MTH-682 Automata
Assignment (2): Context-Free Languages

Mostafa Hassanein

20 November 2025

2.1

Recall the CFG G4 that we gave in Example 2.4. For convenience, let's rename its variables with single letters as follows.

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

Give parse trees and derivations for each string.

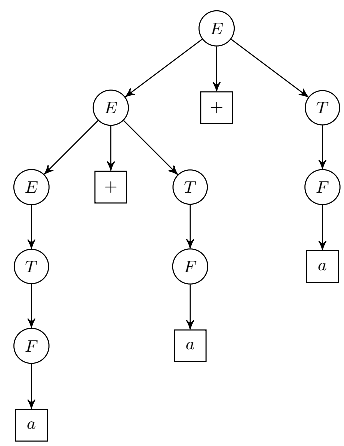
c. $a + a + a$

Solution:

Derivation:
We derive the string $a + a + a$ from the start symbol E as follows:

$$
\begin{aligned}
E &\Rightarrow E + T \\
&\Rightarrow E + T + T \\
&\Rightarrow T + T + T \\
&\Rightarrow F + T + T \\
&\Rightarrow a + T + T \\
&\Rightarrow a + F + T \\
&\Rightarrow a + a + T \\
&\Rightarrow a + a + F \\
&\Rightarrow a + a + a
\end{aligned}
$$

Parse Tree:



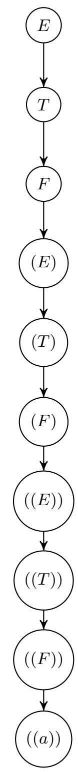
d. $((a))$

Solution:

Derivation:
We derive the string $((a))$ from the start symbol E as follows:

$$\begin{aligned} E &\Rightarrow T \\ &\Rightarrow F \\ &\Rightarrow (E) \\ &\Rightarrow (T) \\ &\Rightarrow (F) \\ &\Rightarrow ((E)) \\ &\Rightarrow ((T)) \\ &\Rightarrow ((F)) \\ &\Rightarrow ((a)) \end{aligned}$$

Parse Tree:



2.4

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

c. $\{w|$ the length of w is odd$\}$

<u>Solution:</u>

$$S \rightarrow ASA \mid A$$
$$A \rightarrow 0 \mid 1$$

f. The empty set

<u>Solution:</u>

$$S \rightarrow S$$

This language has a single rule that infinitely recurses and never terminates to any terminal symbols. Therefore, no words belong to this language.

2.5

Give informal descriptions and state diagrams of pushdown automata for the languages in Exercise 2.4.

c. $\{w|$ the length of w is odd$\}$

$$S \to ASA \mid A$$
$$A \to 0 \mid 1$$

<u>Solution:</u>

Informal Description:

- The PDA starts in state q_{start}.

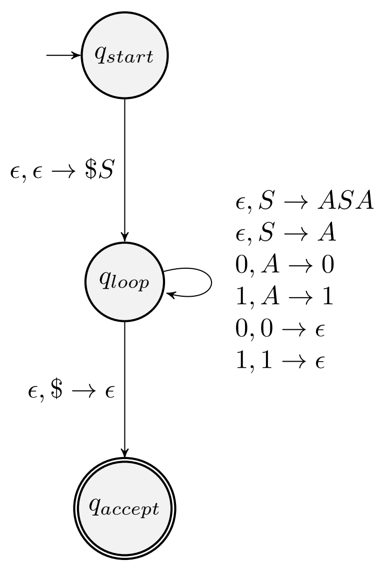
- It then pushes the symbols $\$S$ onto the stack and moves to state q_{loop}.

- On q_{loop}, it contains self transitions for all: 1. derivation rules, and 2. terminals.

- When all stack symbols have been consumed and only \$ remains, q_{loop} transitions to q_{accept}.

- The PDA accepts the input string if both: 1. all symbols in the input string have been consumed, 2. the PDA is in the q_{accept} state.

