UNIVERSIDAD CARLOS III DE MADRID

FORMAL LANGUAGES AND AUTOMATA THEORY

COMPUTER SCIENCE DEGREE. CONTINUOUS ASSESSMENT - PARTIAL 3

Date: 2011 December 14th

Last name(s): ____

First name: ____

NIA: ____

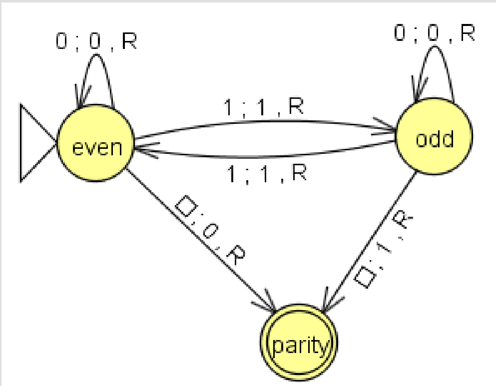
Duration: 45 minutes

Exercise 1 (3 points)

a. Construct a Turing Machine to add the parity bit to a binary string. The TM must add a "0" if the number of 1s of the input string is odd and a "1" if this number is even. The parity bit must be added at the end.

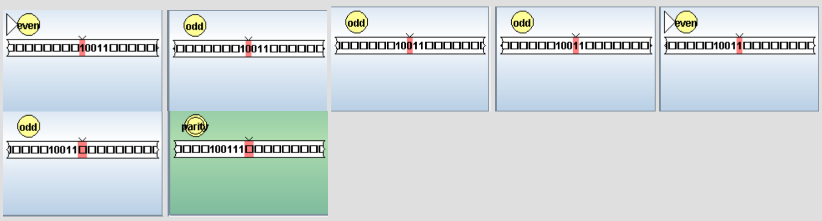
This problem has been solved at class (problem 5, proposed exercises Unit 6). A possible Turing Machine to solve the required problem is as follows:

TM= ({0,1}, {0,1,□}, □, {even, odd, parity}, even, {parity}, f)



b. Represent formally the transitions for the input string "10011".

Transitions are as follows:



c. For the Following Turing Machine, TM = ({0,1,b}, {0,1}, b, {p,q}, p, f, {q}) where f is defined by the following transition function

f	1	0	b
p	q1R	p0R	pbS
q	p1S	p1S	qbS

Which is the condition to accept input words (i.e., required state and input symbol in the tape)?

The inputs words are accepted when being in the state *q* there is a *b* in the current cell in the tape.

Exercise 2 (4 points)

Given the grammar

G = {{S, A}, {x, y, z}, S, P}, with P:
S ::= xyAS | z
yAx ::= xyA
yAy ::= yyA
Az ::= z

a. Which is the type of grammar in the Chomsky hierarchy? Explain it in detail.

It is a Type-0 grammar. It is not a Type-2 or a Type-3 grammar because there are productions with more than one symbol on the left side. It is not a Type-1 grammar due to the production rule Az→ z (there are more symbols on the left side than on the right side).

b. Determine the language that is generated by the grammar.

We can generate derivations with the production rules in the grammar to determine the language:

S → **z**
S → xyAS → xyAz → **xyz**
S → xyAS → xyAxyAS → xxyAyAS → xxyyAAz → xxyyAz → **xxyyz**
…

This way, the language that is generated consists of the set of words that begin with 0 or more **x** followed by the same number of **y** and end with a **z**:

$$L = \{x^n y^n z \mid n >= 0\}$$

c. Find an equivalent Type-2 grammar.

An equivalent Type-2 grammar is:

S' → Az
A → xAy | λ

d. Determine an equivalent Push-Down automaton for the grammar obtained in the previous section.

First of all we have to obtain an equivalent well-formed grammar:

$S' \rightarrow Az \mid z$
$A \rightarrow xAy \mid xy$

Then, we have to obtain an equivalent grammar in GNF:

$S' \rightarrow xABC \mid xBC \mid z$
$A \rightarrow xAB \mid xB$
$B \rightarrow y$
$C \rightarrow z$

An equivalent PDA recognizing words in the language when the stack is empty is as follows:

$PDA_E = (\{x,y,z\}, \{S,A,B,C\}, \{q\}, S', q, f, \Phi)$

$f(q,x,S') = \{(q, ABC), (q, BC)$
$f(q,z,S') = (q, \lambda)$
$f(q,x,A) = \{(q, AB), (q, B)\}$
$f(q,y,B) = (q, \lambda)$
$f(q,z,C) = (q, \lambda)$

e. Explain in detail if it is possible to represent this language by means of the following formal machines and structures: a Turing Machine, a Deterministic Finite Automaton, a Type-1 grammar.

A Turing Machine can recognize the language generated by this grammar, given that these machines can recognize every Type-0, Type-1, Type-2, and Type-3 languages. This language can also be recognized by means of a Push-Down Automaton given that these automata are used to recognize Type-2 languages. It is also possible to recognize this language by means of a Type-1 grammar, given that Type-1 languages include Type-2 languages. This language cannot be recognized by means of a deterministic Finite Automaton given that this is not a regular language.