

guesses (count = 0) ? guess = 0
 while (condition) {

if (word[i] == guess)

{ if (temp[i] != guess)

temp[i] = guess

*(correct)++; -- ch

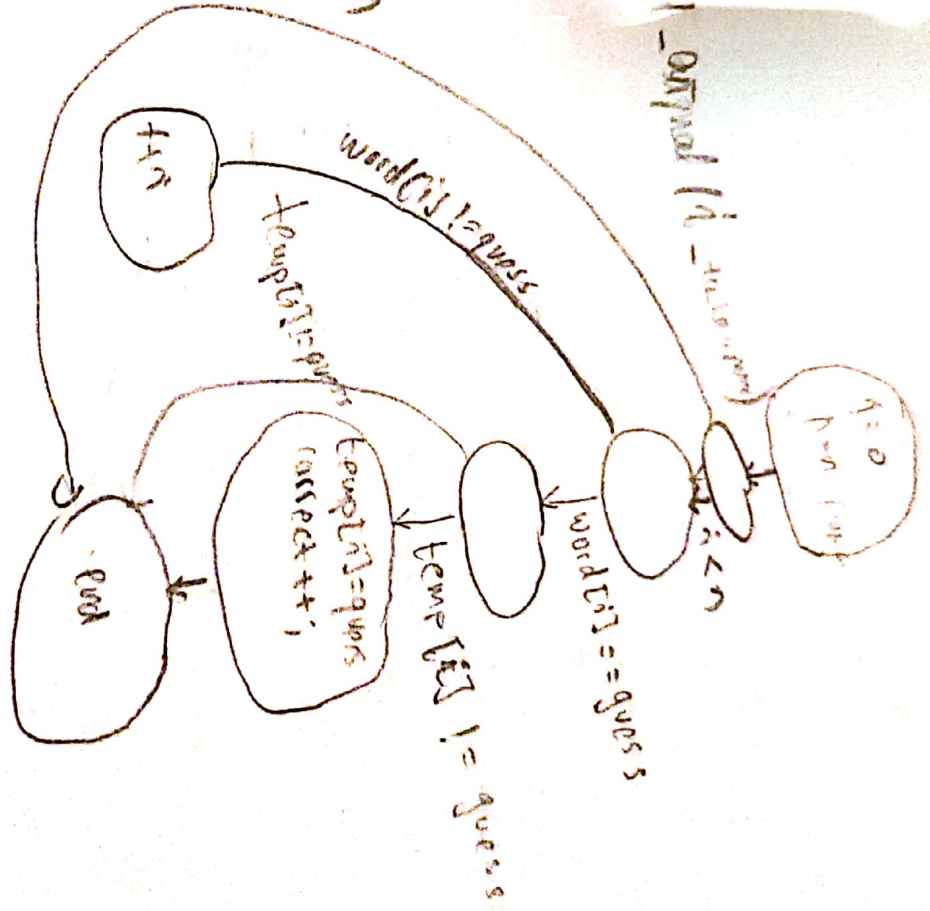
return;

else return;

