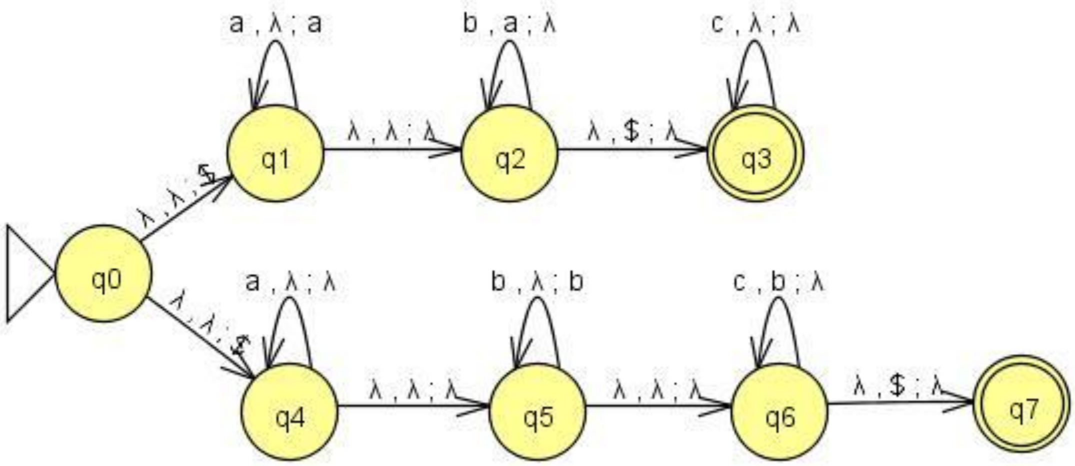
CS510 Fall 2021 Assignment #3 solution

1. **PDA: Exercise 2.10** (Note: in addition to informal description create your PDA on JFLAP and test it on multiple inputs.)

Solution:

1. Nondeterministically branch to either Step 2 or Step 5.
2. Read and push *a*’s
3. Read *b*’s, while popping *a*’s.
4. If *b*’s finish when stack is empty, skip *c*’s on input and **accept**.
5. Skip *a*’s on input.
6. Read and push *b*’s.
7. Read *c*’s, while popping *b*’s.
8. If *c*’s finish when stack is empty, **accept**.



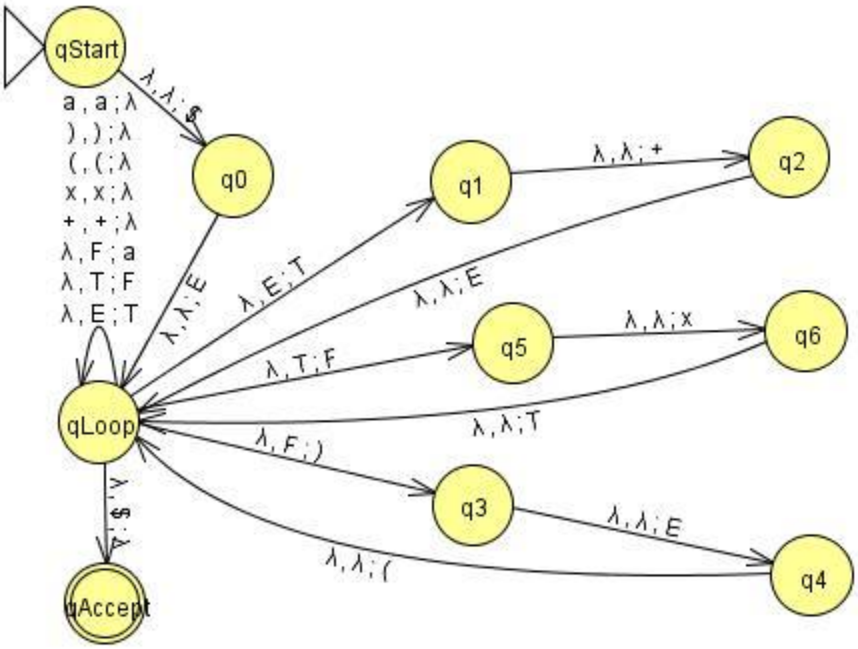
2. **Pumping lemma for CFL: Problem 2.32**

