Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group

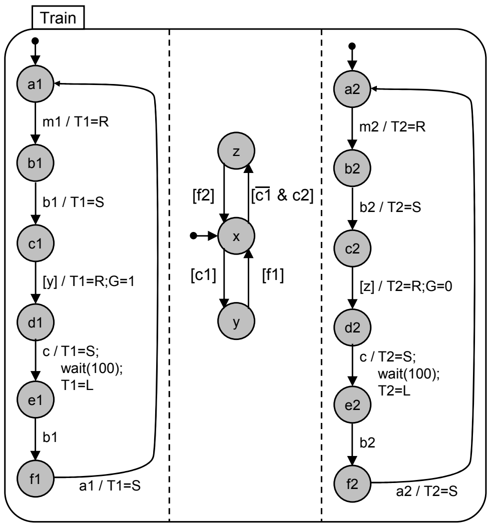
WS 2006/2007

Dr. C. Stamm / Roland Flury / Stefan Schmid

Discrete Event Systems
Solution to Exercise 6

1 The Winter Train Problem

We can model each train individually and combine the corresponding sub-states using an AND-super-state, see the figure below. Additionally, in order to "synchronize" the trains, a third sub-state is needed (shown in the middle) which implements a mutual exclusion: For instance, if there is no train between Stans and Engelberg and if train 1 is in state c1, T1 can enter the critical section and train 2 has to wait. (Notice that if both trains are in states c1 and c2 respectively, T1 has priority.)



2 CNF

a) First, insert a non-terminal S' to ensure that the start symbol S is only used on the left hand of any production. We obtain:

$$\begin{array}{rcl} S & \rightarrow & S'AS' \mid A \\ S' & \rightarrow & S'AS' \mid A \\ A & \rightarrow & 0 \mid 1 \end{array}$$

Then, replace the production $S' \rightarrow S'AS'$ with $S \rightarrow S'Q$ and $Q \rightarrow AS'$ (do the same for $S \rightarrow S'AS'$). Also, insert the terminals of A in the production $S \rightarrow A$ to obtain $S \rightarrow 0 \mid 1$. We obtain the following CNF:

$$\begin{array}{rcl} S & \rightarrow & S'Q \mid 0 \mid 1 \\ S' & \rightarrow & S'Q \mid 0 \mid 1 \\ Q & \rightarrow & AS' \\ A & \rightarrow & 0 \mid 1 \end{array}$$

b) First, ensure that the start symbol S does not appear on the right-hand side of any rule:

$$\begin{array}{rcl} S & \rightarrow & S' \\ S' & \rightarrow & T1T \mid T \\ T & \rightarrow & T0S' \mid T1S' \mid U \\ U & \rightarrow & 1U \mid \epsilon \end{array}$$

Then, remove the ϵ-production, first from the last rule to obtain

$$\begin{array}{rcl} S & \rightarrow & S' \\ S' & \rightarrow & T1T \mid T \\ T & \rightarrow & T0S' \mid T1S' \mid U \mid \epsilon \\ U & \rightarrow & 1U \mid 1 \end{array}$$

Then continue moving up the ϵ:

$$\begin{array}{rcl} S & \rightarrow & S' \\ S' & \rightarrow & T1T \mid 1T \mid T1 \mid T \mid 1 \mid \epsilon \\ T & \rightarrow & T0S' \mid 0S' \mid T1S' \mid 1S' \mid U \\ U & \rightarrow & 1U \mid 1 \end{array}$$

... until the ϵ only occurs in production rules starting from S.

$$\begin{array}{rcl} S & \rightarrow & S' \mid \epsilon \\ S' & \rightarrow & T1T \mid 1T \mid T1 \mid T \mid 1 \\ T & \rightarrow & T0S' \mid 0S' \mid T1S' \mid 1S' \mid T0 \mid 0 \mid T1 \mid 1 \mid U \\ U & \rightarrow & 1U \mid 1 \end{array}$$

Then, remove all unit-variable productions:

$$\begin{array}{lcl}
S & \rightarrow & T1T \mid 1T \mid T1 \mid T0S' \mid 0S' \mid T1S' \mid 1S' \mid T0 \mid 0 \mid 1 \mid 1U \mid \epsilon \\
S' & \rightarrow & T1T \mid 1T \mid T1 \mid T0S' \mid 0S' \mid T1S' \mid 1S' \mid T0 \mid 0 \mid 1 \mid 1U \\
T & \rightarrow & T0S' \mid 0S' \mid T1S' \mid 1S' \mid T0 \mid 0 \mid T1 \mid 1 \mid 1U \\
U & \rightarrow & 1U \mid 1
\end{array}$$

