Theory of Computation

Chapter 02

Context- Free Languages

Introduction to the Theory of Computation, 3rd Ed, Michael Sipser
Introduction to Automata Theory Languages and Computation, 2nd, Hopcroft, Motwani, and Ullman
Last modified 10/10/2020

September, 2020
Zinia Sultana
Lecturer, CSE Dept, MIST

So far learned –

- ✓ *finite automata*
  - ✓ *Deterministic*
  - ✓ *Non-deterministic*
- ✓ *Regular expressions*

Non-regular Language

Let's consider the languages:

1. $L1 = \{w \mid w \text{ has even number of 0s}\}$

2. $L2 = \{0^n 1^n \mid n \geq 0\}$

Context Free Grammar

1. *A more powerful method*
2. *Recursive structure*
3. *Application - human language, compiler*

Context Free Grammar

The collection of languages associated with ***context-free grammars*** *are called the* ***context-free languages.***

Context Free Grammar

- *Grammar, G1.*

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

Context Free Grammar

- *Grammar, G1.*

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

- *substitution rules, also called productions.*
- *variables.*
- *terminals.*
- *start variable.*

Context Free Grammar

- *Derivation – The sequence of substitutions to obtain a string is called a **derivation.***
- *Parse tree – representing the same information of the derivation pictorially*

CFG - derivation

- *For example, grammar G1 generates the string 000#111*
- *A derivation of string 000#111 in grammar G1 is*

- *Grammar, G1.*

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

CFG – parse tree

- *Parse tree*

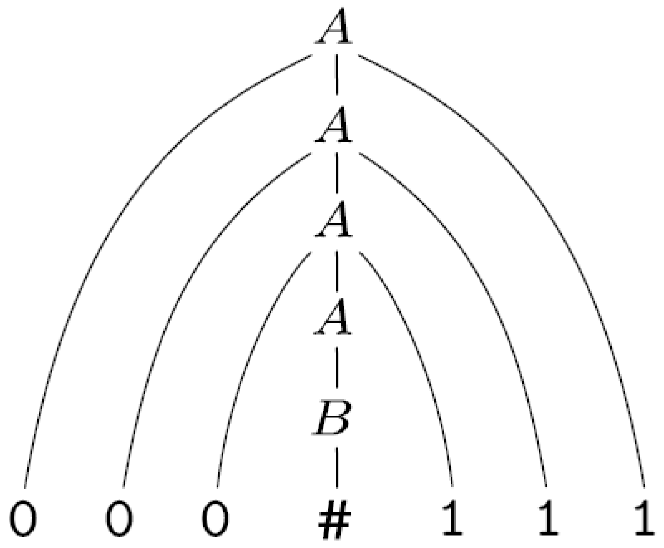


Fig: Parse tree for 000#111 in grammar G1

- *Grammar, G1.*

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

CFG – Example2

Fragment of the English language, Grammar G2

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

CFG – Example2

- Show derivation for the string – **the boy sees a flower** , using CFG G2

CFG – Fromal Definition

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.

CFG – Example2.3

- Grammar G3 = ({S}, {a, b},R, S).
- The set of rules, R, is

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

CFG – Example2.4

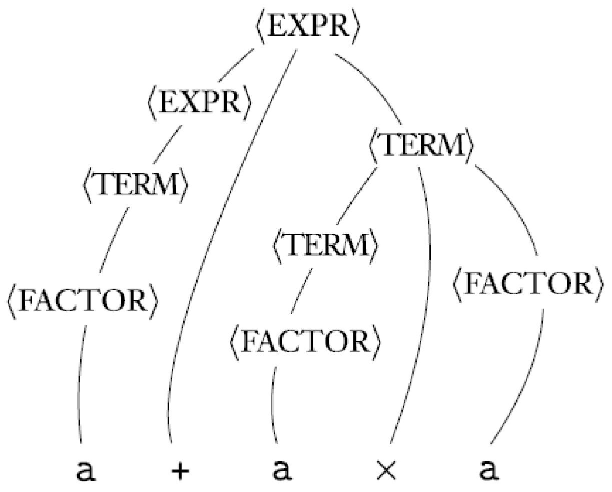
- Consider grammar G4 = (V, Σ, R, ⟨EXPR⟩)
- V is {⟨EXPR⟩, ⟨TERM⟩, ⟨FACTOR⟩} and
- Σ is {a, +, x, (,) }.
- The rules are