-- count

{ (mistake)++; -- ch

i < n



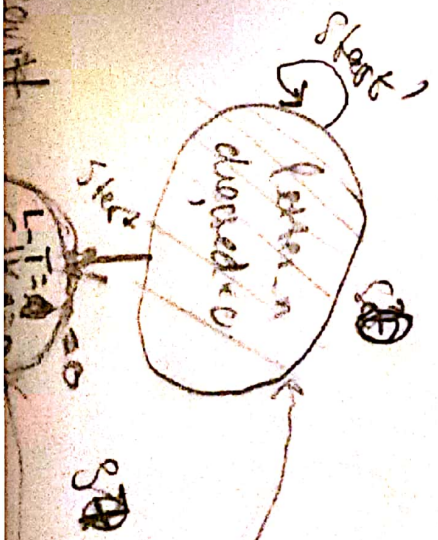
000 → letter 1

001 → 1 2

010 → 1 3

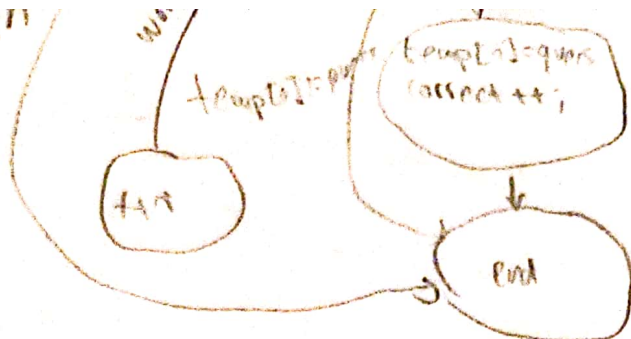
011 → 1 4

0 1 2 3 4 5
 1 2 3 4 5
 1 2 3 4 5



C4

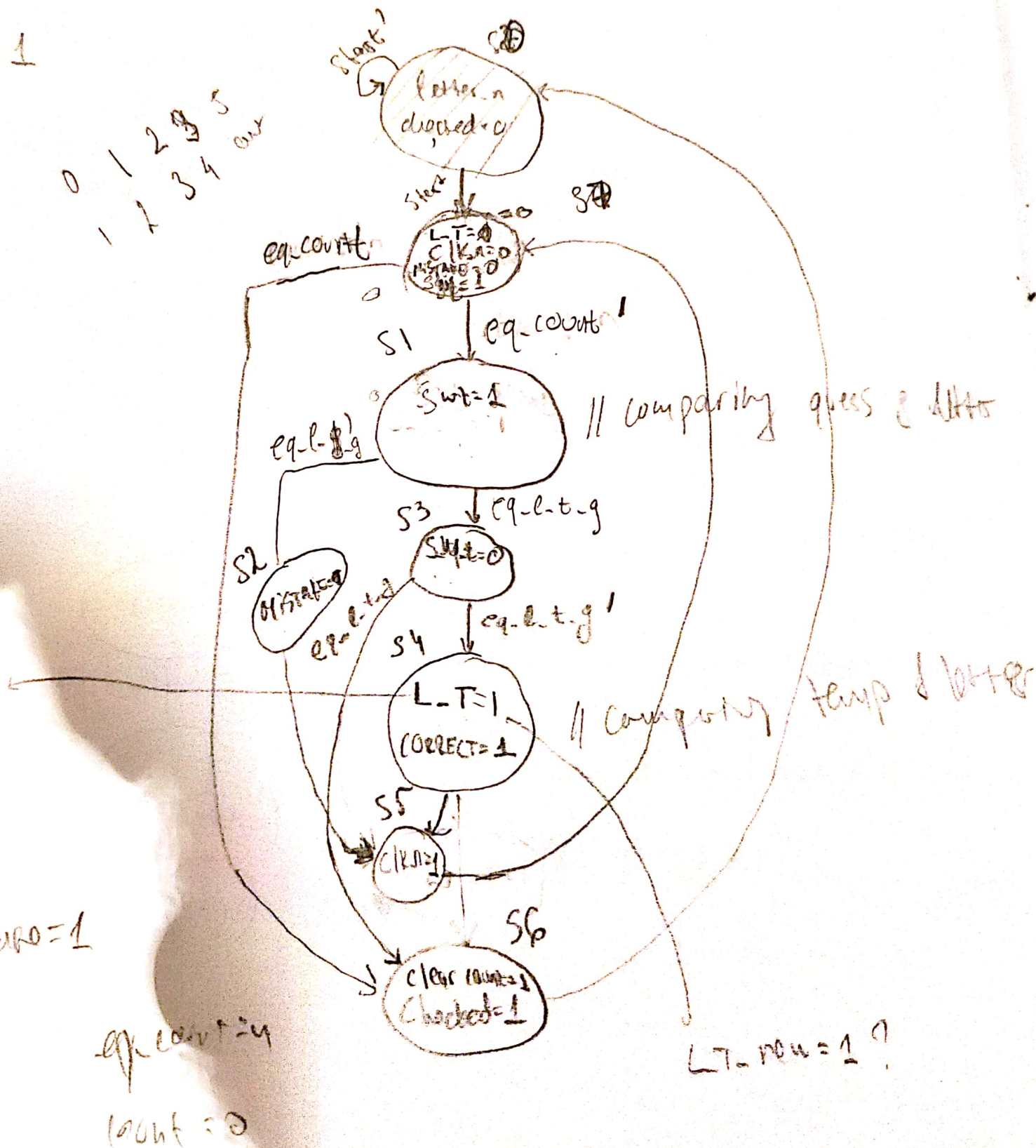
9/12/11



ADD!

