Math 322

Finite State Machine with output.

Design: States = Knowledge

- Input × State → state (transition)
- Input × State → (output)

Language of a Machine alphabet = {0, 1}

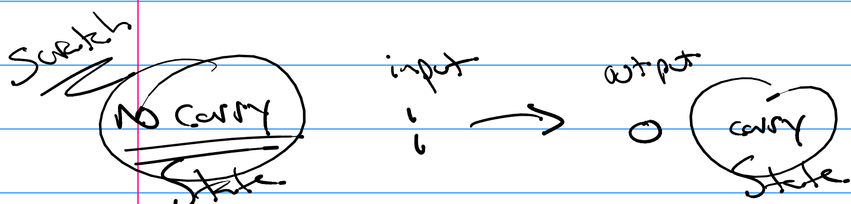
↳ String when input into M the last output symbol is 1.

ex add binary numbers.

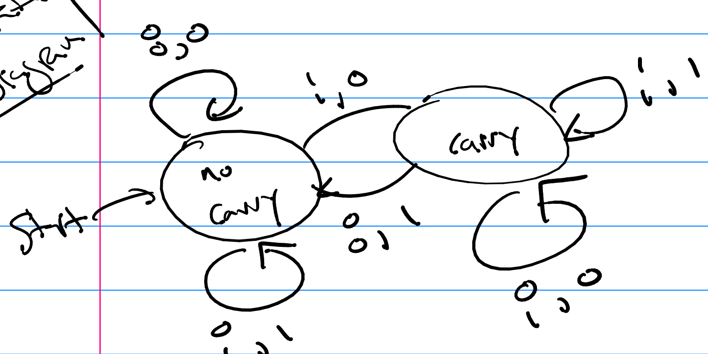
```
  1 1 1 0 0        ← Knowledge of a carry
  0(1,0,1,1,1)_2   ← inputs
+ 0(0,1,1,0,1)_2
  1 0 0 1 0 0      ← outputs
```

$(1,0)_2$ = two

Sketch



State Diagram



State Table

States	f: 1/1	f: 0/1	f: 0/0	g: 1/1	g: 0/1	g: 0/0
no carry	c	nc	nc	0	1	0
carry	c	c	nc	1	0	1

```
  0 1 1
  0 1 1
  -----
  1 1 0
```

13.3 Finite State Automata

Finite State Machine **without** output

$M = (S, I, f, s_0, F)$

S : finite non-empty set of states

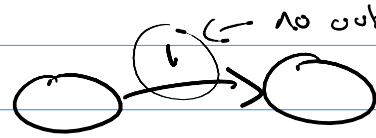
$F \subseteq S$: set of final states

$s_0 \in S$: start state

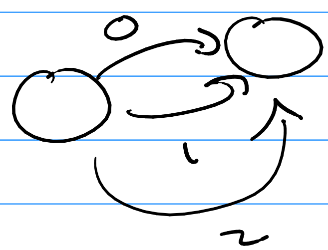
$f : S \times I \to S$ transition function

State Diagrams : ① modify states of F to ◎

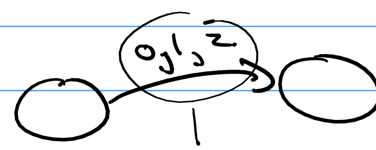
② no output so



b/c of no output symbol we will be lazy for similar transitions...



just use



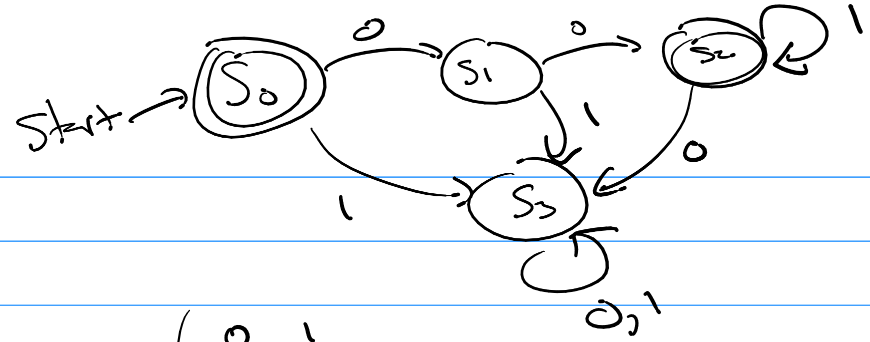
shorthand for 3 arrows.

Main use of F.S.A is string recognition.

Def - M recognizes a string $x \in I^*$ if x takes s_0 to a final state.

