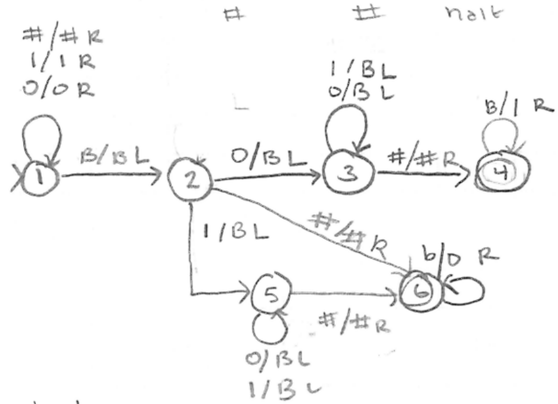
1. Construct a Turing Machine (with no macros) to compute the following number-theoretic functions

(a) Binary even(n) = { 1 if n is even; 0 otherwise }

	read	write	move	goto
11	0	0	right	same
	1	1	left	next
	#	#	halt	

11	0	0	R	same
	1	1	R	same
	#	#	R	same
	b	b	L	next
12	0	b	L	next
	1	b	L	15
	#	#	R	16
13	0	b	L	same
	1	b	L	same
	#	#	R	next
14	b	1	halt	
15	0	b	L	same
	1	b	L	same
	#	#	R	next
16	b	0	halt	



n = 101

1 0 1 — q_1
1 0 1 — q_1
1 0 1 B — q_1
1 0 1 B — q_1
1 0 1 B — q_1

1 0 1 B — q_2
1 0 B B — q_2
1 B B B — q_2
B B B B — q_2
B B B B — q_6

0 B B B — q_6
0 0 B B — q_6
0 0 0 B — q_6
0 0 0 0 — q_6 Halt

= 101 odd

n = 10

1 0 B — q_1 | # 1 0 B — q_1 | # 1 0 B — q_1 | # 1 0 B — q_1

1 0 B — q_2 | # 1 B B — q_3 | # B B B — q_3 | # B B B — q_4

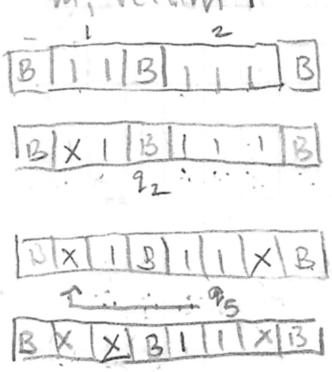
1 B B — q_4 | # 1 1 B — q_4 | # 1 1 1 — q_4 Halt

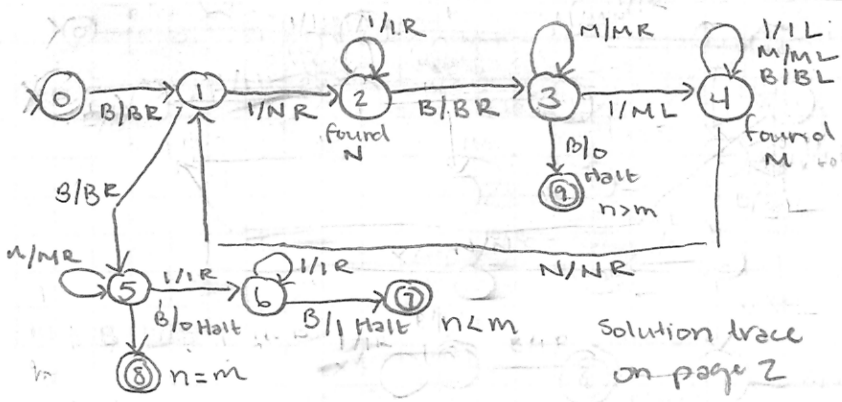
10 even

(b) lt(n,m) = { 1 if n < m; 0 ow }

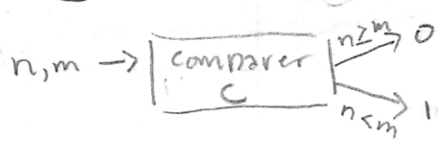
I will use unary representation of numbers for lt