Add dyadic variable rules to replace any longer non-dyadic or non-variable production. We start by removing the non-terminals from non-variable productions:

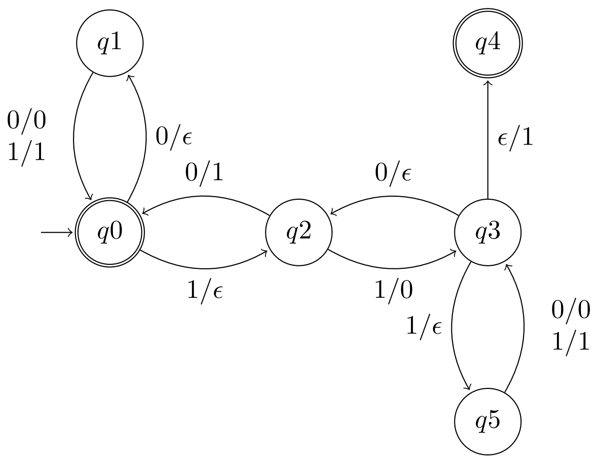
$$\begin{array}{lcl}
S & \rightarrow & TBT \mid BT \mid TB \mid TAS' \mid AS' \mid TBS' \mid BS' \mid TA \mid 0 \mid 1 \mid BU \mid \epsilon \\
S' & \rightarrow & TBT \mid BT \mid TB \mid TAS' \mid AS' \mid TBS' \mid BS' \mid TA \mid 0 \mid 1 \mid BU \\
T & \rightarrow & TAS' \mid AS' \mid TBS' \mid BS' \mid TA \mid 0 \mid TB \mid 1 \mid BU \\
U & \rightarrow & BU \mid 1 \\
A & \rightarrow & 0 \\
B & \rightarrow & 1
\end{array}$$

Finally, we split production rules whose RHS contains more than 2 non-terminals:

$$\begin{array}{lcl}
S & \rightarrow & QT \mid BT \mid TB \mid PS' \mid AS' \mid QS' \mid BS' \mid TA \mid 0 \mid 1 \mid BU \mid \epsilon \\
S' & \rightarrow & QT \mid BT \mid TB \mid PS' \mid AS' \mid QS' \mid BS' \mid TA \mid 0 \mid 1 \mid BU \\
T & \rightarrow & PS' \mid AS' \mid QS' \mid BS' \mid TA \mid 0 \mid TB \mid 1 \mid BU \\
U & \rightarrow & BU \mid 1 \\
A & \rightarrow & 0 \\
B & \rightarrow & 1 \\
P & \rightarrow & TA \\
Q & \rightarrow & TB
\end{array}$$

3 Transducer and Turing Machine

a) The proposed automaton (which is deterministic!) reads two successive symbols (bits) of the input and outputs the sum. If there is a carry-over, we end up in state $q3$, where the output is adapted accordingly.



b) The machine performs the following actions:

1 Move the head to the LSB of b. For convenience of explanation, assume there is a variable i, initially set to 0. After this step, the TM head points to $b[i]$.

2 Replace the digit at the head with A or B, if the digit is a 0 or a 1, respectively. (That's how we store the value of digit $b[i]$ and can find back later on.)

3 Move to the left until we find the + sign. Then, continue moving left until we hit the first digit. (Note: this digit corresponds to $a[i]$). Depending on the value of this digit, go into state $q5$ or $q6$, and remove the digit $a[i]$, by writing a $\Box$.

