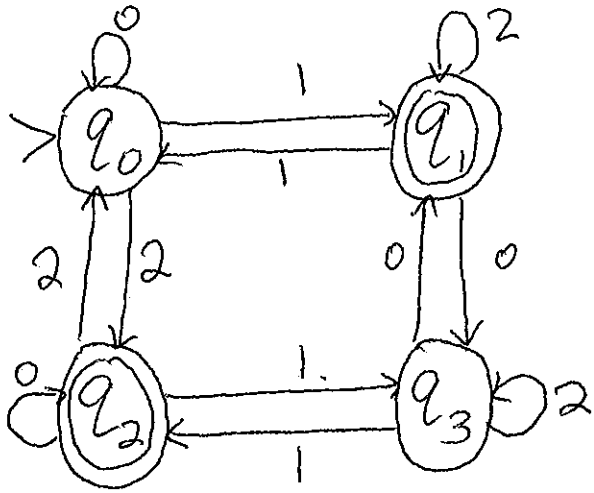
COT 4210: Discrete Structures II
Exam #1
February 7, 2013

Name: SOLUTION

Lecturer: Arup Guha

(Directions: Please justify your answer to each question. No answer, even if it is correct, will be given full credit without the proper justification.)

1) (12 pts) Design a DFA to accept all ternary strings over the alphabet {0, 1, 2} that are equivalent to 1 mod 4 or 2 mod 4. Remember that a ternary string is in base 3, so the value $2102_3 = 2x3^3 + 1x3^2 + 0x3^1 + 2x3^0 = 65$ in base 10, which is part of the language described.



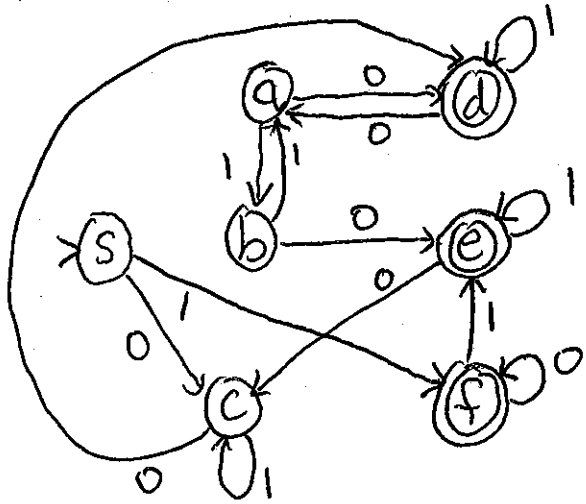
Grading

1pt label start state
2pts have 4 states
2pts label 2 correct final states
1pt every transition exists
6pts - all transitions, 1/2 pt each round down

2) (12 pts) Use the DFA Minimization algorithm shown in the Sudkamp text to minimize the following DFA described below.

$Q = \{s, a, b, c, d, e, f\}$
$\Sigma = \{0, 1\}$
$q_0 = s$
$F = \{d, e, f\}$

δ	0	1
s	c	f
a	d	b
b	e	a
c	d	c
d	a	d
e	c	e
f	f	e



Not accept
s, a, b, c

[s,a] split → on 0
[s,b] split → on 0
[s,c] split → on 0

3pts

Accept
d, e, f

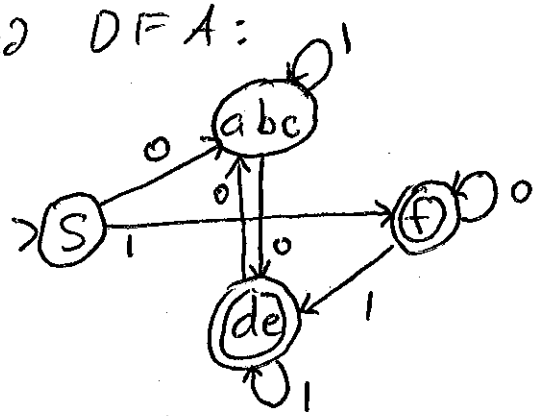
2pts
[d,e] : S[c,a] = {(d,e)} on 0
[d,f] split on → 0
[e,f] : split on → 0

S[d,e] = {[a,b]} on 0
S[b,c] = {[a,c]} on 1
S[e,d] = {[b,c]} on 0
S[a,c] = {[b,c]} on 1

3pts

4 states needed: {s}, {a,b,c}, {d,e}, {f}

Minimized DFA:



Grading
Work for splitting s, a, b, c - 6pts
Work for splitting d, e, f - 2pts
Final DFA - 4pts

3) (15 pts) Let $L = \{0^a1^b \mid a, b > 0 \text{ and } \gcd(a, b) > 1\}$. Prove via the pumping lemma or otherwise, that L is not regular.