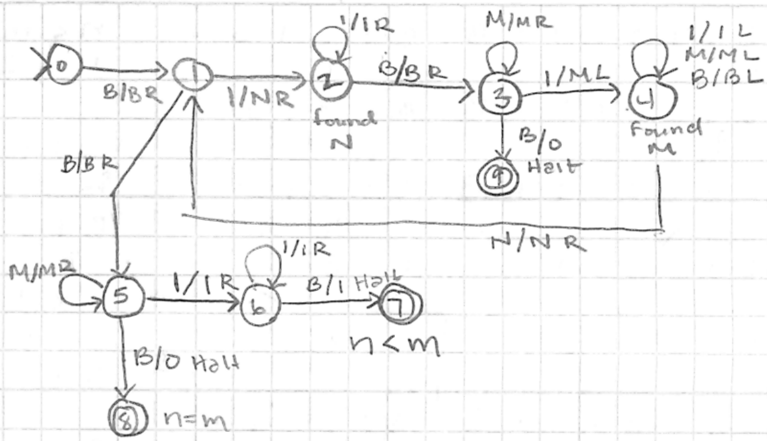
strategy
- exam both args
- if arg n and m are equal, return 0
- if arg n is less than m, return 1





Solution trace on page 2





$n=1 \quad m=2$

1 1 B 1 1 1
q_1
1 1 B 1 1 1
N 1 B 1 1 1
q_2
N 1 B 1 1 1
q_2
N 1 B 1 1 1 B
q_3
N 1 B M 1 1 B
q_4
N 1 B M 1 1 B
q_4
N 1 B M 1 1 B
q_4
N 1 B M 1 1 B
q_1
N N B M 1 1 B
q_2
N N B M 1 1 B
q_3
N N B M 1 1 B
q_3
N N B M M 1 B
q_4
N N B M M 1 B
q_4
N N B M M 1 B
q_4
N N B M M 1 B
q_1
N N B M M 1 B
q_5
N N B M M 1 B
q_5
N N B M M 1 B
q_5
N N B M M 1 B
q_6
N N B M M 1 1
Halt

$n=0, m=0$

B 1 B 1 B
q_0
B 1 B 1 B
q_1
B N B 1 B
q_2
B N B 1 B
q_3
B N B M B
q_4
B N B M B
q_4
B N B M B
q_1
B N B M B
q_5
B N B M B
q_5
B N B M 0
Halt

$n=m$

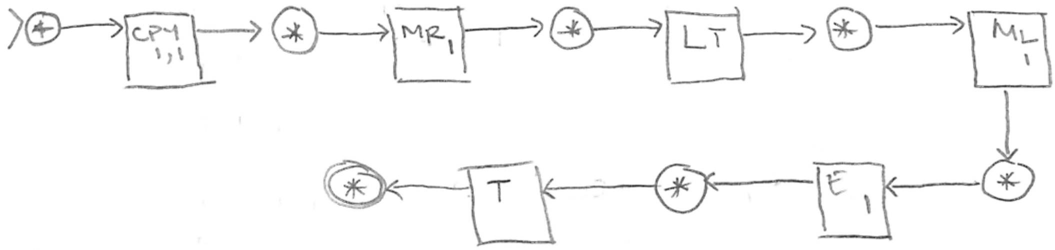
$n=1, m=0$

B 1 1 B 1 B
q_0
B 1 1 B 1 B
q_1
B N 1 B 1 B
q_2
B N 1 B 1 B
q_2
B N 1 B 1 B
q_3
B N 1 B M B
q_4
B N 1 B M B
q_4
B N 1 B M B
q_4
B N 1 B M B
q_1
B N N B M B
q_2
B N N B M B
q_3
B N N B M B
q_3
B N N B M 0 Halt