State Diagram:



f. The empty set

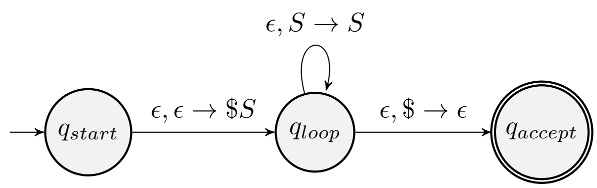
$$S \to S$$

Solution:

Informal Description:

- The PDA starts in state q_{start}.
- It then pushes the symbols $\$S$ onto the stack and moves to state q_{loop}.
- On q_{loop}, it contains an infinitely recursing self-transition that pops S and pushes it back again: $\epsilon, S \to S$.
- Thus, the symbol S is never consumed from the stack, and therefore the PDA never transitions to q_{accept}.

State Diagram:



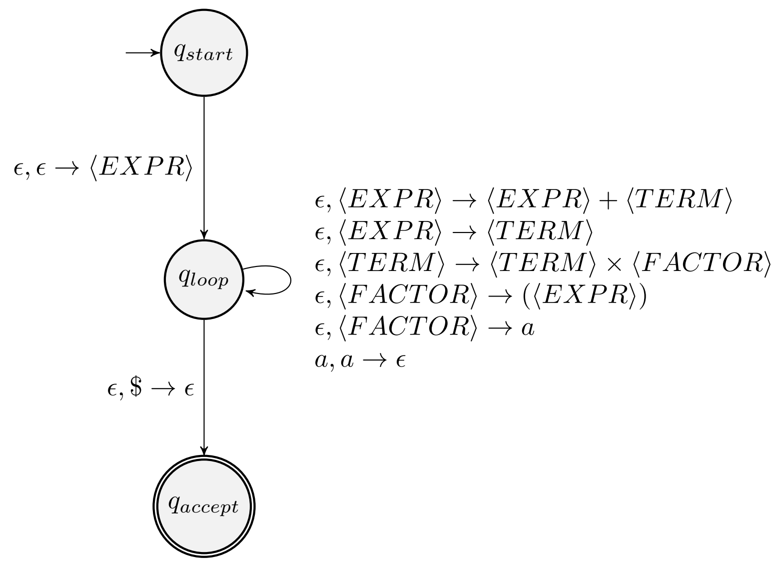
2.11

Convert the CFG G4 given in Exercise 2.1 to an equivalent PDA, using the procedure given in Theorem 2.20.

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

<u>Solution:</u>

State Diagram:



2.13

Let $G = (V, \Sigma, R, S)$ be the following grammar. $V = \{S, T, U\}$; $\Sigma = \{0, \#\}$; and R is the set of rules:

$$S \rightarrow TT|U$$
$$T \rightarrow 0T|T0|\#$$
$$U \rightarrow 0U00|\#$$

a. Describe $L(G)$ in English.
b. Prove that $L(G)$ is not regular.

Solution:

a.

It's much simpler to describe $L(G)$ mathematically:
$L(G) = L(H) \cup L(K)$, where:
$L(H) = \{w|w = 0^i\#o^{2i} \;\; \forall i \geq 1\}$
$L(K) = \{w|w = 0^i\#0^j\#0^k, \; \forall i, k \geq 1, j \geq 0\}$

b.

Proof.
Since regular languages are closed under the union operation and $L(G)$ is the union of 2 languages, then it is sufficient to show that any one of them is not regular.

We will show that $L(H)$ is not regular. We use the pumping lemma (for regular languages) to give a proof by contradiction.

Suppose for the sake of contradiction that $L(H)$ is regular. Since $L(H)$ is infinite, then there exists a pumping length p.

Take the string $w = 0^p\#0^{2p} \in L(H)$.
Since $|w| = 3p + 1 \geq p$, then w can be split as: $w = xyz$, where $|y| > 0$ and $|xy \leq p$.
$\implies y = 0^k, \; 0 < k \leq p$
$\implies w = 0^{p-k}0^k\#0^{2p}$
$\implies xy^0z \in L(H)$
$\implies 0^{p-k}\#0^{2p} \in L(H)$
$\implies 2(p-k) = 2p$
$\implies k = 0$
A contradiction.

Therefore, $L(H)$ is not regular.

