Math 322

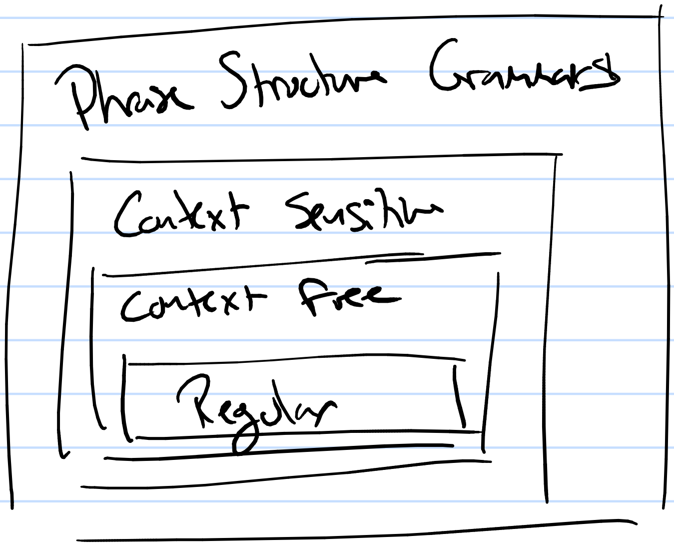
Ch 4

Grammars

$\Sigma, N, T, S, \underline{\underline{P}}$

$\uparrow$

productions



$L(G) = \{ w \in T^* \mid S \overset{*}{\Rightarrow} w \}$

3.4

So far we have made sets of strings (Languages, regular expressions, etc)

$L = \{2, 3, 5, 7, 11, 13, \ldots\}$

$\cancel{1}$ $\boxed{2}$ $\boxed{3}$ $\cancel{4}$ $\boxed{5}$ $\cancel{6}$ $\boxed{7}$ $\cancel{8}$ $\cancel{9}$ $\cancel{10}$

$\boxed{11}$ $\cancel{14}$ $\boxed{13}$ $\cancel{14}$ $\cancel{15}$ $\ldots$

Finite State Machines

Finite State Machine

Type 1 Finite State Machine with output

Type 2 Finite State Machine without output

called Finite State Automata (FSA)

FSM w/ output

(ex) candy Machine

given 2¢ candy machine that has a return $ button, a give candy button, and a 1¢ coin slot.