$n > m$

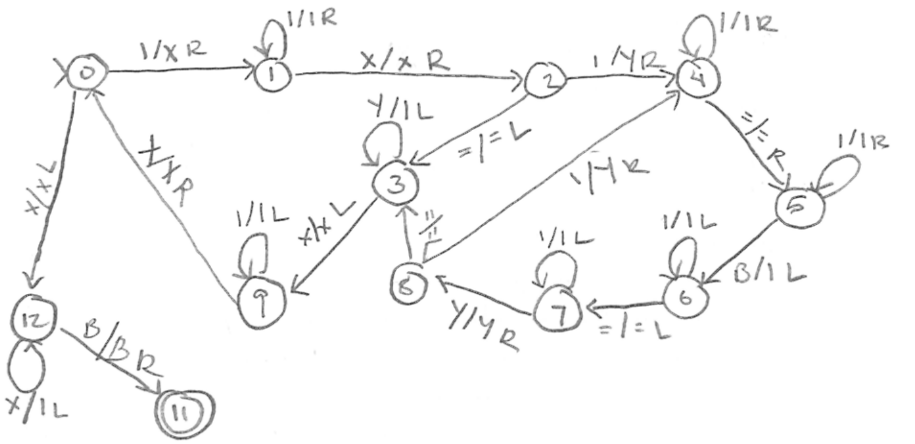
2. Design a machine that computes
$gt(n,m) = \begin{cases} 1 & \text{if } n > m \\ 0 & \text{otherwise} \end{cases}$

$gt(n,m) = lt(m,n)$; using lt defined in 1b)



Machine	Config
	$\underline{B}\bar{n}B\bar{m}B$
$CPY_{1,1}$	$\underline{B}\bar{n}B\bar{m}B\bar{n}B$
MR_1	$B\bar{n}\underline{B}\bar{m}B\bar{n}B$
LT	$B\bar{n}\underline{B}\overline{lt(m,n)}B$
ML_1	$\underline{B}\bar{n}B\overline{lt(m,n)}B$
E_1	$\underline{B}\ldots B\overline{lt(m,n)}B$
T	$\underline{B}\overline{lt(m,n)}B$

3. Trace the actions of the machine MULT for computations with input:
(a) $n=0, m=4$
(b) $n=1, m=0$
(c) $n=2, m=2$



Example 1

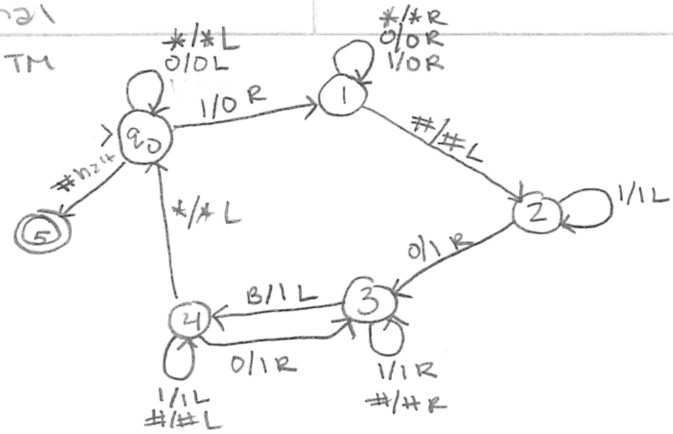
Example 2

States	0	1	#	*	b
q_0	$q_0,0,L$	$q_1,0,R$	q_5, halt	$q_0,*,L$	
q_1	$q_1,0,R$	$q_1,0,R$	$q_2,\#,L$	$q_1,*,R$	
q_2	$q_3,1,R$	$q_2,1,L$			
q_3		$q_3,1,R$	$q_3,\#,R$		$q_4,1,L$
q_4	$q_5,1,R$	$q_4,1,L$	$q_4,\#,L$	$q_0,*,L$	

Configuration for Ex. 2 Multiplier TM

b	#	1	1	1	*	1	1	#	b	b	b

$\boxed{q_0}$ (under the *) — product (the b b b cells)