$\square$

2.14

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

$$A \to BAB \mid B \mid \epsilon$$
$$B \to 00 \mid \epsilon$$

Solution:

Step 1: Add a new start variable S_0:

$$S_0 \to A$$
$$A \to BAB \mid B \mid \epsilon$$
$$B \to 00 \mid \epsilon$$

Step 2: Eliminate ϵ rules:

Step 2.1: Eliminate $B \to \epsilon$:

$$S_0 \to A$$
$$A \to BAB \mid B \mid \epsilon \mid BA \mid AB \; A$$
$$B \to 00$$

Step 2.2: Eliminate $A \to \epsilon$:

$$S_0 \to A \mid \epsilon$$
$$A \to BAB \mid B \mid BA \mid AB \; A$$
$$B \to 00$$

Step 3: Eliminate unit rules:

Step 3.1: Eliminate $A \to A$:

$$S_0 \to A \mid \epsilon$$
$$A \to BAB \mid B \mid BA \mid AB$$
$$B \to 00$$

Step 3.2: Eliminate $S_0 \to A$:

$$S_0 \to BAB \mid B \mid BA \mid AB \mid \epsilon$$
$$A \to BAB \mid B \mid BA \mid AB$$
$$B \to 00$$

Step 3.3: Eliminate $A \to B$ and $S_0 \to B$:

$$
\begin{aligned}
S_0 &\to BAB \mid 00 \mid BA \mid AB \mid \epsilon \\
A &\to BAB \mid 00 \mid BA \mid AB \\
B &\to 00
\end{aligned}
$$

Step 4: Introduce new variables to put the rules into proper Chomsky Normal Form form

$$
\begin{aligned}
S_0 &\to BA_1 \mid 00 \mid BA \mid AB \mid \epsilon \\
A &\to BA_1 \mid 00 \mid BA \mid AB \\
A_1 &\to AB \\
B &\to 00
\end{aligned}
$$

2.23

Let $D = \{xy|\, x, y \in \{0,1\}^*$ and $|x| = |y|$ but $x \neq y\}$.
Show that D is a context-free language.

Solution:

Proof Idea: Observe that in the language D, the left and right halves (x and y) differ at some position: either x has a 0 where y has a 1, or x has a 1 where y has a 0. Construct a grammar that is the union of these 2 cases.

Proof.
We prove that D is context-free by defining a CFG for D.

Define a CFG that recognizes the union of these 2 languages:
1. S_{01}: The language whose words contain a 0 on the left half mismathed with a 1 in the corresponding position on the right half.
2. S_{10}: The language whose words contain a 1 on the left half mismathed with a 0 in the corresponding position on the right half.

Define:

$$V = \{S, S_{01}, S_{10}, M, C\}$$
$$\Sigma = \{0, 1\}$$
$$R = \left\{\begin{array}{l} S \to S_{01} \mid S_{10}, \\ S_{01} \to 0M1 \mid 0S_{01}0 \mid 0S_{01}1 \mid 1S_{01}0 \mid 1S_{01}1, \\ S_{10} \to 1M0 \mid 0S_{10}0 \mid 0S_{10}1 \mid 1S_{10}0 \mid 1S_{10}1, \\ M \to CC \mid \epsilon, \\ C \to 0 \mid 1 \end{array}\right\}$$

The grammar $G = (V, \Sigma, R, S)$ is a CFG that recognizes the language D, therefore D is a context-free language.

□

2.24

Let $E = \{a^i b^j \,|\, i \neq j \text{ and } 2i \neq j\}$.
Show that E is a context-free language.

<u>Solution:</u>

Proof Idea: Show that E is defined as the union of 2 context-free languages: F and H. Therefore, E must be a context-free language.

Proof.
Define:

$$F = \{a^i b^j \,|\, i \neq j\}$$
$$H = \{a^i b^j \,|\, i \neq 2j\}$$

This implies:

$$\begin{aligned} E &= F \cap H \\ &= \bar{F} \cup \bar{H} \qquad \text{(By DeMorgan's law)} \end{aligned}$$

Where:

$$\bar{F} = \{a^i b^j \,|\, i = j\}$$
$$\bar{H} = \{a^i b^j \,|\, i = 2j\}$$

But clearly $\bar{F}$ and $\bar{H}$ are both <u>deterministic</u> context-free languages (DCFLs).

