COMP 330 Homework 2

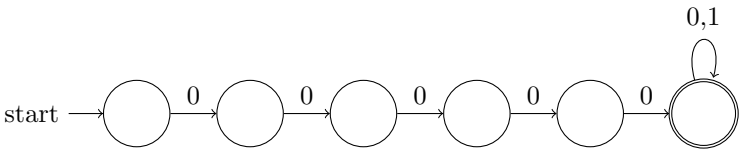
Ryan Sowa, ID: 260886668

1. (35 points) For each of the following languages give a proof that it is or is not regular

   (a)

   $$\{0^m 1^n \mid \ m \geq 5 \text{ and } n \geq 0\}.$$

   Proof. This language is regular, because the following NFA can be constructed for this language:

   □

   (b)

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

   Proof. Let us call this language L. Suppose by contradiction that L is regular. This implies that there must exist a DFA M for L. Let x be the number of states in M. Since we only have x states and we have $0^x \in L$, we must have to revisit a state. In other words, there exists some $j, k \in \mathbb{N}$ where $0 \leq j < j + k \leq x$ such that 0^j and 0^{j+k} end up in the same state. However, suppose $0^{j+k}1^{\sqrt{j+k}} \in L$. From the reasoning above, $0^j 1^{\sqrt{j+k}}$ must also be in L. However, since

   $$j \not\geq j + k$$

   , $(k > 0)$ we have found a contradiction $\Rightarrow\!\!\!\!\!\!\Leftarrow$, and we can conclude that L is NOT a regular language.

   □

(c) The set of strings in $\{0,1\}^*$ which are **not** of the form ww for some $w \in \{0,1\}^*$.

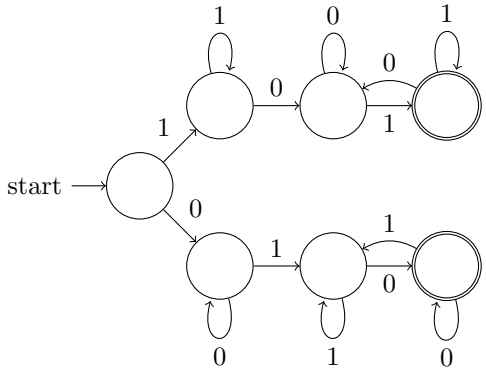
Proof. Let us call this language L. Suppose by contradiction that L is regular. This implies that there must exist a DFA M for L. Let x be the number of states in M. Since we only have x states and we have $0^x \in L$, we must have to revisit a state. In other words, there exists some $j, k \in \mathbb{N}$ where $0 \leq j < j + k \leq x$ such that 0^j and 0^{j+k} end up in the same state. However, suppose $0^{j+k}10^j1 \in L$. From the reasoning above, 0^j10^j1 must also be in L. However, this expression is of the form ww, so it is not in $L \Rightarrow\Leftarrow$, and we can conclude that L is NOT a regular language.

□

(d) Over the alphabet $\sum = \{0,1\}$:

$L = \{x \mid x$ contains the same number of 01's and 10's as substrings$\}$

Proof. This language is regular, since the following DFA can be constructed for it:



□

(e) Over the alphabet $\sum = \{0,1,2\}$:

$L = \{x \mid x$ contains the same number of 01's and 10's as substrings$\}$

Proof. Let us call this language L. Suppose by contradiction that L is regular. This implies L must have a pumping length, P such that any string $S \in L$,where $|S| \geq P$, can be divided into 3 parts, x, y and z. Without loss of generality, suppose $S = 1020^P1^P$ and let $P = 3$. We get the string $S = 102000111$, and we divide the string

into $x = \varepsilon$, $y = 102$, and $z = 000111$. By the Pumping Lemma, xy^iz must also be in L if L is regular. However, take $i = 2$. Then, we get the string 102102000111, which does not contain the same number of 01's and 10's as substrings $\Rightarrow\Leftarrow$. Therefore, we can conclude by the Pumping Lemma that this language is NOT regular.