We will show that L isn't regular by proving that L doesn't satisfy the pumping lemma for regular languages.

Assume to the contrary that L is regular. Then, L satisfies the pumping lemma for regular languages. Let p = the pumping length of L.

Let q be a prime number with $q > p$. (Since there are an infinite number of primes, we are guaranteed of q's existence.)

Consider $s = 0^{2q}1^q$. $|s| = 3q > 3p > p$. (4 pts)

According to the pumping lemma, there exists a way to partition s into xyz such that $|xy| \leq p$, $|y| > 0$ and $xy^iz \in L$. Thus, any valid partition takes the following form:

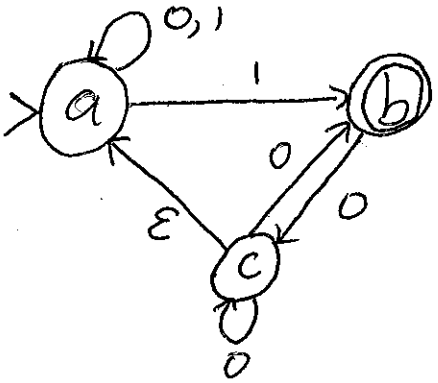
$x = 0^i$, $y = 0^j$, $z = 0^{2q-i-j}1^q$. (with $i+j \leq p$ and $j > 0$) (2 pts)

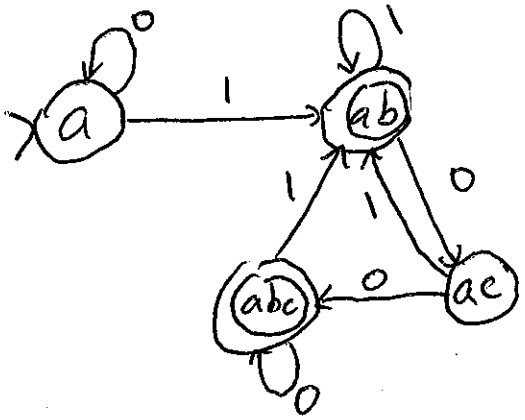
Consider $xy^2z = 0^{2q+j}1^q$. Since $j > 0$ and $j \leq p < q$, $2q+j > 2q$ and $2q+j < 3q$. Since q is prime and $q \nmid (2q+j)$, $\gcd(2q+j, q) = 1$ proving that $xy^2z \notin L$. Thus, s doesn't satisfy the pumping lemma. It follows that L is not regular. (9 pts)

4) (12 pts) Convert the NFA formally described below into a DFA. Please provide either a formal description or a drawing of the converted DFA. Please use the algorithm shown in class.

$Q = \{a, b, c\}$
$\Sigma = \{0, 1\}$
$q_0 = a$
$F = \{b\}$

δ	0	1	ε
a	{a}	{a,b}	{}
b	{c}	{}	{}
c	{b,c}	{}	{a}





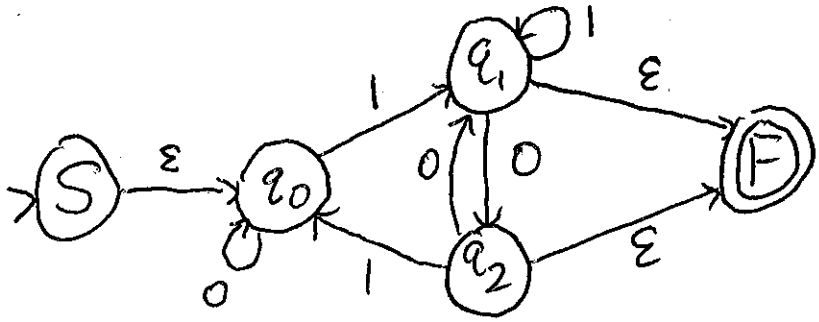
Grading

1 pt per state (4 pts)
½ pt per transition (4 pts), round down
2 pts label start state
2 pts correct final states

5) (12 pts) Consider the process of deriving a regular expression for the language described by the 3-state DFA below. In this process, the DFA gets converted into a 5 state GNFA and then states get ripped out. Show the 4 state GNFA that occurs in this process when q_1 is the first state ripped out of the 5 state GNFA.