Using the property of closure under complementation for DCFL, we get that both F and H are DCFLs.

Finally, using the property of closure under union for context-free languages, we conclude that E is a DFCL.

□

2.26

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

Solution:

Proof. By Strong Induction

Base Case ($n = 1$):

In this case the string w is a single terminal symbol.

In Chomsky Normal Form, such words are derived directly from the start variable with no extra derivations needed.

Thus, it takes a single derivation. This matches the formula $2(1) - 1 = 1$.

Inductive Step:

By the strong induction hypotheis, assume that the formula holds for all strings in $L(G)$ of length $1 \leq n \leq k$.

Let $w \in L(G)$ where $|w| = k + 1$.

$\implies$ $|w| \geq 2$

$\implies$ The first derivation rule of w has the form: $S \to AB$, where S is the start variable; A, B are non-start variables. Additionally, $w = ps$, where $A \Rightarrow^* p$ and $B \Rightarrow^* s$ and $p, s \neq \epsilon$

$\implies$

$$\begin{aligned}
DerivationSteps(w) &= 1 + DerivationSteps(p) + DerivationSteps(s) \\
&= 1 + DerivationSteps(p) + DerivationSteps(k + 1 - p) \\
&= 1 + [2p - 1] + [2\,(k + 1 - p) - 1] \\
&= 1 + [2p - 1] + [2\,(k + 1 - p) - 1] \\
&= 1 + 2k \\
&= 2(k + 1) - 1.
\end{aligned}$$

Therefore, the formula holds for $k + 1$.

□

2.27

Let $G = (V, \Sigma, R, \langle STMT \rangle)$ be the following grammar.

$$\begin{aligned}
\langle STMT \rangle &\to \langle ASSIGN \rangle \,|\, \langle IF-THEN \rangle \,|\, \langle IF-THEN-ELSE \rangle \\
\langle IF-THEN \rangle &\to \text{if condition then} \, \langle STMT \rangle \\
\langle IF-THEN-ELSE \rangle &\to \text{if condition then} \, \langle STMT \rangle \, \text{else} \, \langle STMT \rangle \\
\langle ASSIGN \rangle &\to a := 1
\end{aligned}$$

$\Sigma = \{if, \, condition, \, then, \, else, \, a := 1\}$
$V = \{< STMT >, < IF-THEN >, < IF-ELSE-THEN >, < ASSIGN >\}$

G is a natural-looking grammar for a fragment of a programming language, but G is ambiguous.

a. Show that G is ambiguous.
b. Give a new unambiguous grammar for the same language.

a.

<u>Solution:</u>

We prove this by showing a counter example.
The following word in G has 2 different parse trees:

$w =$ if condition then if condition then a:=1 else a:=1

In one parse tree the else belongs the outer if, and in another parse tree it belongs to the inner if. Therefore, this grammar is ambiguous.
This problem is called: the dangling else ambiguity.

b.

<u>Solution:</u>

$$\begin{aligned}
\langle STMT \rangle &\to \langle MATCHED \rangle \;|\; \langle UNMATCHED \rangle \\
\langle MATCHED \rangle &\to \langle ASSIGN \rangle \;|\, \text{if condition then } \; \langle MATCHED \rangle \; \text{ else } \; \langle MATCHED \rangle \\
\langle UNMATCHED \rangle &\to \text{if condition then } \; \langle STMT \rangle \;|\, \text{if condition then } \; \langle MATCHED \rangle \, else \, \langle UNMATCHED \rangle \\
\langle ASSIGN \rangle &\to a := 1
\end{aligned}$$

2.30

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

d.

$$L = \Big\{t1\#t2\#\dots\#t_k |\, k \geq 2, \forall i\, t_i \in \{a,b\}^* \ \wedge \ \exists i,j : (t_i = t_j \ \wedge \ i \neq j)\Big\}$$

Solution:

Proof. We use the pumping lemma for context-free languages to show that L is not context-free.

Assume for the sake of contradiction that L is a CFL. Let p be the pumping length given by the pumping lemma.

Consider the string $s = a^p b^p \# a^p b^p$.

Clearly, $s \in L$ because it has $k = 2$ parts, and $t_1 = a^p b^p = t_2$.