□

(f) The first two Fibonacci numbers are 0 and 1, and each subsequent number is the sum of the previous two: 0, 1, 1, 2, 3, 5, 8, 13,. . .. Now the language in question is

$$\{0^n \mid n \text{ is a Fibonacci number}\}.$$

Proof. Let us call this language L. Suppose by contradiction that L is regular. This implies that there must exist a DFA M for L. Let x be the number of states in M. Since we only have x states and we have $0^x \in L$, we must have to revisit a state. In other words, there exists some $j, k \in \mathbb{N}$ where $0 \leq j < j + k \leq x$ such that 0^j and 0^{j+k} end up in the same state. However, suppose $0^j \in L$. From the reasoning above, 0^{j+nk} where $n \in \mathbb{Z}$ must also be in L. However, this implies that each term in the Fibonacci sequence differs by exactly k, which is trivially false $\Rightarrow\Leftarrow$. Therefore, we can conclude that L is NOT a regular language.

□

(g) The set of strings in $\{0, 1\}^*$ which are <u>not</u> palindromes:

$$\{w \in \{0, 1\}^* \mid w \neq w^R\}.$$

Proof. Let us call this language L. Suppose by contradiction that L is regular. This implies L must have a pumping length, P such that any string $S \in L$,where $|S| \geq P$, can be divided into 3 parts, x, y and z. Without loss of generality, suppose $S = 0^P100^P$ and let $P = 3$. We get the string $S = 00010000$, and we divide the string into $x = 00$, $y = 0$, and $z = 10000$. By the Pumping Lemma, xy^iz must also be in L if L is regular. However, take $i = 2$. Then, we get the string 000010000, which is a palindrome (not in L) $\Rightarrow\Leftarrow$. Therefore, we can conclude by the Pumping Lemma that this language is NOT regular.

□

2. (a) (3 points) Find a left-most derivation for *aaabbabbba* in the following context-free grammar:

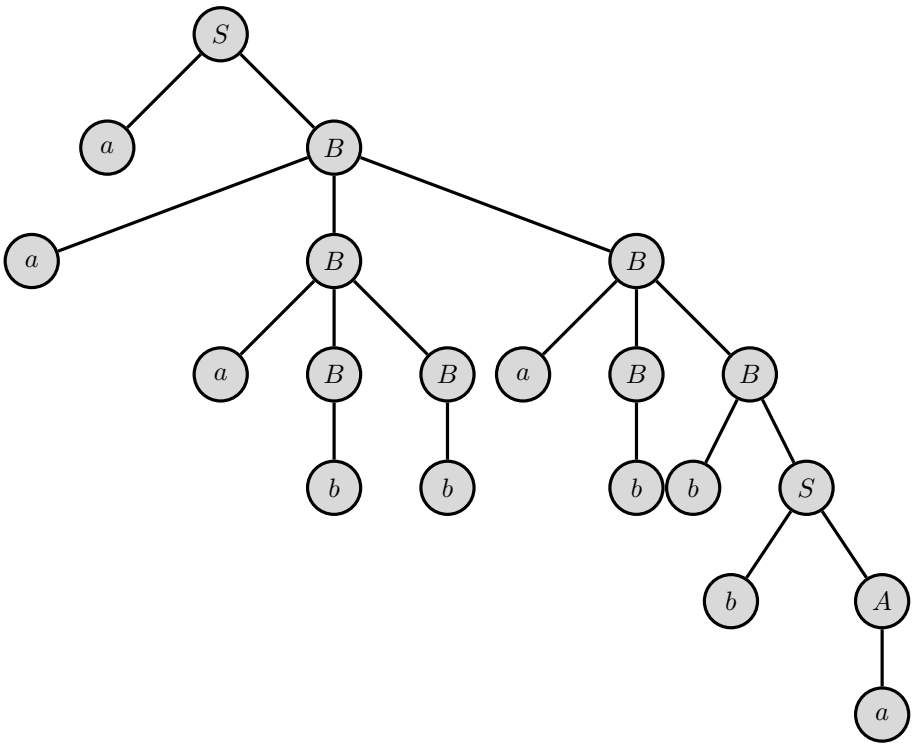
$$S \to aB \mid bA$$
$$A \to a \mid aS \mid bAA$$
$$B \to b \mid bS \mid aBB$$

The left-most derivation for *aaabbabbba* in the following context-free grammar is shown below:

$$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aaaBBB \Rightarrow aaabBB \Rightarrow aaabbB \Rightarrow aaabbaBB$$

$$\Rightarrow aaabbabB \Rightarrow aaabbabbS \Rightarrow aaabbabbbA \Rightarrow aaabbabbba$$

(b) (2 points) Draw the corresponding parse-tree of your left-most deviation.



