Computability and Complexity, CISC 462 - Assignment 1 (Fall 2018, K. Salomaa)
Due in lecture 10:30 AM, Thursday September 27

1. Consider the Turing machine

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$$

where $Q = \{q_0, q_{accept}, q_{reject}\}$, $\Sigma = \{a, b\}$, $\Gamma = \Sigma \cup \{\sqcup\}$ and δ is defined by
$\delta(q_0, a) = (q_0, b, R)$
$\delta(q_0, b) = (q_0, b, L)$
$\delta(q_0, \sqcup) = (q_{accept}, \sqcup, L)$.

Give the sequence of configurations that M enters when started on the input string *abba*. (Recall from the definition of TM computations: what happens if M attempts to move left when the tape head is scanning the leftmost square? Check the definition in the textbook.)

What is the language recognized by M? Here it is sufficient to describe the language as a set, and it is not required to actually prove that your answer is correct.

2. Consider the Turing machine M_0 depicted in Figure 1 (next page) with input alphabet $\Sigma = \{0\}$ and tape alphabet $\Gamma = \{0, x, \sqcup\}$. In each of the parts, give the sequence of configurations that M_0 enters when started on the indicated input string.

   (a) 0

   (b) 00

   (c) 000

   (d) 000000

3. Recall that in CISC-223 we discussed the notion of "deterministic state transition diagram", a.k.a. "deterministic finite automaton" (DFA)[1]. DFAs are described using state diagrams analogously as we do for Turing machines (TMs) but in DFA state diagrams the transitions are labeled simply by the input symbol consumed by the machine.

   Here (see (a) below) we consider the question how a DFA state diagram can be converted to an equivalent TM state diagram[2]. The TM should operate "in the same way" as the original DFA and the transformation should just guarantee that the TM state diagram is syntactically correct.

[1] DFAs are discussed also in chapter 1 of our textbook.
[2] The converse transformation is, in general, not possible.

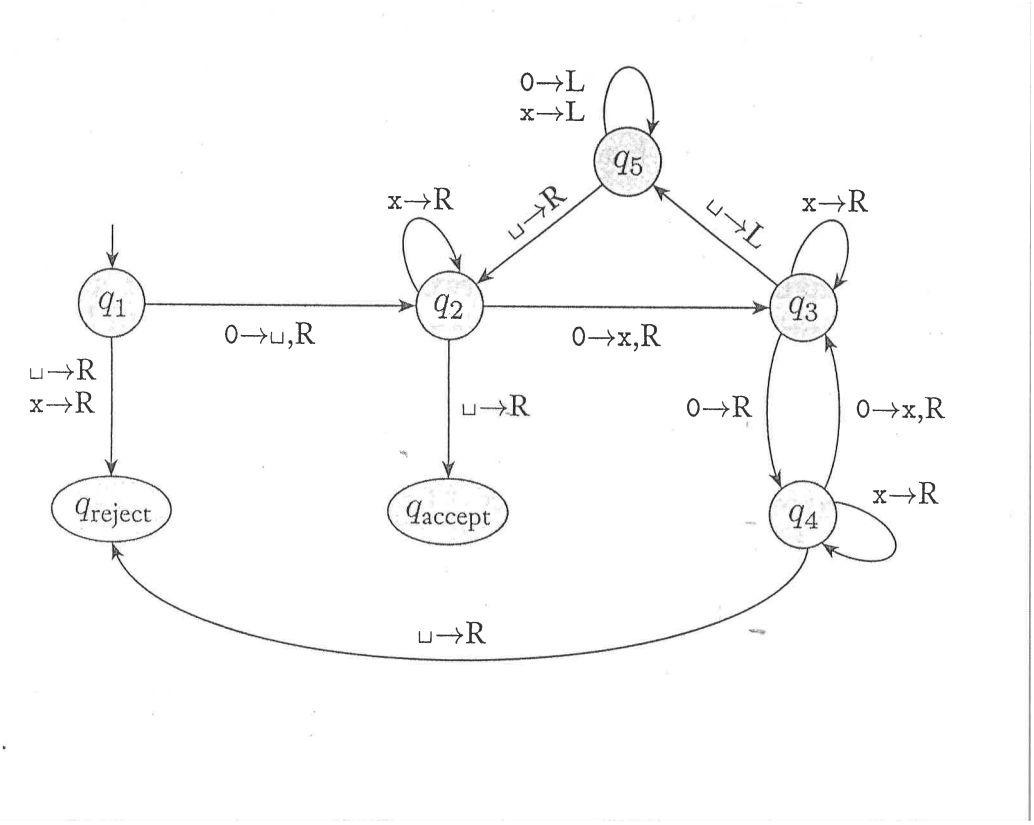

Figure 1: Turing machine M_0 for question 2. The tape alphabet is $\{0, x, \sqcup\}$.

(a) Consider a DFA A given by its state diagram. Explain how from the state diagram of A we obtain a "TM state diagram" that describes a TM M such that $L(M) = L(A)$. In particular, you should explain the following:

- If A has a transition from state q to state p labeled by input symbol b, what should be the corresponding transition label in the state diagram for M?
- How is acceptance of a string taken care of in M? (Recall that a TM halts when it enters an accept state.)

(b) Give a complete state diagram for a deterministic Turing machine that decides the language over $\Sigma = \{a, b\}$ consisting of all strings of length at least three.

4. Recall from CISC-223 the notion of a *pushdown automaton* (PDA).[3]

A *two-stack PDA* A is an extension where the PDA uses two stacks (or pushdown stores). Each transition of A can perform a push or pop operation independently on each of the two stacks (or not change the contents of the stack).

[3]Pushdown automata are described also in chapter 2 of our textbook.

Show how a two-stack PDA can simulate a Turing machine (as defined in Def. 3.3, page 168). A high level description of how the simulation works is sufficient.

5. Give a *complete construction* (that is, a state transition diagram) of a single-tape deterministic TM M that performs the following "right shift" operation.

   - As input M receives a string over $\{a, b\}$.
   - M halts (in the "accept" state) after shifting the entire input string one square to the right. After the shift, the first tape square will contain the blank symbol.
   - For example, if M receives *aabab* as input, the tape contents at the end of the computation is $\sqcup$*aabab* (followed by more blanks)
   - *Give the sequence of configurations* that your machine enters when started on input string *aabab*

6. Give an implementation description of a single-tape deterministic TM that decides the following language over the alphabet $\Sigma = \{a, b\}$:

   $$\{w \in \Sigma^* \mid |w|_a = 2 \cdot |w|_b\}$$

   Here $|w|_x$ denotes the number of symbols x ($\in \Sigma$) occurring in the string w.

   Note: "implementation description" means that you can use English prose to describe the way that the Turing machine moves its head and rewrites symbols on its tape.

Regulations on Assignments

- As described on the course homepage, all assignments must be based on *individual work.*
- The assignments are graded according to the correctness, preciseness and elegance of the solutions.
- If, as part of your solution, you rely on results from the textbook you should clearly state which result(s) you are using.
- Each question is worth 10 marks and the assignment is marked out of 60 marks.