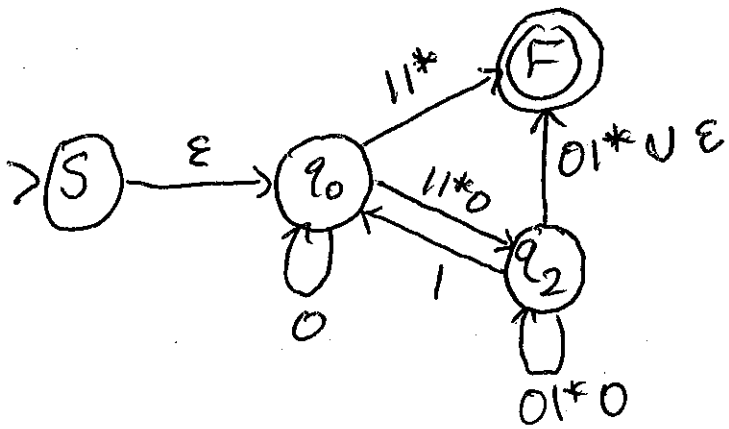
$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{0, 1\}$
Start state = q_0
$F = \{q_1, q_2\}$

δ	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_1	q_0



$q_0 \to q_0$: add nothing
$q_0 \to q_2$: add 11^*0 (2 pts)
$q_0 \to F$: add 11^* (2 pts)
$q_2 \to q_0$: add nothing
$q_2 \to q_2$: add 01^*0 (2 pts)
$q_2 \to F$: add 01^* (2 pts)

New diagram: (4 pts for final drawing)



6) (12 pts) A parabolic word is one that can be partitioned into two parts so that the first part is in alphabetic order and the second part is in reverse alphabetic order. For example, aaabbccccbaa is a parabolic word since it can be partitioned into aaabbccc and cbaa. (Note that multiple valid partitions may exist.) Design a context free grammar over the alphabet {a, b, c} that precisely describes the language of all parabolic words. (Note: ε should be in the language you design.) Briefly justify why your grammar produces the desired language.

$S \to ATA$

$T \to BUB$

$U \to Uc \mid \varepsilon$

$B \to Bb \mid \varepsilon$

$A \to Aa \mid \varepsilon$

Start = S

Grading
label start - 1pt
valid grammar - 2pts
produces ε - 1pt
produces all other strings in lang - 4pts
doesn't produce extra strings - 4pts

Variables A and B produce a^* and b^* respectively. The start symbol allows for a's on both ends. The variable T allows for b's inside the a's. The variable U allows for c's inside of the b's.

Note: This language is regular. A regular expression for it is $a^*b^*c^*b^*a^*$.

Grading
2pts L_1,
2pts L_2,
4pts justification

7) (8 pts) Give an example of two languages L_1 and L_2 such that neither language is regular but both $L_1 \cup L_2$ and $L_1 \cap L_2$ are regular. Briefly justify why the languages you have chosen for L_1 and L_2 are not regular but why $L_1 \cup L_2$ and $L_1 \cap L_2$ are regular.

Let $L_1 = \{a^n b^n \mid n \geq 0\}$ and $L_2 = \overline{L_1}$. From class, we know that L_1 is not regular. If L_2 were regular, its complement, L_1, would be as well. Since L_1 is not regular, it must be the case that L_2 is not regular. (Note: In the book it is proved if L is regular then $\overline{L}$ is regular.)

$L_1 \cup L_2 = \Sigma^*$, by definition of complement. This is regular described by the R.E. $(a \cup b)^*$.

$L_1 \cap L_2 = \emptyset$, by definition of complement. $\emptyset$ is a finite language. Thus it's regular.

8) (15 pts) Use mathematical (strong) induction to 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.

Use induction on the length of the string w, n.

(3pts) Base case: $n = 1$. For any string of length 1 in a CNF grammar, the only valid derivation is $S \to a$, where $a \in \Sigma$ and S is the start symbol. This is 1 step.

$2n - 1 = 2(1) - 1 = 1$ ✓

Thus, the base case holds.

(2pts) Inductive hypothesis: Assume for all positive integers $k \leq n$ that deriving a string of length k takes exactly $2k-1$ steps, for any CNF grammar.

(2pts) Inductive step: Prove for a string of length $n+1$, that all of its derivations take exactly $2(n+1)-1 = 2n+1$ steps, for any CNF grammar.

For any string, W, of length $n+1$ described by a CNF, it can be split into two parts x and y ($w = xy$) with $|x| \geq 1$ and $|y| \geq 1$ (so $|x| \leq n$, $|y| \leq n$) such that there is a rule $S \to AB$ and $A \xrightarrow{*} x$ and $B \xrightarrow{*} y$. In fact, all derivations satisfy this restriction. Thus, in all derivations of w, we must take this 1 step, followed by exactly

(1pts) $2|x| - 1$ and $2|y| - 1$ steps, respectively, based on the inductive

(4 pts) hypothesis. Adding, we get:

$$\begin{aligned} & 1 + 2|x| - 1 + 2|y| - 1 \\ &= 2(|x| + |y|) - 1 \\ &= 2(n+1) - 1 \end{aligned}$$

$= 2n+1$, as desired. (3 pts)

9) (2 pts) Who created the Chomsky hierarchy of languages? Chomsky

(2pts)