

Class: COMP 3040 - Foundation of Computer Science
 Name: DangNhi Ngoc Ngo
 Student ID: 01553277
 Homework 3

Homework 3: Chapter 2 - CFG, PDA

2.1 Recall the CFG G_4 that we gave in Example 2.4. For convenience, let's rename its variables with single letters as follows.

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

Give parse trees and derivations of each string.

a. a

The derivation is $E \Rightarrow T \Rightarrow F \Rightarrow a$.



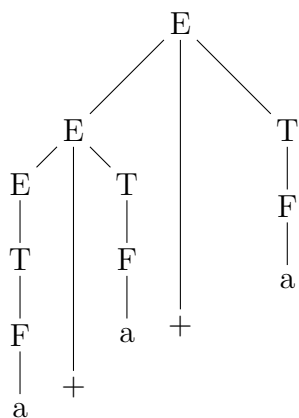
b. $a + a$

The derivation is $E \Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow a + T \Rightarrow a + F \Rightarrow a + a$.



c. $a + a + a$

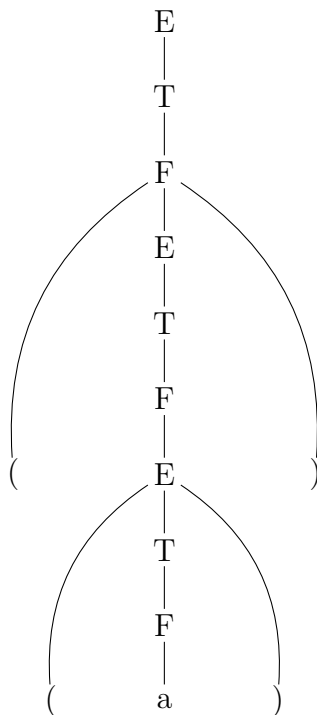
The derivation is $E \Rightarrow E + T \Rightarrow E + T + T \Rightarrow T + T + T \Rightarrow F + T + T \Rightarrow a + T + T \Rightarrow a + F + T \Rightarrow a + a + T \Rightarrow a + a + F \Rightarrow a + a + a$.



d. $((a))$

The derivation is

$E \Rightarrow T \Rightarrow F \Rightarrow (E) \Rightarrow (T) \Rightarrow (F) \Rightarrow ((E)) \Rightarrow ((T)) \Rightarrow ((F)) \Rightarrow ((a))$



2.4 Give context-free grammars that generate the following languages. In all parts, the alphabet Σ is $\{0, 1\}$.

b. $\{w \mid w \text{ starts and ends with the same symbol}\}$

$S \rightarrow 0R0 \mid 1R1 \mid \varepsilon$

$R \rightarrow 0R \mid 1R \mid \varepsilon$

c. $\{w \mid \text{the length of } w \text{ is odd}\}$
 $S \rightarrow 0 \mid 1 \mid 00S \mid 01S \mid 10S \mid 11S$

e. $\{w \mid w^R, \text{ that is, } w \text{ is a palindrome}\}$

$S \rightarrow B1B$

$B \rightarrow BB \mid 0B1 \mid 1B0 \mid 1 \mid \varepsilon$

S forces an extra 1, because B creates strings that satisfy to have at least as many 1s as 0s

f. The empty set

$S \rightarrow S$

2.5 Give informal descriptions and state diagrams of pushdown automate for the language in Exercise 2.4.

b. $\{w \mid w \text{ starts and ends with the same symbol}\}$

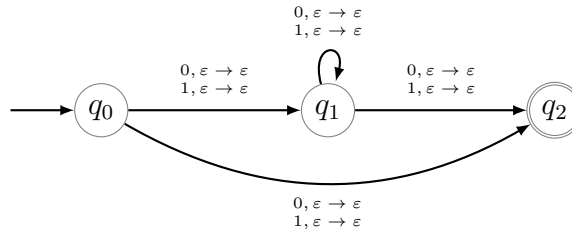
Informal Description:

If the string has only one symbol (by nondeterministically guessing), we can accept it without using the stack.

