University of Victoria

CSC 320 - Spring 2023

Foundations of Computer Science

Tutorial 05

Teaching Team

Learning Outcomes:

- Become familiar with Context Free Grammars.
- Convert a Context Free Grammar into Chomsky Normal Form.
- Use a Pushdown Automata to describe a language.

Interesting Article:

"A Formalisation of the Cocke-Younger-Kasami Algorithm" [1]

February 14th, 2023

Question 02

Consider the following language over $\Sigma = \{0, 1\}$, find a set of rules that defines a CFG that recognizes the language:

$$L_2 = \{w \mid w \text{ starts and ends with the same symbol}\}$$

Intuitively, we can determine what is and isn't in the language:

NOT : 10, 100, 011, 0101

IN: ε, 101, 00, 11, 1, 0, 1001001

Thus, we can see an arising pattern within our accepted strings.

We have 3 cases:

$$0\Sigma^*0 \text{ or } 1\Sigma^*1 \text{ or } \varepsilon$$

where Σ^* is $\{0,1\}^*$ or $(0 \cup 1)^*$

Next, we start writing our CFG:

$G = (V, \Sigma, R, S)$, where

V : finite set of variables

Σ : finite set of terminals

R : finite set of rules

S : $S \in V$ is the start variable

$$\boxed{\begin{array}{l} S \longrightarrow 0X0 \mid 1X1 \mid \varepsilon \mid 0 \mid 1 \\ X \longrightarrow 0X \mid 1X \mid \varepsilon \end{array}}$$

This is sufficient since we are asked to find a set of rules, but it is good habit to include the formal definition.

Thus we obtain

$$G_2 = (\{S, X\}, \{0,1\}, R, S)$$

We can check our work to convince ourselfs that the grammar is correct.

S, 0X0, 00X0, 000X0, 0000 ✓

S, 1 ✓

S, 1X1, 11X1, 110X1, 1100X1, 11001 ✓

Question 03

Consider the following language over $\Sigma = \{0, 1\}$, find a set of rules that defines a CFG that recognizes the language:

$$L_3 = \emptyset$$

We note that th accepted language for L_3 is th empty set.

So, we can use th following rules to describe L_3:

$$\boxed{S \longrightarrow S}$$

Thus, nothing is accepted.

We would then have th following grammar..

$$G_3 = (\{S\}, \{0,1\}, R, S)$$

Question 05

Consider the following language over $\Sigma = \{0, 1\}$, find a set of rules that defines a Context Free Grammar (CFG) that recognizes the language:

$$L_5 = \{0^n 1^m \mid 2n \leq m \leq 3n\}$$

In plain english, we see that 1's are at least twice the amount of 0's, but not more than three times the amount of 0's.

Let us consider what is and isn't in the language:

NOT: Say $m = 4$ and $n = 6$ then $2(6) \leq 4 \leq 3(6)$
$12 \leq 4 \leq 18$, which does not satisfy the condition.

0000001111,

IN: if $n = 2$ then $2(2) \leq m \leq 3(2)$
$4 \leq m \leq 6$, so m can be 4, 5, or 6.

001111, 0011111, 00111111,

Next, we observe the pattern from our intuition of what is and isn't included. We want to either include two 1's or three 1's for each 0 we add to our string w.

$$\boxed{S \longrightarrow 0S11 \mid 0S111 \mid \varepsilon}$$

Explanation not entirely necessary, but the purpose of learning is to be able to convince the reader that your solution is correct and to explain your process.

Formally our CFG is of the form:

