Assignment 6: Machines and traces

Solutions.

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

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

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

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.

S	$\xrightarrow{> (+)}$	M		C	$\xrightarrow{\mathsf{c}\ (+)}$	C	
M	$\xrightarrow{\mathsf{a}\ (>)}$	A		C	$\xrightarrow{\sqcup\ (-)}$	R	
M	$\xrightarrow{\mathsf{b}\ (>)}$	B		R	$\xrightarrow{\mathsf{c}\ (\sqcup)}$	W	
A	$\xrightarrow{\sigma\ (+)}$	A	$(\sigma = >, \mathsf{a})$	W	$\xrightarrow{\sigma\ (-)}$	W	$(\sigma \neq >)$
A	$\xrightarrow{\mathsf{b}\ (+)}$	B		W	$\xrightarrow{> (+)}$	M	
B	$\xrightarrow{\mathsf{b}\ (+)}$	B		M	$\xrightarrow{\sqcup\ (\sqcup)}$	Y	
B	$\xrightarrow{\mathsf{c}\ (+)}$	C					

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

Solution.

$$
\begin{aligned}
(S, \underline{>}\mathsf{abcc}\sqcup) &\Rightarrow (M, >\underline{\mathsf{a}}\mathsf{bcc}\sqcup) \\
&\Rightarrow (A, >\underline{>}\mathsf{bcc}\sqcup) \\
&\Rightarrow (A, >>\underline{\mathsf{b}}\mathsf{cc}\sqcup) \\
&\Rightarrow (B, >>\mathsf{b}\underline{\mathsf{c}}\mathsf{c}\sqcup) \\
&\Rightarrow (C, >>\mathsf{bc}\underline{\mathsf{c}}\sqcup) \\
&\Rightarrow (C, >>\mathsf{bcc}\underline{\sqcup}) \\
&\Rightarrow (R, >>\mathsf{bc}\underline{\mathsf{c}}\sqcup)
\end{aligned}
$$

$$
\begin{aligned}
&\Rightarrow (W, >>\mathsf{bc}\underline{\sqcup}\,\sqcup) \\
&\qquad \cdots \\
&\Rightarrow (W, >\underline{>}\mathsf{bc}\sqcup\,\sqcup) \\
&\Rightarrow (M, >>\underline{\mathsf{b}}\mathsf{c}\sqcup\,\sqcup) \\
&\Rightarrow (B, >>\underline{>}\mathsf{c}\sqcup\,\sqcup) \\
&\Rightarrow (B, >>>\underline{\mathsf{c}}\sqcup\,\sqcup)
\end{aligned}
$$

$$
\begin{aligned}
&\Rightarrow (C, >>>\mathsf{c}\underline{\sqcup}\,\sqcup) \\
&\Rightarrow (R, >>>\underline{\mathsf{c}}\sqcup\,\sqcup) \\
&\Rightarrow (W, >>>\underline{\sqcup}\,\sqcup\,\sqcup) \\
&\Rightarrow (W, >>\underline{>}\sqcup\,\sqcup\,\sqcup) \\
&\Rightarrow (M, >>>\underline{\sqcup}\,\sqcup\,\sqcup) \\
&\Rightarrow (Y, >>>\underline{\sqcup}\,\sqcup\,\sqcup)
\end{aligned}
$$

(iii) Give the computation-trace for $\texttt{abaccc}$.

Solution.

$$\begin{aligned}(S, \underline{>}\texttt{abaccc}\sqcup) &\Rightarrow (M, >\underline{\texttt{a}}\texttt{baccc}\sqcup)\\ &\Rightarrow (A, >\underline{>}\texttt{baccc}\sqcup)\\ &\Rightarrow (A, >>\underline{\texttt{b}}\texttt{accc}\sqcup)\\ &\Rightarrow (B, >>\texttt{b}\underline{\texttt{a}}\texttt{ccc}\sqcup)\end{aligned}$$

The last configuration is terminal, because there is no transition for state B and symbol $\texttt{a}$.