Otherwise, pushing the first symbol read into the stack, then reading every other symbol and guessing if that is the last symbol read.

We can accept in the case the last symbol read matches with the symbol in the stack and no more input. If not, we reject it.

State diagrams:



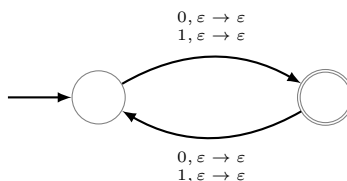
- c. $\{w \mid \text{the length of } w \text{ is odd}\}$

Informal Description:

In this case, we do not need to use the stack.

We can read the input and only accept if the length is odd, that is after the first symbol read or every other symbol read thereafter if it is the final symbol read.

State diagrams:



- e. $\{w \mid w^R, \text{ that is, } w \text{ is a palindrome}\}$

Informal Description:

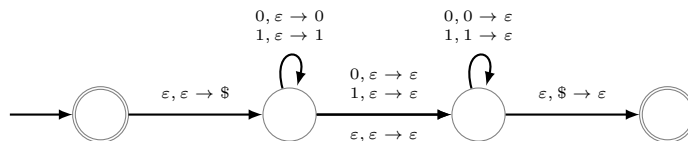
In this case, we use the stack to push the symbols read into it.

We can use the nondeterministically guess if the middle of the string has been reached, or if the next symbol read is the middle of the string and will not be put in the stack.

If they match the input symbols read, we will pop the symbols out of the stack.

In addition, if the stack empties as the input are finished, and the symbols popped are the same with the symbols that were pushed before, we will accept. Otherwise, we reject it.

State diagrams:



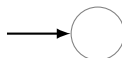
- f. The empty set

Informal Description:

The PDA consists of one state that does not accept.

The CFG can not accept any strings, including the empty because there is no derivations terminating.

State diagrams:



2.6 Give context-free grammars generating the following languages.

- b. The complement of the language $\{a^n b^n \mid n \geq 0\}$

Rewrite the complement as:

$$\{a^n b^m : n \neq m\} \cup \{(a \cup b)^* ba(a \cup b)^*\}$$

L_1 is the leftmost language.

L_2 is the rightmost language.

The context-free grammar that generate L_1 is:

$$\begin{aligned} S_1 &\rightarrow aS_1b \mid T \mid U \\ T &\rightarrow aT \mid a \\ U &\rightarrow Ub \mid b \end{aligned}$$

The context-free grammar that generate L_2 is:

$$\begin{aligned} S_2 &\rightarrow RbaR \\ R &\rightarrow a \mid b \mid \varepsilon \end{aligned}$$

So, the final context-free grammar G that generate $L_1 \cup L_2$ is:

$$\begin{aligned} S &\rightarrow T \mid aU \mid Vb \\ T &\rightarrow aT \mid Ta \mid bT \mid Tb \mid ba \\ U &\rightarrow aU \mid W \\ V &\rightarrow Vb \mid W \\ W &\rightarrow aWb \mid \varepsilon \end{aligned}$$

- d. $\{x_1 \# x_2 \# \dots \# x_k \mid k \geq 1, \text{ each } x_i \in \{a, b\}^*, \text{ and for some } i \text{ and } j, x_i = x_j^R\}$

Rewrite:

$$\dots \# \dots \# w \# \dots \# w^R \# \dots \# \dots$$

Let A be a non-terminal that generates $\{a, b, \#\}^*$, i.e.

$$A \rightarrow aA \mid bA \mid \#A \mid \varepsilon$$

Then, the part $w \# \dots \# w^R$ can be generated as follows.