$$G_5 = (\{S\}, \{0,1\}, R, S).$$

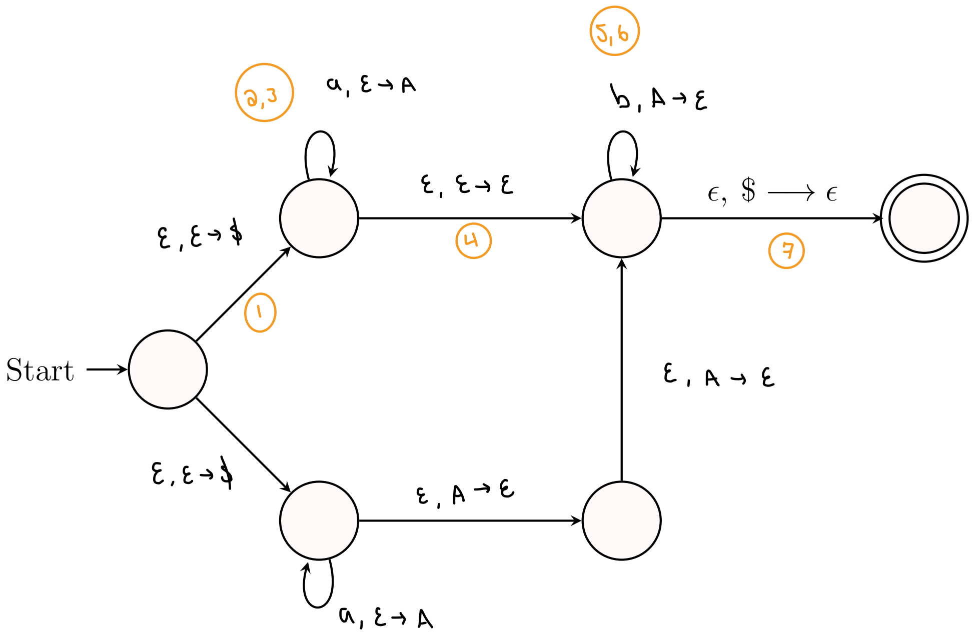
Note: The rule $S \to \varepsilon$, what would happen if we remove the rule? What would our string w look like?

Question: Could we correctly add a rule $S \longrightarrow 0$ and/or remove $S \longrightarrow \varepsilon$? Why couldn't we change the rules and would there be cases where the string create still satisfies the requirements?

Question 10

Complete the state diagram by adding missing transitions so that it describes a PDA that recognizes the following language:

$$L = \{a^m b^n \mid m, n \geq 0 \text{ and (either } m = n \text{ or } m = n + 2)\}$$



Example :

Let $m = 2$ and $n = 2$, our string of th form $a^m b^n$ is

aabb.

1

$

Stack

aabb

2

A

$

Stack

aabb

3

A

A

$

Stack

aabb

4

A

A

$

Stack

aabb

5

A

$

Stack

aabb

6

$

Stack

aabb

7

Stack

aabb

Thus, we see that "aabb" is accepted in our PDA.

Question 11

Derive or generate the string "aabaa" for the following grammar:

R:

$$S \longrightarrow aAS \mid aSS \mid \epsilon$$
$$A \longrightarrow SbA \mid ba$$

$G = (V, \Sigma, R, S)$

$V = \{S, A\}$

$\Sigma = \{a, b\}$

$R = R$

$S = S$

To begin, we will follow left-most derivation:

— At each step, replace left-most variable.

Let us start at S

$S \longrightarrow aSS \longrightarrow a\,\underline{aSS}\,S \longrightarrow a\,\underline{a}\,\underline{ba}\,\underline{S}\,S \longrightarrow a\,\underline{a}\,\underline{ba}\,\underline{aSS}\,S$

$\longrightarrow$ we replace SSS by ϵ respectively.

$\longrightarrow$ aabaa.

Thus, we can derive "aabaa" from our set of rules for grammar G. ▲

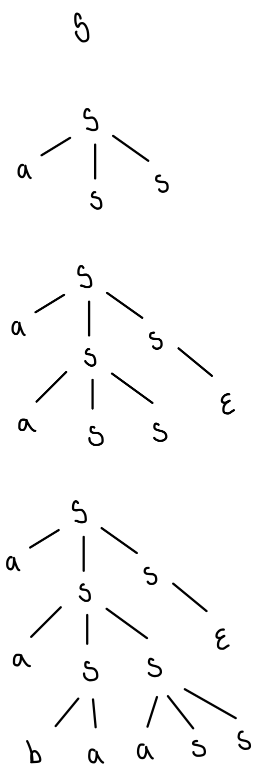
Additionally, we can represent the above as a parse tree for this derivation.

