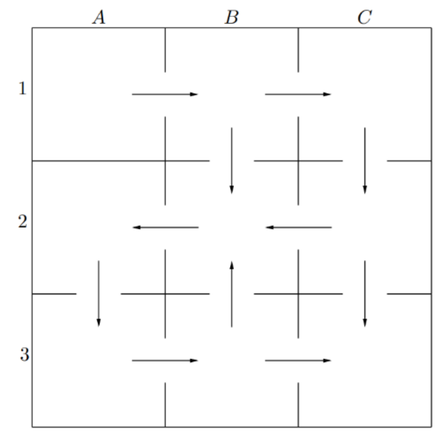
Jon Rippe | Automata HW #2

1



- Possible routes from box A1 to C3, just by looking at it:
  - $EESS$
  - $ES(B2)$
  - $EESW(B2)$
- Possible routes from q_4:
  - $WSEE$
  - $WSEN(B2)$
- Create equations & solve:

$$L = L_{A1} = EESS \cup ESL_{B2} \cup EESWL_{B2} = EESS \cup (ES \cup EESW)L_{B2}$$

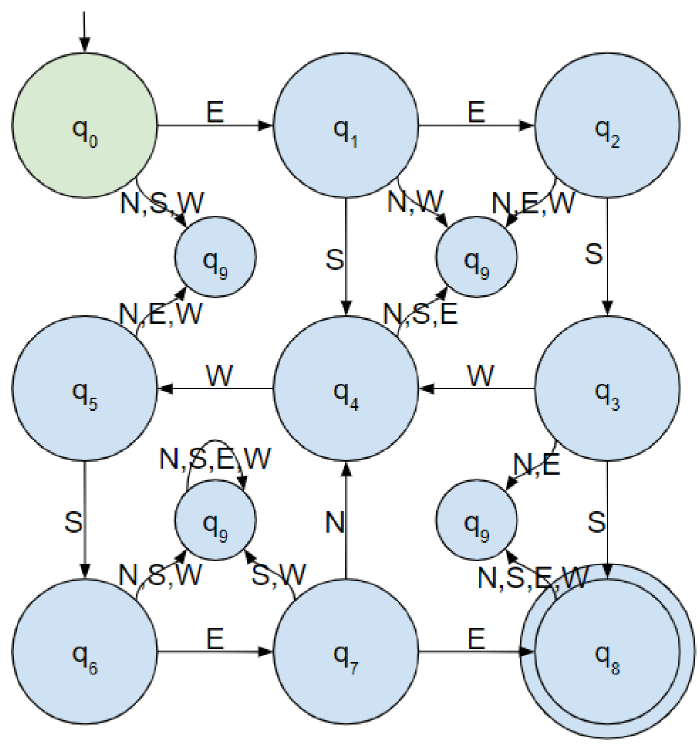
$$L_{B2} = WSENL_{B2} \cup WSEE = (WSEN)^*WSEE, \qquad L = BL \cup C = B^*C$$

$$L = EESS \cup (ES \cup EESW)(WSEN)^*WSEE$$

- DFA recognizing L:

$M = (Q, \Sigma, \delta, q, F)$
$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$
$\Sigma = \{N, S, E, W\}$
$q = q_0$
$F = \{q_8\}$
δ is given by the table:

	N	S	E	W
q_0	q_9	q_9	q_1	q_9
q_1	q_9	q_4	q_2	q_9
q_2	q_9	q_3	q_9	q_9
q_3	q_9	q_8	q_9	q_4
q_4	q_9	q_9	q_9	q_5
q_5	q_9	q_6	q_9	q_9
q_6	q_9	q_9	q_7	q_9
q_7	q_4	q_9	q_8	q_9
q_8	q_9	q_9	q_9	q_9
q_9	q_9	q_9	q_9	q_9



