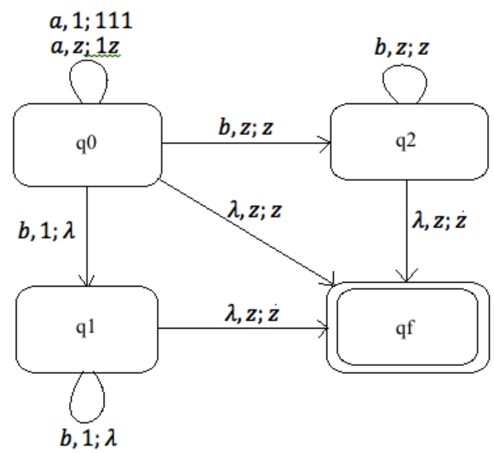
Cosc 362, Spring 2017

Test #3 – Take-Home Solutions

1. Consider the pushdown automaton M with states $\{q_0, q_1, q_2 q_F\}$, imput alphabet $\{a, b\}$, stack alphabet $\{1, z\}$, stack start symbol z, initial state q_0, and final state q_F, with the following transition function:

$$\delta(q_0, a, z) = (q_0, 1z)$$
$$\delta(q_0, \lambda, z) = (q_f, z)$$
$$\delta(q_0, a, 1) = (q_0, 111)$$
$$\delta(q_0, b, z) = (q_2, z)$$
$$\delta(q_0, b, 1) = (q_1, \lambda)$$
$$\delta(q_1, b, 1) = (q_1, \lambda)$$
$$\delta(q_1, \lambda, z) = (q_f, z)$$
$$\delta(q_2, b, z) = (q_2, z)$$
$$\delta(q_2, \lambda, z) = (q_f, z)$$



a. Sketch a transition graph for M

Answer: see diagram to the right

b. Determine which of the following strings are accepted by M: $bb, abb, bbb, aabb$. (Show your work!)

$(q_0, bb, z) \vdash (q_2, bb, z) \vdash (q_2, b, z) \vdash (q_2, \lambda, z) \vdash (q_f, \lambda, z)$ - so bb is accepted

$(q_0, abb, z) \vdash (q_0, bb, 1z) \vdash (q_1, b, z)$ - we're stuck here, so abb is not accepted

bbb is accepted, aabb is not accepted

c. Describe the language that is accepted by M.

Answer: it's the set of all strings of the form $b^n, n \geq 0$ or $a^n b^{2n-1}, n \geq 1$

2. Consider the grammar

$$S \to AB \mid AC \mid a$$
$$A \to SA \mid AA \mid a$$
$$B \to BA \mid CC$$
$$C \to b$$

Use the CYK algorithm to show that the string $abbaa$ is generated by this grammar.

Answers: sets of variables should be as follows:
$V_{11} = \{S, A\}, V_{22} = \{C\}, V_{33} = \{C\}, V_{44} = \{S, A\}, V_{55} = \{S, A\}$
$V_{12} = \{S\}, V_{23} = \{B\}, V_{34} = \emptyset, V_{45} = \{A\}$
$V_{13} = \{S\}, V_{24} = \{B\}, V_{35} = \emptyset$
$V_{14} = \{S, A\}, V_{25} = \{B\}$
$V_{15} = \{S, A\}$

Since the start variable is in V_{15}, $abbaa$ is in the language generated by this grammar.

3.

a. Design (with a list of transition rules or a transition graph, whichever you prefer) a Turing machine, M, with tape alphabet $\Gamma = \{a, b, \sqcup\}$, that carries out the following algorithm:

- If the input string begins with a, immediately halt in a favorable state, without modifying the input, and with the read/write head observing the first letter of the input string
- If the input begins with b, replace it with an a; then, move through the entire string, replacing each b with an a and each a with a b, and then halt in a favorable state, with the read/write head observing the first letter of the modified string
- If the input string is empty, move the read/write head to the right forever, never entering a halting state

b. Using “building block” machines from the Combining Turing Machines handout, create a block diagram for the machine you designed in part (a).

Note on block diagrams (discussed in class, but maybe not clear from the handout) – in the design of a block diagram, it is usually assumed that whenever a “building block” machine halts, it halts in a favorable state, so you don’t need to make this explicit in your diagram.

Solution: Discussed in class.

4. Consider the grammar, G, with productions

$$S \to aSB \mid bA$$
$$A \to aAB \mid aB$$
$$B \to a \mid b$$

a. Design (with a list of transition rules or a transition graph, whichever you prefer) a npda that accepts the language generated by G.

Solution: The following transitions (with initial state q_0 and final state q_F) describe an npda that accepts $L(G)$:

$$\delta(q_0, \lambda, z) = (q_1, Sz)$$
$$\delta(q_1, a, S) = (q_1, SB)$$
$$\delta(q_1, b, S) = (q_1, A)$$
$$\delta(q_1, a, A) = (q_1, AB)$$
$$\delta(q_1, a, A) = (q_1, B)$$
$$\delta(q_1, a, B) = (q_1, \lambda)$$
$$\delta(q_1, b, B) = (q_1, \lambda)$$
$$\delta(q_1, \lambda, z) = (q_F, z)$$

b. Find a derivation for the string *abaabab* in G, and show the corresponding sequence moves made by the npda you designed in part (a) by which it accepts the input *abaabab*

Solution: $S \to aSB \to abAB \to abaABB \to abaaBBB \to abaabBB \to abaabaB \to abaabab$

$$(q_0, abaabab, z) \vdash (q_1, abaabab, Sz) \vdash (q_1, baabab, SBz) \vdash (q_1, aabab, ABz) \vdash (q_1, abab, ABBz)$$
$$\vdash (q_1, bab, BBBz) \vdash (q_1, ab, BBz) \vdash (q_1, b, Bz) \vdash (q_1, \lambda, z) \vdash (q_F, \lambda, z)$$

5. Use the encoding process described in Section 10.4 of the text to encode the Turing machine with the following set of transition rules:

$$\delta(q_1, a_2) = (q_2, a_1, L)$$
$$\delta(q_1, a_3) = (q_3, a_2, R)$$
$$\delta(q_3, a_2) = (q_2, a_1, L)$$

Answer: if we use 1 for L and 11 for R (this was not entirely clear from the text), we'd have:

10110110101 00 101110111011011 00 1110110110101 000

Note: The convention of 00 between rules and 000 at the end of the machine's code was from class, not from the text. If your answer is similar to the one here based on different (but reasonable) assumptions, then it is fine.

Comment: The main point here is to reiterate the idea that, given a consistent set of rules for doing so, every Turing machine may be converted into a unique binary string. In this way, we can show that the set of all Turing machines is countable.