MSc in Computer Science
MSc in Mathematics and the Foundations of Computer Science

Michaelmas Term 2017

Foundations of CS

Exercise class 1, focuses on finite automata

1. This is Sipser Problem 1.6 parts c,f,i (page 84):

   Give state diagrams of DFAs recognizing the following languages; in all parts the alphabet should be $\{0, 1\}$:

   - $\{w|w$ contains the substring 0101, i.e. $w = x0101y$ for some x and $y\}$
   - $\{w|w$ does not contain the substring 110 $\}$
   - $\{w|w$ every odd position of w is a $1\}$

2. This is Sipser Problem 1.41 or 1.42 (depends on edition). It's a good exercise in writing a precise proof using the standard notation.

   For languages A and B (each with alphabet Σ), let the shuffle of A and B be the language:

   $\{w|w = a_1b_1 \dots a_kb_k$ where $a_1 \dots a_k$ is a word in A and $b_1 \dots b_k$ is a word in B, and each a_i and b_i is a word in Σ^* $\}$

   Show that the class of languages recognized by DFAs is closed under shuffle.

3. Show that given two languages recognized by DFAs, L_1 and L_2, their intersection is recognized by a DFA. You should prove correctness, at the level of detail of the example proof given on the webpage.

4. Give an algorithm (informally, in pseudo-code) for each of the following problems. (Note that presentation of algorithms in pseudo-code is an important skill in FCS, as it is in many other CS contexts.)

   - Given an NFA A, the algorithm decides whether A accepts some string (that is, the algorithm decides the *emptiness problem for regular languages*).
   - Given two NFAs A_1 and A_2, the algorithm decides whether or not A_1 and A_2 recognize the same language.
   - Given an NFA A, the algorithm decides whether or not $L(A)$ contains *some* string whose length is a composite number (i.e. whose length is not a prime number).