Justin Li
Dr. Krehbiel
CSCI 161
11/14/20

HW 8

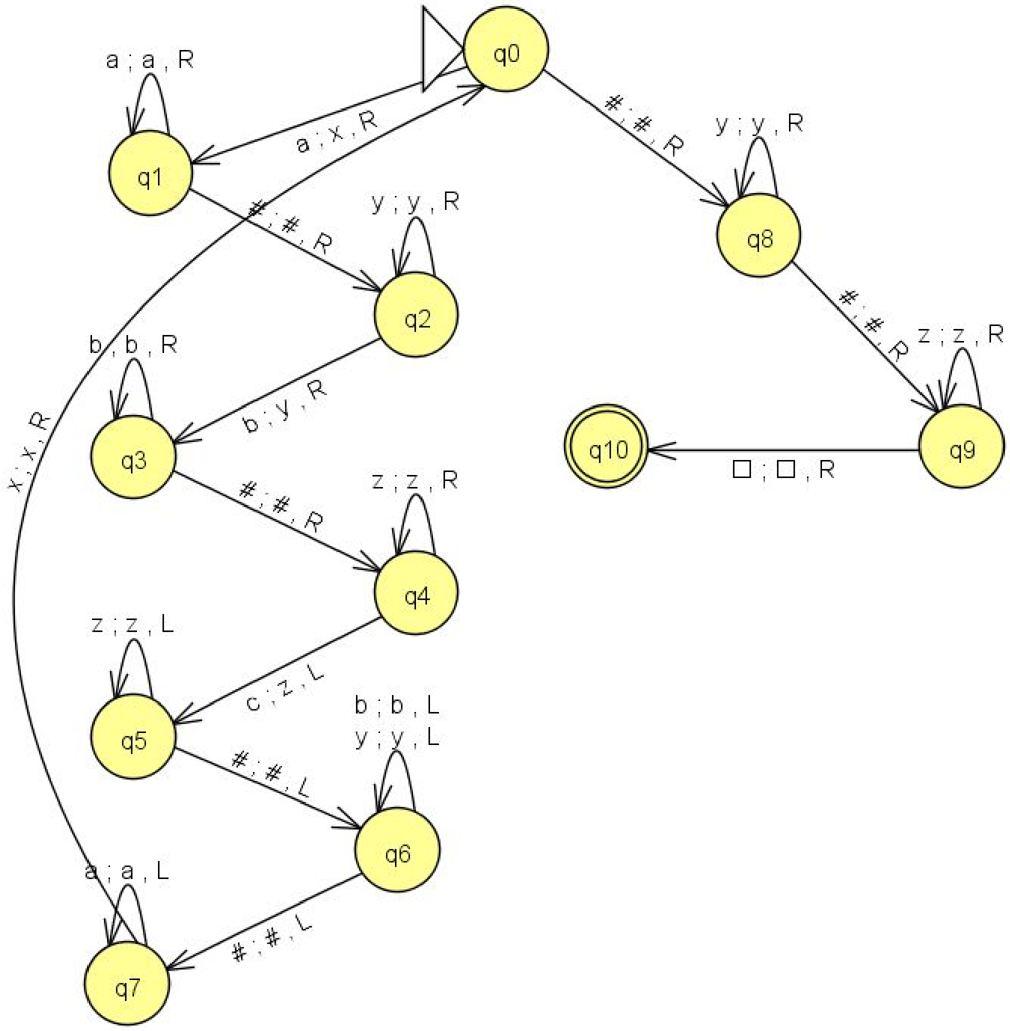
1) High Level Pseudocode:

   Match each a on the L (mark as x) with a b on the R (mark as y) past # symbol, and then a c on the R (mark as z) past another # symbol. Stop when I don't have any more a's after the L # moving L. Scanning to the R, if every character is marked, then accept.

   Low Level Pseudocode:

```
Until current is #: //looking at leftmost unmarked, or first # if done
    Mark a
    Scan R past unmarked a's // may not be any
    Move R on #
    Scan R past marked b's // may not be any
    Mark b and move R
    Scan R past unmarked b's // may not be any
    Move R on #
    Scan R past marked c's // may not be any
    Mark c
    Scan L past marked c's // there is at least 1
    Move L on #
    Scan L past unmarked b's // there may not be any
    Scan L past marked b's // there is at least 1
    Move L on #
    Scan L past unmarked a's // there may not be any
    Move R
Move R on #
Scan R past marked b's // may not be any
Move R on #
Scan R past marked c's // may not be any
If current is _: ACCEPT
```

2)



3) a) _xxx0x0x
   b) _x0x0x0
   c) _xxx0x

4) a) Match each 0 on the L (mark as x) with a 0 on the R (mark as x) past the # symbol. Or, match each 1 on the L (mark as x) with a 1 on the R (mark as x) past the # symbol. Stop when you don't have any more unmarked characters after the # moving L. Then, scan once more to the right, and if every character is marked, then accept.

Run-time analysis:
n iterations (left of the #), $\le 4n+4$ steps per iteration, $n+2$ on final scan
Total: $n(4n+4) + n+2 = 4n^2 + 5n + 2 \rightarrow O(n^2)$

b) Do the same as in part a, starting with reading either a 0 or a 1 (and marking it) but instead of reading past a #, for the elements on the R, we must test all possible cases of matching 0s or 1s, put all of those on different tapes, and then scan back left for all of these cases. We then repeat the same process for each of these tapes, and ignore all of the tapes with computations that lead to a reject state. Go until one accept state is reached, and accept that computation history and string.

c) In order to simulate the nondeterministic turing machine described in part b, use a 3 tape deterministic one that tries all of these computation histories (in BFS order), using an input tape, a simulation tape, and an address tape in order to test out all of the nondeterministic options for the Turing Machine for L2.