Anyi Guo
Msc. Data Science Part Time

Student ID: 13154855

01/2019

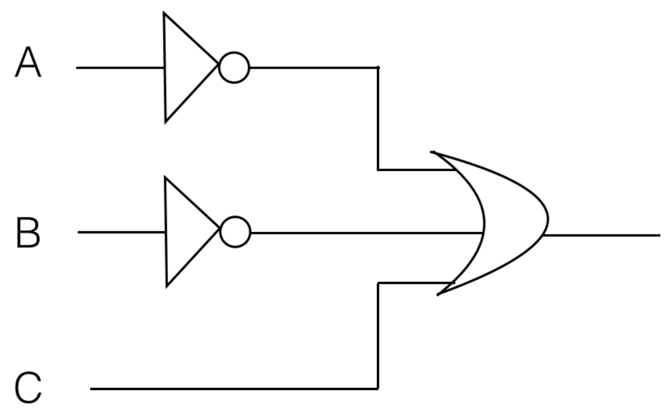
Coursework 1
Fundamentals of Computing

1.

(a)

A	B	C	B—>A	A ∧ B ∧ C	∨	¬B∨C	(B—>A) -> ((A ∧ B ∧ C) ∨ (¬B∨C))
1	1	1	1	1	1	1	1
1	1	0	1	0	0	0	0
1	0	0	1	0	1	1	1
1	0	1	1	0	1	1	1
0	1	1	0	0	1	1	1
0	1	0	0	0	0	0	1
0	0	1	1	0	1	1	1
0	0	0	1	0	1	1	1

(b) Boolean equation: $(\neg A) \vee (\neg B) \vee c$



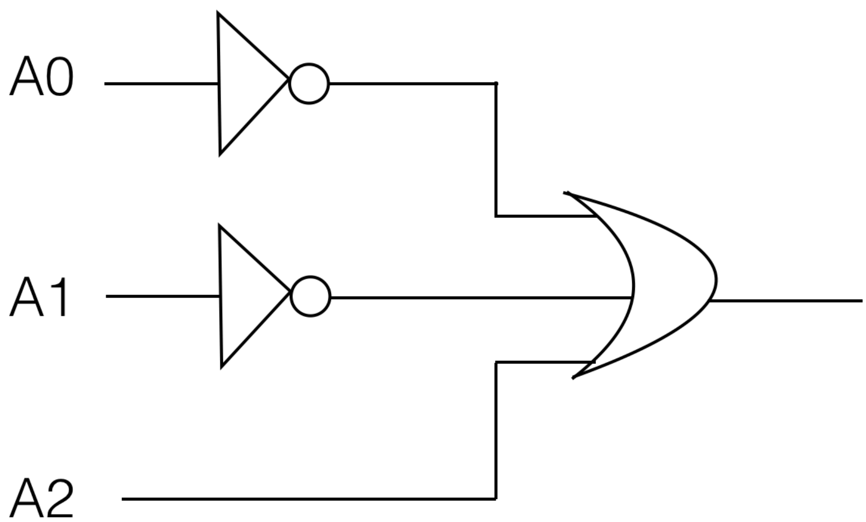
(c) It is not correct, because when $C -> A$ and $A -> B$ are true, there is one case when B is not true (highlighted in blue)

A	B	C	C->A	A->B
1	1	1	1	1
1	1	0	1	1
1	0	0	1	0
1	0	1	1	0
0	1	1	0	1
0	1	0	1	1
0	0	1	0	1
0	0	0	1	1

2.

Decimal	A2	A1	A0	Y (if result < 3)
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
0	1	0	0	1
-1	1	0	1	1
-2	1	1	0	1
-3	1	1	1	1

Boolean equation: $A2 \vee (\neg A1) \vee (\neg A0)$



Anyi Guo Student ID: 13154855 01/2019
Msc. Data Science Part Time

3.
A= « Jones didn't meet with Smith last night »
B= « Smith is the murderer »
C= « Jone is lying »
D=« Murder took place after midnight »

The conclusion « Smith was the murderer » (proposition B) is not correct because there are multiple cases when all 3 statements are true (« A-> B v C », « B -> A ∧ D », « D -> B v C »), but B is false, for example case 1,3,4,11,12 (highlighted in red)

