Midterm - Tuesday, Feb. 8

Quiz 7 - today

HW 4 - due Saturday

HW 5 - due Monday 11:00 am

Review Session - Monday 7pm

<u>Example</u>: Turing machine that enumerates the set $\{1^n : n \text{ even}\} = \{\epsilon, 11, 1111, 111111, \dots\}$

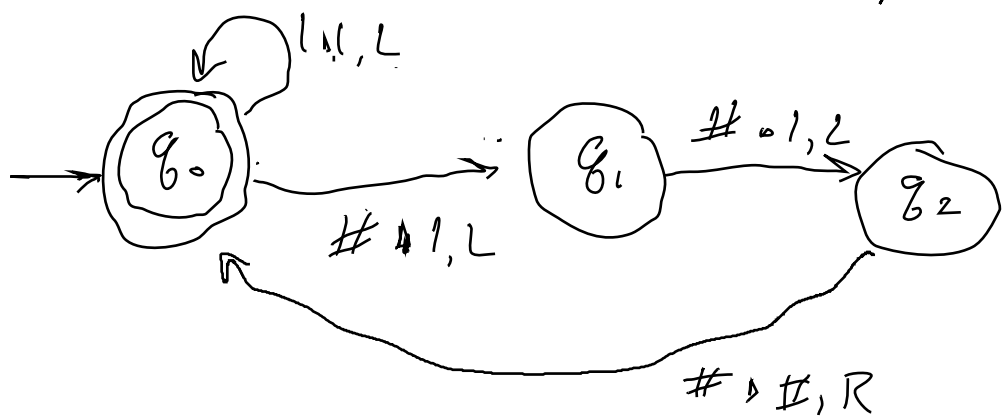
$\Sigma = \{1\}$ $\quad \Gamma = \{1, \#\}$ - Starts on blank tape

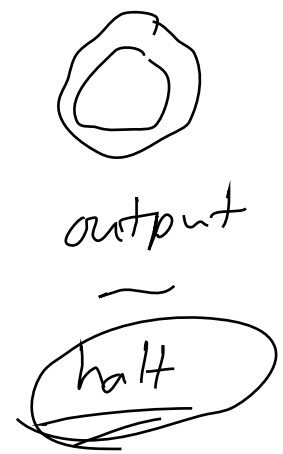
Turing machine: $(\Gamma, \Sigma, Q, q_0, \delta, Q_{halt}, q_{out})$

$Q_{halt} = \emptyset$ $\qquad$ In my example: $q_{out} = q_0$

$\#\#\#\#$
$\uparrow$
$q_0$

$\#\#1\#$
$\uparrow$
$q_1$




$\#\#\#\#$
$\uparrow$
$q_0$

$\#11\#$
$\uparrow$
$q_2$

$\#11\#$
$\uparrow$
$q_0$

$\#11$

3 type of machines

$Q_{halt} = \{q_{acc}, q_{rej}\}$ - (semi) decides a relation

$Q_{halt} = \{q_{out}\}$ - computes or partial computes a function

$Q_{halt} = \phi$ - enumerates a set or a relation.