2

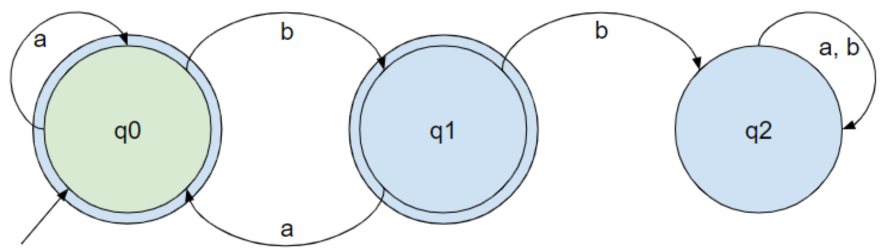
Let $\Sigma = \{a\}$. Let $L = \{a^p | p \text{ is prime}\} \subseteq \Sigma^*$.
Lemma: L is not a regular language.
Proof:
For the sake of contradiction, let's assume L is regular.
Consider the string $s = a^p$ where $p \geq 1$ is the pumping length. Clearly $|s| = p$ and $s \in L$.
By the pumping lemma, $s = xyz$ such that $y \neq \epsilon, |xy| \leq p, xy^iz \in A$ for all $i \geq 0$.
It can be seen that $x = a^q, y = a^r, z = a^s$, where $q, s \geq 0$ and $r \geq 1$
Thus, $|s| = |x| + |y| + |z| = q + r + s = p$, and $|xy^iz| = q + ri + s = n$ where n is prime.
Using algebra, $n = q + ri + s = q + ri + s + r - r = p + ri - r = p + r(i - 1)$.
Because n is prime for all $i \geq 0, n$ is prime for $i = p + 1$, and $p + r\big((p + 1) - 1\big)$ is prime.
Using algebra, $n = p + r\big((p + 1) - 1\big) = p + rp = p(1 + r)$.
This contradicts that n is prime. Thus, L is not regular. ■

3

- DFA accepting F:

$M = (Q, \Sigma, \delta, q, F)$
$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$q = q_0$
$F = \{q_0, q_1\}$
δ is given by the table:

	a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_2



- Regular Expression describing F:

$L_{q_0} = \varepsilon \cup aL_{q_0} \cup bL_{q_1}, \qquad L_{q_1} = \varepsilon \cup aL_{q_0} \cup bL_{q_2}, \qquad L_{q_2} = aL_{q_2} \cup bL_{q_2}$
Because q_2 is inescapable and not a final state:
$L_{q_0} = \varepsilon \cup aL_{q_0} \cup bL_{q_1}, \qquad L_{q_1} = \varepsilon \cup aL_{q_0} \cancel{\cup bL_{q_2}}, \qquad \cancel{L_{q_2} = aL_{q_2} \cup bL_{q_2}}$

$L = L_{q_0} \cup L_{q_1} = \varepsilon \cup aL_{q_0} \cup bL_{q_1} \cup \varepsilon \cup aL_{q_0} = \varepsilon \cup aL_{q_0} \cup bL_{q_1} = L_{q_0}$
$L = L_{q_0} = \varepsilon \cup aL_{q_0} \cup bL_{q_1} = \varepsilon \cup aL \cup b(\varepsilon \cup aL) = \varepsilon \cup aL \cup b\varepsilon \cup baL = (a \cup ba)L \cup \varepsilon \cup b$
$= (a \cup ba)^*(\varepsilon \cup b) = ((a^*ba)^*a^*)(\varepsilon \cup b) = (a^*ba)^*a^* \cup (a^*ba)^*a^*b$

- Let $\Sigma = \{a, b\}$. Let $F \subseteq$
Σ^* the set of all words over Σ that do not contain immediate repetitions of b's. For all $n \geq 0$, let $F_n =$
$\{w \in F | |w| = n\}$.
Lemma: For $n \geq 1$, the cardinality of F_n is the n-th Fibonacci number.
$\text{fib}(0) = 1$
$\text{fib}(1) = 2$
$\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2), \qquad \forall n \geq 2$
Consider the function $B(n) = \text{fib}(n)$, where $B(n) = |A|, A =$ set of all possible string arrangements.
$B(0) = |\{\emptyset\}| = 1, \qquad B(1) = |\{0,1\}| = 2$
Consider a string of length n where $n = |\text{possible strings ending with } 0| + |\text{possible strings ending with } 1|$.
If the n-th digit is 0, then the (n-th $- 1$) digit can be 0 or 1, and $|\text{possible strings ending with } 0| = B(n-1)$.
If the n-th digit is 1, then the (n-th $- 1$) digit must be 0, and $|\text{possible strings ending with } 1| = B(n-2)$.
Thus, $B(n) = B(n-1) + B(n-2)$ and $B(n) = \text{fib}(n)$. ■

