Grado en Informática
Formal Languages and Automata Theory

DEPARTAMENTO DE INFORMÁTICA
UNIVERSIDAD CARLOS III DE MADRID

Formal Languages and Automata Theory
Exercises Turing Machines
Unit 7

1. Design a Turing Machine to replace 0's with 1's in the input string.
2. Design a Turing Machine to calculate the 1-complement of a binary number (i.e. replace 0's with 1's and 1's with 0's).
3. Design a Turing Machine to obtain the successor of a number in unary codification. Consider that the unary representation of 0 is the empty string, 1 is represented by 1, 2 is represented by 11, etc.
4. Design a Turing Machine to obtain the predecessor of a number in unary codification. Consider the same representation described in the previous exercise.
5. Design a Turing Machine to calculate the parity of a binary number, i.e. add a 0 at the end if the number of 1's in the input string is even or a 1 is this number is odd.
6. Design a Turing Machine to be a unary counter of characters in the language with alphabet $\Sigma = \{a,b,c\}$, i.e., the machine must generate as 1's as output as characters in the input word. Consider the same representation for 0 defined in the exercise 3.
7. Design a Turing Machine to generate a copy of a string with symbols {A, B, C}. For instance, given the input "bAABCAb", the resulting input tape would be "bAABCAAABCAb", where b represents the blank symbol.
8. Design a Turing Machine which takes a input string with M 1's and N A's (M≤N), and replaces the M first A's with B's. For instance, given the input "b11AAAAAb" it would generate the input tape "b11BBAAAb", where b represents the blank symbol (i.e., empty cells in the tape).
9. Design a Turing Machine which takes two input words generated with the alphabet {0, 1, 2} separated using the symbol {#}, and verifies whether they are the same. For instance, given the input b2101#2101b the Turing Machine would inform that both words are the same, where b represents empty cells in the tape.
10. Design a Turing Machine to recognize the language $L=a^nb^n$.
11. Design a Turing Machine to recognize the language $L=a^nb^nc^n$.
12. Design a Turing Machine which obtains the successor of a binary number.
13. Design a Turing Machine which obtains the predecessor of a binary number.
14. Design a Turing Machine which translates a 1's sequence in unary codification to its equivalent binary codification (0's and 1's).