Technische Universität München **Winter term 2015/16**
I7
Prof. J. Esparza / Dr. D. Suwimonteerabuth / Dr. C. Broadbent **11.12.2015**

Automata and Formal Languages — Homework 10

Due **Friday** 18th December 2015 (TA: Christopher Broadbent)

As announced in the last class, we will also have a short written quiz in the first half hour of this class. This does *not* count towards anything, so you do not need to worry. However, it will be very helpful for us to guage how well you are following the material, and it should also be useful practice for you.

Exercise 10.1

Consider a circular railway[1] divided into 8 tracks: $0 \to 1 \to \ldots \to 7 \to 0$. In the railway circulate three trains, modeled by three automata T_1, T_2, and T_3. Each automaton T_i has states $\{q_{i,0}, \ldots, q_{i,7}\}$, alphabet $\{enter[i,j] \mid 0 \leq j \leq 7\}$ (where $enter[i,j]$ models that train i enters track j), transition relation $\{(q_{i,j}, enter[i, j \oplus 1], q_{i,j\oplus 1}) \mid 0 \leq j \leq 7\}$, and initial state $q_{i,2i}$, where $\oplus$ denotes addition modulo 8. In other words, initially the trains occupy the tracks 2, 4, and 6.

Define automata $C_0, \ldots, C_7$ (the *local controllers*) to make sure that two trains can never be on the same or adjacent tracks (i.e., there must always be at least one empty track between two trains). Each controler C_j can only have knowledge of the state of the tracks $j \ominus 1$, j, and $j \oplus 1$, there must be no deadlocks, and every train must eventually visit every track. More formally, the network of automata $\mathcal{A} = \langle C_0, \ldots, C_7, T_1, T_2, T_3 \rangle$ must satisfy the following specification:

- For $j = 0, \ldots, 7$: C_j has alphabet $\{enter[i, j \ominus 1], enter[i,j], enter[i, j \oplus 1], \mid 1 \leq i \leq 3\}$.
  (C_j only knows the state of tracks $j \ominus 1$, j, and $j \oplus 1$.)
- For $i = 1, 2, 3$: $L(\mathcal{A}) \mid_{\Sigma_i} = (\ enter[i, 2i]\ enter[i, 2i \oplus 1] \ldots\ enter[i, 2i \oplus 7]\ )^*$.
  (No deadlocks, and every train eventually visits every segment.)
- For every word $w \in L(\mathcal{A})$: if $w = w_1\ enter[i,j]\ enter[i',j']\ w_2$ and $i' \neq i$, then $|j - j'| \notin \{0, 1, 7\}$.
  (No two trains on the same or adjacent tracks.)

Exercise 10.2

Construct Büchi automata and ω-regular expressions, as small as possible, recognizing the following languages over the alphabet $\{a, b, c\}$.

(1) $\{w \in \{a,b,c\}^\omega \mid \{a,b\} \supseteq inf(w)\}$

(2) $\{w \in \{a,b,c\}^\omega \mid \{a,b\} = inf(w)\}$

(3) $\{w \in \{a,b,c\}^\omega \mid \{a,b\} \subseteq inf(w)\}$

(4) $\{w \in \{a,b,c\}^\omega \mid \{a,b,c\} = inf(w)\}$

(5) $\{w \in \{a,b,c\}^\omega \mid \text{if } a \in inf(w) \text{ then } \{b,c\} \subseteq inf(w)\}$

(The set $inf(w)$ is the set of symbols that occur infinitely often in w.)

Exercise 10.3

[This question is harder, but we have already discussed representing addition of integers using finite languages, and the idea from this can be adapted].

As mentioned in the lectures, ω-words can be used to represent real numbers. Just as we can use 'decimal' (base 10) notation to represent real numbers as infinite words (e.g. 3.3^ω represents $3 + \sum_{i=1}^{\infty} \frac{3}{10^i}$, that is to say 'three and a third') so can it be convenient to use a base 2 notation.