3. (10 points) Show that the language of the grammar $S \to 0S1 \mid 1S0 \mid SS \mid \varepsilon$ is

$$\{w \in \{0,1\}^* \mid w \text{ contains the same number of zeros and ones}\}.$$

Proof. Let G be the context-free grammar above. Proving the statement above is equivalent to proving the following proposition:

$P(n)$: For any string in $\{0,1\}^*$ of length n such that $n \in \mathbb{N}$, G can generate all possible strings of length n which contain the number the same number of zeros and ones.

Base case:

We want to prove $P(0)$ or the following statement: For any string of length 0, G can generate all possible strings of length 0 which contain the number the same number of zeros and ones. For a string of length 0, there is only one string containing an equal number of zeros and ones: the empty string. G can generate this string using the rule $S \to \varepsilon$.

Induction Hypothesis:

Assume $P(k)$ is true for some $n = k$.

Inductive Step:

We want to prove $P(k+2)$ or the following statement: For any string of length $k+2$, G can generate all possible strings of length $k+2$ which contain the number the same number of zeros and ones. Since $P(k)$ is true by the induction hypothesis, we need to analyse all the cases where we add a zero and a one (since we are adding 2 and we must have the same number of 0's and 1's) to our string of length k to our string. We can add a zero or a one in one of the following ways: 1. Add a zero at the beginning of the string and a one at the end of the string, 2.Add a one at the beginning of the string and a zero at the end of the string, 3. Add 01 or 10 to the beginning or end of our string.

Case 1: We can achieve this using the: $S \to 0S1$ (let us call this rule 1).

Case 2: We can achieve this using: $S \to 1S0$ (let us call this rule 2).

Case 3: We apply the rule: $S \to SS$. We can then apply rule 1 or rule 2 on the first "S" to add 01 or 10, respectively at the beginning of the string. Similarly, we can then apply rule 1 or rule 2 on the second "S" to add 01 or 10, respectively at the end of the string.

When we have reached the amount of 1's and 0's desired, we can replace all S's with ε's.

Since we have proved all the cases, we can conclude that the proposition is true.

□

4. (20 points) Construct a context free grammar for the set of all words w over the alphabet $\{0, 1\}$ such that each prefix of w has at least as many

0's as 1's. You have to prove that (i) every such word can be generated with your grammar, and (ii) every word generated by your grammar has the desired property.

$$S \to 0S \mid 0S1 \mid SS \mid \varepsilon$$

(i)

Proof. Let the context free grammar we constructed above be denoted by G and the language above L.
$P(n)$: All strings of length n, such that $n \in \mathbb{N}$, in L can be generated by G.

Base Case: We want to prove $P(0)$ or the statement: All strings of length 0 in L can be generated by G. The only possible string in L of length 0 is the empty string, and G can generate the empty string with the rule $S \to \varepsilon$.

Induction Hypothesis: Assume $P(k)$ is true for some $n = k$.

Induction Step: We want to prove $P(k+1)$ or the statement: All strings of length $k+1$ in L can be generated by G. By our induction hypothesis, we know that all strings of length k in L can be generated by G. Let us take some string of length k in L and call it S. To reach a string of length $k+1$ we can either 1. add a 0 to S or 2. add a 1 to S.

Case 1: Adding a 0 to S must be in L since the number of 0's in S must be still greater than or equal to the number of 1's. Our context free grammar can generate this string using the rules $S \to SS \to \ S0S \to S0$.

Case 2: According to our language, if we are adding a 1, we must have a greater number of 0's than 1's in S, since the total number of 0's in our string of $k+1$ must be greater than or equal to the total number of 1's. Let m be the amount 0's in S minus the amount of 1's in S. Since $m > 0$, we can pair an extra 0 up with this added 1 using the rule $S \to 0S1$. Therefore, adding a 1 can also be generated by G.