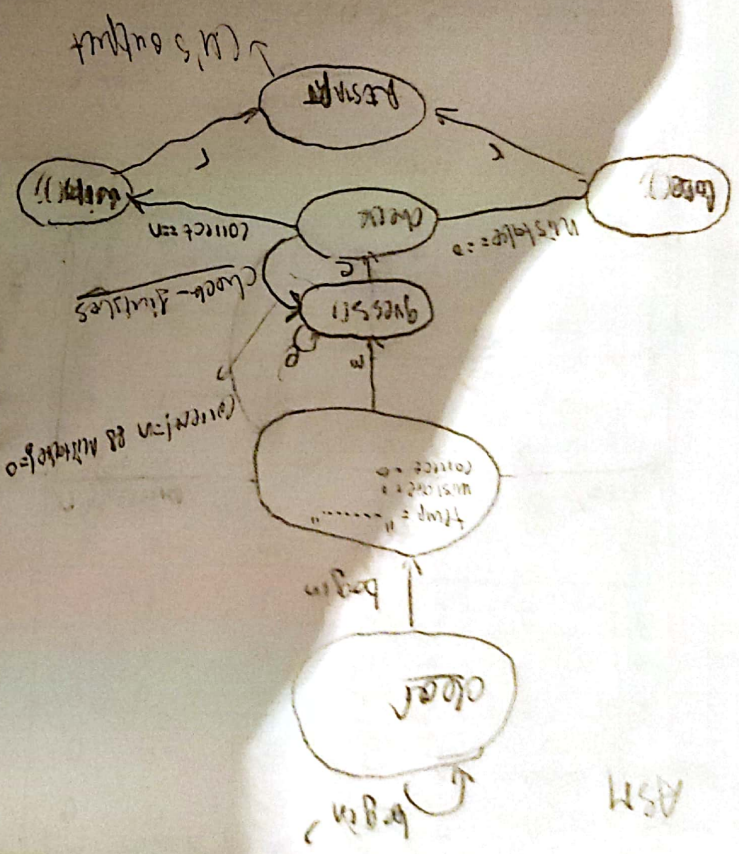
ref 1

0 1 2 3 4 out
1 2 3 4 out



Decoder - output

000	000	000	000	000	000	000	000	000	000
001	000	000	000	000	000	000	000	000	000
010	000	000	000	000	000	000	000	000	000
011	000	000	000	000	000	000	000	000	000
100	000	000	000	000	000	000	000	000	000
101	000	000	000	000	000	000	000	000	000
110	000	000	000	000	000	000	000	000	000
111	000	000	000	000	000	000	000	000	000