Solution:

Assume that C is context free and get its pumping length *p* from the pumping lemma. Let $s = 1^p3^p2^p4^p$. Because $s \in C$, it can be split into *uvxyz* satisfying the conditions of the lemma. By condition 3, *vxy* cannot contain both 1’s and 2’s, and cannot contain both 3’s and 4’s. Hence, uv^2xy^2z does not have equal number of 1’s and 2’s or of 3’s and 4’s, and therefore won’t be a member of *C*, so *s* cannot be pumped and contradiction is reached. Therefore, C is not context-free.

3. **Construct PDA from CFG: Exercise 2.11** (Create your PDA on JFLAP and test it on multiple inputs.)

Solution:



4. **Turing machine: Exercise 3.1(a)**

Solution:

$q_1 0$, $\sqcup q_2 \sqcup$, $\sqcup\sqcup q_{accept}$

5. **Turing machine: Exercise 3.2(c)**

Solution:

$q_1 1\#\#1,\ x q_3 \#\#1,\ \ x\# q_5 \#1,\ \ x\#\# q_{reject} 1.$

6. **Turing machine: Exercise 3.8 (b)**

Solution:

"On input string w:

1. Scan the tape and mark the first 0 which has not been marked. If there is no unmarked 0, go to stage 5.
2. Continue scanning and mark the next unmarked 0. If there is not any on the tape, **reject**. Otherwise, move the head to the front of the tape.
3. Scan the tape and mark the first 1 which has not been marked. If there is no unmarked 1, **reject**.
4. Move the head to the front of the tape and repeat stage1.
5. Move the head to the front of the tape. Scan the tape for any unmarked 1's. If none, **accept**. Otherwise, **reject**."

7. **Decidable language: Exercise 3.15 (b)(d)**

Solution to **Exercise 3.15 (b):**

For any two decidable languages L_1 and L_2, let M_1 and M_2 be the TMs that decide them. We construct a NTM M' that decides the concatenation of L_1 and L_2:

" On input w:

1. For each way to cut w into two parts $w = w_1w_2$:
2. Run M_1 on w_1.
3. Run M_2 on w_2.
4. If both accept, **accept**. Otherwise, continue with next w_1, w_2.
5. All cuts have been tried without success, so **reject**."

We try every possible cut of w. If we ever come across a cut such that the first part is accepted by M_1 and the second part is accepted by M_2, w is in the concatenation of L_1 and L_2. So M' accept w. Otherwise, w does not belong to the concatenation of the language and is rejected.

Solution to **Exercise 3.15 (d):**

For any decidable language L, let M be the TM that decides it. We construct a TM M' that decides the complement of L:

" On input w:

1. Run M on w. If M accepts, **reject**; if M rejects, **accept**."

Since M' does the opposite of whatever M does, it decides the complement of L.

8. **Turing-recognizable language: Exercise 3.16 (c)**

Solution:

For any Turing-recognizable languages L, let M be the TM that recognizes it. We construct a NTM M' that recognizes the star of L:
" On input w:

1. Nondeterministically cut w into parts so that $w = w_1 w_2 \ldots wn$:
2. Run M on w_i for all i. If M accepts all of them, **accept**. If it halts and rejects any of them, **reject**."

If there is a way to cut w into substrings such that M accepts all the substrings, w belongs to the star of L and M' will accept w after a finite number of steps.

9. **Formulate Church-Turing thesis. Is it a mathematical theorem? Explain your answer.**

Solution:

Church-Turing thesis: Intuitive notion of algorithms equals Turing machine algorithms.

It is not a theorem because it does not have a proof.