$$B \rightarrow aBa \mid bBb \mid \# \mid \#A\#$$

The context-free grammars that generates is as follows:

$$\begin{aligned} S &\rightarrow A\#B\#A \mid B\#A \mid A\#B \mid B \\ B &\rightarrow aBa \mid bBb \mid \# \mid \#A\# \\ A &\rightarrow aA \mid bA \mid \#A \mid \varepsilon \end{aligned}$$

2.9 Give a context-free grammar that generates the language.

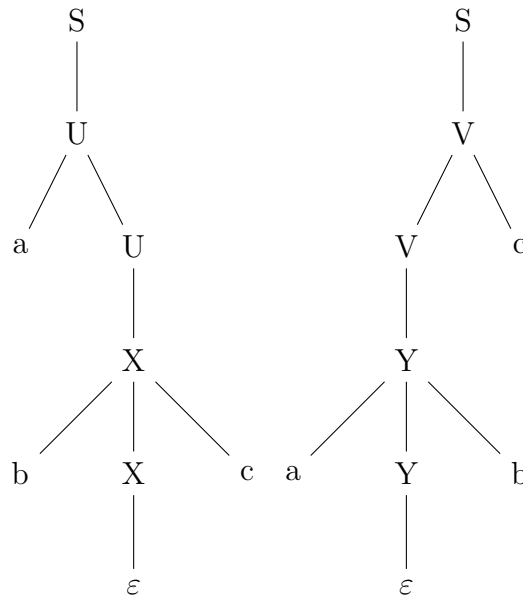
$$A = \{a^i b^j c^k \mid i = j \text{ or } j = k \text{ where } i, j, k \geq 0\}.$$

Is your grammar ambiguous? Why or why not?

Rules:

$$\begin{aligned} S &\rightarrow U \mid V \\ U &\rightarrow aU \mid X \\ V &\rightarrow Vc \mid Y \\ X &\rightarrow bXc \mid \varepsilon \\ Y &\rightarrow aYb \mid \varepsilon \end{aligned}$$

The grammar is ambiguous because we can have these two derivations.



2.10 Give an informal description of a pushdown automaton that recognizes the language A in Exercise 2.9.

Informal description of a pushdown automaton that recognizes the language A .

- Nondeterministically branch to step 2 or step 6.
- Read and push a's.
- Read b's, while popping a's out of the stack.
- Skip c's on input and *accept* in the case b's finish when stack is empty.
- Skip a's on input.
- Read and push b's.
- Read c's, while popping b's out of the stack.

- *Accept* if c 's finish when stack is empty.

2.11 Convert the CFG G_4 given in Exercise 2.1 to an equivalent PDA, using the procedure given in Theorem 2.20.

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

The formal definition of the equivalent PDA = $(Q, \Sigma, \Gamma, \delta, q_1, F)$ where"

$Q = \{q_1, q_2\}$; $\Sigma = \{+, \times, (,), a\}$

$\Gamma = \{E, T, F\} \cup \Sigma$; $F = \{q_2\}$.

The transition function $\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow P(Q \times \Gamma_\varepsilon)$

$$\delta(q, x, y) = \begin{cases} \{(q_2, \varepsilon)\} & \text{if } q = q_1, x = \varepsilon, y = \$ \\ \{(q_1, E + T), (q_1, T)\} & \text{if } q = q_1, x = \varepsilon, y = E \\ \{(q_1, T \times F), (q_1, F)\} & \text{if } q = q_1, x = \varepsilon, y = T \\ \{(q_1, (E)), (q_1, a)\} & \text{if } q = q_1, x = \varepsilon, y = F \\ \{(q_1, \varepsilon)\} & \text{if } q = q_1, x = y \end{cases}$$

2.12 Convert the CFG G given in Exercise 2.3 to an equivalent PDA, using the procedure given in Theorem 2.20.

(a) The formal definition of the equivalent PDA

$(Q, \Sigma, \Gamma, \delta, q_1, F)$

$Q = \{q_1, q_2\}$; $\Sigma = \{a, b\}$

$\Gamma = \{R, S, T, X\} \cup \Sigma$; $F = \{q_2\}$.