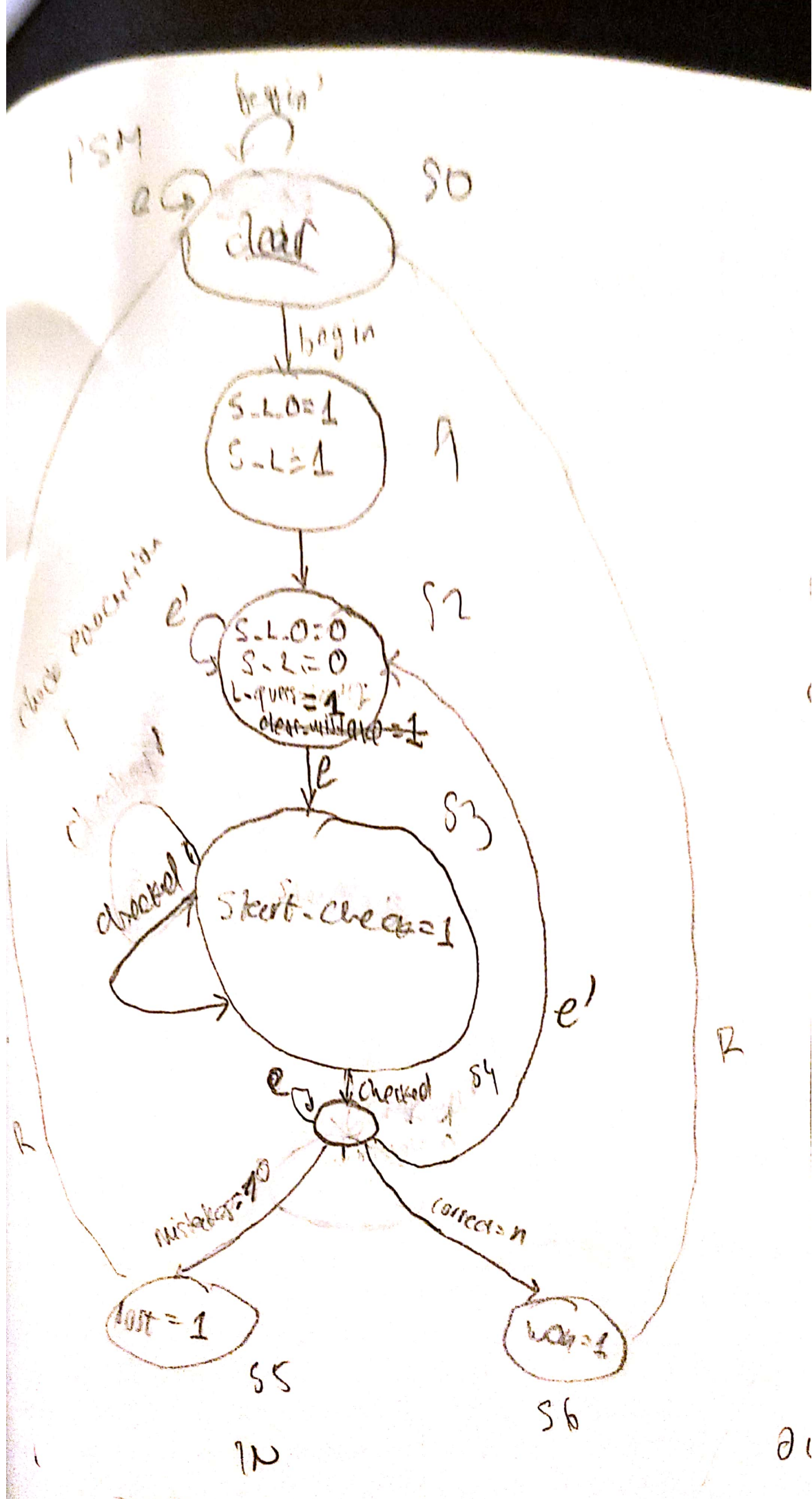


000 - 1
 001 - 2
 010 - 3
 011 - 4
 100 - 5
 101 - 6
 110 - 7
 111 - 8

while (mistakes < 10 && correct == 0)
 {
 guess = getinput();
 check(word, temp, n, mistakes, &correct);
 if (mistakes == n)
 {
 mistakes++;
 }
 }
 if (correct == 0)
 {
 temp = "-----";
 }
 while (begin)

Decoder - 4096 m

000	00000001
001	00000010
010	00000100
011	00000100
100	00001000
101	00010000
110	00100000
111	01000000
1000	11111111
1001	00000000



DATA PART

(correct-hat \sim correct == 1 in check)

Mistake-incr \sim mistake == 1 in check

S-L-1 \sim correct == 1 in check

S-count = \sim current count in check

L-T-undat = 1 \sim correct == 1

Q1: (out check)

	S2	S1	S0	begin	P	res	and	left	win	res	mi	no	S2	res	L	res	dec	mi	S1	w
S0	0	0	0	1	0	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0
S1	0	0	0	x	1	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0
S2	0	1	0	x	0	x	x	x	x	0	1	0	0	1	0	0	0	0	0	0
S3	0	1	1	x	x	x	0	x	x	0	1	1	0	0	0	0	0	0	0	0
S4	1	0	0	x	y	x	x	1	x	1	0	0	0	0	0	0	0	0	0	0
S5	1	0	1	x	x	x	x	x	x	1	0	0	0	0	0	0	0	0	0	0
S6	1	1	0	x	x	x	x	x	x	1	1	0	0	0	0	0	0	0	0	0