$\boxed{n=0 \quad m=4}$

b # 1 * 1 1 1 1 1 # b
q_0

b # 1 * 1 1 1 1 1 # b
q_0

b # 0 * 1 1 1 1 1 # b
q_1

b # 0 * 1 1 1 1 1 # b
q_1

b # 0 * 0 1 1 1 1 # b
$q_1 \longrightarrow$

b # 0 * 0 0 0 0 0 # b
q_1

b # 0 * 0 0 0 0 0 # b
q_2

b # 0 * 0 0 0 0 1 # b
q_3

b # 0 * 0 0 0 0 1 # b
q_3

b # 0 * 0 0 0 0 1 # 1
q_4

b # 0 * 0 0 0 0 1 # 1
q_4

b # 0 * 0 0 0 0 1 # 1
q_4

b # 0 * 0 0 0 1 1 # 1
$q_3 \longrightarrow$

b # 0 * 0 0 0 1 1 # 1 1
$q_4 \longleftarrow q_4$

b # 0 * 0 0 1 1 1 # 1 1
$q_3 \longrightarrow q_3$

b # 0 * 0 0 1 1 1 # 1 1 1
$q_4 \longleftarrow q_4$

b # 0 * 0 1 1 1 1 # 1 1 1
$q_3 \longrightarrow q_3$

b # 0 * 0 1 1 1 1 # 1 1 1 1
$q_4 \longleftarrow q_4$

b # 0 * 1 1 1 1 1 # 1 1 1 1
$q_3 \longrightarrow q_3$

b # 0 * 1 1 1 1 1 # 1 1 1 1 1
$q_4 \longleftarrow q_4$

b # 0 * 1 1 1 1 1 # 1 1 1 1 1
q_0

b # 0 * 1 1 1 1 1 # 1 1 1 1 1
q_0

b # 0 * 1 1 1 1 1 # 1 1 1 1 1
q_5 halt

$\boxed{n=1 \quad m=0}$

b # 1 1 * 1 # b
q_0

b # 1 1 * 1 # b
q_0

b # 1 0 * 1 # b
q_1

b # 1 0 * 1 # b
q_1

b # 1 0 * 0 # b
q_1

b # 1 0 * 0 # b
q_2

b # 1 0 * 1 # b
q_3

b # 1 0 * 1 # b
q_3

b # 1 0 * 1 # 1
$\longleftarrow q_4$

b # 1 0 * 1 # 1
q_4

b # 1 0 * 1 # 1
q_0

b # 1 0 * 1 # 1
q_0

b # 0 0 * 1 # 1
$q_1 \longrightarrow q_1$

b # 0 0 * 0 # 1
q_1

b # 0 0 * 0 # 1
q_2

b # 0 0 * 1 # 1
$q_3 \longrightarrow q_3$

b # 0 0 * 1 # 1 1
$q_4 \longleftarrow q_4$

b # 0 0 * 1 # 1 1
$q_0 \longleftarrow q_0$

b # 0 0 * 1 # 1 1
q_5 halt

This machine is not configured to represent 0 in unary as in the book. so 1=1, 2=11, 3=111, etc.