More specifically we will represent real numbers as words in the language $(0+1)^* \,.\, (0+1)^\omega$ (over the alphabet $\{\ 0, 1, .\ \}$). This time the representation uses *most significant bit first representation.* Given $r \in \mathbb{R}$ the set of encodings of r is given by:

$$encs(r) := \{\ a_k a_{k-1} \cdots a_0 . b_1 b_2 \cdots \in (0+1)^* \,.\, (0+1)^\omega \mid r = \sum_{i=0}^{k} a_i . 2^i \quad + \quad \sum_{i=1}^{\infty} b_i . 2^{-i}\ \}$$

[1] If you want to get in a seasonal mood, feel free to pretend that the railway is located at the North Pole and has some connection to reindeer, slays and Santa Claus.

For example:

$$00100.0^\omega \in encs(4) \qquad 100.0^\omega \in encs(4)$$

$$0.01^\omega \in encs(0.5) \qquad 0.10^\omega \in encs(0.5) \qquad 10.(01)^\omega \in encs(2.33\cdots) \qquad 01.(10)^\omega \in encs(1.66\cdots)$$

We can represent a binary function on the real numbers by a language over the alphabet $\{ 0,1 \} \times \{ 0,1 \} \times \{ 0,1 \} \cup \{ . \}$ that is a subset of

$$L := (\{ 0,1 \} \times \{ 0,1 \} \times \{ 0,1 \})^* \,.\, (\{ 0,1 \} \times \{ 0,1 \} \times \{ 0,1 \})^\omega$$

In particular the addition relation can be represented by such a language L_+:

$$L_+ := \left\{ (a_k, b_k, c_k) \cdots (a_0, b_0, c_0).(a'_1, b'_1, c'_1) \cdots \in L \;\middle|\; \begin{array}{l} a_k \cdots a_0.a'_1 \cdots \in encs(r_1) \text{ and} \\ b_k \cdots b_0.b'_1 \cdots \in encs(r_2) \text{ and} \\ c_k \cdots c_0.c'_1 \cdots \in encs(r_3) \text{ for} \\ r_1, r_2, r_3 \in \mathbb{R} \text{ s.t. } r_1 + r_2 = r_3 \end{array} \right\}$$

For example, the following corresponds to the fact that $2.33\cdots + 1.66\cdots = 4$:

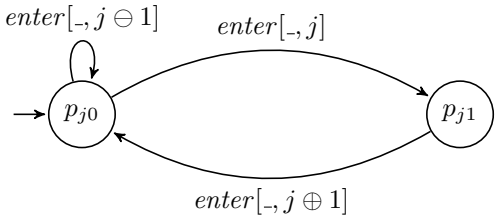
$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} . \left(\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right)^\omega \quad \in \quad L_+$$

Show that real addition is ω-regular (i.e. that L_+ is recognised by a non-deterministic Büchi automaton).

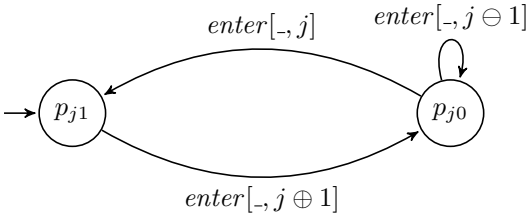
Solution 10.1

The controller C_j will be responsible for ensuring that no train enters the $(j \ominus 1)$th track when the jth track is occupied. It has two states. When C_j is in state p_{j0} the jth track is unoccupied and when it is in state p_{j1} it is occupied.

Thus for $j \notin \{ 2, 4, 6 \}$ we take C_j to be:



and for $j \in \{ 2, 4, 6 \}$ we take C_j to be:



Solution 10.2

The automata are shown in Figure 1. We sketch the argument of why the recognize the specified languages. (1) The