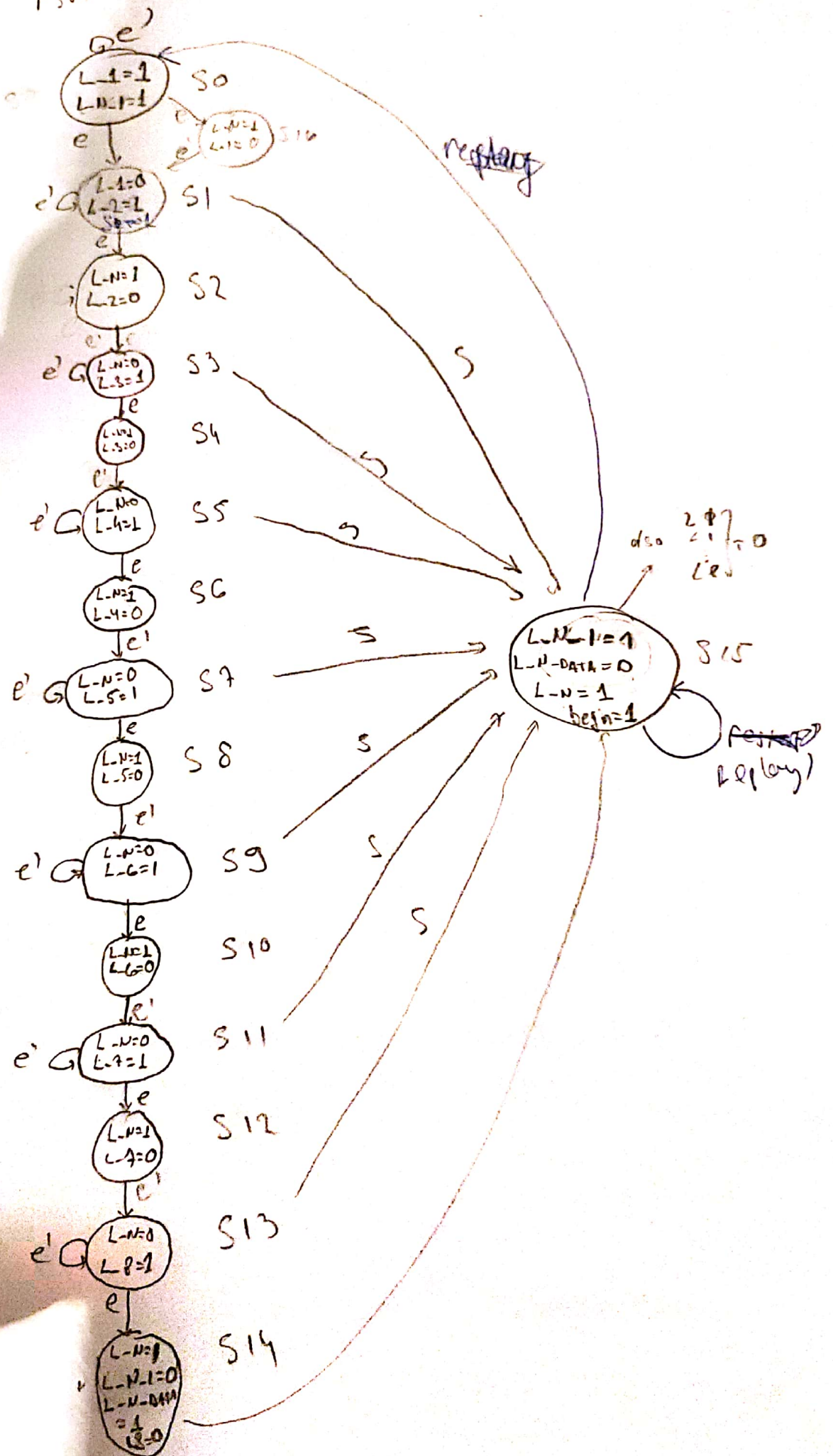
~~M2 = S0~~ & S3 overtook + S5 (res) + S6 (res) + ~~S4~~ (lost) + S4 (lost) + S4 (lost) + S5 (res)

M1 = S1 + S2 + S3 overtook + S4 (lost) + S4 (lost) + S6 (res)