4

$G = (\{S, A, B, C\}, \{a, b\}, R, S)$

$S \to ABa$
$S \to CbB$
$A \to CaA$
$A \to a$
$B \to S$
$B \to b$
$C \to Bb$
$C \to A$

Eliminate S on r.h.s. (add S_0 as new start)
$S_0 \to S$
$S \to ABa$
$S \to CbB$
$A \to CaA$
$A \to a$
$B \to S$
$B \to b$
$C \to Bb$
$C \to A$

Eliminate unit rules ($S_0 \to S, B \to S, C \to A$)
$S_0 \to ABa|CbB$
$S \to ABa$
$S \to CbB$
$A \to CaA$
$A \to a$
$B \to ABa|CbB$
$B \to b$
$C \to Bb$
$C \to CaA|a$

Eliminate >2 symbols on r.h.s ($S_0 \to ABa|CbB, S \to ABa, S \to CbB, A \to CaA, B \to ABa|CbB, C \to CaA$)
$S_0 \to S_1a|S_2B$
$S \to S_3a$
$S \to S_4B$
$A \to A_1A$
$A \to a$
$B \to B_1a|B_2B$
$B \to b$
$C \to Bb$
$C \to C_1A|a$
$S_1 \to AB$
$S_2 \to Cb$
$S_3 \to AB$
$S_4 \to Cb$
$A_1 \to Ca$
$B_1 \to AB$
$B_2 \to Cb$
$C_1 \to Ca$

Eliminate $=2$ on r.h.s not both variables ($S_0 \to S_1a, S \to S_3a, B \to B_1a, C \to Bb, S_2 \to Cb, S_4 \to Cb, A_1 \to Ca, B_2 \to Cb, C_1 \to Ca$)
$S_0 \to S_1S_5|S_2B$
$S \to S_3S_6$
$S \to S_4B$
$A \to A_1A$
$A \to a$
$B \to B_1B_3|B_2B$
$B \to b$
$C \to BC_2$
$C \to C_1A|a$
$S_1 \to AB$
$S_2 \to CS_7$
$S_3 \to AB$
$S_4 \to CS_8$
$S_5 \to a$
$S_6 \to a$
$S_7 \to b$
$S_8 \to b$
$A_1 \to CA_2$
$A_2 \to a$
$B_1 \to AB$
$B_2 \to CB_4$
$B_3 \to a$
$B_4 \to b$
$C_1 \to CC_3$
$C_2 \to b$
$C_3 \to a$

$G' = (\{S_0, S, A, B, C, S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, A_1, A_2, B_1, B_2, B_3, B_4, C_1, C_2, C_3\}, \{a, b\}, R', S_0)$
$R' =$ (above work), $\quad L(G') = L(G)$

5

$$M=\left(\{0,1\},\{0,1,\square\},\{q_0,q_1,q_2,q_3,q_{accept},q_{reject}\},\delta,q_0,q_{accept},q_{reject}\right)$$

q_0: Start state. Tapes 1 and 2 are at the leftmost position.

q_{accept}: Accept state, q_{reject}: Reject state

δ:

