Homework 2 Questions

Problem 1: (20 points) Use the pumping lemma to show that the following languages are not regular.

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

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

If we assume that A_1 is regular, the Pumping Lemma definition tells us that any string s in A_1 can be 'pumped' at least a 'pumping length' of p , then divided into three pieces $s = xyz$. For it to be regular, it must satisfy the following conditions:

(1) $\forall i \geq 0, xy^i z \in A$
(2) $|y| \geq 0$
(3) $|xy| \leq p$

We can test these conditions by doing the following:

- Assume A is regular
- Let p be the pumping length
- Choose a string s in language A_1 to test: $s = 0^P 1^P 2^P$
  - As we assume A_1 is regular, s is a member of A_1
  - Because $|s| = 3p$, and $3p \geq p$, then $|s| \geq p$
  - Since both statements above are true, the Pumping Lemma definition tells us that s can be split into three pieces $s = xyz$

- For conditions (2) $|y| \geq 0$ and (3) $|xy| \leq p$ to be met, piece y must contain only 0's
  - For example, consider $p = 3$

$$0^P 1^P 2^P = 0^3 1^3 2^3$$

$$0^3 1^3 2^3 = 000111222$$

  - For $|xy| \leq 3, y$ can only contain 0's

$$\underset{x}{0}\,\underset{y}{00}\,\underset{z}{111000}$$

- According to condition (1) $\forall i \geq 0, xy^i z \in A$
  - assume $i = 2$

$$xy^i z \in A$$
$$xy^2 z \in A$$

  - Visual Representation

$$\underset{x}{0}\,\underset{y}{00}\,\underset{y}{00}\,\underset{z}{111000}$$

- $xy^2z \notin A$
  - However, there is no possible division of 00000111000 that will result in the required $0^n1^n2^n$ format. This is evident in the visual representation above as it has the format $0^5 1^3 2^3$ and $5 \neq 3$. As i increases, this also remains true since additional zeros are placed at the beginning of the string, creating a further imbalance between 0^n and 1^n2^n (which should be identical n's). Therefore, language A_1 does not meet condition (1) for Pumping Lemma.

Since not all pumping lemma conditions are met, the assumption that A_1 is regular is contradicted and proves that A_1 is not regular.

Q.E.D

b. $A_{2=}\{www|\ w \in \{a,b\}^*\}$

$$\boxed{A_2 = \{www\ |w\ \in \{a,b\}^*\}}$$

If we assume that A_2 is regular, the Pumping Lemma definition tells us that any string s in A_2 can be 'pumped' at least a 'pumping length' of p , then divided into three pieces $s = xyz$. For it to be regular, it must satisfy the following conditions:

(1) $\forall i \geq 0, xy^iz \in A$
(2) $|y| \geq 0$
(3) $|xy| \leq p$

We can test these conditions by doing the following:

- Assume A_2 is regular
- Let p be the pumping length
- Choose a string s in language A_2 to test: $s = a^Pba^Pba^Pb$
  - As we assume A_2 is regular, s is a member of A_2
  - Because $|s| = 3p + 3$, and $3p + 3 \geq p$, then $|s| \geq p$
  - Since both statements above are true, the Pumping Lemma definition tells us that s can be split into three pieces $s = xyz$

- For conditions (2) $|y| \geq 0$ and (3) $|xy| \leq p$ to be met, piece y must contain only a's
  - For example, consider $p = 3$

$$a^Pba^Pba^Pb = a^3ba^3ba^3b$$

$$a^3ba^3ba^3b = aaabaaabaaab$$

  - For $|xy| \leq 3$, y can only contain a's

$$aaabaaabaaab$$
$$x\ y\qquad z$$

- According to condition (1) $\forall i \geq 0, xy^iz \in A$
  - assume $i = 2$

$$xy^iz \in A$$
$$xy^2z \in A$$

- Visual Representation

$$aaaaabaaabaaab$$
$$x\ y\ y\ \quad z$$

- $xy^2z \notin A$
  - However, there is no possible division of $aaaaabaaabaaab$ that will result in the required www format. This is evident in the visual representation above as it has the format $a^5ba^3ba^3b$ and $5 \neq 3$. As i increases, this also remains true since additional a's are placed at the beginning of the string, creating a further imbalance between the first $w = a^Pb$ and the last two $ww = a^3ba^3b$ (which should be identical w's). Therefore, language A_2 does not meet condition (1) for Pumping Lemma.

