02140 Languages and Parsing: Mandatory Exercise Set 1

Martin Fränzle and Michael R. Hansen

September 22, 2004

Exercise 1

A friend, who loves to twiddle with hardware, has constructed a variant of a shift register. In his version, the bits are shifted to the right, but the *complement* of the bit being "shifted out" of the rightmost position is reinserted at the leftmost position. This can be described as follows for the four-bit case:

$$\begin{aligned} b'_3 &= \neg b_0 \\ b'_2 &= b_3 \\ b'_1 &= b_2 \\ b'_0 &= b_1 \end{aligned}$$

Here, b_0 to b_3 represent the rightmost to the leftmost bit, and the primed variants denote the values of these bits in the successor state. The transition graph of this hardware device is shown in figure 1. Your friend claims that his circuit, when started in state $b_0 = b_3 = 0, b_1 = b_2 = 1$, will never enter a state that has exactly 3 bits set to 1.

1. Model your friend's claim as a language emptiness problem of a DFA. To do so, you have to
   (a) define a state set representing the hardware states,
   (b) find an appropriate alphabet,[1]
   (c) formalize the transition function,
   (d) encode the claim by appropriate selection of the initial state and the final state set.

   Please state explicitly in what correspondence the language emptiness problem of your DFA is to the original problem (i.e., is the DFA's language empty iff your friend is right, or, vice versa, iff your friend is wrong?).
2. Construct your automaton in SML and use SML to decide whether your friend is right.
3. Consider now a six-bit variant of the shift register and the claim that the circuit will never enter a state where exactly 5 bits are set to 1, when it is started in a state where $b_0 = b_2 = b_4 = 0$ and $b_1 = b_3 = b_5 = 1$. Use SML to decide whether this claim is right or wrong.
4. How can you use the automata procedures of Sept. 20 for finding out about the cycle length of the shift register (i.e., whether it will go through all 2^6 possible states when started in state $b_0 = b_2 = b_4 = 0$ and $b_1 = b_3 = b_5 = 1$, or — if not — how many states it will cycle through)? Describe your idea and build a program calculating the cycle length.

[1] Note that the circuit has no external inputs and thus has exactly one outgoing transition per state.

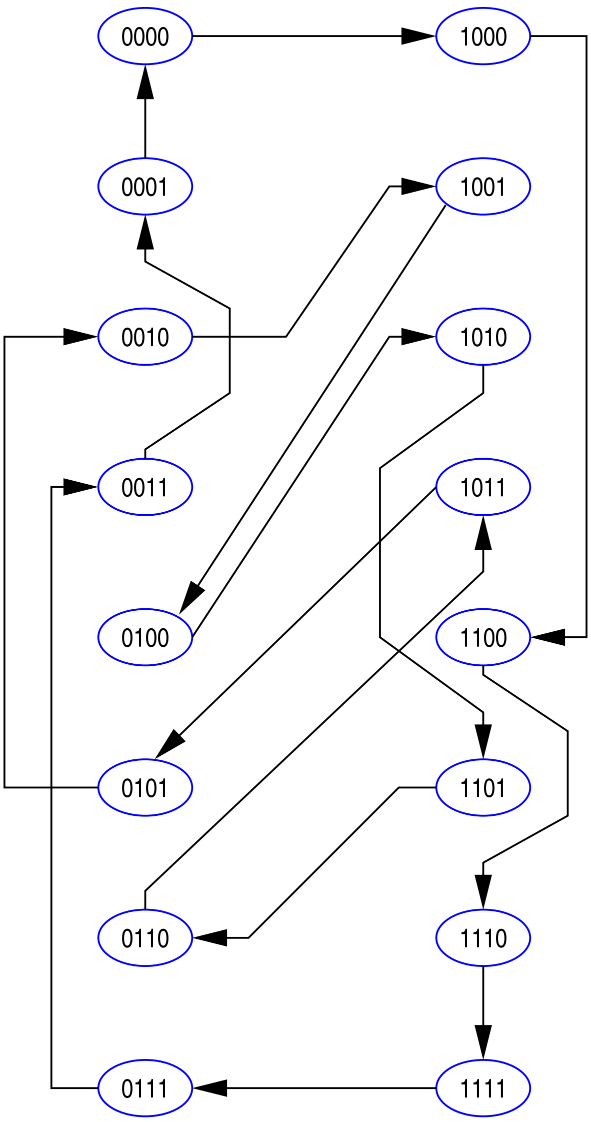

Figure 1: Transition graph of the modified shift register

Exercise 2

Let $A = (Q_A, \Sigma, \delta_A, q_0^A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, q_0^B, F_B)$ be DFAs. The product automaton $A \otimes B$ is defined as $A \otimes B = (Q_A \times Q_B, \Sigma, \delta_{A\otimes B}, (q_0^A, q_0^B), F_A \times F_B)$ with

$$\delta_{A\otimes B}((p, q), a) = (\delta_A(p, a), \delta_B(q, a))$$

for each $p \in Q_A$, $q \in Q_B$ and $a \in \Sigma$. It is easy to prove that $L(A \otimes B) = L(A) \cap L(B)$, provided that

$$\hat{\delta}_{A\otimes B}((p, q), w) = (\hat{\delta}_A(p, w), \hat{\delta}_B(q, w)) \tag{1}$$

holds for each $p \in Q_A$, $q \in Q_B$ and $w \in \Sigma^*$.

Prove equation (1) by induction on strings.

Exercise 3

Use the construction given in the lectures of September 23[2] for translating the following regular expressions to ε-NFAs.

1. $(a + (b + c))^*(ab + ac)$
2. $(0(0 + 1)) + (00 + 01)$

Report Requirements

Your work on this assignment must be documented by a brief report.

- The **hard deadline is Friday, October 8, at 13.00** (i.e. 1 pm), where your report must be put into the box named 02140, which is placed in just inside the west entrance of Building 322.
- The report will be marked according to the 13 scale.
- The *printed* report should comprise the following:
  - Your names and student numbers. Allowed group sizes are 2 to 3.
  - A detailed description of all the objects you constructed, including print-outs and file names (including your user id) for the SML programs, so that we can test your programs in the G-databar. The corresponding files have to be made group-readable.
  - Spelled-out proofs[3], whenever proofs are asked for.
- It is allowed to discuss with other groups. However, the reports must be written independently. ***Any copying of text or programs will be considered as fraud.***

[2] Also to be found in Sect. 3.2.3 of the HMU book.

[3] I.e., something like "Let x, y, z be an arbitrary split $w = xyz$ of w with .... We take $k = \ldots$" instead of staccato just mentioning the objects without making clear the quantifier nesting.