q_0: start state $q_0\square\square\square \to q_{reject}$ $q_0\square 0\square \to q_{reject}$ $q_0\square 1\square \to q_{reject}$ $q_0 0\square\square \to q_{reject}$ $q_0 1\square\square \to q_{reject}$ $q_0 00\square \to q_1 00\square RRN$ $q_0 01\square \to q_1 01\square RRN$ $q_0 10\square \to q_1 10\square RRN$ $q_0 11\square \to q_1 11\square RRN$	q_1: moving to LSB's $q_1\square\square\square \to q_2\square\square\square LLN$ $q_1\square 0\square \to q_1\square 0\square NRN$ $q_1\square 1\square \to q_1\square 1\square NRN$ $q_1 0\square\square \to q_1 0\square\square RNN$ $q_1 1\square\square \to q_1 1\square\square RNN$ $q_1 00\square \to q_1 00\square RRN$ $q_1 01\square \to q_1 01\square RRN$ $q_1 10\square \to q_1 10\square RRN$ $q_1 11\square \to q_1 11\square RRN$	q_2: adding (no overflow) $q_2 00\square \to q_2 000LLL$ $q_2 01\square \to q_2 011LLL$ $q_2 10\square \to q_2 101LLL$ $q_2 11\square \to q_3 110LLL$ $q_2\square 0\square \to q_2\square 00NLL$ $q_2\square 1\square \to q_2\square 11NLL$ $q_2 0\square\square \to q_2 0\square 0LNL$ $q_2 1\square\square \to q_2 1\square 1LNL$ $q_2\square\square\square \to q_{accept}$
q_3: adding (overflow) $q_3 00\square \to q_2 001LLL$ $q_3 01\square \to q_3 010LLL$ $q_3 10\square \to q_3 100LLL$ $q_3 11\square \to q_3 111LLL$	$q_3\square 0\square \to q_2\square 01NLL$ $q_3\square 1\square \to q_3\square 10NLL$ $q_3 0\square\square \to q_2 0\square 1LNL$ $q_3 1\square\square \to q_3 1\square 0LNL$ $q_3\square\square\square \to q_2\square\square 1NNL$	$q_n XX0 \to$ cannot happen $q_n XX1 \to$ cannot happen

6

State	Tape 1	Tape 2	Tape 3
q_0	$\square\ \underset{\blacktriangle}{1}\ 0\ 1\ 0\ 1\ 0\ \square$	$\square\ \underset{\blacktriangle}{1}\ 1\ 1\ \square$	$\square\ \square\ \square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ \square$
q_1	$\square\ 1\ \underset{\blacktriangle}{0}\ 1\ 0\ 1\ 0\ \square$	$\square\ 1\ \underset{\blacktriangle}{1}\ 1\ \square$	$\square\ \square\ \square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ \square$
q_1	$\square\ 1\ 0\ \underset{\blacktriangle}{1}\ 0\ 1\ 0\ \square$	$\square\ 1\ 1\ \underset{\blacktriangle}{1}\ \square$	$\square\ \square\ \square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ \square$
q_1	$\square\ 1\ 0\ 1\ \underset{\blacktriangle}{0}\ 1\ 0\ \square$	$\square\ 1\ 1\ 1\ \underset{\blacktriangle}{\square}$	$\square\ \square\ \square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ \square$
q_1	$\square\ 1\ 0\ 1\ 0\ \underset{\blacktriangle}{1}\ 0\ \square$	$\square\ 1\ 1\ 1\ \underset{\blacktriangle}{\square}$	$\square\ \square\ \square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ \square$
q_1	$\square\ 1\ 0\ 1\ 0\ 1\ \underset{\blacktriangle}{0}\ \square$	$\square\ 1\ 1\ 1\ \underset{\blacktriangle}{\square}$	$\square\ \square\ \square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ \square$
q_1	$\square\ 1\ 0\ 1\ 0\ 1\ 0\ \underset{\blacktriangle}{\square}$	$\square\ 1\ 1\ 1\ \underset{\blacktriangle}{\square}$	$\square\ \square\ \square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ \square$
q_2	$\square\ 1\ 0\ 1\ 0\ 1\ \underset{\blacktriangle}{0}\ \square$	$\square\ 1\ 1\ \underset{\blacktriangle}{1}\ \square$	$\square\ \square\ \square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ \square$
q_2	$\square\ 1\ 0\ 1\ 0\ \underset{\blacktriangle}{1}\ 0\ \square$	$\square\ 1\ \underset{\blacktriangle}{1}\ 1\ \square$	$\square\ \square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ 1\ \square$
q_3	$\square\ 1\ 0\ 1\ \underset{\blacktriangle}{0}\ 1\ 0\ \square$	$\square\ \underset{\blacktriangle}{1}\ 1\ 1\ \square$	$\square\ \square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ 0\ 1\ \square$
q_3	$\square\ 1\ 0\ \underset{\blacktriangle}{1}\ 0\ 1\ 0\ \square$	$\underset{\blacktriangle}{\square}\ 1\ 1\ 1\ \square$	$\square\ \square\ \square\ \square\ \underset{\blacktriangle}{\square}\ 0\ 0\ 1\ \square$
q_3	$\square\ 1\ \underset{\blacktriangle}{0}\ 1\ 0\ 1\ 0\ \square$	$\underset{\blacktriangle}{\square}\ 1\ 1\ 1\ \square$	$\square\ \square\ \square\ \underset{\blacktriangle}{\square}\ 0\ 0\ 0\ 1\ \square$
q_2	$\square\ \underset{\blacktriangle}{1}\ 0\ 1\ 0\ 1\ 0\ \square$	$\underset{\blacktriangle}{\square}\ 1\ 1\ 1\ \square$	$\square\ \square\ \underset{\blacktriangle}{\square}\ 1\ 0\ 0\ 0\ 1\ \square$
q_2	$\underset{\blacktriangle}{\square}\ 1\ 0\ 1\ 0\ 1\ 0\ \square$	$\underset{\blacktriangle}{\square}\ 1\ 1\ 1\ \square$	$\square\ \underset{\blacktriangle}{\square}\ 1\ 1\ 0\ 0\ 0\ 1\ \square$
q_{accept}			$110001=49=42+7$