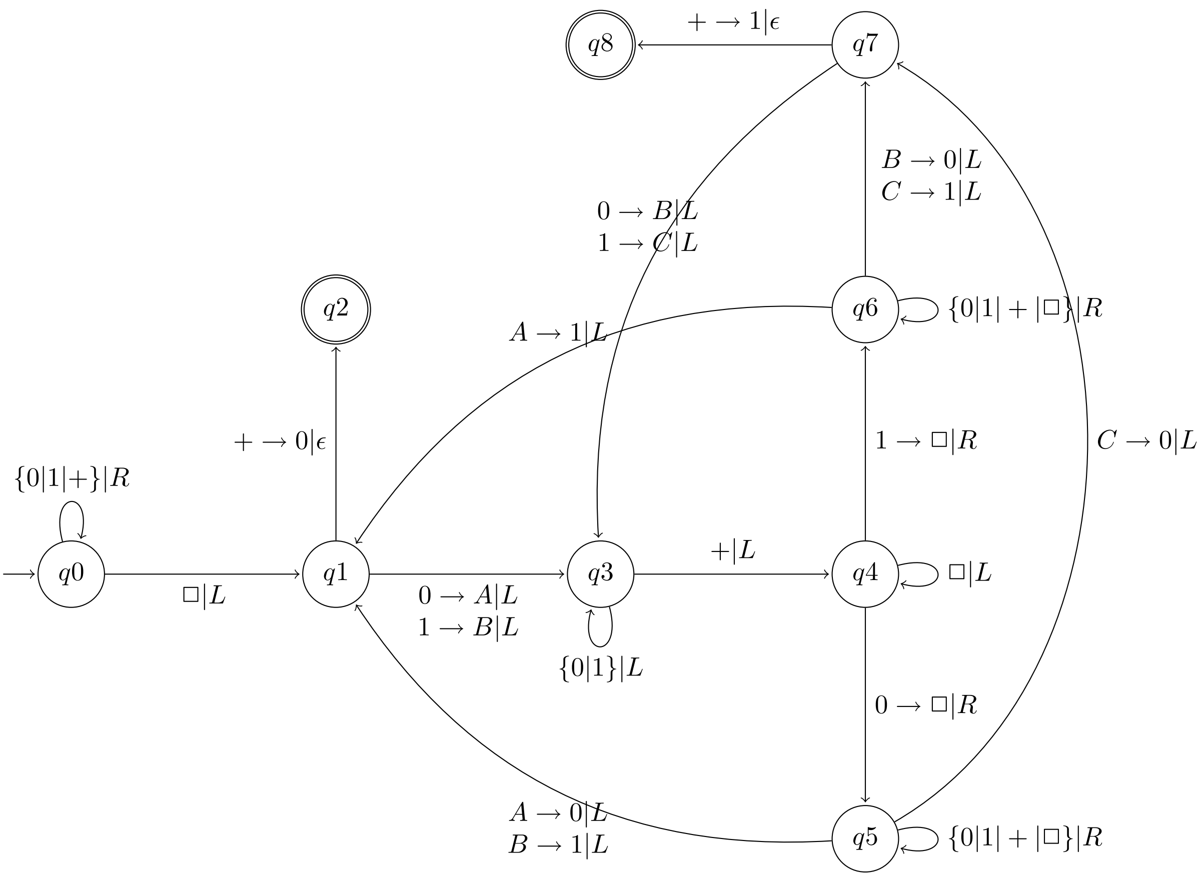
4 Move right until we hit an A or B (or C, which we explain later). At that point, we have the information of $a[i]$ and $b[i]$ and can determine the sum. If $a[i] + b[i] \geq 2$ (we get a reminder), go to state $q7$. (Note that $q1$ corresponds to $q7$: we're in $q7$ if there is a reminder, otherwise we're in $q1$.)

(5) Now, we're done with the digit at offset i. Increment i by one. (This is no action of the TM, it is only for the sake of explanation.)

6 Continue until we're in $q1$ or $q7$ and read a + sign, in which case we write the current reminder and terminate (accept).

6' Some more explanation to $q7$: In this state, we have a carry-over from the previous sum. Thus, $b[i]$ plus this carry over may already sum up to 2, in which case we write a C on the tape.

We use the following notation for transitions: $\alpha \to \beta|\gamma$: read α from the tape at the current position, then write a β and finally move left if $\gamma = L$ or move right if $\gamma = R$. We abbreviate transitions of the form $\alpha \to \alpha|\gamma$ and write $\alpha|\gamma$ (these transitions do not modify the content of the tape).



c) The proposed Turing machine decrements the value of a until $a = 0$. In each step, it adds a '1' to the output:

1 Move the TM head to the right of a and place a \$ sign. We will use this marker to return to the LSB of a.

2 Look at the LSB of a. If it is '1', we change it to 0 (transition between $q1$ and $q3$) and move to the right. Then, we continue moving to the right until we hit a $\square$, which is changed to a '1' (transition $q4$ to $q5$). Finally, we move back to the LSB of a.

3 If the LSB of a is 0, we search for the first '1' in a from the right (transition $q1$ to $q2$ and loop on $q2$).

3.1 If we find a '1', we change it to '0'. While moving back to the \$ symbol, we change all '0' to '1' (self-loop on $q3$). Then, we proceed as in point 2 after passing the \$ symbol.

3.2 If we don't find a '1' in a at all (transition $q2$ to $q6$), we start the cleanup procedure: Remove all 0 on the right of the \$ symbol, and finally remove the \$ symbol itself and move to the right of u.