Case #	A	B	C	D	B v C	A ∧ D	A-> B v C	B -> A ∧ D	D -> B v C
1	0	0	0	0	0	0	1	1	1
2	0	0	0	1	0	0	1	1	0
3	0	0	1	0	1	0	1	1	1
4	0	0	1	1	1	0	1	1	1
5	0	1	0	0	1	0	1	0	1
6	0	1	0	1	1	0	1	0	1
7	0	1	1	0	1	0	1	0	1
8	0	1	1	1	1	0	1	0	1
9	1	0	0	0	0	0	0	1	1
10	1	0	0	1	0	1	0	1	0
11	1	0	1	0	1	0	1	1	1
12	1	0	1	1	1	1	1	1	1
13	1	1	0	0	1	0	1	0	1
14	1	1	0	1	1	1	1	1	1
15	1	1	1	0	1	0	1	0	1
16	1	1	1	1	1	1	1	1	1

Anyi Guo Student ID: 13154855 01/2019
Msc. Data Science Part Time

4.

(a) a two's complement integer

$-2^{31}+2^{30}+2^{24}+2^{23}+2^{21}+2^{20}$

(b) an unsigned integer

$2^{31}+2^{30}+2^{24}+2^{23}+2^{21}+2^{20}$

(c) a single precision IEEE 754 floating-point number

1 10000011 011 0000 0000 0000 0000 0000

5.

(a) -107 as two's complement 32-bit binary number

$107 = 64+32+8+2+1$

$= 2^6+2^5+2^3+2^1+2^0$

$= 000\ldots1101011$

The boolean negation is:

1111…0010100, after which we add 1:

1111…(twenty-one 1s here)…0010101

(b) -107 as an IEEE 754 32-bit floating-point number

1 1000 0101 10101100 0000 0000 0000 000

(c) -15.375 as an IEEE 754 32-bit floating-point number

1 1000 0010 111011 0000 0000 0000 0000 0

6.
(a) one-to-one but not onto

This function's y would never be 0, which is in the codomain N.

$$f(x) = \begin{cases} 2x & x > 0 \\ -2x + 1 & x <= 0 \end{cases}$$

(b) onto but not one-to-one

$$f(x) = \begin{cases} x & x >= 0 \\ -x & x < 0 \end{cases}$$

(c) both one-to-one and onto

$$f(x) = \begin{cases} 2x & x >= 0 \\ -(2x + 1) & x < 0 \end{cases}$$

(d) neither one-to-one nor onto

$f(x) = 1$

7.

(a). $f(x) = (x + 1)/(x - 2)$

It is neither one-to-one, nor onto.

(b). $f(m, n) = m + n - 1$

It is one-to-one and onto.

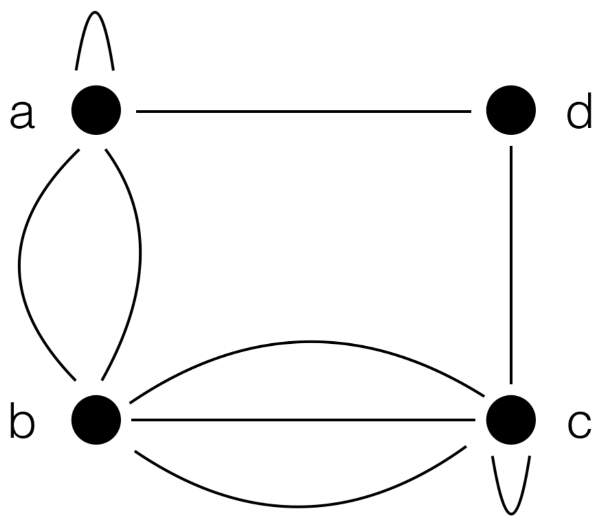
(c). $f(m, n) = |m| - |n| - 1$

It is not one-to-one, but it is onto.

Anyi Guo Student ID: 13154855 01/2019
Msc. Data Science Part Time

8. Graph

(a) This is not a simple graph because there is more than one edge between two different vertices (e.g. a&b, b&c)



(b)

- $G3$ is not isomorphic to $G1$ and $G2$ because $G1$ and $G2$ contain a simple cycle of length 3, while $G3$ contains only simple cycles of length 4 and above.
-
- $G1$ and $G2$ are isomorphic. The function f defined by taking:
  $f(a) = X, f(e) = U, f(f) = V, f(c) = W, f(b) = Y, f(d) = Z$

Anyi Guo Student ID: 13154855 01/2019
Msc. Data Science Part Time

9.
(a)
i. bb
This string is accepted: $(s, bb), (p, bb), (q, b), (p, b), (q, \varepsilon) --> accepted$