The transition function $\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow P(Q \times \Gamma_\varepsilon)$

$$\delta(q, x, y) = \begin{cases} \{(q_2, \varepsilon)\} & \text{if } q = q_1, x = \varepsilon, y = \$ \\ \{(q_1, XRX), (q_1, S)\} & \text{if } q = q_1, x = \varepsilon, y = R \\ \{(q_1, aTb), (q_1, bTa)\} & \text{if } q = q_1, x = \varepsilon, y = S \\ \{(q_1, (XTX)), (q_1, X), (q_1, \varepsilon)\} & \text{if } q = q_1, x = \varepsilon, y = T \\ \{(q_1, a), (q_1, b)\} & \text{if } q = q_1, x = \varepsilon, y = X \\ \{(q_1, \varepsilon)\} & \text{if } q = q_1, x = y \end{cases}$$

2.13 a. Describe $L(G)$ in English.

$L(G)$ is the set of strings that consists of zeros and #s that either have only 2 #s and any number of zeros or zeros separated by a single # and the number of zeros to the right of the # is double the number of zeros to the left.

b. Prove that $L(G)$ is not regular.

Intersecting $L(G)$ with the set of strings of the form $0^* \# 0^*$.

For example,

$$A = L(G) \cap 0^* \# 0^*$$

Assumed $L(G)$ is regular, then A must also be regular.

Let p be the pumping length for A .

Then the string $w = 0^p \# 0^{2p}$ is equal to xyz where the absolute value of $xy \leq p$, y is not the empty string, and $xy^i z$ is an element of $L(G)$ for any $i \geq 0$.

If x contains '#', then y is to the right of #. Pumping down would make the number of zeros on the right less than the number of zeros on the left. The resulting string does not belong to A .

If y contains '#', then pumping y would make the string contain no #'s. The resulting string does not belong to A .

If z contains '#'. then again pumping y makes the number of zeros on the right more than twice the amount of zeros on the left. The resulting string does not belong to A .

Conclusion: A is not regular because A failed the pumping lemma, which results in $L(G)$ is not regular, either.

2.14 Convert the following CFG into an equivalent CFG in Chomsky normal form, using the procedure given in Theorem 2.9.

$$\begin{aligned} A &\rightarrow BAB \mid B \mid \varepsilon \\ B &\rightarrow 00 \mid \varepsilon \end{aligned}$$

Solution:

$$\begin{aligned} S_0 &\rightarrow BA_1 \mid BA \mid AB \mid UU \mid BB \mid \varepsilon \\ A &\rightarrow BA_2 \mid BA \mid AB \mid UU \mid BB \\ B &\rightarrow UU \\ U &\rightarrow 0 \\ A_1 &\rightarrow AB \\ A_2 &\rightarrow AB \end{aligned}$$

2.26 Show that if G is a CFG in Chomsky normal form, then for any string $w \in L(G)$ of length $n \geq 1$, exactly $2n - 1$ steps are required for any derivation of w

Lemma: For all $n \geq 1$, if G derives a sentential form w in n steps, then $n = 2T(w) + NT(w) - 1$, where $NT(w)$ is the number of nonterminals in w and $T(w)$ is the number of terminals in w .

If w has no nonterminals, the result will be as expected.

Proof by induction:

$n = 1$, and deriving w must give one of two forms: $S \Rightarrow a$, where $S \rightarrow A$ is a rule of G and $w = a$ or $S \Rightarrow AB$ where $S \rightarrow AB$ is a rule of G and $w = AB$.

If the first form $2T(w) + NT(w) - 1 = 2 + 0 - 1 = n$, and in the second case, $2T(w) + NT(w) - 1 = 0 + 2 - 1 = n$.