Step 1

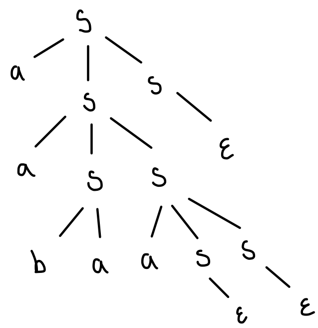
Step 2

Step 3

Step 4



Step 5



Step 6

Observe all the leaf nodes in order from left-to-right.

aabaa $\varepsilon\varepsilon\varepsilon$

= aabaa

▲

Question 12

Convert the following CFG into Chomsky Normal Form:

$$S \longrightarrow AAA \mid \epsilon$$
$$A \longrightarrow aa \mid Aa \mid \epsilon$$

Remember: To convert a CFG into Chomsky Normal Form (CNF) we can follow the following "algorithm" or steps.

1. Add new start variable $S_0 \longrightarrow S$.
2. Remove epsilon (ε).
3. Remove unit rules.
4. Add terminal rules
5. Clean up long rules.

Every rule of form:
① $A \longrightarrow BC$
② $A \longrightarrow a$
③ $S_0 \longrightarrow \varepsilon$ only!

We begin with step 1 of the algorithm:

① $S_0 \longrightarrow S$
$S \longrightarrow AAA \mid \varepsilon$
$A \longrightarrow aa \mid Aa \mid \varepsilon$

2a $S_0 \longrightarrow S \mid \varepsilon$
$S \longrightarrow AAA \mid \varepsilon$
$A \longrightarrow aa \mid Aa \mid \varepsilon$

Since we removed $S \longrightarrow \varepsilon$ in step 2a we will not re-add th transition.

2b $S_0 \longrightarrow S \mid \varepsilon$
$S \longrightarrow AAA \mid AA \mid A \mid \varepsilon$
$A \longrightarrow aa \mid Aa \mid a$

3a $S_0 \longrightarrow AAA \mid AA \mid A \mid \varepsilon$
$S \longrightarrow AAA \mid AA \mid A$
$A \longrightarrow aa \mid Aa \mid a$

3b $S_0 \longrightarrow AAA \mid AA \mid aa \mid Aa \mid a \mid \varepsilon$
$S \longrightarrow AAA \mid AA \mid aa \mid Aa \mid a$
$A \longrightarrow aa \mid Aa \mid a$

4a $S_0 \longrightarrow AAA \mid AA \mid aa \mid Aa \mid a \mid \varepsilon$
$S \longrightarrow AAA \mid AA \mid aa \mid Aa \mid a$
$A \longrightarrow aa \mid Aa \mid a$
$X \longrightarrow a$

4b $S_0 \longrightarrow AAA \mid AA \mid XX \mid AX \mid a \mid \varepsilon$
$S \longrightarrow AAA \mid AA \mid XX \mid AX \mid a$
$A \longrightarrow XX \mid AX \mid a$
$X \longrightarrow a$

5a $S_0 \longrightarrow AAA \mid AA \mid XX \mid AX \mid a \mid \varepsilon$
$S \longrightarrow AAA \mid AA \mid XX \mid AX \mid a$
$A \longrightarrow XX \mid AX \mid a$
$X \longrightarrow a$
$B \longrightarrow AA$

5b $S_0 \longrightarrow AB \mid AA \mid XX \mid AX \mid a \mid \varepsilon$
$S \longrightarrow AB \mid AA \mid XX \mid AX \mid a$
$A \longrightarrow XX \mid AX \mid a$
$X \longrightarrow a$
$B \longrightarrow AA$

And now we have the following rule set:

$$\boxed{\begin{aligned} S_0 &\longrightarrow AB \mid AA \mid XX \mid AX \mid a \mid \varepsilon \\ S &\longrightarrow AB \mid AA \mid XX \mid AX \mid a \\ A &\longrightarrow XX \mid AX \mid a \\ X &\longrightarrow a \\ B &\longrightarrow AA \end{aligned}}$$

Where $G = (V, \Sigma, R, S)$ is $V: \{S_0, S, A, X, B\}, \{a\}, R, S_0\}$

Resources

[1] M. Bortin, “A Formalisation of the Cocke-Younger-Kasami Algorithm,” *Archive of Formal Proofs*, 2016, `https://isa-afp.org/entries/CYK.html`, Formal proof development, ISSN: 2150-914x.