1st finite states $= S$

knowledge states

2nd start state $= S_0$

3rd Input Alphabet $= I$

(ex) 1¢, c, r

penny, give candy, return money

4th Output Alphabet $= O$

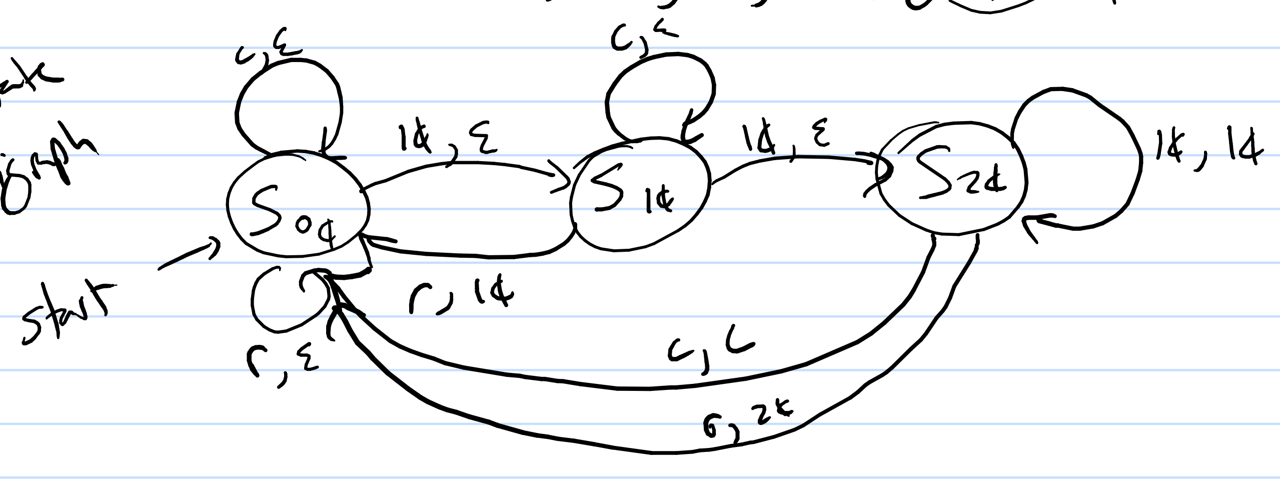
(ex) ε, 1¢, 2¢, c

$$M = (S, S_0, I, O, f, g)$$

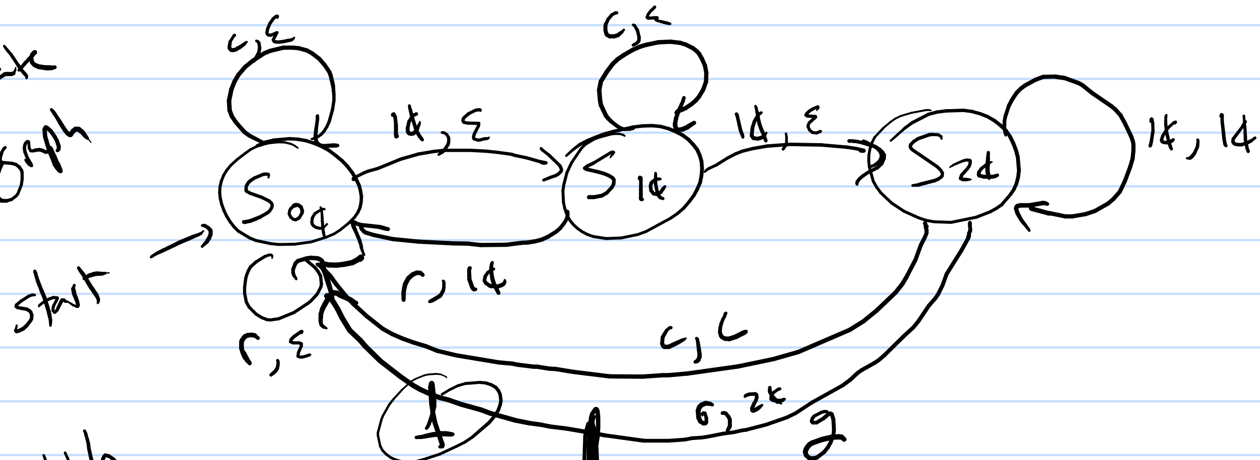
f: input to newstate

g: input to output

State Digraph



State Digraph



State table

States	1¢	c	r	1¢	c	r
0¢	1¢	0¢	0¢	ε	ε	ε
1¢	2¢	1¢	0¢	ε	ε	1¢
2¢	2¢	0¢	0¢	1¢	c	2¢

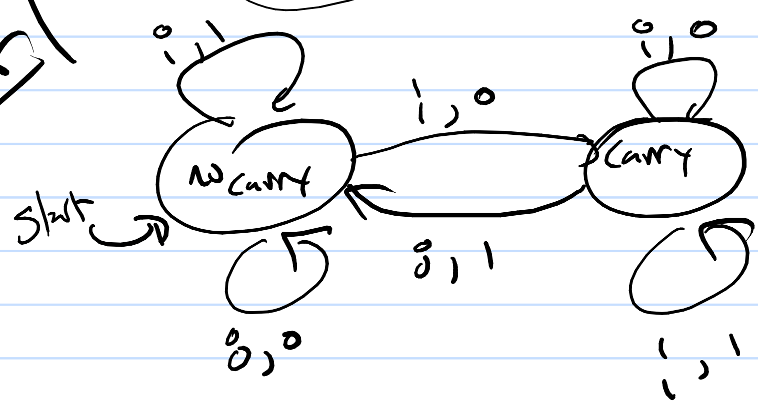
ex binary adder

0	0	1	1
0	1	0	1

2 states:

no carry

carry



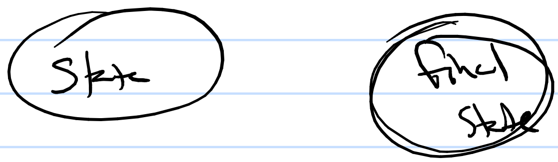
Modify these machines to have no output but the states are partitioned into Final States and Non-final states