Turing machine that copies a string

<u>Start</u> $\# \underset{\uparrow}{w} \#$ End with $\# w \# w \#$

$\#0100\# \mapsto \#0100\#0100\#$

$\Sigma = \{0,1\}$ $\Gamma = \{0, 1, 0', 1', \#\}$

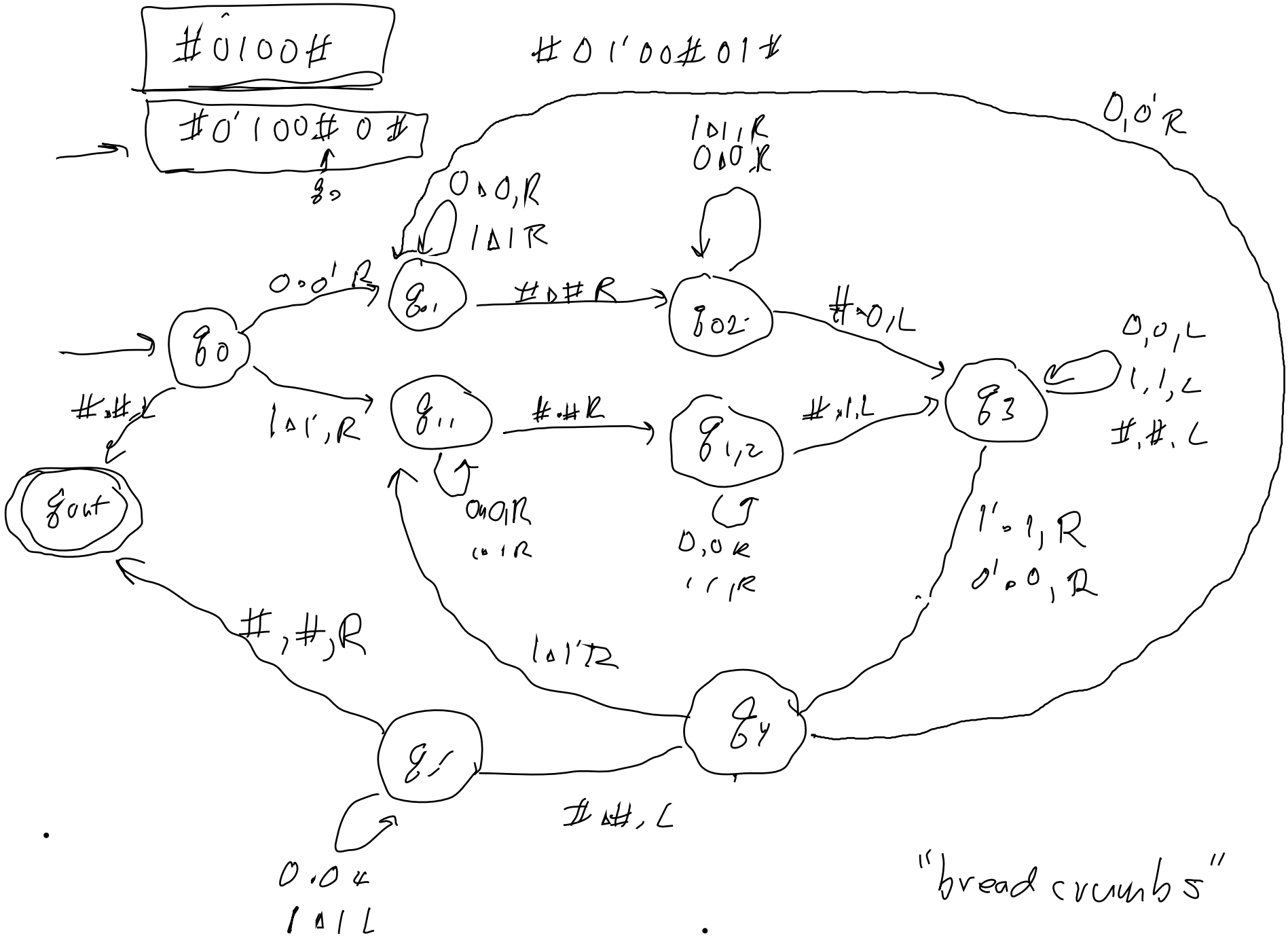
<u>Strategy</u>

$q_0$: At a symbol to be copied next
(If #, done - Add prime
→ remember 0 or 1 (i) + move right

$q_{0,1}, q_{0,2}$ } remember $i=0$ or $i=1$ (to be copied)
$q_{1,1}, q_{1,2}$ } + scanning right around to 2nd #
+ copy the symbol $i$

$q_3$, - scan leftward back to 0' or 1'
change back to 0 or 1

$q_4, q_5$ - handle the next 0 or 1 (similarly to $q_0$)


#0100#
#0'100#0#
q0
#01'00#01#
0,0'R
1,1,R
0,0,R
0,0,R
1,1,R
0,0'R
q01
#,#,R
q02
#,0,L
q0
0,0,L
1,1,L
#,#,L
#,#,L
1,1',R
q11
#,#R
q12
#,1,L
q3
qout
0,0,R
1,1R
0,0R
1,1R
1',1,R
0',0,R
#,#,R
1,1'R
q4
q5
#,#,L
0,0L
1,1L
"bread crumbs"

**Theorem**: the function $w \mapsto ww$ is Turing computable.

$0100 \mapsto 01000100$

**Theorem** $\{ w\#w : w \in \Sigma^* \}$ is Turing decidable

$\{ ww : w \in \Sigma^* \}$ " " "

(Non-negative) Integers – binary notation – permit leading 0's

**Theorem**: $n \mapsto n+1$ is Turing computable

$1100 \mapsto 1101$ $\quad 12+1=13$

$1011 \mapsto 1100$ $\quad 11+1=12$

$\#\#1111\#1\ldots$

$\#10000\#$

Scan to end (right end)

Change 1's to 0's

Change 1st encountered 0 or # to 1

Then change nothing else. → Scan left to #
Step 1 symbol Right

**Theorem** $n, m \longmapsto n+m$ is Turing computable

$\# \underbrace{0101}_{n} \# \underbrace{001001}_{m} \#$

$n, m$ - are getting changed

**Loop** If $m=0$, output the ~~number~~ left number $n$

Subtract 1 from $m$

Add 1 to $n$

Endloop

**Theorem:** $n, m \longmapsto n \cdot m$ is Turing computable

Overall idea

$P$

$0 \# \underbrace{n'}_{\text{same size as } n} \# \underbrace{n}_{} \# \underbrace{m}_{} \#$

where the answer will be

**Loop**

If $m=0$, output $N$ and halt

Set $n' = n$ (copy)

Add $n'$ to $P$

Subtract 1 from $m$

End of loop.

So far

if-then / conditional branching

loops

basic copying, arithmetic operations

<u>Still needed</u> Arbitrary memory access

e.g. Arrays of value a

Accessing data by its address

$\$ w \# v \$ w' \# v \$$

Value $v$ is stored at location $w$

(a in the variable

named $w$)

From this: Church-Turing thesis.