  ⟨EXPR⟩ → ⟨EXPR⟩ **+** ⟨TERM⟩ | ⟨TERM⟩
  ⟨TERM⟩ → ⟨TERM⟩ **x** ⟨FACTOR⟩ | ⟨FACTOR⟩
  ⟨FACTOR⟩ → (⟨EXPR⟩) | a

CFG – Example2.4

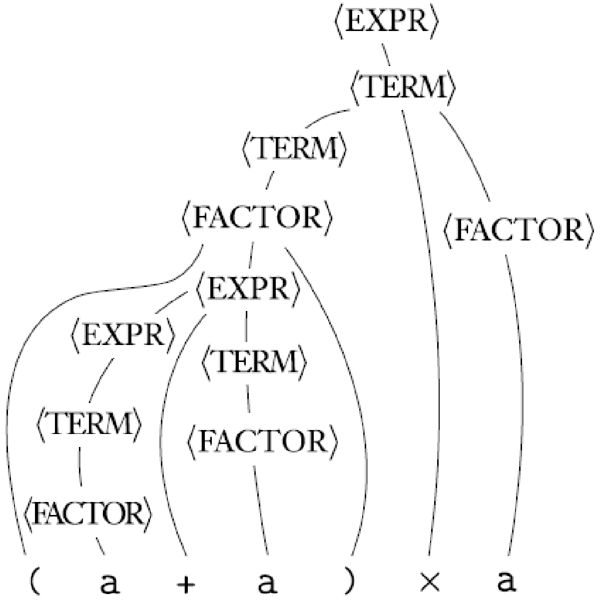
- Derivation and Parse trees for the strings a+axa

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



CFG – Example2.4

- Derivation and Parse trees for the strings **(a+a)xa**



E → E **+** T | T
T → T **x** F | F
F → (E) | a

Designing Context-Free Grammar

Designing Context-Free Grammars

Sipser, 2.1, p-106

- L1 = $\{0^n 1^n \mid n \geq 0\} \cup \{1^n 0^n \mid n \geq 0\}$

Designing Context-Free Grammars - 2

- L2 = $\{0^n 1^{2n} \mid n \geq 0\}$

Designing Context-Free Grammars -3

- L3 = $\{0^n 1^m \mid m, n \geq 0,\ \ 2n \leq m \leq 3n\}$

Designing Context-Free Grammars -4

- L4 = $\{0^n 1^m \mid m, n \geq 0,\ n \neq m\}$

Designing Context-Free Grammars -5

- L5 = $\{w \mid w \in \{a, b\}^* \mid n_a(w) = n_b(w)\}$

Designing Context-Free Grammars - 6

- L6 = $\{w \mid w \in \{0,1\}^* and\ of\ even\ length\}$

Designing Context-Free Grammars -7

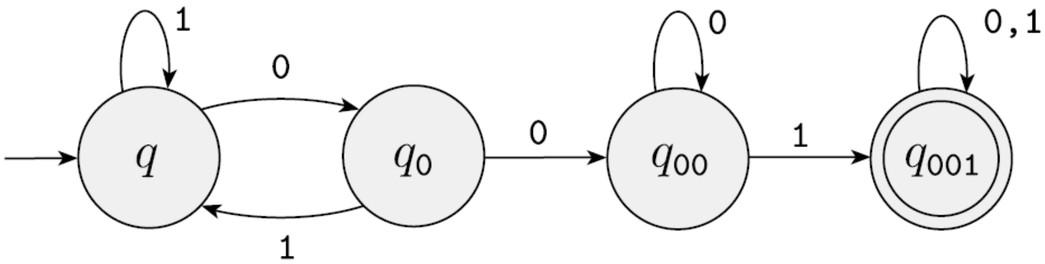
- L7 = $\{a^n b^m c^k \mid m, n, k \geq 0 \; and \; n = m + k\}$

Designing Context-Free Grammars -8

- L8 = $\{w \mid w \in \{0,1\}^* and\ w\ is\ palindrome\}$

Designing Context-Free Grammars –from DFA

- Accepts strings containing 001



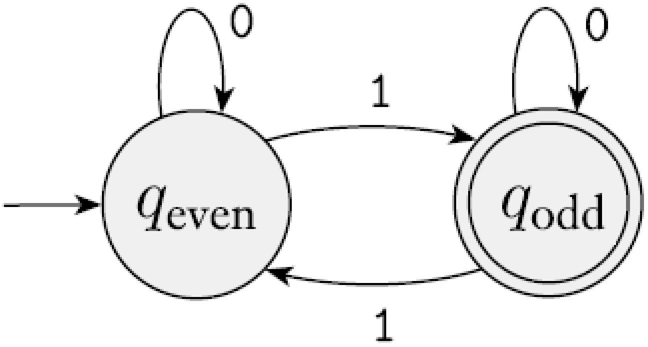
- CFG:

Designing Context-Free Grammars –from DFA

- Make a variable Ri for each state qi of the DFA
- Add the rule Ri → aRj to the CFG if δ(qi, a) = qj is a transition in the DFA
- Add the rule Ri → ε if qi is an accept state of the DFA
- Make R0 the start variable of the grammar, where q0 is the start state of the machine
- Verify that the resulting CFG generates the same language that the DFA recognizes.