M0 = S0 begin + S2C + S3 overtook + S4 lost + S5 (res)

Added

FSM



being 0s since there is loss 0s (and then use an invariant)

$$\begin{aligned}
 n_0 &= s_1' s_2' s_1' s_0' (e f e') + s_1' s_2' s_1' s_0' \text{ start}' (e f e') + s_1' s_2' s_1' s_0' + s_1' s_2' s_1' s_0' \text{ start}' (e f e') + s_1' s_2' s_1' s_0' + s_1' s_2' s_1' s_0' \text{ start}' (e f e') + s_1' s_2' s_1' s_0' + s_1' s_2' s_1' s_0' \text{ start}' (e f e') \\
 &= s_1' s_2' s_1' s_0' + s_1' s_2' s_1' s_0' \text{ start}' + s_1' s_2' s_1' s_0' + s_1' s_2' s_1' s_0' \text{ start}' + s_1' s_2' s_1' s_0' + s_1' s_2' s_1' s_0' \text{ start}' + s_1' s_2' s_1' s_0' + s_1' s_2' s_1' s_0' \text{ start}' \\
 m &= s_0 + s_1 \text{ start}' + s_2 + s_3 e' \text{ start}' + s_4 + s_5 e' \text{ start}' + s_6 + s_7 e' \text{ start}' + s_8 + s_9 e' \text{ start}' + s_{10} + s_{11} e' \text{ start}' + s_{12} + s_{13} e' \text{ start}' \\
 n_1 &= s_0 + s_1 e' \text{ start}' + s_3 e' \text{ start}' + s_4 + s_5 e' \text{ start}' + s_7 e' \text{ start}' + s_8 + s_9 e' \text{ start}' + s_{11} e' \text{ start}' + s_{12} + s_{13} e' \text{ start}' \\
 n_0 &= s_0 e' + s_1 e' \text{ start}' + s_3 e' \text{ start}' + s_5 e' \text{ start}' + s_7 e' \text{ start}' + s_9 e' \text{ start}' + s_{11} e' \text{ start}' + s_{13} e' \text{ start}' \\
 &= s_0 e' + e' \text{ start}' (s_1 + s_3 + s_5 + s_7 + s_9 + s_{11} + s_{13}) + e' (s_2 + s_4 + s_6 + s_8 + s_{10} + s_{12} + s_{14}) \\
 n_1 &= s_{16} e' + s_{17} e' = e' (s_{16} + s_{17})
 \end{aligned}$$

s_3	s_2	s_1	s_0	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	$k-n$	Increment	$L-8$
s_0				1	0	0	0	0	0	0	0	0	1	0
s_1				0	1	0	0	0	0	0	0	0	1	0
s_2				0	0	0	0	0	0	0	0	1	1	0
s_3				0	0	1	0	0	0	0	0	0	1	0
s_4				0	0	0	0	0	0	0	0	1	1	0
s_5				0	0	0	1	0	0	0	0	0	1	0
s_6				0	0	0	0	0	0	0	0	1	1	0
s_7				0	0	0	0	1	0	0	0	0	1	0
s_8				0	0	0	0	0	0	0	0	1	1	0
s_9				0	0	0	0	0	1	0	0	0	1	0
s_{10}				0	0	0	0	0	0	0	0	1	1	0
s_{11}				0	0	0	0	0	0	1	0	0	1	0
s_{12}				0	0	0	0	0	0	0	0	1	1	0
s_{13}				0	0	0	0	0	0	0	0	0	1	0
s_{14}				0	0	0	0	0	0	0	0	0	0	1
s_{15}				0	0	0	0	0	0	0	0	0	0	0

$L_1 = s_0$
 $L_2 = s_1$
 $L_3 = s_3$
 $L_4 = s_5$
 $L_5 = s_7$
 $L_6 = s_9$
 $L_7 = s_{11}$
 $L_8 = s_{13}$
 $L-N = s_2 + s_4 + s_6 + s_8 + s_{10} + s_{12}$
 $\text{Increment} = s_{14} \Rightarrow \text{Increment} = s_{14}$
 $L-8 = s_{14}$

