Math 322

13.1 Phrase-Structure Grammars make languages $L(G)$

Machines: (new **goal**: take a string from T^* and it will "recognize" if a given string is of a language)

Finite State Machines — **ver #1** with output; **ver #2** without output

with output.

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

represent "knowledge" of events to machine.

S is a non-empty finite set of states and S_0 is the start state.

I input alphabet

O output alphabet

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

$g : S \times I \to O$ output function

representing M

State table

States	f: inputs	g: inputs
S_0	next state	output
S_1		
$\vdots$		
S_n		

ex's 2¢ candy machine

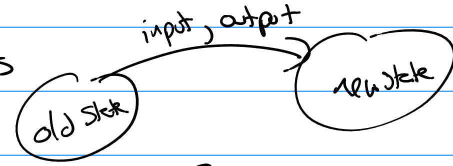
$I = \{1¢,\ \text{return} = r,\ \text{candy} = c\}$

$O = \{n,\ 1¢,\ 2¢,\ \text{candy} = c\}$

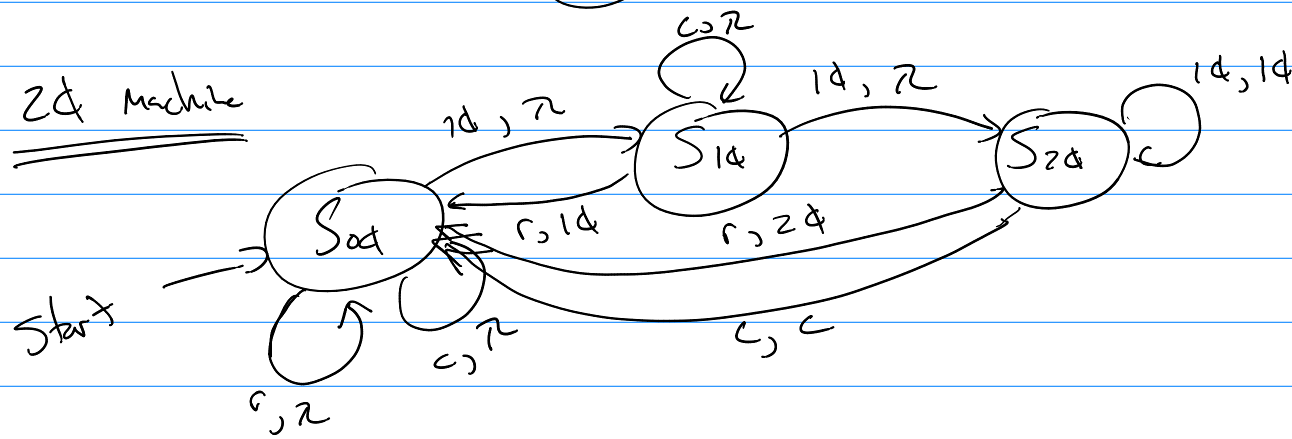
	f			g		
States	1¢	r	c	1¢	r	c
$S_{0¢}$	$S_{1¢}$	$S_{0¢}$	$S_{0¢}$	n	n	n
$S_{1¢}$	$S_{2¢}$	$S_{0¢}$	$S_{1¢}$	n	1¢	n
$S_{2¢}$	$S_{2¢}$	$S_{0¢}$	$S_{0¢}$	1¢	2¢	C

State diagram: (1) (digraph, except vertices are name)

(2) edge sets

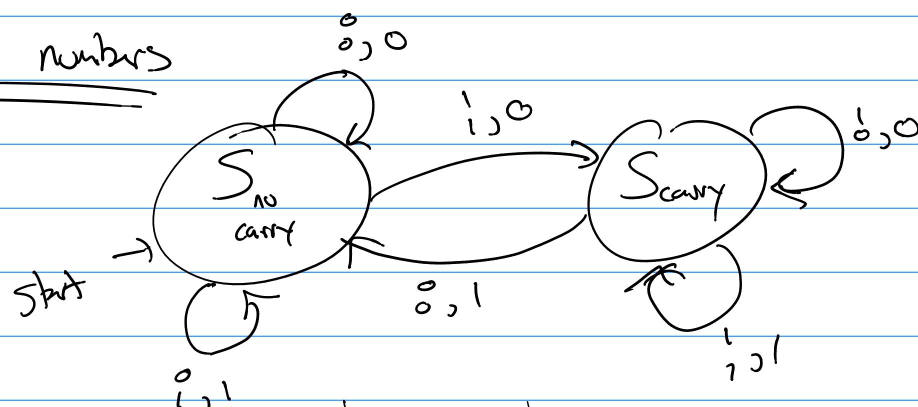


ex 25¢ Machine



ex add binary numbers

$$\begin{array}{r} {}^{1\,1}\;\; \\ 011010 \\ +\,001100 \\ \hline 100110 \end{array}$$



