Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
P. Schneider, C. Zheng

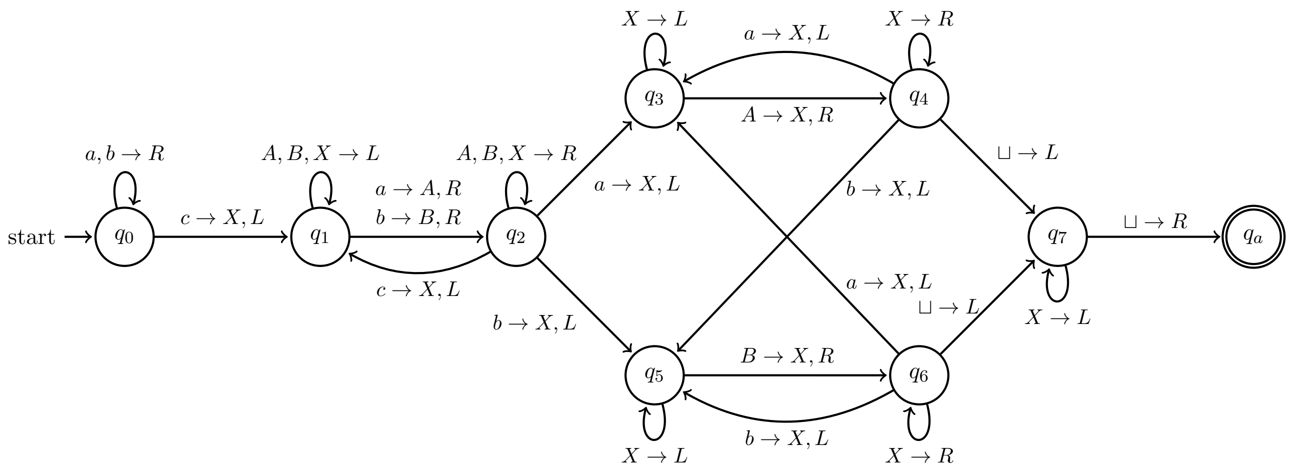
December 5, 2016

Theoretical Computer Science - Bridging Course
Winter Term 2016
Exercise Sheet 6

Hand in (electronically or hard copy) before your weekly meeting but not later than 23:59, Wednesday, December 7, 2016

Exercise 1: Interpret Turing Machines (4.5 + 1.5 points)

Consider the following Turing machine M given as state diagram. An edge label $(\mathcal{A} \to Y, H)$ with $\mathcal{A} \subseteq \Gamma, Y \in \Gamma, H \in \{L, R\}$ means that if a symbol $X \in \mathcal{A}$ is read on the tape, it is replaced with Y and the read/write head of the Turing machine is moved according to H. In this notation the symbol Y can be omitted, which means the Turing machine leaves the symbol $X \in \mathcal{A}$ it currently reads unchanged. If M reads $\sqcup \in \Gamma$ this means that the current cell on the tape is empty. M halts in a state if there is no eligible transition from that state. An input string written on M's tape before being started is accepted, if and only if M halts in q_a. Note that we consider a Turing machine with an infinite tape in both directions, which is equally mighty and often easier to work with.



(a) Simulate M with inputs $s_1 = abccba$, $s_2 = aacccaa$, $s_3 = abccab$ until it halts. Give the sequence of configurations which the simulation of M on s_i produces. You may omit configurations where no symbol is replaced. Which of these inputs are in $L(M)$?

(b) Give the language $L(M)$ of strings that are accepted by M.

Exercise 2: Construct Turing Machines (4+3 points)

(a) Draw a state diagram of a Turing machine M recognizing the language $\{a^n b^n a^n \mid n \geq 0\}$ over the alphabet $\Sigma = \{a, b\}$.

(b) Simulate your Turing machine with the input strings $s_1 = aabbaa$ and $s_2 = abaaa$. Give the sequence of configurations for these inputs. You may omit configurations where no symbol is replaced.

Exercise 3: RE and Set Operations (3+3 points)

The concept of Turing machines is very powerful. Indeed there are 'universal' Turing machines that can emulate other Turing machines M on an input s, when given a codification of $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ together with the input s. Such universal Turing machines can base their own behavior on the behavior of the Turing-machines they simulate.
A Turing machine can even do several tasks 'simultaneously' due to the concept of multitape Turing machines which have several tapes and read/write heads that can operate independently on their respective tapes. Such a multitape Turing machine can in turn be emulated by a single tape Turing machine.
Now consider the set of Recursive Enumerable Languages (RE). $L \in$ RE, iff there exists a Turing machine M such that: M with input string s halts in the accepting state, if and only if $s \in L$ (in short: $L = L(M)$). Based on the insight above, show the following.

(a) If $L_1, L_2 \in$ RE then $L_1 \cup L_2 \in$ RE.

(b) If $L_1, L_2 \in$ RE then $L_1 \cap L_2 \in$ RE.

Hint: Argument informally what a Turing machine for the respective language should do. Keep in mind that for any Language $L \in RE$ the corresponding Turing machine M with $L(M) = L$ does not necessarily need to halt for input $s \notin L$.