Actual representation
n=0, m=4

b # * 1 1 1 1 # b
q_0

b # * 1 1 1 1 # b
q_0

b # * 1 1 1 1 # b
q_5 halt

n=1, m=0

b # 1 * # b
q_0

b # 1 * # b
q_0

b # 0 * # b
q_1

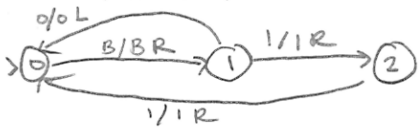
b # 0 * # b
q_1

b # 0 * # b
q_2 halt

(c) m=2, n=2

```
b # 1 1 * 1 1 # b
      q0
b # 1 1 * 1 1 # b
        q0
b # 1 0 * 1 1 #
        q1 ---> q1
b # 1 0 * 0 0 #
            q2
b # 1 0 * 0 1 #
              q3->
b # 1 0 * 0 1 # 1
        q4 <- q4
b # 1 0 * 1 1 # 1
            q3 ---> q3
b # 1 0 * 1 1 # 1 1
      q4 <----- q4
b # 1 0 * 1 1 # 1 1
      q0
b # 1 0 * 1 1 # 1 1
    q0
b # 0 0 * 1 1 # 1 1
      q1 ---> q1
b # 0 0 * 0 0 # 1 1
            q2
b # 0 0 * 0 1 # 1 1 1
              q3 ---> q3
b # 0 0 * 0 1 # 1 1 1
        q4 <----- q4
b # 0 0 * 1 1 # 1 1 1
          q3 -----> q3
b # 0 0 * 1 1 # 1 1 1 1
      q4 <------- q4
b # 0 0 * 1 1 # 1 1 1 1
 q0 <- q0
b # 0 0 * 1 1 # 1 1 1 1
 q5 halt
```

Given a TM



Transition table

	q_i	sym		q_j		sym		move	
$\delta(q_0, B) = [q_1, B, R]$	1 0	111	0	11	0	111	0	11	00
$\delta(q_1, 0) = [q_0, 0, L]$	11 0	1	0	1	0	1	0	1	00
$\delta(q_1, 1) = [q_2, 1, R]$	11 0	11	0	111	0	11	0	11	00
$\delta(q_2, 1) = [q_0, 1, L]$	111 0	11	0	1	0	11	0	1	00

5. Let G be the context sensitive grammar

G: S → SBA | a
BA → AB
aA → aaB
B → b

a) Give a derivation of aabb

S → SBA
→ S(AB)
→ a(AB)
→ (aA)B
→ aaBB
→ aabb ✓

b) $L(G) = \{a^{i+1}b^{2i} \mid i \geq 0\}$

c) CFG(L(G)) = S → aSbb | a

6. Design a two tape TM that determines if two strings u and v over {0,1} are identical. The computation begins with BuBvB on the tape

see page 6

7. Construct a TM that decides whether a string over $\{0,1\}^*$ is the encoding of a nondeterministic TM. What would be required to change this to a machine that decides whether the input is the representation of a deterministic TM.

Let the encoding of M be defined by

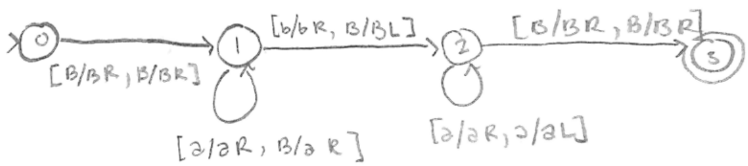
symbol	enc(x)
B	111
0	1
1	11
q_0	1
q_1	11
⋮	
L	1
R	11

where 0 divides components of the TM

00 divides transitions of the TM

000 prefix/suffix of the TM

continue on pg. 7.



B a b a   B a b a B
0→   1
B B B B   B a B B B
0→   1

B a b a   B a b a B   B a b a B   B a b a B B
1→   2→   2   3
B a B B   B a B B B   B a B B B   B a B B B B
←1   ←2   2→   3