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.]

Solution. Start state S, accept state Y.

$$\begin{array}{rcll}
S & \xrightarrow{>\,(+)} & M & \\
M & \xrightarrow{\sigma\,(>)} & C_\sigma & \sigma = \texttt{a},\texttt{b}) \\
C_\sigma & \xrightarrow{\tau\,(+)} & C_\sigma & (\sigma = \texttt{a},\texttt{b},\ \tau \neq \sqcup) \\
C_\sigma & \xrightarrow{\sqcup\,(-)} & H_\sigma & (\sigma = \texttt{a},\texttt{b}) \\
H_\sigma & \xrightarrow{\sigma\,(\sqcup)} & R & (\sigma = \texttt{a},\texttt{b}) \\
R & \xrightarrow{\sigma\,(-)} & R & (\sigma = \texttt{a},\texttt{b}) \\
R & \xrightarrow{>\,(+)} & M & \\
M & \xrightarrow{\sqcup\,(\sqcup)} & Y &
\end{array}$$

(b) Give the computation trace of your acceptor for $\texttt{abba}$.

Solution.

$(S, \underline{>}\texttt{abba})$

$$\begin{array}{llllll}
\Rightarrow & (M, >\underline{\texttt{a}}\texttt{bba}) & \Rightarrow & (R, >>\texttt{bb}\underline{\sqcup}\,\sqcup) & \Rightarrow & (C_b, >>>\texttt{b}\underline{\sqcup}\,\sqcup) \\
\Rightarrow & (C_a, >\underline{>}\texttt{bba}) & \Rightarrow & (R, >>\texttt{b}\underline{\texttt{b}}\sqcup\,\sqcup) & \Rightarrow & (H_b, >>>\underline{\texttt{b}}\sqcup\,\sqcup) \\
\Rightarrow & (C_a, >>\underline{\texttt{b}}\texttt{ba}) & \Rightarrow & (R, >>\underline{\texttt{b}}\texttt{b}\sqcup\,\sqcup) & \Rightarrow & (R, >>>\underline{\sqcup}\,\sqcup\,\sqcup) \\
\Rightarrow & (C_a, >>\texttt{b}\underline{\texttt{b}}\texttt{a}) & \Rightarrow & (R, >\underline{>}\texttt{bb}\sqcup\,\sqcup) & \Rightarrow & (R, >>\underline{>}\sqcup\,\sqcup) \\
\Rightarrow & (C_a, >>\texttt{bb}\underline{\texttt{a}}) & \Rightarrow & (M, >>\underline{\texttt{b}}\texttt{b}\sqcup\,\sqcup) & \Rightarrow & (M, >>>\underline{\sqcup}\,\sqcup\,\sqcup) \\
\Rightarrow & (C_a, >>\texttt{bba}\underline{\sqcup}) & \Rightarrow & (C_b, >>\underline{>}\texttt{b}\sqcup\,\sqcup) & \Rightarrow & (Y, >>>\underline{\sqcup}\,\sqcup\,\sqcup) \\
\Rightarrow & (H_a, >>\texttt{bb}\underline{\texttt{a}}\sqcup) & \Rightarrow & (C_b, >>>\underline{\texttt{b}}\sqcup\,\sqcup) & &
\end{array}$$

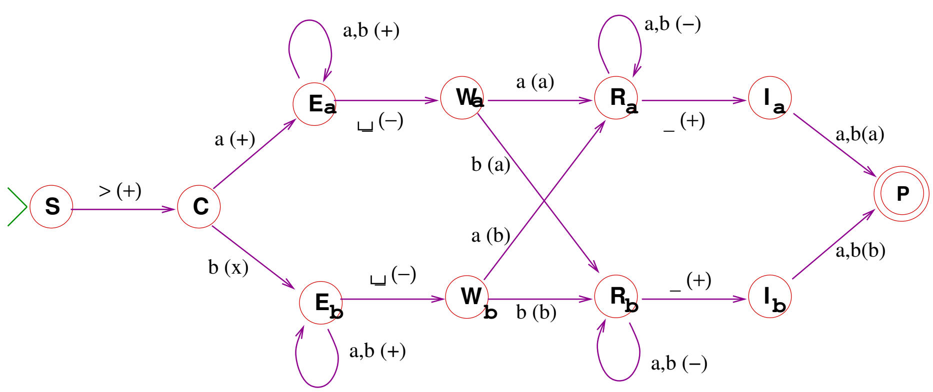
B.