Designing Context-Free Grammars –from DFA

- ❑ Try yourself: Desing a CFG for the following DFA



Ambiguity

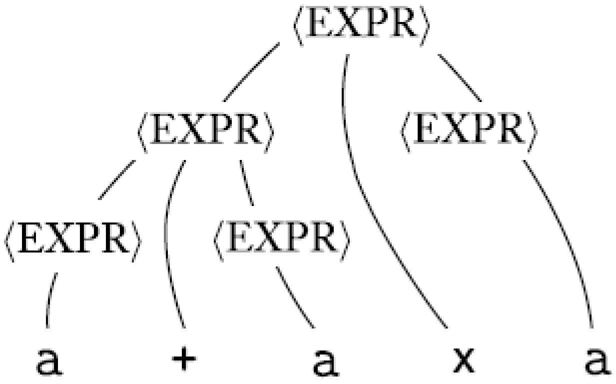
- Grammar, G5.

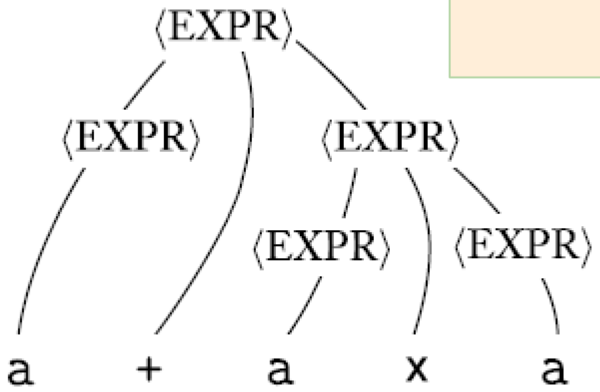
```
<EXPR> → <EXPR> + <EXPR>  |
         <EXPR> x <EXPR>  |
         (<EXPR>)         |
          a
```

Ambiguity

- Parse tree of string a + a x a

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





Leftmost and Rightmost Derivations

- We want to restrict the number of choices we have in deriving a string.
- It is often useful to require that at each step we replace the leftmost variable by one of its production bodies.
- Such a derivation is called a leftmost derivation.

Leftmost and Rightmost Derivations

- Similarly, it is possible to require that at each step the rightmost variable is replaced by one of its bodies.
- If so, we call the derivation rightmost.

Ambiguity

DEFINITION 2.7

A string w is derived ***ambiguously*** in context-free grammar G if it has two or more different leftmost derivations. Grammar G is ***ambiguous*** if it generates some string ambiguously.

Ambiguity

- Sometimes when we have an ambiguous grammar we can find an unambiguous grammar that generates the same language.
- Some context-free languages, however, can be generated only by ambiguous grammars.
- Such languages are called **inherently ambiguous**.

- The language $\{a^i b^j c^k \mid i, j, k \geq 0 \text{ } and \text{ } i = j \text{ } or \text{ } j = k\}$ is inherently ambiguous.

Chomsky Normal Form - CNF

DEFINITION 2.8

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

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

where a is any terminal and A, B, and C are any variables—except that B and C may not be the start variable. In addition, we permit the rule $S \to \varepsilon$, where S is the start variable.

Chomsky Normal Form - CNF

THEOREM 2.9

Any context-free language is generated by a context-free grammar in Chomsky normal form.

Chomsky Normal Form - CNF

- ❑ Add a new start variable.
- ❑ Eliminate all ε-***rules*** of the form a → ε.
- ❑ Eliminate all ***unit rules*** of the form a → b.
- ❑ Finally, convert the remaining rules into the proper form.

Chomsky Normal Form – CNF [Example-1]

- Grammar, G5.
  $S \rightarrow ASA \mid aB$
  $A \rightarrow B \mid S$
  $B \rightarrow b \mid \varepsilon$

Chomsky Normal Form – CNF [Example-2]

- Grammar,

  $A \rightarrow BAB \mid B \mid \varepsilon$

  $B \rightarrow 00 \mid \varepsilon$

For Practice

- Try to design CFG for all the regular language you have seen so far.
- Some language are listed below. Design CFG for each of the following Languages:
  1. $\{0^n 1^m$ and $n \geq m\}$
  2. $\{ (0 \cup 1)0^*(11)^*\}$
  3. $\{w \in \{0, 1\}^* \mid w$ contains at least three 1s $\}$
  4. $\{w \in \{0, 1\}^* \mid$ the length of w is odd and the middle symbol is 0 $\}$
  5. $\{w \in \{0, 1\}^* \mid w = w^R$ and length of w is even $\}$
  6. $\{ a^i b^j c^k \mid i, j, k \geq 0$, and $i = j$ or $i = k \}$
  7. $\{ a^i b^j c^k \mid i, j, k \geq 0$, and $i+j = k \}$
  8. { strings of properly balanced left and right bracket and Σ={ [,] } }
  9. $\{w \in \{0, 1\}^* \mid$ w starts and ends with the same symbol$\}$
  10. $\{w \in \{0, 1\}^* \mid$ length of w is odd$\}$
  11. $\{w \in \{a, b\}^* \mid$ string with twice as many a's as b's$\}$
  12. $\{ a^i b^j c^{2j} \mid i, j \geq 0\}$
  13. {All strings over Σ={a, b} with more a's than b's}
- Try converting the above designed CFG into CNF.

END