Mat Shi

I pledge my honor that I have abided by the Stevens Honor System

CS 334 Problem Set 4

1. Step 1:

	1	2	3	4	5	6	7	8	9
1				✓					
2				✓					
3				✓					
4					✓	✓	✓	✓	✓
5									
6									
7									
8									
9									

Step 2:

	1	2	3	4	5	6	7	8	9
1		✓	✓	✓		✓		✓	
2			✓	✓	✓	✓	✓		✓
3				✓	✓		✓	✓	✓
4					✓	✓	✓	✓	✓
5						✓		✓	
6							✓	✓	✓
7								✓	
8									✓
9									

Step 3:

	1	2	3	4	5	6	7	8	9
1		✓	✓	✓		✓		✓	
2			✓	✓	✓	✓	✓		✓
3				✓	✓		✓	✓	✓
4					✓	✓	✓	✓	✓
5						✓		✓	
6							✓	✓	✓
7								✓	
8									✓
9									

→ No new pairs marked

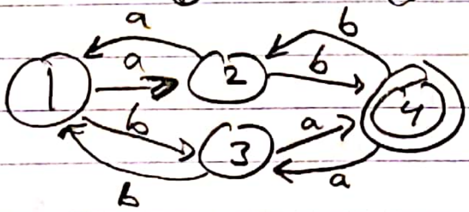
1=5, 1=7, 1=9, 2=8, 3=6, 5=7, 5=9, 7=9.

Class 1 = 1, 5, 7, 9

Class 2 = 2, 8,

Class 3 = 3, 6

Class 4 = 4.



2. Let A be a regular language, and let D be a DFA that accepts A. Therefore, for every string not in A, D ends up in a reject state. In other words, for every string in A's complement, $\bar{A}$, D ends in a reject state. Thus, if we flip the accept and reject states in D to produce $\bar{D}$, this DFA will end in an accept state for all strings in $\bar{A}$. Thus, a DFA accepts $\bar{A}$, proving A being regular implies $\bar{A}$ is also regular.

2b. Let A and B be regular languages. Let $\bar{B}$ be the complement of B. By the proof in the previous problem, $\bar{B}$ is regular. $A - B$ is equivalent ~~to~~ the expression $A \cap \bar{B}$. Intersection of two ~~sets~~, C and D, $C \cap D$, is equivalent to $(\overline{\bar{C} \cup \bar{D}})$. Thus, $A \cap \bar{B} = (\overline{\bar{A} \cup B})$, where $\bar{A}$ is the complement of A. $\bar{A}$ is also regular. Thus, because regular languages are closed under union, $\bar{A} \cup B$ is regular. Furthermore, because regular languages are also closed under complement, $(\overline{\bar{A} \cup B})$ is also regular. Thus, $A - B$ is regular if A and B are regular languages.

2c. Let us partition the input strings into L into three parts u, v, w, where u and v are part of the alphabet, and w is the rest of the input string. There are then three cases for strings accepted by L. Let pumping length $p = 2$.

Case 1: $uv = aa$.
In this case, $w =$ a string with length ≥ 0 that has strictly any number of a's, then b's, then c's. In this case, we assign x, y, z from the pumping lemma as such: $x = u = a$, $y = v = a$, $z = w$. This satisfies all three conditions, as $xy^iz \in A\ \forall i \geq 0$, as xy^i matches a^i and z matches b^jc^k in the reg. ex., $|y| = 1 > 0$, and $|xy| = 2 \leq p$, as $p = 2$.

Case 2: $u \neq a$, so $u = b$ or $u = c$
In this case, let $uvw = s$. For string s to be accepted by L, there are three combinations for u, v, and w. If $u = c$, v must be c, and w is a string of length ≥ 0, of only c's. If $u = b$, v can be either b or c. If it is c, w is the same as if $u = c$. If $v = b$, then $w =$ string of length ≥ 0 of strictly b's then c's. Let these be case A, B, and C respectively. All three properties of the pumping lemma are still fulfilled. $|y| = 1 > 0$, $|xy| = 2 \geq p$, as $p = 2$. $xy^iz \in A\ \forall i \geq 0$, as $i = 0$ in the a^i of the reg. ex, and:

Case A:
$x = u = c$, $y^i = v^i = c^i\ \forall i \geq 0$, $z = w = c^j$, $j \geq 0$, → this all matches c^k, and thus is $\in L$.

Scanned by CamScanner

case B:

$x = u = b$, $y^i = v^i = c^i$ $\forall i \geq 0$, $z = w = c^j$, $j \geq 0$, and since x matches b^j, and $y^i z$ matches c^k, case $B \in L$.

case C:

$x = u = b$, $y^i = v^i = b^i$ $\forall i \geq 0$, $z = w = b^j c^k$ $j,k \geq 0$, and since xy^i matches b^j from the regex, and z matches $b^j c^k$ from the regex, Case $C \in L$.

Thus, case 2 fulfills all properties.

Case 3: $u = a$, $v \neq a$, thus $v = b$ or $v = c$.

$u = a$, thus fulfilling a^i. However, in this case, $i = 1$, which means $j = k$ for the regex $b^j c^k$. Thus, in order to satisfy the regex, $v \neq c$, as j can never equal k if $v = c$. Thus, $v = b$. However, in this case, set $x = \epsilon$, $y = u = a$, and $z = v$ plus a string with k number of b's and then $k+1$ number of c's, for $k \geq 0$. Thus, $xy^i z \in L$, $\forall i \geq 0$, as $xy^i = a^i$ which matches a^i in the regex, and $z = vw = b^j c^j$ which matches the $b^j c^k$, $j = k$ in the regex. Furthermore, $|y| = 1 > 0$, and $|xy| = 1 \leq p$, as $p = 2$.

Therefore, through all three cases, all conditions on the pumping lemma are satisfied.

2d. Assume that L is regular. Because regular languages are closed under union, $\{ab^i c^i : i \geq 0\}$ must be regular. Because regular languages are closed under concatenation, $\{b^i c^i : i \geq 0\}$ must also be regular, and therefore, must satisfy the pumping lemma. Let $\{b^i c^i : i \geq 0\}$ be the language L_R. Because L_R is regular, it must be recognized by a DFA with p states for some number $p \geq 1$. Considering the string $b^p c^p$, $b^p c^p$ is recognized by L_R. Then, let $x = \epsilon$, $y = b^p$, and $z = c^p$, for $p \geq 1$. This satisfies $|p| > 1$ and $|xy| \leq p$, as $|xy| = p$. However, as y is pumped, $xyyz$ is not accepted by L_R as $xyyz = b^p b^p c^p$, and $p + p \neq p$. Thus, there is a contradiction, which means L_R is not regular and L is not regular.

e) This has to do with the rules surrounding $p \to q$. If we let p = "L is a regular language" and q = "L satisfies all conditions of the pumping lemma," we can show that we do not contradict the pumping lemma. In this case, p was false, which automatically makes the statement $p \to q$ true, as q can now be any result. Thus, (c) and (d) do not contradict the pumping lemma.