(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.

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

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 $\notin \Sigma$).

Solution.

$$\begin{array}{rcll} S & \xrightarrow{\sigma(+)} & S & (\sigma \neq \sqcup) \\ S & \xrightarrow{\sqcup(-)} & R & \\ R & \xrightarrow{\sigma(\sqcup)} & M_\sigma & \\ M_\sigma & \xrightarrow{\tau(-)} & M_\sigma & (\tau \neq >) \\ M_\sigma & \xrightarrow{>(+)} & I_\sigma & \\ I_\sigma & \xrightarrow{\tau(\sigma)} & N_\tau & (\tau \neq \sqcup) \\ N_\tau & \xrightarrow{\sigma(+)} & I_\tau & \\ I_\sigma & \xrightarrow{\sqcup(\sigma)} & P & \end{array}$$

(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 $\#\mathtt{abcde}$ the output is $\mathtt{edcba}\#$.

Solution.

$$\begin{array}{rcll} S & \xrightarrow{\sigma(+)} & S & (\sigma \neq \sqcup) \\ S & \xrightarrow{\sqcup(-)} & R & \\ R & \xrightarrow{\sigma(\sqcup)} & M_\sigma & (\sigma \neq \#) \\ M_\sigma & \xrightarrow{\tau(-)} & M_\sigma & (\tau \neq \#) \\ M_\sigma & \xrightarrow{\#(\sigma)} & N_\# & \\ N_\sigma & \xrightarrow{\tau(+)} & I_\sigma & \\ I_\sigma & \xrightarrow{\tau(\sigma)} & N_\tau & (\tau \neq \sqcup) \\ I_\sigma & \xrightarrow{\sqcup(-)} & R & \\ R & \xrightarrow{\#(-)} & P & \end{array}$$

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.

Solution. States and symbols are as below, with S and P the start and output states, respectively. For each $\sigma \in \Sigma$ stipulate a fresh auxiliary letter $\overline{\sigma}$, and let $\bar{\Sigma} = \{\overline{\sigma} \mid \sigma \in \Sigma\}$
Algorithm: Matching successive letters in the first half with a deletion of their mirror image in the input. Print if there is no next-letter to match (state A below). Abort if the next-letter is followed by a blank because the input has odd length (state C).

Transitions:

$$
\begin{array}{lcll}
S & \xrightarrow{\sigma\,(+)} & A & (\sigma \in \Sigma \cup \{>\}) \\
A & \xrightarrow{\sqcup\,(\sqcup)} & P & \\
A & \xrightarrow{\sigma\,(\overline{\sigma})} & B & (\sigma \in \Sigma) \\
B & \xrightarrow{\overline{\sigma}\,(+)} & C & (\overline{\sigma} \in \overline{\Sigma}) \\
C & \xrightarrow{\sigma\,(+)} & D & (\sigma \in \Sigma) \\
D & \xrightarrow{\sigma\,(+)} & D & (\sigma \in \Sigma \cup \{\sqcup\}) \\
D & \xrightarrow{\sqcup\,(-)} & E & (\sigma \in \Sigma \cup \{\sqcup\}) \\
E & \xrightarrow{\sigma\,(\sqcup)} & F & (\sigma \in \Sigma) \\
F & \xrightarrow{\sigma\,(-)} & F & (\sigma \in \Sigma \cup \bar{\Sigma}) \\
F & \xrightarrow{\overline{\sigma}\,(\sigma)} & S & (\sigma \in \Sigma)
\end{array}
$$