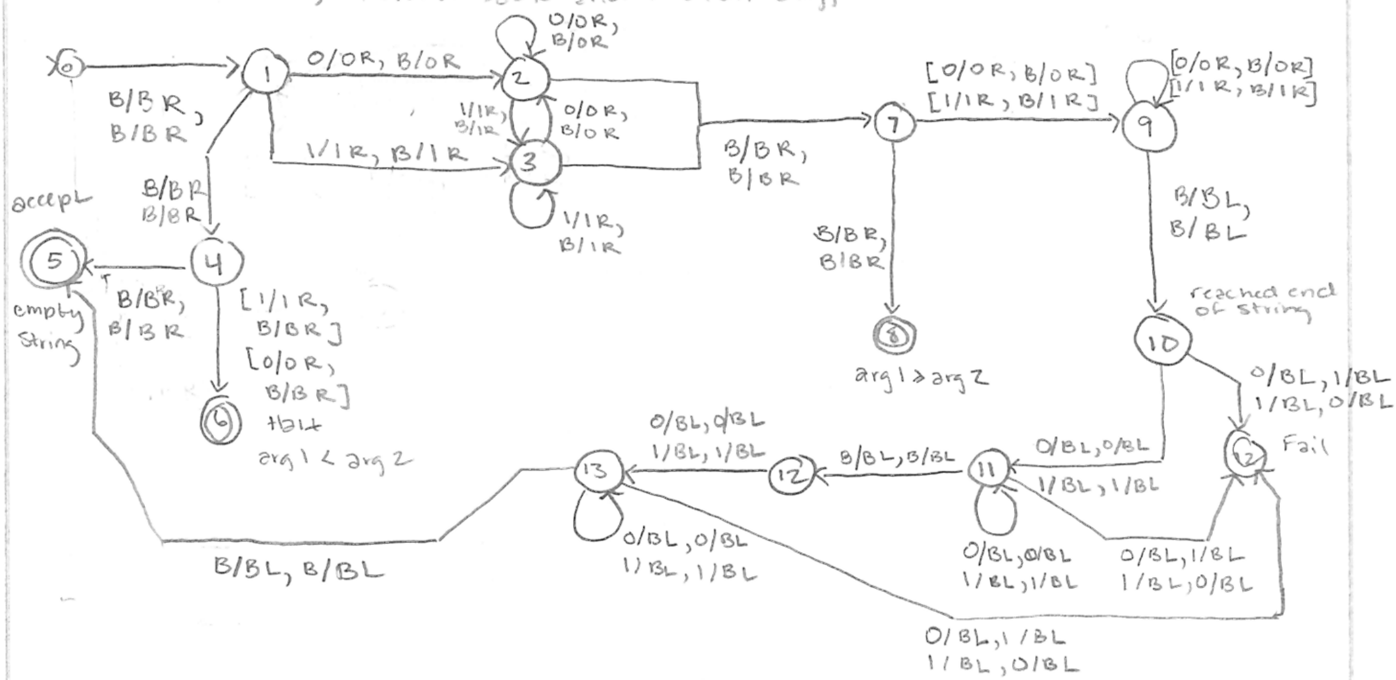
6 Solution

B 0 B 0 B
→q
B B B B B
→q

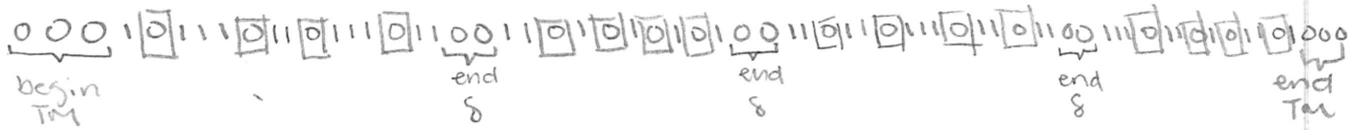
1) Move off first B
2) Copy all symbols from arg 1 to tape 2
3) Reach arg separator, "B"
4) Copy all symbols from arg 2 to tape 2
5) Reach end of strings
6) Traverse back and match args

