Assignment 6: Machines and traces

Solved practice problems numbered in red, assigned problems in green.

A. **(i)** Construct an LBA that recognizes the language

$$L = \{\mathtt{a}^n\mathtt{b}^m\mathtt{c}^{n+m} \mid n, m \geqslant 1\}.$$

Solution. We match a string of $\mathtt{a}$'s and $\mathtt{b}$'s with a string of $\mathtt{c}$'s, deleting a terminal $\mathtt{c}$ for each initial $\mathtt{a}$ and $\mathtt{b}$. Success requires that $\mathtt{a}$'s precede $\mathtt{b}$'s and $\mathtt{b}$'s precede $\mathtt{c}$'s. Here is an LBA in modular format, with S as the start state and Y as the accept state.

$$
\begin{array}{lll|lll}
S \xrightarrow{\ >\,(+)\ } M & & & C \xrightarrow{\ \mathtt{c}\,(+)\ } C & & \\
M \xrightarrow{\ \mathtt{a}\,(>)\ } A & & & C \xrightarrow{\ \sqcup\,(-)\ } R & & \\
M \xrightarrow{\ \mathtt{b}\,(>)\ } B & & & R \xrightarrow{\ \mathtt{c}\,(\sqcup)\ } W & & \\
A \xrightarrow{\ \sigma\,(+)\ } A & (\sigma = >, \mathtt{a}) & & W \xrightarrow{\ \sigma\,(-)\ } W & (\sigma \neq >) & \\
A \xrightarrow{\ \mathtt{b}\,(+)\ } B & & & W \xrightarrow{\ >\,(+)\ } M & & \\
B \xrightarrow{\ \mathtt{b}\,(+)\ } B & & & M \xrightarrow{\ \sqcup\,(\sqcup)\ } Y & & \\
B \xrightarrow{\ \mathtt{c}\,(+)\ } C & & & & &
\end{array}
$$

(ii) Give the computation-trace for $\mathtt{abcc}$.

Solution.

$$
\begin{array}{ll|l|l}
(S, \underline{>}\mathtt{abcc}\sqcup) & \Rightarrow (M, >\underline{\mathtt{a}}\mathtt{bcc}\sqcup) & \Rightarrow (W, >>\mathtt{bc}\underline{\sqcup}\,\sqcup) & \Rightarrow (C, >>>\mathtt{c}\underline{\sqcup}\,\sqcup) \\
 & \Rightarrow (A, >\underline{>}\mathtt{bcc}\sqcup) & \cdots & \Rightarrow (R, >>>\underline{\mathtt{c}}\sqcup\,\sqcup) \\
 & \Rightarrow (A, >>\underline{\mathtt{b}}\mathtt{cc}\sqcup) & \Rightarrow (W, >\underline{>}\mathtt{bc}\sqcup\,\sqcup) & \Rightarrow (W, >>>\underline{\sqcup}\,\sqcup\,\sqcup) \\
 & \Rightarrow (B, >>\mathtt{b}\underline{\mathtt{c}}\mathtt{c}\sqcup) & \Rightarrow (M, >>\underline{\mathtt{b}}\mathtt{c}\sqcup\,\sqcup) & \Rightarrow (W, >>\underline{>}\sqcup\,\sqcup\,\sqcup) \\
 & \Rightarrow (C, >>\mathtt{bc}\underline{\mathtt{c}}\sqcup) & \Rightarrow (B, >>\underline{>}\mathtt{c}\sqcup\,\sqcup) & \Rightarrow (M, >>>\underline{\sqcup}\,\sqcup\,\sqcup) \\
 & \Rightarrow (C, >>\mathtt{bcc}\underline{\sqcup}) & \Rightarrow (B, >>>\underline{\mathtt{c}}\sqcup\,\sqcup) & \Rightarrow (Y, >>>\underline{\sqcup}\,\sqcup\,\sqcup) \\
 & \Rightarrow (R, >>\mathtt{bc}\underline{\mathtt{c}}\sqcup) & &
\end{array}
$$

(iii) Give the computation-trace for **abaccc**.

Solution. $(S, \underline{>}\texttt{abaccc}\sqcup) \Rightarrow (M, \texttt{>}\underline{\texttt{a}}\texttt{baccc}\sqcup)$
$\Rightarrow (A, \texttt{>}\underline{>}\texttt{baccc}\sqcup)$
$\Rightarrow (A, \texttt{>>}\underline{\texttt{b}}\texttt{accc}\sqcup)$
$\Rightarrow (B, \texttt{>>b}\underline{\texttt{a}}\texttt{ccc}\sqcup)$

The last configuration is terminal, because there is no transition for state B and symbol a.