Since not all pumping lemma conditions are met, the assumption that A_2 is regular is contradicted and proves that A_2 is not regular.

Q.E.D

Problem 2: (20 points) Give context-free grammars that generate the following languages. In all parts $\Sigma = \{0,1\}$

1. {w | w starts and ends with the same symbol}

S	→	**0 | 1 | 1*T*1 | 0*T*0**
T	→	$0T \mid 1T \mid \varepsilon$

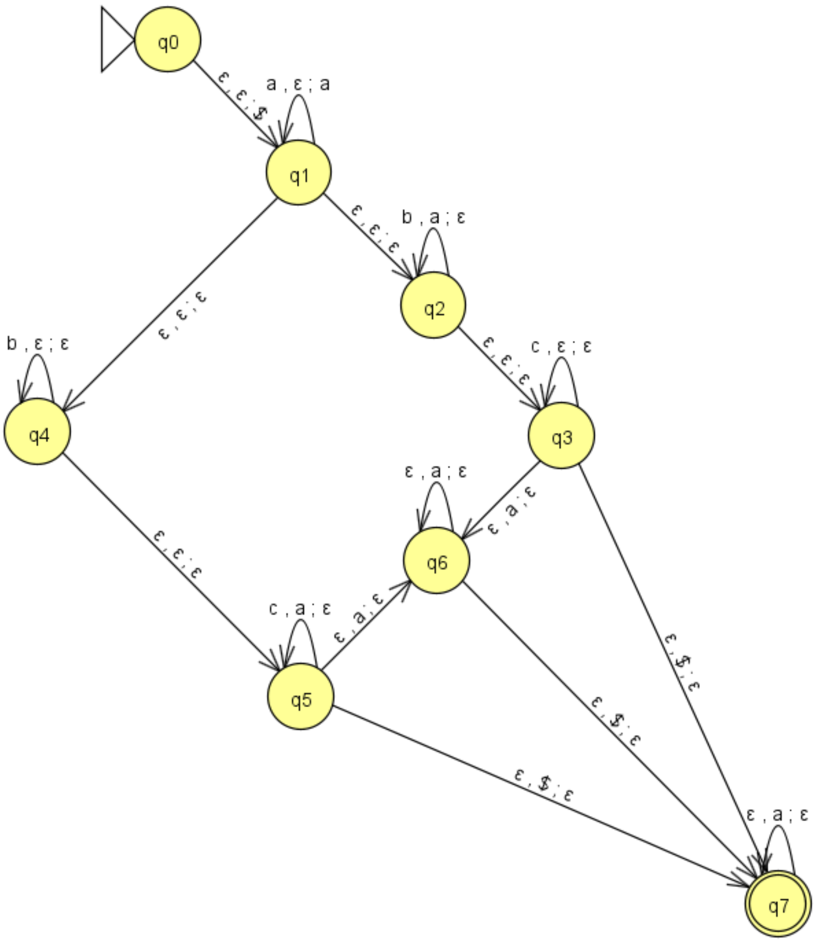
2. {w | the length of w is odd|

S	→	**0 | 1 | *SSS***

Problem 3: (10 points) Please design and draw a PDA which recognizes $\{a^ib^jc^k \mid i,j,k \geq 0 \ and \ i \geq j \ or \ i \geq k\}$

PDA 6-Tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$

$Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$	Set of States
$\Sigma = \{a, b, c\}$	Input Alphabet
$\Gamma = \{\$, a\}$	Stack Alphabet
q_0	Start State
$F = \{q_7\}$	Set of Accept States



Input	Result
a	Accept
b	Accept
c	Accept
ab	Accept
aab	Accept
ac	Accept
aac	Accept
abc	Accept
aabbcc	Accept
aabc	Accept
aaabbc	Accept
aaabcc	Accept
aabbbc	Accept
abb	Accept
acc	Accept
	Accept
bc	Reject
bccc	Reject
abbcc	Reject
aabbbccc	Reject

Problem 4: (10 points) Convert the following CFG into an equivalent CFG in Chomsky normal form, using the procedure given in Theorem 2.9.

