Introduction to Computer Science
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: CH08-320101
Date: 2017-11-28
Due: 2017-12-05

ICS Problem Sheet #11

Problem 11.1: *finite state machine* (1+1+1+1 = 4 points)

The language L over the alphabet $\Sigma = \{a, b\}$ consists of all non-empty words where any occurance of a is immediately preceeded and followed by a b. For example, the following words b, bb, bab, $bbab$, $babb$, $babab$, and $bbbabbbbbab$ all belong to L. Define a finite state machine that recognizes L.

1. Define the finite state machine formally. Provide a complete definition of the transition function, that is, make sure that for every state a transition is defined for every element of Σ.
2. Provide a graph representation of the finite state machine.
3. Write a Haskell program (following the template shown in class) that decides whether a given string belongs to the language L.
4. Define a regular grammar G that generates L, i.e., $L = L(G)$.

Problem 11.2: *turing machines to increment, decrement, and add numbers* (2+2+2 = 6 points)

In this problem, you are expected to construct several Turing machines. For each Turing machine, provide a high-level description how it works and provide the graph representation. (You can leave out the formal definition if the graph is complete.)

a) Write a Turing machine T_{inc} that can add 1 to a binary encoded number stored on the tape of the Turing machine. The binary number is enclosed by the symbol $\$$ and you can assume that the binary number starts with a 0 (i.e., there is no overflow to consider). For example, the input $\$0100\$$ is transformed by T_{inc} into $\$0101\$$ and the input $\$0111\$$ is transformed by T_{inc} into $\$1000\$$. The turing machine starts with the head located at the $\$$ sign left of the number.

b) Write a Turing machine T_{dec} that can subtract 1 from a binary encoded positive number stored on the tape of the Turing machine. The binary number is enclosed by the symbol $\$$. For example, the input $\$0100\$$ is transformed by T_{dec} into $\$0011\$$ and the input $\$0111\$$ is transformed by T_{dec} into $\$0110\$$. The turing machine starts with the head located at the $\$$ sign left of the number.

   Hint: You can invert all bits of the number, then add one to the number, then invert all bits of the number again.

c) Write a Turing machine T_{add} that can add two binary encoded numbers on the tape of the Turing machine. The binary numbers are each enclosed by the symbol $\$$ and you can assume that the binary numbers have a sufficient number of leading 0s to hold the sum (i.e., there is no overflow to consider). For example, the input $\$0100\$0010\$$ is transformed by T_{add} into $\$0000\$0110\$$, i.e., the first number was added to the second number.

   Hint: You can construct T_{add} out of T_{inc} and T_{dec}: While the first number is not zero, decrement the first number and increment the second number. Please name your states such that it is clear to which part of your textual description they belong.

You may find it useful to write a Haskell program to simulate your Turing machines (i.e., following the example shown in class). This way you can run tests against your Turing machine to verify it is working correctly. Feel free to submit your Haskell code so that we can verify your design of the turing machines.

Note that it is your responsibility to document things properly. If you hand in something we cannot understand, you will likely get zero points.