Induction step:

$$\begin{aligned} n &= 2T(w) + NT(w) - 1 \\ n &= 2T(x) + NT(x) - 1 \text{ (because } S \Rightarrow^n x \Rightarrow w \text{)} \end{aligned}$$

Case 1: $NT(w) = NT(x) - 1$ and $T(w) = T(x) + 1$
 $2T(w) + NT(w) - 1 = 2(T(x) + 1) + (NT(x) - 1) - 1 = 2T(x) + NT(x) = n + 1$

Case 2: $NT(w) = NT(x) + 1$
and $T(w) = T(x)$ so $2T(w) + NT(w) - 1 = 2T(x) + NT(w) = n + 1$

In both case, $2T(w) + NT(w) - 1 = n + 1$

2.30 Use the pumping lemma to show that the following languages are not context free.

a. $\{ 0^n 1^n 0^n 1^n \mid n \geq 0 \}$

Assumed that L is context-free with L is a string consisting a zero followed by a one followed by a zero followed by a one.

Let p denote the pumping length of L .

Consider the string $0^p 1^p 0^p 1^p \in L$.

Write this string as $uvxyz$ where $|vxy| \leq p$ and $|vy| > 1$.

There are two cases:

Case 1: Either v or y contains more than one type of character. Pumping up once to uv^2xy^2z inserts an extra $0^i 1^j$ sequence into s -i.e, which results in containing at least 3 zero segments and 3 one segments. For example, if $v = 0011$, then pumping once would result in an additional 00 segment followed by an additional 11 segment. But, strings in L contain only four segments.

Case 2: Both v and y contain only one type of character. $vxy \leq p$, vxy contains letters from at most 2 segments of s . Pumping once to uv^2xy^2z increases the number of characters in one or two of the segments, but leaves the remaining

segments untouched. Therefore, the number of characters in each segment will be unbalanced.

- d. $\{t_1\#t_2\#\dots\#t_k \mid k \geq 2, \text{ each } t_i \in \{a, b\}^*, \text{ and } t_i t_j \text{ for some } i \neq j\}$

Assumed L is context-free

Let p denote its pumping length.

Consider $s = 0^p 1^p \# 0^p 1^p \in L$. Rewrite s as $uvxyz$ where $|vxy| \leq p$ and $|vy| > 1$ with the pumping lemma

vxy lies entirely on one side of the $\#$ symbol. Then, pumping once to uv^2xy^2z results in a string where $t_1 \neq t_2$, so $uv^2xy^2z \notin L$.

vxy contains the $\#$ symbol. If either v or y contains the $\#$ symbol, then we can pump s down to uv^0xy^0z , which will not contain the $\#$ symbol and hence will not be in L . Otherwise, the $\#$ symbol is contained in x , v is a substring of 1^p , and y is a substring of 0^p (since $|vxy| \leq p$). Pumping s down to uv^0xy^0z reduces either the number of ones in t_1 or the number of zeros in t_2 or both. As a result, $t_1 \neq t_2$ for uv^0xy^0z , so the string is not in L .

- 2.31** Let B be the language of all palindromes over $\{0, 1\}$ containing equal numbers of 0s and 1s. Show that B is not context free.

Solution:

Assumed that B is a context Free Language and obtains a contradiction.

Let p be the pumping constant of B that is guaranteed to exist by the pumping lemma. Select the string $s = 0^p 1^p 1^p 0^p$.

s is a member of B and of length at least p . We can show that no matter how we divide s into $uvxyz$, one of the conditions of the lemma is violated. Pumping lemma, $|vxy| \leq p$, we can only place vxy in the following ways:

- (1) vxy completely falls in the first 0^p (If we pump v and y , then the new string is no longer a palindrome, and the number of 0s will be greater than the number of 1s, which is a contradiction).
- (2) vxy falls between 0^p and 1^p (In this case, v will only contain 0s while y will only contain 1s. So if we pump s , the new string is no longer palindrome, which is a contradiction).
- (3) vxy completely falls in the first 1^p (Similar to (1), after pumping s , the number of 1s will be greater than the number of 0s, which is a contradiction).
- (4) vxy falls between 1^p and 1^p (After pumping s , the number of 1s will be greater than the number of 0s, which is a contradiction).
- (5) vxy completely falls in the second 1^p (This case is same with (3)).
- (6) vxy falls between 1^p and 0^p (Similar to (2), v will only contain 1s while y will only contain 0s. So if we pump s , the new string is no longer palindrome, which is a contradiction).