inputs $\frac{0}{0}, \frac{1}{0} = \frac{0}{1}, \frac{1}{1}$

outputs 0, 1

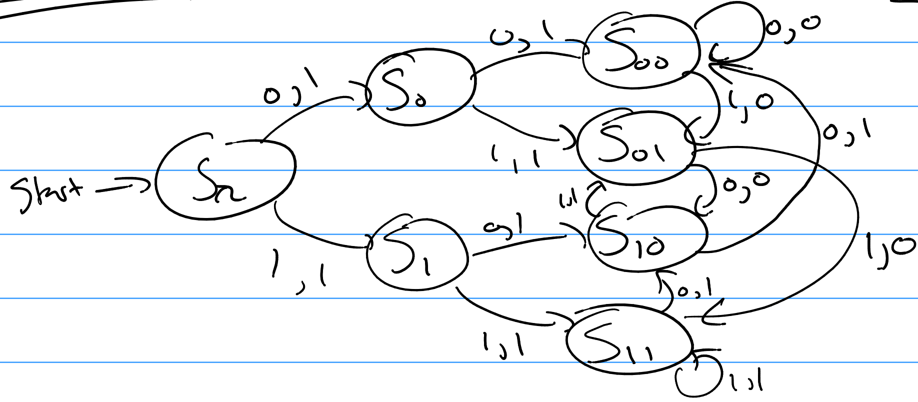
table

	$\frac{0}{0}$	$\frac{1}{0}$	$\frac{1}{1}$	$\frac{0}{0}$	$\frac{1}{0}$	$\frac{1}{1}$
S_{no}	S_{no}	S_{no}	S_{yes}	0	1	0
S_{yes}	S_{no}	S_{yes}	S_{yes}	1	0	1

2-bit delay (by 11 (Input String))

ex 1011001....

11 1011001...



Language and Finite-State Machines with output.

- M is said to recognize an input string when your machine outputs a 1 as the last output for the given input string.

- $L(M)$ is set of all input strings that M recognizes.

13.3 Finite State Machines with no output

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

$S \equiv$ States

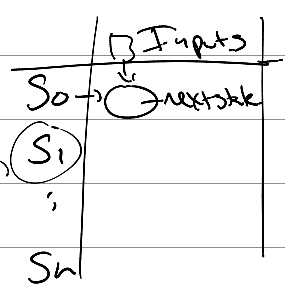
$I \equiv$ input symbols

$f : S \times I \to S$

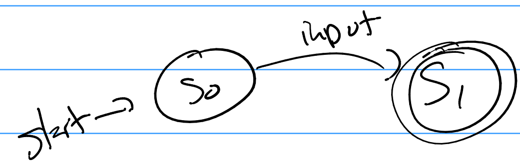
$s_0 \in S$ is the start state

$F \subseteq S$ is the set of final states

State table:



maybe circle final states in F

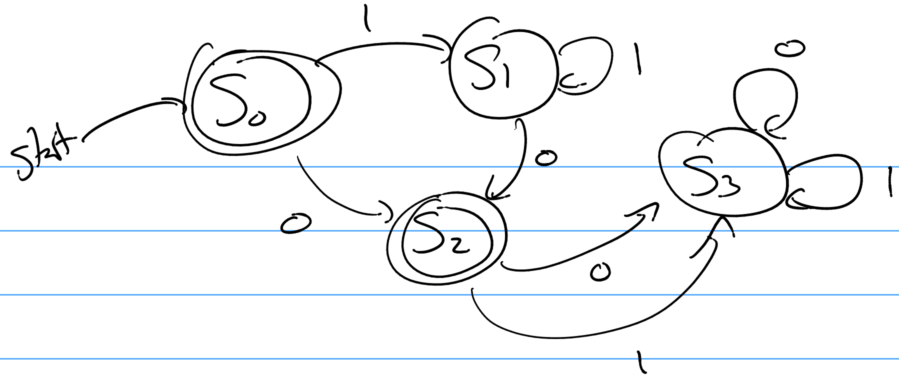


Main purpose of finite state machines without output is language recognition.

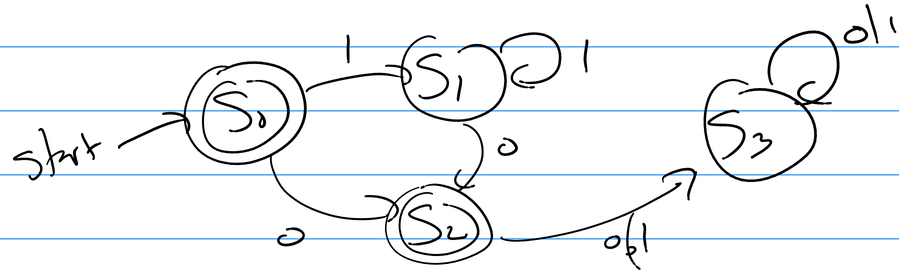
→ call them finite state automatons (FSA)

$L(M)$ is set of all input strings that have M end in a final state.

ex:



Simplified version



$$L(M) = \lambda,\ 0,\ 10,\ 110,\ 1110,\ 11110,\ \ldots$$

$$1^0 0 \quad 1^1 0 \quad 1^2 0 \quad 1^3 0 \quad 1^4 0$$

Set of strings

How to write sets of strings based on concatenation and union (or)

Def: A, B are sets. AB is the set of all strings ab (string concatenation) of $a \in A \wedge b \in B$

ex $A = \{0, 00\}$ $B = \{1, 10, 01\}$

$$AB = \{01, 010, 001, 001, 0010, 0001\}$$

$$= \{01, 010, 001, 0010, 0001\}$$

$$= \{0, 00\}\{1, 10, 01\}$$

Def: $A^0 = \{\lambda\}$

$A^{n+1} = A^n A$

ex all bit strings $= \{0,1\}^0$ or $\{0,1\}^1$ or $\{0,1\}^2$ or ...

Def: **Kleene closure** $A^* = \bigcup_{k=0}^{\infty} A^k$

Now show: $L(M) = \lambda, 0, 10, 110, 1110, 11110, \ldots$

$\underbrace{1^0 0 \quad 1^1 0 \quad 1^2 0 \quad 1^3 0 \quad 1^4 0}$

$\underbrace{\{1^0, 1^1, 1^2, 1^3, \ldots\} 0}$

$1^* 0$

$$\boxed{L(M) = \lambda, 1^* 0}$$