
- **terminals** (analogous to input alphabets)
- one variable designated as the **start variable**
  - usually occurs on the LHS of the topmost rule

G_1 contains 3 rules; the variables are A and B, where A is the start variable; the terminals are 0, 1, #

Generation of strings from a grammar

We use a grammar to describe a language by generating each string of that language using the following procedure:

1. Write down the start variable (usually the variable on the LHS of the top rule)
2. Find a variable that is written down and a rule that starts with that variable. Replace the written down variable with the RHS of that rule.
3. Repeat step 2 until no variables remain.

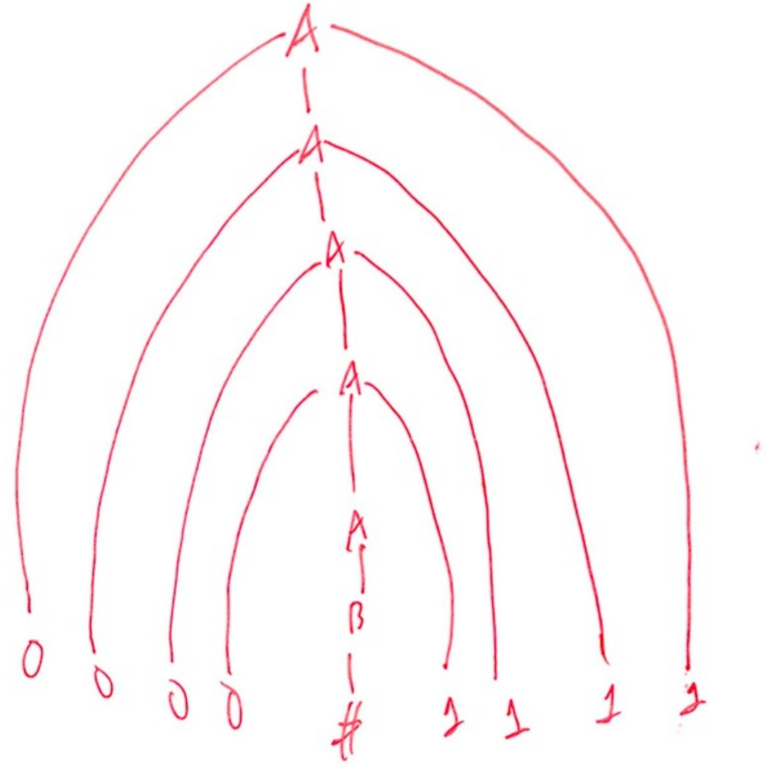
The sequence of substitutions to obtain a string is called a ***derivation***.

Example – for G_1

$$A \rightarrow 0A1$$
$$A \rightarrow B$$
$$B \rightarrow \#$$

$$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111$$
$$\Rightarrow 0000A1111 \Rightarrow 0000B1111$$
$$\Rightarrow 0000\#1111$$

parse tree



The language of a grammar

- All strings generated in this way constitute the language of the grammar
- We write $L(G_1)$ for the language of grammar G_1. Some experimentation with the grammar G_1 shows us that

  $L(G_1) = \{0^n \# 1^n \mid n \geq 0\}$
- Any language that can be generated by some CFG is called a *context-free language* (CFL)
- When presenting a CFG, we usually abbreviate several rules with the same left-hand variable into a single line using the symbol "|" as an "or"
  - E.g: $A \longrightarrow 0A1$ and $A \longrightarrow B$ is abbreviated into a single line as

    $A \longrightarrow 0A1 \mid B$

A second example of CFG – G_2

a fragment of the English language

$$
\begin{aligned}
\langle \text{SENTENCE} \rangle &\to \langle \text{NOUN-PHRASE} \rangle \langle \text{VERB-PHRASE} \rangle \\
\langle \text{NOUN-PHRASE} \rangle &\to \langle \text{CMPLX-NOUN} \rangle \mid \langle \text{CMPLX-NOUN} \rangle \langle \text{PREP-PHRASE} \rangle \\
\langle \text{VERB-PHRASE} \rangle &\to \langle \text{CMPLX-VERB} \rangle \mid \langle \text{CMPLX-VERB} \rangle \langle \text{PREP-PHRASE} \rangle \\
\langle \text{PREP-PHRASE} \rangle &\to \langle \text{PREP} \rangle \langle \text{CMPLX-NOUN} \rangle \\
\langle \text{CMPLX-NOUN} \rangle &\to \langle \text{ARTICLE} \rangle \langle \text{NOUN} \rangle \\
\langle \text{CMPLX-VERB} \rangle &\to \langle \text{VERB} \rangle \mid \langle \text{VERB} \rangle \langle \text{NOUN-PHRASE} \rangle \\
\langle \text{ARTICLE} \rangle &\to \texttt{a} \mid \texttt{the} \\
\langle \text{NOUN} \rangle &\to \texttt{boy} \mid \texttt{girl} \mid \texttt{flower} \\
\langle \text{VERB} \rangle &\to \texttt{touches} \mid \texttt{likes} \mid \texttt{sees} \\
\langle \text{PREP} \rangle &\to \texttt{with}
\end{aligned}
$$