- $L(M)$ the language of M is the set of all such x that M recognizes.

ex



0 1010111101

$S_1\ S_3\ S_3\ \ldots$

table

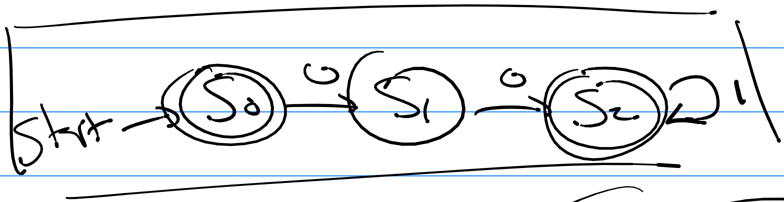
	0	1
S_0	S_1	S_3
S_1	S_2	S_3
S_2	S_3	S_2
S_3	S_3	S_3

String M recog.

$L(M) \rightarrow$

λ
00
001
0011
00111
⋮

ex



$L(M) = \{\lambda, 00, 001, 0011, \ldots\}$

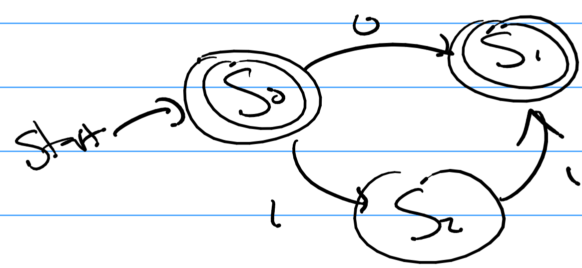
table

	0	1
S_0	S_1	
S_1	S_2	
S_2		S_2

not a function!

it is a partial function

Make M (creatively) such that $L(M) = \{\lambda, 0, 11\}$



this is a partial function.

total function:

Notation for strings:

A is a set of symbols

B is a set of symbols

① A^* Kleene closure $A^* = \{\lambda, \boxed{\text{singletons of } A}, \boxed{\text{take two syms of } A}, \ldots\}$

(ex) $A = \{0, a, \square\}$

$A^* = \{\lambda, 0, a, \square, 0a, 0\square, 00, a0, aa, a\square,$
$\square 0, \square a, \square\square, 000, \ldots, \square\square\square, \ldots\}$

→ $\{0,1\}^*$ is <u>all</u> bitstrings

② $AB = \{x_A x_B \mid x_A \in A, x_B \in B\}$

$\{0,1\}\{00,2,3\} = \{000, 02, 03, 100, 12, 13\}$

③ $A \cup B = \{e \mid e \in A \text{ or } e \in B\}$

$\{0,1\} \cup \{00,2,3\} = \{0, 1, 00, 2, 3\}$

So from above (ex) $L(M) = \{\lambda, 00, 001, 0011, 00111, \ldots\}$

$$L(M) = \boxed{\{\lambda, \underbrace{001^*}_{00\{\lambda, 1, 11, 111, \ldots\}}\}}$$

$L(M) = 01, 0111, 011111, \ldots = 01(11)^*$

$= 01^{2n-1}, n = 1, 2, 3, \ldots$

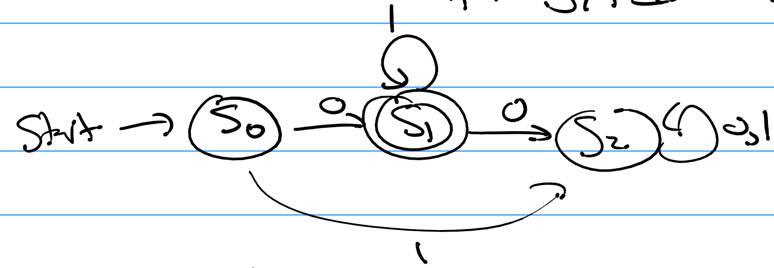
Def equivalent machines. M_1, M_2

if $L(M_1) = L(M_2)$ call them equivalent.

Deterministic F.S.A. $M = (S, I, f, s_0, F)$

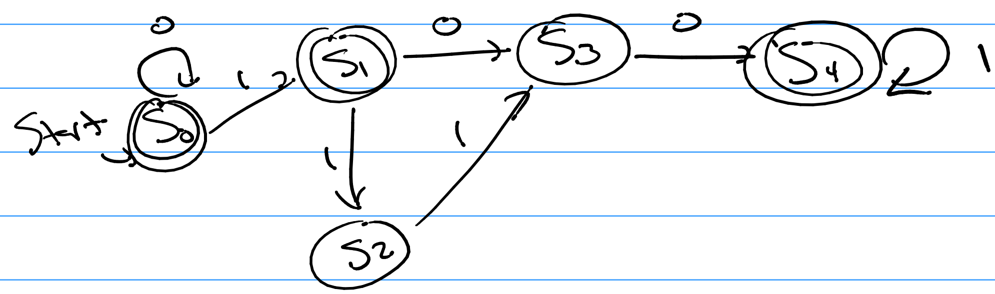
$f : S \times I \to S$

ex



$L(M) = \{0, 01, 011, 0111, \ldots\} = 01^*$

ex



$L(M) = \{0^*, 0^*1, 0^*100 1^*, 0^*11101^*\}$

Non-Deterministic FSA

$f : S \times I \to P(S)$