Finite State Automaton

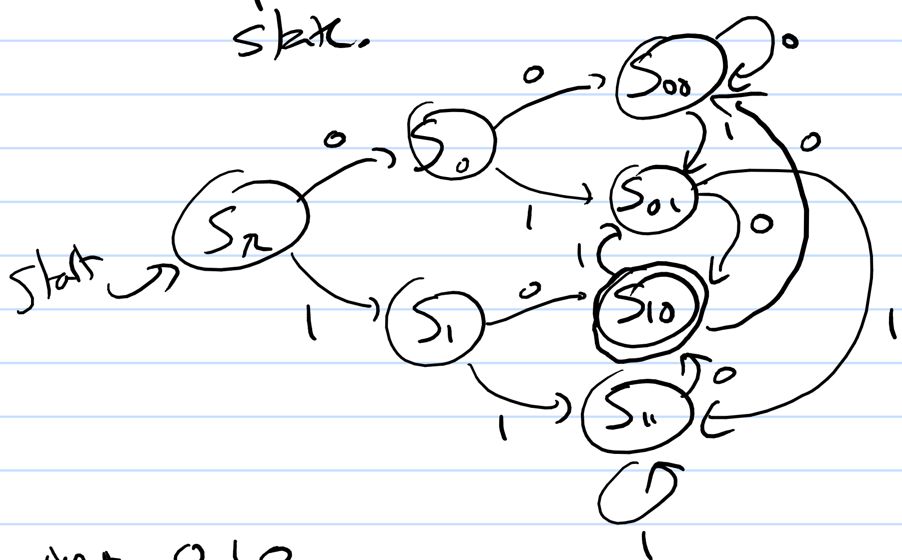
$FSA = (S, I, f, s_0, F \subseteq S)$

input alphabet

set of final states



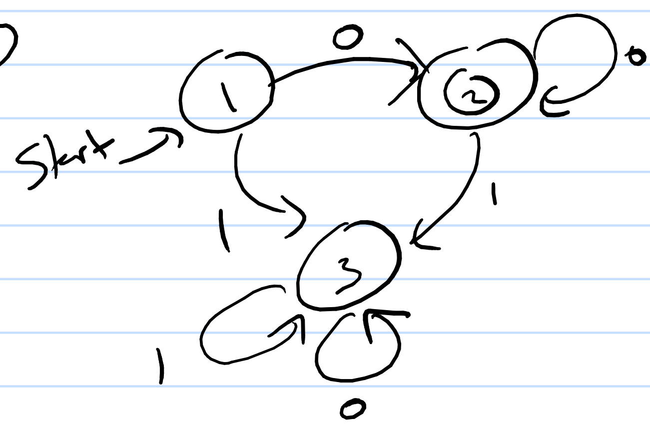
Language of a FSA is the set of all strings of inputs that take the start state to a final state.



$L(M) = (0|1)^* 10$

(ex) input 010
input 000010

(ex)



L(M)

list final states

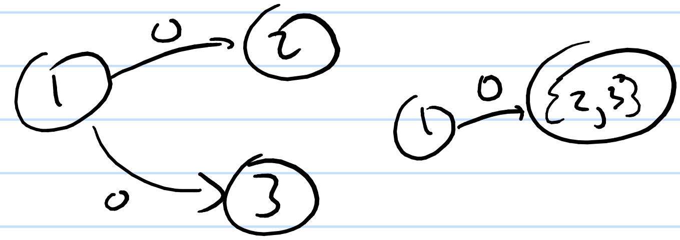
$s_2 : 00^*$

Deterministic FSA

$f: S \times I \to S$

is a function

Non-Deterministic FSA



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