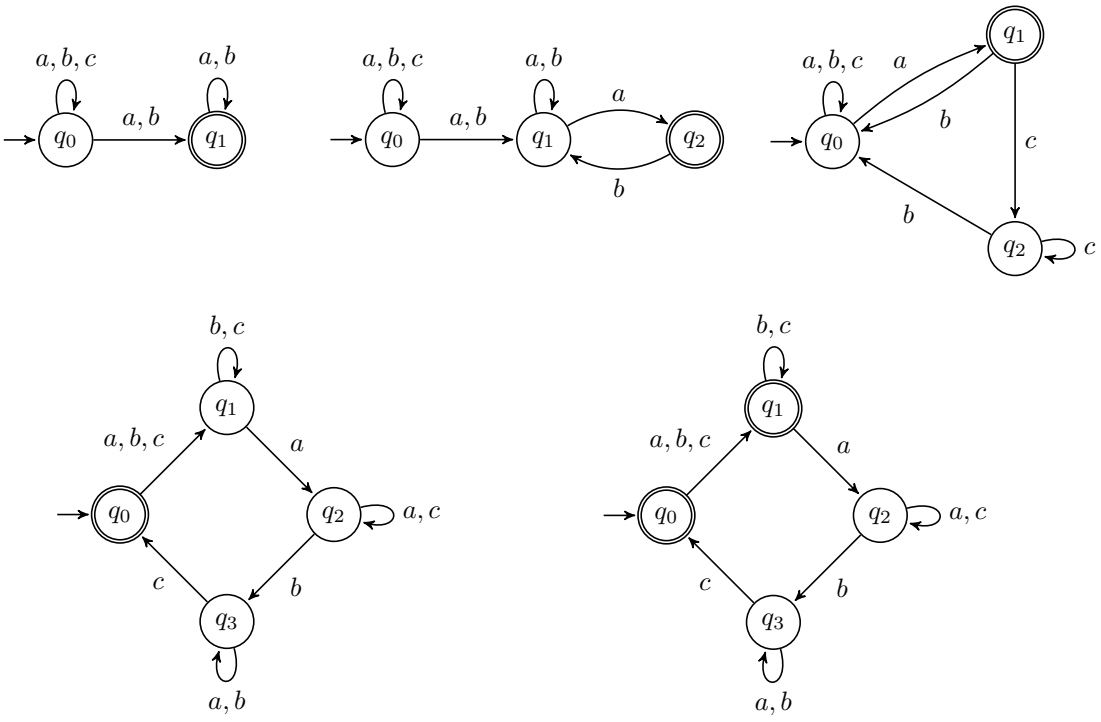


Figure 1: Automata for (1)-(5).

automaton must recognize the set of words containing only finitely many c. Every word with finitely many cs is acepted: the automaton just moves to q_1 after the last c. Conversely, every accepting run must eventually move to q_1, and so the word accepted contains only finitely many c.

(2) The automaton must recognize the ω-words containing infinitely many a, infinitely many b, but only finitely many c.

Every such ω-word is accepted by the automaton in the figure: the automaton moves to q_1 after the last c. The rest of the word contains only a and b, both inifnitely many times. So the word contains infinitely many occurrences of ab. At each

of them the automaton takes the loop through q_2. Conversely, every accepted word contains only finitely many c, because after moving to q_1 no further c can be read, and both infinitely many as and bs, because every accepting run must visit q_2 infinitely often, and each visit contributes an a and a b.
(3) The automaton recognizes all words containing infinitely many as and infinitely many bs, and either finitely or infinitely many cs. To show that every such word is accepted by the automaton we have to modify the argument of (2): now every word in the language contains infinitely many subwords of ac^*b, and the automaton accepts the word by moving to q_1 at each of these subwords. For the converse, it is clear that every visit to q_1 requires to read an a and a b, and so every accepted word contains both letters infinitely often. Notice that we cannot remove q_2 and add a self-loop labeled by c to q_1, because then the automaton would accept for instance ac^ω.
(4) The automaton recognizes all words containing infinitely many as, bs, and cs. The argument is similar to that of (2), but now we ensure that between and two visits to the final state the automaton has read at least one a, one b, and one c. Observe that the order doesn't matter: if all three letters occur infinitely often, we know that after any letter we will eventually see again any of the others.
(5) We add a new final state to the automaton for (4). Every word accepted by (4) is accepted now. The new accepting runs eventually stay on q_1, and accept all the words containing finitely many as.

Here are ω-regular expressions for the five languages:

(1) $(a+b+c)^*(a+b)^\omega$

(2) $(a+b+c)^*(aa^*bb^*)^\omega$

(3) $\big((b+c)^*a(a+c)^*b\big)^\omega$

(4) $\big((b+c)^*a(a+c)^*b(a+b)^*c\big)^\omega$

(5) $(a+b+c)^*\Big(b+c+a(a+c)^*b(a+b)^*c\Big)^\omega$