G_2 has

- 10 variables (the capitalized grammatical terms written in brackets)
- 27 terminals (the standard English alphabet plus a space character)
- 18 rules

Derivation of the string "a boy sees" using G_2

$$
\begin{aligned}
\langle\text{SENTENCE}\rangle &\Rightarrow \langle\text{NOUN-PHRASE}\rangle\langle\text{VERB-PHRASE}\rangle \\
&\Rightarrow \langle\text{CMPLX-NOUN}\rangle\langle\text{VERB-PHRASE}\rangle \\
&\Rightarrow \langle\text{ARTICLE}\rangle\langle\text{NOUN}\rangle\langle\text{VERB-PHRASE}\rangle \\
&\Rightarrow \texttt{a}\ \langle\text{NOUN}\rangle\langle\text{VERB-PHRASE}\rangle \\
&\Rightarrow \texttt{a boy}\ \langle\text{VERB-PHRASE}\rangle \\
&\Rightarrow \texttt{a boy}\ \langle\text{CMPLX-VERB}\rangle \\
&\Rightarrow \texttt{a boy}\ \langle\text{VERB}\rangle \\
&\Rightarrow \texttt{a boy sees}
\end{aligned}
$$

Formal definition of a CFG

DEFINITION 2.2

A ***context-free grammar*** is a 4-tuple (V, Σ, R, S), where

1. V is a finite set called the ***variables***,
2. Σ is a finite set, disjoint from V, called the ***terminals***,
3. R is a finite set of ***rules***, with each rule being a variable and a string of variables and terminals, and
4. $S \in V$ is the start variable.

Some notations

- If u, v and w are strings of variables and terminals, and $A \longrightarrow w$ is a rule of the grammar,
  - We say that uAv ***yields*** uwv, and write it as $uAv \Rightarrow uwv$
  - We say that u ***drives*** v, and write it as $u \overset{*}{\Rightarrow} v$, if $u = v$ or if a sequence $u_1, u_2, \ldots, u_k$ exists for $k \geq 0$ and $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots. \Rightarrow u_k \Rightarrow v$
- The language of the grammar is $\{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$

More examples of CFGs -- G_3

Consider grammar $G_3 = (\{S\}, \{\mathtt{a}, \mathtt{b}\}, R, S)$. The set of rules, R, is

$$S \rightarrow \mathtt{a}S\mathtt{b} \mid SS \mid \varepsilon.$$

Which of the following strings belong to $L(G_3)$?

aabb
abab
abba
aababb
abbabb

Example – G_4

Consider grammar $G_4 = (V, \Sigma, R, \langle \text{EXPR} \rangle)$.

V is $\{\langle \text{EXPR} \rangle, \langle \text{TERM} \rangle, \langle \text{FACTOR} \rangle\}$ and Σ is $\{\mathtt{a}, +, \times, (,)\}$. The rules are

$$\begin{aligned}
\langle \text{EXPR} \rangle &\to \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \mid \langle \text{TERM} \rangle \\
\langle \text{TERM} \rangle &\to \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle \mid \langle \text{FACTOR} \rangle \\
\langle \text{FACTOR} \rangle &\to (\langle \text{EXPR} \rangle) \mid \mathtt{a}
\end{aligned}$$

Consider the strings $a+a \times a$ and $(a+a) \times a$

Can these strings be generated with grammar G_4?

Consider grammar $G_4 = (V, \Sigma, R, \langle \text{EXPR} \rangle)$.

V is $\{\langle \text{EXPR} \rangle, \langle \text{TERM} \rangle, \langle \text{FACTOR} \rangle\}$ and Σ is $\{\texttt{a}, \texttt{+}, \texttt{x}, \texttt{(}, \texttt{)}\}$. The rules are

$$\langle \text{EXPR} \rangle \to \langle \text{EXPR} \rangle \texttt{+} \langle \text{TERM} \rangle \mid \langle \text{TERM} \rangle$$
$$\langle \text{TERM} \rangle \to \langle \text{TERM} \rangle \texttt{x} \langle \text{FACTOR} \rangle \mid \langle \text{FACTOR} \rangle$$
$$\langle \text{FACTOR} \rangle \to \texttt{(} \langle \text{EXPR} \rangle \texttt{)} \mid \texttt{a}$$

Parse trees

Parse trees of the strings a+a × a and (a+a) × a