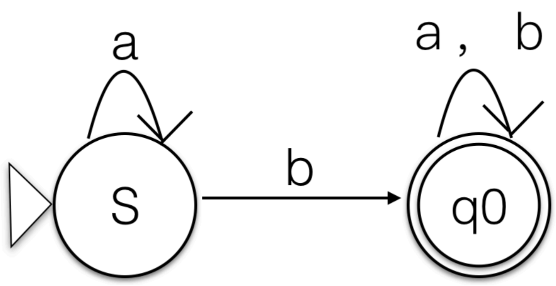
ii. aa
This string is not accepted: every computation on it is stuck somewhere.
$(s, aa), (s, a), (s, \varepsilon), (p, \varepsilon) --> stuck$
$(s, aa), (p, aa) --> stuck$
$(s, aa), (s, a), (s, \varepsilon) --> stuck$
$(s, aa), (p, a) --> stuck$

iii. ab
This string is accepted: $(s, ab), (p, b), (q, \varepsilon) --> accepted$

iiii. aba
This string is accepted: $(s, aba), (p, ba), (q, a), (q, \varepsilon) --> accepted$

v. ε
This string is not accepted:
$(s, \varepsilon) --> stuck$
$(s, \varepsilon), (p, \varepsilon) --> stuck$

(b)

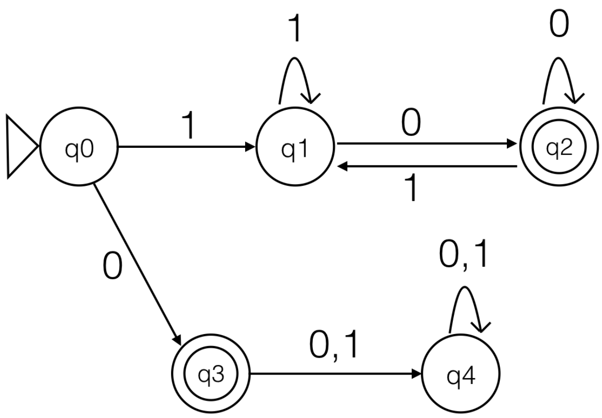


(c) regular expression: $a*ba*b*$

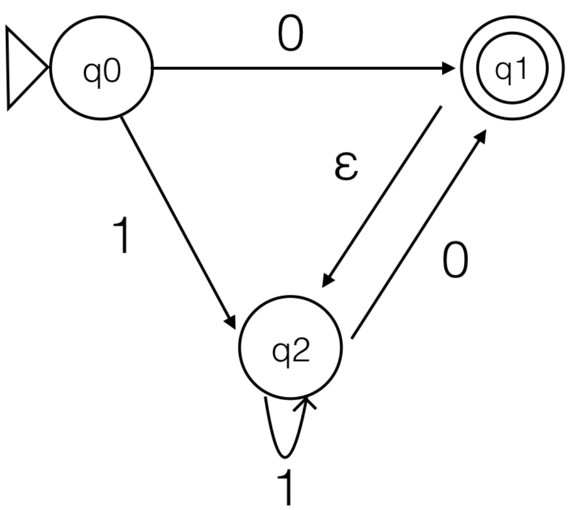
(d) context-free grammar
$S -> aS$
$S -> bA$
$A -> aA$
$A -> \varepsilon$

10.

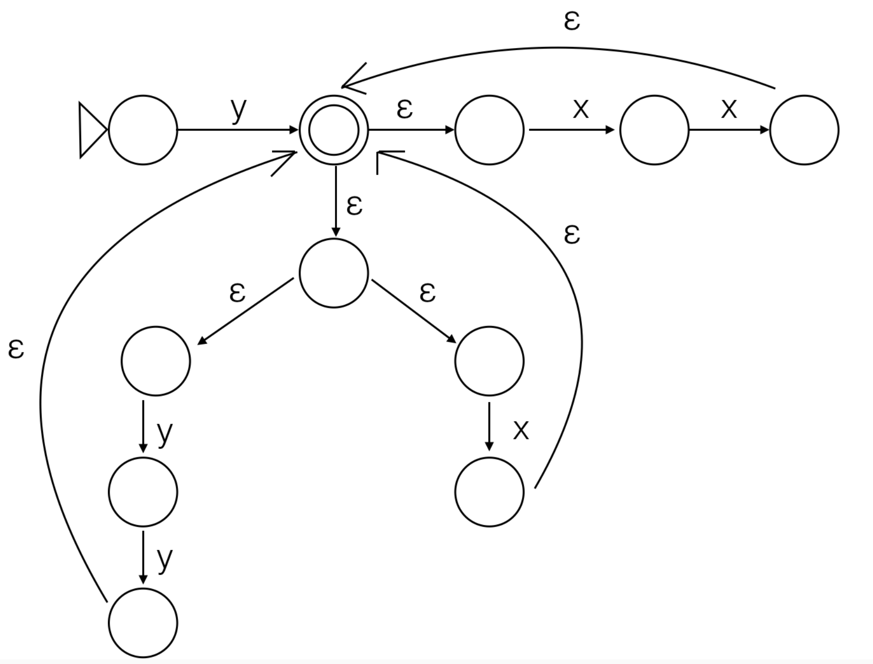
(a)