7

$M = (\{0,1\}, \{0,1,\square\}, \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_{accept}, q_{reject}\}, \delta, q_0, q_{accept}, q_{reject})$

q_0: Start state. Tapes 1 and 2 are at the leftmost position.

q_{accept}: Accept state, q_{reject}: Reject state

δ:

q_0: start state $q_0\square\square\square \to q_{reject}$ $q_0\square 0\square \to q_{reject}$ $q_0\square 1\square \to q_{reject}$ $q_0 0\square\square \to q_{reject}$ $q_0 1\square\square \to q_{reject}$ $q_0 00\square \to q_1 00\square RRN$ $q_0 01\square \to q_1 01\square RRN$ $q_0 10\square \to q_1 10\square RRN$ $q_0 11\square \to q_1 11\square RRN$	q_1: moving to LSB's $q_1\square\square\square \to q_2\square\square\square LLN$ $q_1\square 0\square \to q_1\square 0\square NRN$ $q_1\square 1\square \to q_1\square 1\square NRN$ $q_1 0\square\square \to q_1 0\square\square RNN$ $q_1 1\square\square \to q_1 1\square\square RNN$ $q_1 00\square \to q_1 00\square RRN$ $q_1 01\square \to q_1 01\square RRN$ $q_1 10\square \to q_1 10\square RRN$ $q_1 11\square \to q_1 11\square RRN$	q_2: adding (no overflow) $q_2 000 \to q_2 000NLL$ $q_2 001 \to q_2 001NLL$ $q_2 010 \to q_2 010NLL$ $q_2 011 \to q_2 011NLL$ $q_2 100 \to q_2 100NLL$ $q_2 101 \to q_2 101NLL$ $q_2 110 \to q_2 111NLL$ $q_2 111 \to q_3 110NLL$ $q_2 00\square \to q_2 000NLL$ $q_2 01\square \to q_2 010NLL$ $q_2 10\square \to q_2 100NLL$ $q_2 11\square \to q_2 111NLL$ $q_2 0\square\square \to q_4 0\square\square LRR$ $q_2 1\square\square \to q_4 1\square\square LRR$
q_3: adding (overflow) $q_3 100 \to q_2 101NLL$ $q_3 101 \to q_3 100NLL$ $q_3 110 \to q_3 110NLL$ $q_3 111 \to q_3 111NLL$ $q_3 10\square \to q_2 101NLL$ $q_3 11\square \to q_3 110NLL$	q_4: resetting to LSB $q_4 000 \to q_4 000NRR$ $q_4 001 \to q_4 001NRR$ $q_4 010 \to q_4 010NRR$ $q_4 011 \to q_4 011NRR$ $q_4 100 \to q_4 100NRR$ $q_4 101 \to q_4 101NRR$ $q_4 110 \to q_4 110NRR$ $q_4 111 \to q_4 111NRR$ $q_4 0\square 0 \to q_4 0\square 0NNR$ $q_4 0\square 1 \to q_4 0\square 1NNR$ $q_4 1\square 0 \to q_4 1\square 0NNR$ $q_4 1\square 1 \to q_4 1\square 1NNR$ $q_4 0\square\square \to q_5 0\square\square NLL$ $q_4 1\square\square \to q_5 1\square\square NLL$ $q_4\square XX \to q_{accept}$	q_5/q_6 : setting product bit position $q_5 0XX \to q_5 0XXRRN$ $q_5 1XX \to q_5 1XXRRN$ $q_5\square XX \to q_6\square XXLLN$ $q_6 X\square X \to q_6 X\square XLLL$ $q_6 X0X \to q_2 X0XNNN$ $q_6 X1X \to q_2 X1XNNN$