We show that for any decomposition $s = uvxyz$ satisfying $|vxy| \leq p$ and $|vy| > 0$, there exists an $i \geq 0$ such that $uv^i xy^i z \notin L$.

Case 1: The substring vxy does not contain the symbol $\#$.
In this case, vxy must be entirely contained within either the first part (t_1) or the second part (t_2). This is because $|vxy| \leq p$, so it cannot span across the $\#$ without containing it.
Without loss of generality, assume vxy is in the first part. Pump up with $i = 2$. The resulting string is $s' = t_1' \# t_2$. Since $|vy| > 0$, we have $|t_1'| \neq |t_1|$. Since $|t_1| = |t_2|$, it follows that $|t_1'| \neq |t_2|$, and thus $t_1' \neq t_2$. Since there are only 2 parts ($k = 2$), and they are not equal, the condition $\exists i,j : (t_i = t_j \wedge i \neq j)$ is not satisfied. Thus $s' \notin L$.

Case 2: The substring vxy contains the symbol $\#$.
Since $|vxy| \leq p$, vxy must be a substring of the segment $b^p \# a^p$ (the b's from t_1 and a's from t_2).
Subcase 2a: The symbol $\#$ is in x. Then v consists of b's from t_1, and y consists of a's from t_2. Let $v = b^m$ and $y = a^n$, with $m + n > 0$. Pump up with $i = 2$. The resulting string is $s' = a^p b^{p+m} \# a^{p+n} b^p$. For s' to be in L, we must have $t_1' = t_2'$, i.e., $a^p b^{p+m} = a^{p+n} b^p$. Comparing the number of a's, we need $p = p + n \implies n = 0$. Comparing the number of b's, we need $p + m = p \implies m = 0$. This contradicts $|vy| > 0$. Thus $t_1' \neq t_2'$, so $s' \notin L$.

Subcase 2b: The symbol $\#$ is in v or y. Pump down with $i = 0$. This removes the $\#$ from the string. The resulting string has no $\#$ symbol, meaning it consists of a single part ($k = 1$). However, the definition of L requires $k \geq 2$. Thus the pumped string is not in L.

In all cases, we found a pumped string that is not in L. This contradicts the pumping lemma. Therefore, L is not context-free.

$\square$

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.

<u>Solution:</u>

Proof. We use the pumping lemma for context-free languages to show that B is not context-free.

Assume for the sake of contradiction that B is a CFL. Let p be the pumping length given by the pumping lemma.

Consider the string $s = 0^p 1^{2p} 0^p$.

First, we verify that $s \in B$:

- s reads the same forwards and backwards, so it is a palindrome.
- The number of 0s is $p + p = 2p$. The number of 1s is $2p$. Thus, it has equal numbers of 0s and 1s.

Also, $|s| = 4p \geq p$.

We show that for any decomposition $s = uvxyz$ satisfying $|vxy| \leq p$ and $|vy| > 0$, the pumped string is not in B.

Since $|vxy| \leq p$, the substring vxy cannot span across the block of 1s to touch both blocks of 0s (the distance between the two blocks of 0s is $2p > p$).

We consider the cases for the position of vxy:

Case 1: vxy is contained in the first 0^p or overlaps the first 0^p and 1^{2p}.

Pump up with $i = 2$ (uv^2xy^2z). This increases the number of 0s in the first block (or 0s in first block and 1s in middle), but the last block of 0s remains 0^p. The resulting string is not a palindrome because it starts with more 0s than it ends with (or has mismatched structure). Hence, it is not in B.

Case 2: vxy is contained entirely within 1^{2p}.

Pump up with $i = 2$. This increases the number of 1s, but the number of 0s remains $2p$. The number of 1s becomes $2p + |vy| > 2p$. The condition that the number of 0s equals the number of 1s is violated. Hence, it is not in B.

Case 3: vxy overlaps 1^{2p} and the last 0^p, or is contained in the last 0^p.

Pump up with $i = 2$. This increases the number of 0s in the last block, but the first block remains 0^p. The resulting string is not a palindrome. Hence, it is not in B.

In all cases, the pumped string is not in B. This contradicts the pumping lemma. Therefore, B is not context-free.

□