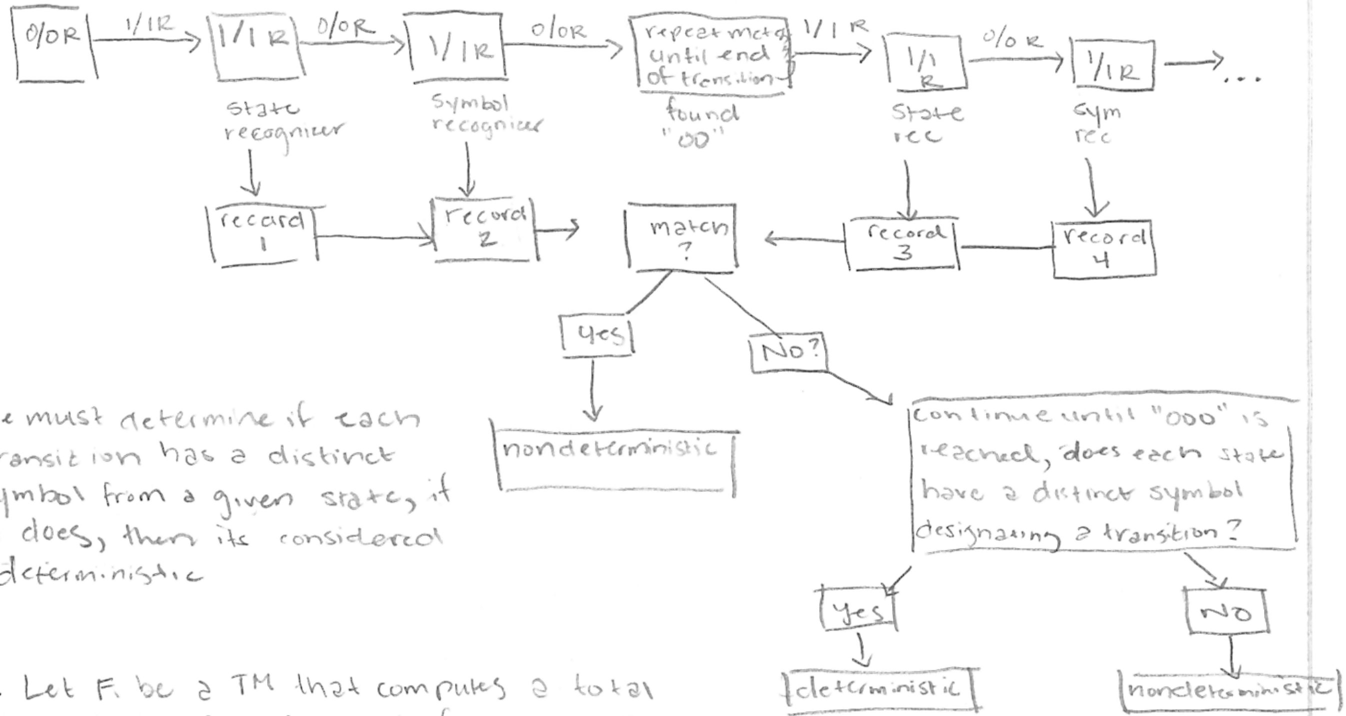
B 0 B 0 B   B 0 B 0 B
B B B B B   B 0 B B B

B 0 B 0 B   B 0 B 0 B
B 0 B B B   B 0 B 0 B



The machine M then is represented by the string

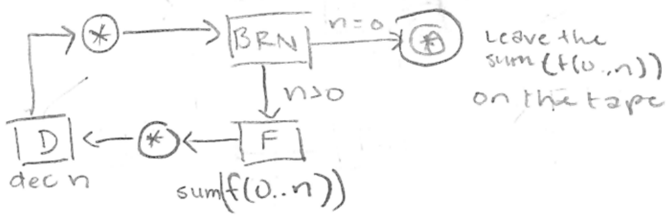




We must determine if each transition has a distinct symbol from a given state, if it does, then its considered deterministic

4. Let F be a TM that computes a total unary number-theoretic function f. Design a machine that computes the function

$$g(n) = \sum_{i=0}^{n} f(i)$$ The summation of function application TM



8. Explain the fundamental difference between the "Halts on the n'th transition problem" from example 11.5.2 and the "Halting Problem" that makes the former decidable and the latter undecidable

Halts on n^{th} transition problem is decidable because there is an algorithmic solution that solves this particular example, described in 11.5.2, whereas the Halting Problem refers to the superset of all possible program-input pairs and a general algorithm that would solve the Halting problem – which doesn't exist, and so it is not decidable.

9.