A	$\rightarrow$	BAB | B | ε
B	$\rightarrow$	00 | ε

1. Create new start variable

S	$\rightarrow$	A
A	$\rightarrow$	BAB | B | ε
B	$\rightarrow$	00 | ε

2. Eliminate all ε-rules of form $A \rightarrow \varepsilon$

S	$\rightarrow$	A | ε
A	$\rightarrow$	BAB | B | A | AB | BA | BB
B	$\rightarrow$	00

ε may appear **only** on the right-hand side of start variable S

3. Eliminate all unit rules of form $A \to B$

S	$\to$	$A \mid \varepsilon$
A	$\to$	$BAB \mid \cancel{B} \mid 00 \mid A \mid AB \mid BA \mid BB$
B	$\to$	00

remove $A \to B$

S	$\to$	$A \mid \varepsilon$
A	$\to$	$BAB \mid 00 \mid \cancel{A} \mid AB \mid BA \mid BB$
B	$\to$	00

remove $A \to A$

S	$\to$	$\cancel{A} \mid BAB \mid 00 \mid AB \mid BA \mid BB \mid \varepsilon$
A	$\to$	$BAB \mid 00 \mid AB \mid BA \mid BB$
B	$\to$	00

remove $S \to A$

4. Replace each rule $A \to \mathrm{u}_1 u_2 \cdots u_k$, where $k \geq 3$ and u_i is a variable or terminal symbol

S	$\to$	$\cancel{BAB} \mid BA_1 \mid 00 \mid AB \mid BA \mid BB \mid \varepsilon$
A	$\to$	$\cancel{BAB} \mid BA_1 \mid 00 \mid AB \mid BA \mid BB$
B	$\to$	00
A_1	$\to$	AB

replace BAB

S	$\to$	$BA_1 \mid 00 \mid AB \mid BA \mid BB \mid \varepsilon$
A	$\to$	$BA_1 \mid 00 \mid AB \mid BA \mid BB$
B	$\to$	$\cancel{00} \mid UU$
A_1	$\to$	AB
U	$\to$	0

replace B

5. Final CFG in Chomsky Normal Form

S	$\to$	$BA_1 \mid 00 \mid AB \mid BA \mid BB \mid \varepsilon$
A	$\to$	$BA_1 \mid 00 \mid AB \mid BA \mid BB$
B	$\to$	UU
A_1	$\to$	AB
U	$\to$	0

Problem 5: (10 points) Use pumping lemma to show that $\{0^n1^n0^n1^n | n \geq 0\}$ is not context free.

$$\boxed{A_3 = \{0^n1^n0^n1^n | n \geq 0\}}$$

The Context- Free Pumping Lemma definition tells us that any string s in A can be 'pumped' at least a 'pumping length' of p , then divided into five pieces $s = uvxyz$. For it to be regular, it must satisfy the following conditions:

(1) $\forall i \geq 0, uv^ixy^iz \in A$
(2) $|vy| \geq 0$
(3) $|vxy| \leq p$

We can test these conditions by doing the following:

- Assume A_3 is a context-free language (CFL)
- Let p be the pumping length
- Choose a string s in language A_3 to test: $s = 0^P1^P0^P1^P$
  - As we assume A_3 is CFL, s is a member of A_3
  - Because $|s| = 4p$, and $4p \geq p$, then $|s| \geq p$
  - Since both statements above are true, the Pumping Lemma definition tells us that s can be split into five pieces $s = uvxyz$

- Case 1: v and x contain only one type of symbol
  - Testing conditions (2) $|vy| \geq 0$ and (3) $|vxy| \leq p$ under Case 1
    - Assume $p = 5$

$$0^P1^P0^P1^P = 0^51^50^51^5$$

$$0^51^50^51^5 = 00000111110000011111$$

    - Split $s = uvxyz$, with $|vxy| \leq 5$

$$\underbrace{000}_{u}\underbrace{00}_{v}\underbrace{1}_{x}\underbrace{11}_{y}\underbrace{110000011111}_{z}$$

$$|vxy| = 5$$

  - Testing condition (1) $\forall i \geq 0, uv^ixy^iz \in A$
    - assume $i = 2$

$$uv^ixy^iz \in A$$
$$uv^2xy^2z \in A$$

    - Visual Representation

$$\underbrace{000}_{u}\underbrace{00}_{v}\underbrace{00}_{v}\underbrace{1}_{x}\underbrace{11}_{y}\underbrace{11}_{y}\underbrace{110000011111}_{z}$$

    - $uv^ixy^iz \notin A$
      - There is no possible division of 0000000111111110000011111 that will result in the required $0^n1^n0^n1^n$ format. This is evident in the visual

representation above as it has the format $0^7 1^7 0^5 1^5$ and $7 \neq 5$. As i increases, this also remains true since additional zeros and ones are placed at the beginning of the string, creating a further imbalance between the first $0^n 1^n$ and the second $0^n 1^n$ (which should be identical n's). Therefore, language A_3 does not meet condition (1) for Pumping Lemma when split in such a way that v and x contain only one type of symbol.