8

State	Tape 1	Tape 2	Tape 3
q_0	□ 1▲ 1 □	□ 1▲ 0 0 0 □ □	□ □ □ □ □ □ □ □▲ □
q_1	□ 1 1▲ □	□ 1 0▲ 0 0 □ □	□ □ □ □ □ □ □ □▲ □
q_1	□ 1 1 □▲	□ 1 0 0▲ 0 □ □	□ □ □ □ □ □ □ □▲ □
q_1	□ 1 1 □▲	□ 1 0 0 0▲ □ □	□ □ □ □ □ □ □ □▲ □
q_1	□ 1 1 □▲	□ 1 0 0 0 □▲ □	□ □ □ □ □ □ □ □▲ □
q_2	□ 1 1▲ □	□ 1 0 0 0▲ □ □	□ □ □ □ □ □ □ □▲ □

q_2	□ 1 $\underset{\blacktriangle}{1}$ □	□ 1 0 $\underset{\blacktriangle}{0}$ 0 □ □	□ □ □ □ □ □ $\underset{\blacktriangle}{\square}$ 0 □
q_2	□ 1 $\underset{\blacktriangle}{1}$ □	□ 1 $\underset{\blacktriangle}{0}$ 0 0 □ □	□ □ □ □ □ $\underset{\blacktriangle}{\square}$ 0 0 □
q_2	□ 1 $\underset{\blacktriangle}{1}$ □	□ $\underset{\blacktriangle}{1}$ 0 0 0 □ □	□ □ □ □ $\underset{\blacktriangle}{\square}$ 0 0 0 □
q_2	□ 1 $\underset{\blacktriangle}{1}$ □	$\underset{\blacktriangle}{\square}$ 1 0 0 0 □ □	□ □ □ $\underset{\blacktriangle}{\square}$ 1 0 0 0 □
q_4	□ $\underset{\blacktriangle}{1}$ 1 □	□ $\underset{\blacktriangle}{1}$ 0 0 0 □ □	□ □ □ □ $\underset{\blacktriangle}{1}$ 0 0 0 □
q_4	□ $\underset{\blacktriangle}{1}$ 1 □	□ 1 $\underset{\blacktriangle}{0}$ 0 0 □ □	□ □ □ □ 1 $\underset{\blacktriangle}{0}$ 0 0 □
q_4	□ $\underset{\blacktriangle}{1}$ 1 □	□ 1 0 $\underset{\blacktriangle}{0}$ 0 □ □	□ □ □ □ 1 0 $\underset{\blacktriangle}{0}$ 0 □
q_4	□ $\underset{\blacktriangle}{1}$ 1 □	□ 1 0 0 $\underset{\blacktriangle}{0}$ □ □	□ □ □ □ 1 0 0 $\underset{\blacktriangle}{0}$ □
q_4	□ $\underset{\blacktriangle}{1}$ 1 □	□ 1 0 0 0 $\underset{\blacktriangle}{\square}$ □	□ □ □ □ 1 0 0 0 $\underset{\blacktriangle}{\square}$
q_5	□ $\underset{\blacktriangle}{1}$ 1 □	□ 1 0 0 $\underset{\blacktriangle}{0}$ □ □	□ □ □ □ 1 0 0 $\underset{\blacktriangle}{0}$ □
q_5	□ 1 $\underset{\blacktriangle}{1}$ □	□ 1 0 0 0 $\underset{\blacktriangle}{\square}$ □	□ □ □ □ 1 0 0 $\underset{\blacktriangle}{0}$ □
q_5	□ 1 1 $\underset{\blacktriangle}{\square}$	□ 1 0 0 0 □ $\underset{\blacktriangle}{\square}$	□ □ □ □ 1 0 0 $\underset{\blacktriangle}{0}$ □
q_6	□ 1 $\underset{\blacktriangle}{1}$ □	□ 1 0 0 0 $\underset{\blacktriangle}{\square}$ □	□ □ □ □ 1 0 0 $\underset{\blacktriangle}{0}$ □
q_6	□ $\underset{\blacktriangle}{1}$ 1 □	□ 1 0 0 $\underset{\blacktriangle}{0}$ □ □	□ □ □ □ 1 0 $\underset{\blacktriangle}{0}$ 0 □
q_2	□ $\underset{\blacktriangle}{1}$ 1 □	□ 1 0 0 $\underset{\blacktriangle}{0}$ □ □	□ □ □ □ 1 0 $\underset{\blacktriangle}{0}$ 0 □