1. (20+10%)

(a) Construct an LBA recognizing $L = \{w \cdot w^R \mid w \in \{\texttt{a}, \texttt{b}\}^* \}$, where w^R is the reverse of w. Define your LBA in a modular format. [Hint: This is similar to the problem of accepting the strings $\texttt{a}^n\texttt{b}^n$ considered in class.]

(b) Give the computation trace of your acceptor for **abba**.

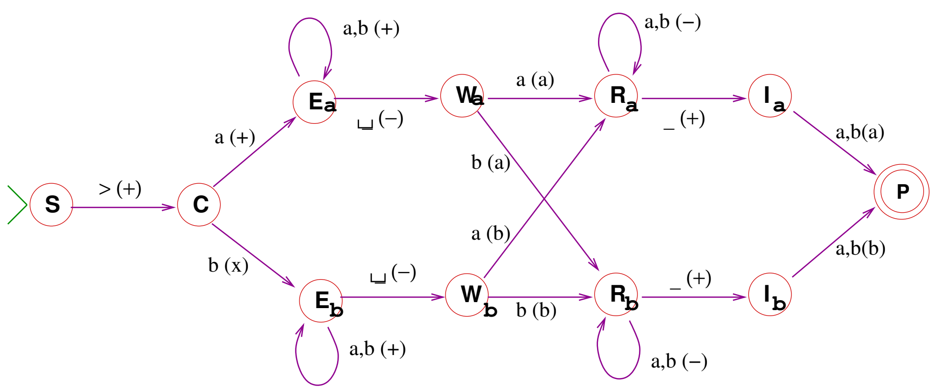
B.

(i) Construct a Turing transducer over the alphabet Σ that swaps the first and last input symbols. For example, for input **abcde** the output is **ebcda**. (Single-letter strings and ε are mapped to themselves.)

Solution.

$$
\begin{array}{rcll}
S & \xrightarrow{>\,(+)} & C & \\
C & \xrightarrow{\sigma\,(+)} & F_\sigma & (\sigma \in \Sigma) \\
F_\sigma & \xrightarrow{\tau\,(+)} & F_\sigma & (\sigma, \tau \in \Sigma) \\
F_\sigma & \xrightarrow{\sqcup\,(-)} & R_\sigma & (\sigma \in \Sigma) \\
R_\sigma & \xrightarrow{\tau\,(\sigma)} & B_\tau & (\tau, \sigma \in \Sigma) \\
B_\tau & \xrightarrow{\sigma\,(-)} & B_\tau & (\tau, \sigma \in \Sigma) \\
B_\tau & \xrightarrow{>\,(+)} & L_\tau & (\tau \in \Sigma) \\
L_\tau & \xrightarrow{\sigma\,(\tau)} & P & (\tau \in \Sigma)
\end{array}
$$

Here is a transition diagram for the transducer above for $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.



(ii) Give the trace of your transducer for the input string **abb**

Solution.

$$
\begin{array}{lll|l}
(S, \underline{>}\mathtt{abb}) & \Rightarrow & C, >\underline{\mathtt{a}}\mathtt{bb}) & \Rightarrow (R_b, >\mathtt{a}\underline{\mathtt{b}}\mathtt{a}\sqcup) \\
 & \Rightarrow & (F_a, >\mathtt{a}\underline{\mathtt{b}}\mathtt{b}) & \Rightarrow (R_b, >\underline{\mathtt{a}}\mathtt{ba}\sqcup) \\
 & \Rightarrow & (F_a, >\mathtt{ab}\underline{\mathtt{b}}) & \Rightarrow (B_b, \underline{\sqcup}\mathtt{aba}\sqcup) \\
 & \Rightarrow & (F_a, >\mathtt{abb}\underline{\sqcup}) & \Rightarrow (L_b, >\underline{\mathtt{a}}\mathtt{ba}\sqcup) \\
 & \Rightarrow & (W_a, >\mathtt{ab}\underline{\mathtt{b}}\sqcup) & \Rightarrow (P, >\underline{\mathtt{b}}\mathtt{ba}\sqcup) \\
 & \Rightarrow & (R_b, >\mathtt{ab}\underline{\mathtt{a}}\sqcup) &
\end{array}
$$

2. (20+20%)

(a) Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Construct a Turing transducer that for input $\#w\sigma$ outputs $\sigma\#w$ (where $\sigma \in \Sigma$, $w \in \Sigma^*$, and $\#$ is a symbol $\notin \Sigma$).

(b) Building on your previous transducer obtain a transducer that for input $\#w$ outputs $w^R\#$, where $w \in \Sigma^*$ and w^R is the reverse of w.
Example: for input #`abcde` the output is `edcba`#.

3. (30%) Define a Turing transducer T for the partial function $f : \{0, 1\}^* \rightharpoonup \{0, 1\}^*$ that on input w of even length returns the first half of w and is undefined for input of odd length. T should terminate on all input, though not necessarily with the print state.