In ^{replay}

Out

	S3	S2	S1	S0	er Start	m1 n3	n2	n1	n0
S0	0 0	0 0	0 0	0 0	0 x x	0 0	0 0	0 0	0 0
	0 0	0 0	0 0	0 0	1 x x	1 0/1	0/1	0/1	1/0
S1	0 0	0 0	0 0	1 1	0 y 0	0 0	0 0	0 0	1 1
	0 0	0 0	0 0	1 1	1 x 0	0 0	0 0	1 1	0 0
	0 0	0 0	0 0	1 1	x x x	1 1	1 1	1 1	1 1
S2	0 0	0 0	1 1	0 0	0 x x	0 0	0 0	1 1	0 0
	0 0	0 0	1 1	0 0	1 x x	0 0	0 0	1 1	0 0
	0 0	0 0	1 1	0 0	0 x 0	0 0	0 0	1 1	1 1
S3	0 0	0 0	1 1	1 1	0 1 0	0 1	0 0	0 0	0 0
	0 0	0 0	1 1	1 1	x x 1	1 1	1 1	1 1	1 1
S4	0 0	1 1	0 0	0 0	0 x x	0 0	1 1	0 0	0 0
	0 0	1 1	0 0	0 0	1 x x	0 0	1 1	0 0	0 0
	0 0	1 1	0 0	0 0	0 x 0	0 0	1 1	0 0	1 1
S5	0 0	1 1	0 0	1 1	1 x 0	0 1	1 1	0 0	0 0
	0 0	1 1	0 0	1 1	x x 1	1 1	1 1	1 1	1 1
S6	0 0	1 1	1 1	0 0	0 x x	0 0	1 1	1 1	0 0
	0 0	1 1	1 1	0 0	1 x x	0 0	1 1	1 1	0 0
	0 0	1 1	1 1	0 0	0 x 0	0 0	1 1	1 1	1 1
S7	0 0	1 1	1 1	1 1	1 x 0	1 0	0 0	0 0	0 0
	0 0	1 1	1 1	1 1	x x 1	1 1	1 1	1 1	1 1
S8	1 0	0 0	0 0	0 0	0 x x	1 0	0 0	0 0	0 0
	1 0	0 0	0 0	0 0	1 x x	1 0	0 0	0 0	0 0
	1 0	0 0	0 0	0 0	0 x 0	1 0	0 0	0 0	1 1
S9	1 0	0 0	0 0	1 1	1 x 0	1 0	0 0	1 1	0 0
	1 0	0 0	0 0	1 1	x x 1	1 1	1 1	1 1	1 1
S10	1 0	0 0	1 1	0 0	0 x x	1 0	0 0	1 1	0 0
	1 0	0 0	1 1	0 0	1 x x	1 0	0 0	1 1	0 0
	1 0	0 0	1 1	0 0	0 x 0	1 0	0 0	1 1	1 1
S11	1 0	0 0	1 1	1 1	1 x 0	1 1	0 0	0 0	0 0
	1 0	0 0	1 1	1 1	x x 1	1 1	1 1	1 1	1 1
S12	1 1	0 0	0 0	0 0	0 x x	1 1	0 0	0 0	0 0
	1 1	0 0	0 0	0 0	1 x x	1 1	0 0	0 0	0 0
	1 1	0 0	0 0	0 0	0 x 0	1 1	0 0	0 0	1 1
S13	1 1	0 0	0 0	1 1	1 x 0	1 1	1 1	0 0	0 0
	1 1	0 0	0 0	1 1	x x 1	1 1	1 1	1 1	1 1
S14	1 1	1 1	0 0	0 0	0 x x	1 1	1 1	0 0	0 0
	1 1	1 1	0 0	0 0	1 x x	1 1	1 1	0 0	0 0
	1 1	1 1	0 0	0 0	0 x 0	1 1	1 1	0 0	1 1
S15	1 1	1 1	1 1	1 1	x 0 x	0 0	0 0	0 0	0 0
	1 1	1 1	1 1	1 1	x 1 x	0 0	0 0	0 0	0 0
	1 1	1 1	1 1	1 1	0 x x	0 0	0 0	0 0	1 1
S16	1 0 0 0 0				1 x x	1 0	0 0	0 0	0 0

ENDING STATE

new / added

nos ... S16.e

c ... + S16.e