(7) vxy completely falls in the second 0^p (This case is same with (1)).

Conclusion: B is not context free.

2.32 Let $\Sigma = \{1, 2, 3, 4\}$ and $C = \{w \in \Sigma^* \mid \text{in } w, \text{ the number of 1s equals the number of 2s, and the number of 3s equals the number of 4s}\}$. Show that C is not context free.

Solution:

$C = \{w \in \{0, 1, 2, 3, 4\}^* \mid \text{in } w \text{ the number of 1s equals the number of 2s and the number of 3s equals the number of 4s}\}$

For a contradiction, assume that C is context free; thus, C has a pumping length p .

Take $s = 1^p 3^p 2^p 4^p \in C$ with $|s| > p$; thus, there exists $uvxyz$ such that (1)

$uv^i xy^i z \in C$ for all $i \geq 0$, (2) $|vy| > 0$ and (3) $|vxy| \leq p$.

Following cases prove that with any values of $uvxyz$, there is a contradiction:

Case 1: vxy contains a 1. Then $uv^2 xy^2 z \notin C$, since it will no longer have the same number of 1s and 2s. This is because from (3), vxy cannot contain any 2s.

Case 2: vxy contains a 2. Then $uv^2 xy^2 z \in C$, since it will no longer have the same number of 1s and 2s. This is because from (3), vxy cannot contain any 1s.

Case 3: vxy contains a 3. Then $uv^2 xy^2 z \in C$, since it will no longer have the same number of 3s and 4s. This is because from (3), vxy cannot contain any 4s.

Case 4: vxy contains a 4. Then $uv^2 xy^2 z \in C$, since it will no longer have the same number of 3s and 4s. This is because from (3), vxy cannot contain any 3s.

From (2) we see that these are all the cases. In each one we contradict (1). Therefore, C is not context free.

2.47 Let $\Sigma = \{0, 1\}$ and let B be the collection of strings that contain at least one 1 in their second half. In other words, $B = \{uv \mid u \in \Sigma^*, v \in \Sigma^*1\Sigma^* \text{ and } |u| \geq |v|\}$

a. Give a PDA that recognizes B .

The PDA M that recognizes B is $(Q, \Sigma, \Gamma, \delta, q_0, F)$, where:

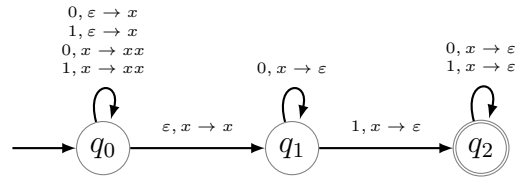
$$Q = \{q_0, q_1, q_2, q_3\},$$

$$\Sigma = \{0, 1\},$$

$$\Gamma = \{x\}$$

$$F = \{q_3\}$$

Transition function δ is presented as the following state diagram for the PDA:



b. CFG is:

$$S \rightarrow UV$$

$$U \rightarrow AB$$

$$V \rightarrow A1A \mid A1B \mid A1U \mid B1U \mid U1U$$

$$A \rightarrow 00^* \mid \varepsilon$$

$$W \rightarrow 11^* \mid \varepsilon$$

Explanation:

- The string S is the concatenation of U and V .
- The string U may consist of any number of 0's and 1's.
- The string V may consist of at least one 1 or only one 1.
- The string A may consist of at least one zero or more than one zeros
- The string B may consist of at least one 1 or more than one 1.