{PS 7}

John Schneiderman & Jessica Noel

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

Problem 1

$L = \{a^i b^k c^i d^k : i, k \geq 0\}$

High-Level Description:

(Since there are 2 stacks, I'll represent one as S1 and the other S2)

Push as onto S1

When the first b is read:

Change state

Push b onto S2

Loop: for every b read, push b onto S2

When the first c is read:

change state

Pop S1

Loop: for every c read, pop S1

When S1 is empty

Change state

Pop S2

Loop: for every d read, pop S2

Enter accept state when S2 is empty

Problem 2

Show $L_{mult} = \{a^i b^{ij} c^j : i, j \geq 0\}$

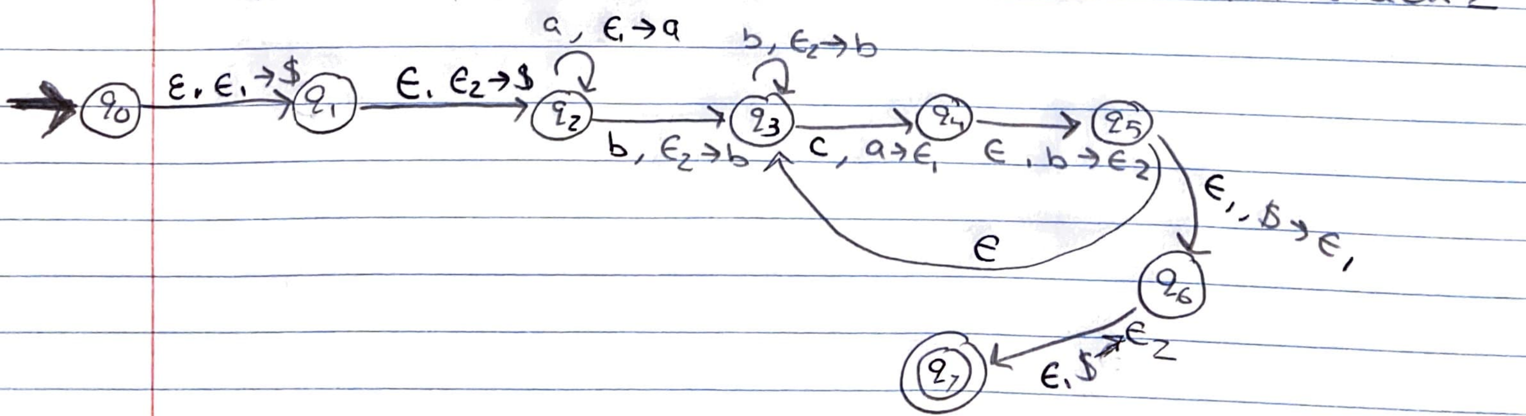
Stack 1 = S1 Stack 2 = S2

High-Level Description

1) Push a's onto S1
2) When the first b is seen:
   - Change state
   - push b onto S2
3) Loop: for every b seen, push b onto S2
4) When the first c is seen:
   - Change State
   - Pop a off S1
   - Change State
   - Pop b off S2
5) Loop: for every c seen, pop b from S2

If S1 isn't empty, return to step 4
else if S2 is empty, hit accept state
else reject state

ϵ_1 = push/pop Stack 1 ϵ_2 = push/pop Stack 2



Problem 3

Prove that the intersection of CFL and a regular language is always context free.

Let L_1 = the context free language & L_2 = regular

L_1 ↓ must exist a PDA P to accept

L_2 ↓ must exist a DFA D to accept

L_3 = intersection of the two languages

The accepting automata for L_3 must have aspects of both the PDA P and DFA D. Since "adding" these two together would result in an automata similar to that of a PDA (this machine would still have a stack to rely on), the resulting language is context-free. L_3 may not necessarily be regular, as the automata is a PDA and not a strict DFA, but is indeed a CFL.

Problem 4

Prove that the language $A/B = \{w : wx \in A, x \in B\}$ where A is a CFL and B is a regular CFL.

Similar to the explanation in #3, the automata that describes A/B must have both properties of the PDA that describes CFL A and regular language B. Let this automata be a PDA P. Since a PDA exists for this language that must mean it is a context-free language

Problem 5

When taking an input into a queue, we push each read symbol onto it one by one. Also put a \$ at the start of the queue (pushed last) to signify machine with a queue, we pop the end of the queue and push the new symbol back to the front if the rule says to. When the \$ is reached, we know the string is done, so the process is over. Queues can be used to simulate turing machines easily.

\$	a	a	b	b	c

=>

x	\$	a	a	b	b

until =>

z	y	y	x	x	\$