When we have reached the amount of 1's and 0's desired, we can replace all S's with ε's.

Since $P(k+1)$ is true, we can conclude that all strings of length n in L can be generated by G.

□

(ii)

Proof. Proposition: Every word generated by our grammar, G has the desired property.

Base Case: The empty string can be generated by G. This is clearly true using the rule $S \to \varepsilon$.

Induction Hypothesis: Suppose S is generated by our grammar and is in the language above, L.

Induction Step: We want to prove that applying any of the rules of G, our new string, T will remain in L. By our induction hypothesis, we know that S will contain at least as many 0's as 1's.

Since $S \to 0S$ is simply adding extra 0's to S, we can confirm that T is still in L as the number of 0's in T will still be greater than or equal to the number of 1's in T (it will be in L). For the rule $S \to 0S1$, we are adding an equal amount of 0's and 1's to S, so know that the number of 0's in T will still be greater than or equal to the number of 1's in T (it will be in L). For the rule $S \to SS$, since we know that S contains at least as many 0's as 1's, if we multiply the number of 0's and the number of 1's in S by 2, T will trivially still contain at least as many 0's as 1's (it will be in L). Lastly, for the rule $S \to \varepsilon$, we are not changing the number of 0's or 1's in S, so T will still be in L.

□

5. (20 points) For each of the following languages construct a context-free grammar that generates that language:

   (a)
   $$\{0,1\}^*.$$
   $$S \to 0S \mid 1S \mid \varepsilon$$

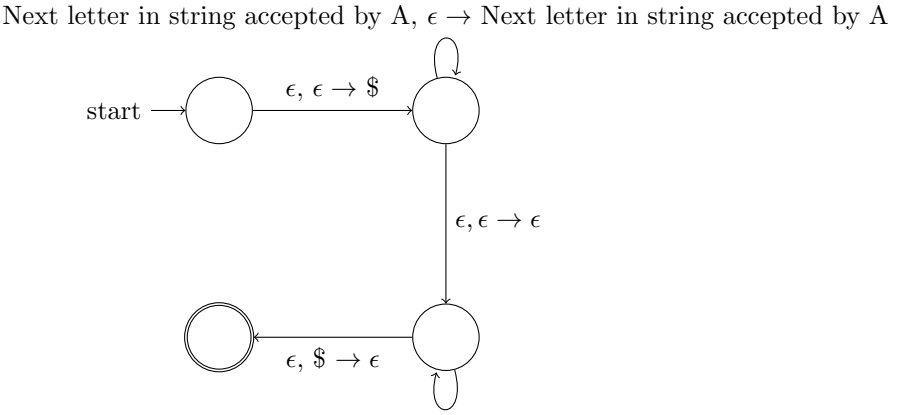
   (b)
   $$\{0^m 1^n \mid m \geq n \text{ and } m - n \text{ is even}\}.$$
   $$S \to 0S1 \mid 00S \mid \varepsilon$$

   (c) The complement of $\{0^n 1^n \mid n \geq 0\}$ over the alphabet $\{0,1\}$.
   $$S \to A0 \mid 1A \mid 0S1$$
   $$A \to 0A \mid 1A \mid \varepsilon$$

   (d) The set of strings in $\{0,1\}^*$ which are <u>not</u> palindromes:
   $$\{w \in \{0,1\}^* \mid w \neq w^R\}.$$
   $$S \to 0A1 \mid 1A0 \mid 0S0 \mid 1S1$$
   $$A \to 0A \mid 1A \mid \varepsilon$$

6. (10 points) Use the equivalence of context-free grammars and push-down automata to show that if A and B are regular languages, then $\{xy \mid x \in A, y \in B, |x| = |y|\}$ is context-free.

Since A and B are regular, the following push-down automata will accept this language:



In this push-down automata, we only accept a string x which is in A followed by a string y which is in B such that $|x| = |y|$. Since push-down automatas are equivalent to context free languages, a context-free language can also be constructed for this language, and hence the language is context-free.