(b)



11.



12.
(a)

S1 is when $i = j$, S2 is when $i = k$(i.e. palindrome):

$S -> S1$
$S -> S2$

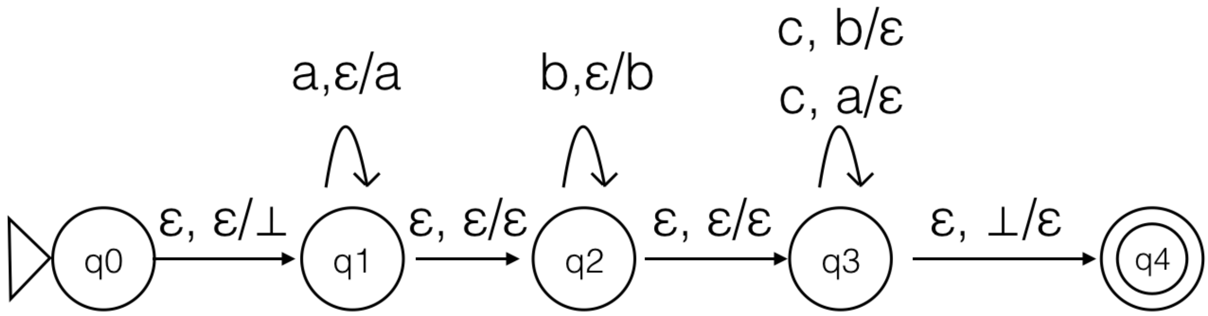
$S1 -> \varepsilon$
$S1 -> S1c$
$S1 -> A$
$A -> aAb$
$A -> \varepsilon$

$S2 -> \varepsilon$
$S2 -> B$
$B -> aBc$
$B -> bB$
$B -> \varepsilon$

(b) This language is not regular because a finite automaton (which has limited memory) must attempt to store the entire prefix a^i before going to b in the middle shows up. Because the i here could be any number, this makes the language not regular.

$$\left\{a^i b^j a^i | i >= 0, j >= 3\right\}$$

13.



14.
(a)

(blank = ␣ , I can't find a way to type this symbol on my computer)

i. Input = 10
$(s,\triangleright\underline{1}0),(q,\triangleright 1\underline{0}),(q,\triangleright,10\underline{\text{blank}}), (p,\triangleright 1\underline{0}),(p,\triangleright 10\underline{\text{blank}})$, halted$(h,\triangleright 10\underline{0})$

ii. input = 111
$(s, \triangleright\underline{1}11), (q,\triangleright 1\underline{1}1),(q,\triangleright 11\underline{1}),(q,\triangleright 111\underline{\text{blank}}), (p,\triangleright 11\underline{1})$, halted$(h,\triangleright 11\underline{1})$

iii. Input = 110
$(s,\triangleright\underline{1}10),(q,\triangleright 1\underline{1}0),(q,\triangleright 11\underline{0}),(q,\triangleright 110\underline{\text{blank}}), (p,\triangleright 11\underline{0}),(p,\triangleright 110\underline{\text{blank}})$, halted$(h,110\underline{\text{blank}})$

