York University CSE 2001 September 22, 2009

Homework Assignment #2
Due: October 1, 4:00 p.m.

Consider the alphabet $\left\{\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\1\end{pmatrix}, \begin{pmatrix}1\\1\\0\end{pmatrix}, \begin{pmatrix}1\\1\\1\end{pmatrix}\right\}$. We shall use strings in this alphabet to describe three integers: one for the top row of bits, one for the middle row and one for the bottom row. Each integer is represented in binary, but it is represented backwards (with the least significant bit at the right end instead of the left end). For example, to represent the three integers 24, 11 and 13 (whose ordinary binary representations are 11000, 1011 and 1101), we would use the string $\begin{pmatrix}0\\1\\1\end{pmatrix}\begin{pmatrix}0\\1\\0\end{pmatrix}\begin{pmatrix}0\\0\\1\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}$: the top row is the reverse of 11000, the middle row is the reverse of 01011 (the 0 in the 16's position is there just to make all the strings have the same length), and the bottom row is the reverse of 01101.

Let $SUBTRACT$ be the language of all strings that represent correct binary subtractions, that is, those strings where the top integer minus the middle integer is equal to the bottom integer.

For example, the string $\begin{pmatrix}0\\1\\1\end{pmatrix}\begin{pmatrix}0\\1\\0\end{pmatrix}\begin{pmatrix}0\\0\\1\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}$ is in $SUBTRACT$, since

$$\begin{array}{cccccc} & 1 & 1 & 0 & 0 & 0 \\ - & 0 & 1 & 0 & 1 & 1 \\ \hline & 0 & 1 & 1 & 0 & 1. \end{array}$$

(In decimal notation, this just means $24 - 11 = 13$.) Draw the transition diagram of a deterministic finite automaton that recognizes $SUBTRACT$.

You need not prove your automaton is correct. However you must give, for each state of your finite automaton, a precise description of all the strings that can lead to that state.

Hint: You should actually figure out what each state of your automaton represents (in other words, which strings take the automaton to that state) *before* drawing the transition diagram. Think about what the automaton has to remember as it processes the input.

Supplementary Programming Exercise. This part of the assignment is just for fun: you do not have to hand it in. Write a java programme that opens a file, and checks whether the number of occurrences of the character 'x' in the file is odd or even. Try to write your programme using as little memory as you can. You should be able to do it using only one character variable to store the character that you are currently processing and one boolean variable. When you have finished writing your programme, use the file containing the source code of your programme as an input to the programme to test your programme.