q_2	□ $\underset{\blacktriangle}{1}$ 1 □	□ 1 0 $\underset{\blacktriangle}{0}$ 0 □ □	□ □ □ □ 1 $\underset{\blacktriangle}{0}$ 0 0 □
q_2	□ $\underset{\blacktriangle}{1}$ 1 □	□ 1 $\underset{\blacktriangle}{0}$ 0 0 □ □	□ □ □ □ $\underset{\blacktriangle}{1}$ 0 0 0 □
q_2	□ $\underset{\blacktriangle}{1}$ 1 □	□ $\underset{\blacktriangle}{1}$ 0 0 0 □ □	□ □ □ $\underset{\blacktriangle}{\square}$ 1 0 0 0 □
q_2	□ $\underset{\blacktriangle}{1}$ 1 □	$\underset{\blacktriangle}{\square}$ 1 0 0 0 □ □	□ □ $\underset{\blacktriangle}{\square}$ 1 1 0 0 0 □
q_4	$\underset{\blacktriangle}{\square}$ 1 1 □	□ $\underset{\blacktriangle}{1}$ 0 0 0 □ □	□ □ □ $\underset{\blacktriangle}{1}$ 1 0 0 0 □
q_{accept}			$11000 = 24 = 3 * 8$

9

- Per the Church-Turing Thesis, anything that can be written in a coding language (e.g., Java, C++, pseudo-code) can be converted into a Turing machine. Using this Theorem, we can show that subtraction and Euclidean division via Turing machine is possible by writing code that performs the same operation:

    - Subtraction is so much like addition that I could likely implement it with a Turing machine no more complicated than my shift-addition multiplication machine in #7. A machine capable of subtraction would work very similar to the addition machine described in #5 and would only require a few extra states that convert the second tape to its 2's complement before performing an addition operation:

```
While (tape2 != ☐): //need a state for this operation
       flip bit
       move tape2 Right
move tape2 Left

//need 2 states for this operation (one for carry, one for no carry)
carry = add 1 to tape2 value
While (tape2 != ☐):
       move tape2 Left
       if(carry):
              carry = add 1 to tape2 value
move tape2 Right

//We now have an addition problem that can be solved by our defined machine.
```

```
int subtract(int num1, int num2){
       num2 = get2sCompliment(num2);
       answer = num1 + num2;
       return answer;
}
```

    - Using the above algorithm for subtraction, we can create a new Euclidean division algorithm:

```
Tuple euclDivide(int num1, int num2){
       quotient = 0;
       While(num1 > num2){
              quotient++;
              num1 -= num2;
       }
       remainder = num1;
       return (quotient, remainder);
}
```