- Case 2: Either v or x contain more than one type of symbol
  - Testing conditions (2) $|vy| \geq 0$ and (3) $|vxy| \leq p$ under Case 2
    - Assume $p = 5$

$$0^P 1^P 0^P 1^P = 0^5 1^5 0^5 1^5$$

$$0^5 1^5 0^5 1^5 = 00000111110000011111$$

    - Split $s = uvxyz$, with $|vxy| \leq 5$

$$\underbrace{0000}_{u}\underbrace{01}_{v}\underbrace{1}_{x}\underbrace{11}_{y}\underbrace{10000011111}_{z}$$

$$|vxy| = 5$$

  - Testing condition (1) $\forall i \geq 0, uv^i xy^i z \in A$
    - assume $i = 2$

$$uv^i xy^i z \in A$$
$$uv^2 xy^2 z \in A$$

    - Visual Representation

$$\underbrace{0000}_{u}\underbrace{01}_{v}\underbrace{01}_{v}\underbrace{1}_{x}\underbrace{11}_{y}\underbrace{11}_{y}\underbrace{10000011111}_{z}$$

    - $uv^i xy^i z \notin A$
      - There is no possible division of 0000010111111110000011111 that will result in the required $0^n 1^n 0^n 1^n$ format. This is evident in the visual representation above as it has the format $0^5 1^1 0^1 1^7 0^5 1^5$ and $0^n 1^n 0^n 1^n 0^n 1^n \neq 0^n 1^n 0^n 1^n$. As i increases, this also remains true since v contains more than one type of symbol; as v^i is pumped, the string places 1's in the incorrect spot, disrupting the language format entirely. Therefore, language A_3 does not meet condition (1) for Pumping Lemma when split in such a way that either v or x contain more than one type of symbol.

Since not all pumping lemma conditions are met in either Case 1 or Case 2, the assumption that A_3 is a CFL is contradicted and proves that A_3 is not context-free.

Q.E.D

Problem 6: (10 points) Answer the following two questions:

- Briefly describe how regular language is used for lexical analysis in a compiler.
  - Regular Language is used by the lexical analyzer to tokenize the source file (Character Stream), using the string patterns defined by the regular language as the scope for the tokens. These tokens define basic program syntax and are used to verify errors in code. For example, if the code contains any characters which do not match a token pattern (valid syntax), a Lexical Error is raised. Examples of Lexical Errors are spelling errors, typos, and illegal characters. After these tokens are sent into a Token stream, they are sent to the Parser, and undergo syntax analysis.

- Briefly describe how context free language is used for parsing (syntax analysis) in a compiler.
  - Context-free language is used for parsing in a compiler to transfer the tokens from the Token Stream into a Parse Tree while verifying the code is syntactically correct. While lexical analysis focuses on individual words, syntax analysis focuses on how these words work together and whether they adhere to the correct grammar rules. Context-free language is what specifies these grammar rules. If the code contains any deviations to these rules, such as structurally or syntactically, a Syntax Warning/Error is raised. Examples of Syntax Errors are unbalanced parenthesis, missing operators, and indentation errors. After Syntax Analysis, the code undergoes Semantic Analysis.

Problem 7: (20 points) Programming Assignments

a. (4 Points) Describe GCC Intermediate Files
   - hello.c
     - This is the original source code file in C. It is a simple program written using the coding language C that prints the line “hello world” and newline character to the console.
   - hello.i
     - This is the GCC preprocessor output file. The preprocessor begins transforming the original source file using Lexical Analysis. Some operations completed during this step includes: removing comments, substitutes macros for their equivalent values, and compiles all conditionals.
   - hello.s
     - The assembly code for the source file. It contains the source file equivalent written in low-level assembly language.
   - hello.o
     - This is the object file. It contains the machine code (binary representation) equivalent of the source code. When there are multiple source files, multiple object files are created. These are then linked together to create the executable.
   - hello
     - This is compiled and linked executable which can be ran using “./hello”. When the program is run, it executes the code originally written in hello.c – it prints the line “hello world” and newline character to the console.

b. Included in HW folder.