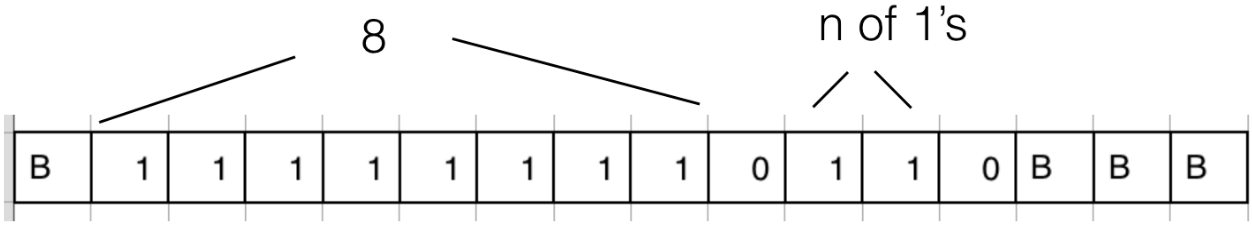
(b) This Turing Machine can be expressed by the following function:

$$f(x) = \begin{cases} 2x & x = even \\ x - 1 & x = odd \end{cases}$$

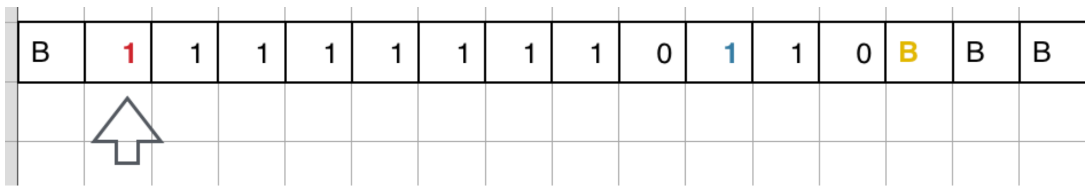
15.
(a) $f(n) = 8n + 2$

part 1: multiplication : $f(n) = 8n$

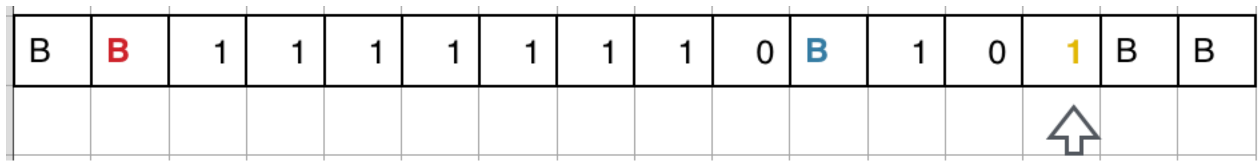
Starting state:



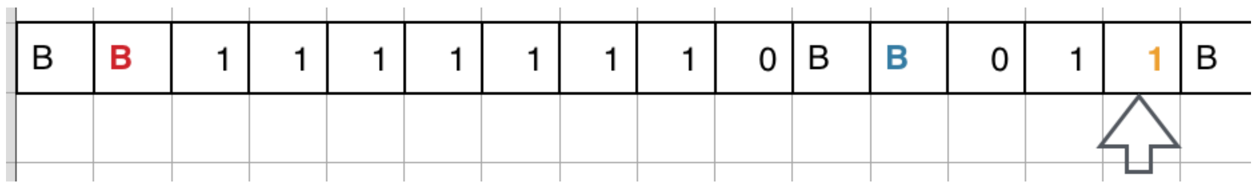
1. Start from leftmost symbol, the machine changes the first 1 (highlighted in red) into Blank and moves right to find the first 0.



2. Once it finds the first 0, it will change the 1 to the right of the first 0 (highlighted in blue) into Blank, go right to find the second 0, and replace the first blank (highlighted in yellow) with 1. This is what the tape looks like after Step 2:



3. Then it goes left to find the first B (in blue in the above chart), change the 1 to the right of this B to Blank, move right to find the first B and replace it with 1. After step 3:



4. Then it goes left to find the first zero. Continue going left and repeat step 2-3 above until all 1's between the two zeros are replaced by B.

5. When there're only B's between the two zeros, change these B's into 1 (highlighted in purple in the chart below), the tape goes left to find the first B, go right to the next 1 (highlighted in light blue in the chart below), and repeat step 1-4 above.

B	B	1	1	1	1	1	1	1	0	1	1	0	1	1	B

6. Halt when there's no more 1 to the left of the first 0 (highlighted in green in the chart below)

B	B	B	B	B	B	B	B	B	0	1	1	0	1	1	1

part 2: addition: $f(n) = n + 2$

1. Start from leftmost symbol, the machine first goes to the right until it reaches a blank
2. then it turns left and flips all 1's to 0's until it reaches the first 0
3. then it changes the first 0 to 1 and halts
4. if it never finds 0, it writes 1 in the leftmost cell , finds the first blank on the right, writes over 0, and halts
5. repeat step 1-4 above one more time

(b) Sorry - I couldn't figure out how to create a transition table for Turing Machine. Can we cover this in the Term 3 revision please?