10

- Using the same argument above, we can argue that any code can be converted into a Turing machine, and multiple Turing machines can be represented by a single Turing machine. The following code specifically performs the operation $\frac{ad+cb}{bd}$ using the logic from our defined Turing machines for addition and multiplication:

```
tape1 = a;
tape2 = b;
tape3 = c;
tape4 = d;
tape5 = TMadd(TMmult(tape1, tape2), TMmult(tape3, tape4));
tape6 = TMmult(tape2, tape4);
```

- For a simplified version $\frac{ac}{bd}$:

```
tape1 = a;
tape2 = b;
tape3 = c;
tape4 = d;
tape5 = TMmult(tape1, tape3);
tape6 = TMmult(tape2, tape4);
```

11

$Theorem: The\ set\ of\ real\ numbers\ \mathbb{R}\ is\ not\ coutable$

$Proof:$

Let $A = \{x \in \mathbb{R} \mid 0 \le x < 1\}$.

Observe that $A \subseteq \mathbb{R}$.

For the sake of contradiction, let's assume A is countable. Then, by definition, the bijection $f: \mathbb{N} \to A$ must exist.

For every $n \in \mathbb{N}$ there exists an equation $f(n) = x_n$

$= 0.d_{n1}d_{n2}d_{n3} \ldots$ where d is a digit from 0 to 9 and there exists a d_{ni} for every $i \in \mathbb{N}$.

We now define an arbitrary real number $x_\mathrm{r} \in A \mid x_\mathrm{r} = 0.d_1d_2d_3 \ldots$ where:

$$d_i = \begin{cases} 1, & \text{if } d_{n_\mathrm{r}i} = 2 \\ 2, & \text{if } d_{n_\mathrm{r}i} \neq 2 \end{cases}$$

Because $x_\mathrm{r} \in A$, there must exist a number $n_\mathrm{r} \in \mathbb{N}$ such that $f(n_\mathrm{r}) = x_\mathrm{r}$. Thus, $f(n_\mathrm{r}) = x_\mathrm{r} = 0.d_{n_\mathrm{r}1}d_{n_\mathrm{r}2}d_{n_\mathrm{r}3} \ldots$

$= 0.d_1d_2d_3 \ldots$

It can be seen that $d_i = d_{n_\mathrm{r}i}$. This is a contradiction of our defined real number, and thus A is not countable.

Because $A \subseteq \mathbb{R}$, $\mathbb{R}$ is also not countable.■

In class, we showed that each Turing machine can be mapped to an integer number through a function $gT \to \mathbb{N}$, meaning for each Turing machine M, $g(M) =$ a unique Gödel Number.

Corollary: Because the size of $\mathbb{R}$ is greater than the size of $\mathbb{N}$ and the number of Turing Machines is equal to the size of $\mathbb{N}$, the size of $\mathbb{R}$ must be greater than the number of Turing machines, and there are more real numbers than Turing machines.

12

First, let's see if we can keep track of both n and i by changing the way we do the summation. Because addition is commutative, we can add down from a number n to 0 instead of adding up. This is what we'll do, so now our first tape starts at n and decrements after each summation step. When the Turing machine goes to check the first tape and finds it to be 0, it terminates in the accept state.

Next, we're going to get rid of all the fractions inside the parentheses by finding a common denominator and rewrite using algebra. I'm not actually going to do this because it would be a nightmare, but if I *had* to build this machine, it would be my next step.

Now our machine works with the numerators of Bellard's formula. It uses addition, subtraction, and multiplication to find the numerator inside the parentheses and store it in tape #2. It then uses multiplication, or another algorithm for determining if our current i is odd/even, to get the other numerator within the summation, multiply it with tape #2, overwriting tape #2's contents. Our machine then does the same with the denominators, storing the resulting values in tape #3, performing an extra calculation for the denominator outside the summation.

This continues until the value stored in tape #1 is 0.

A quick glance at Bellard's formula shows that the denominator grows at a maximum rate of 2^{10n} while the numerator remains the same (aside from alternating positive to negative). This mean that our discrepancy from true π ($|a_n/b_n - \pi|$) should be $\leq 1/c^n$ as $1/2^{10n}$ will shrink at a faster rate than $1/c^n$.