FSM Examples

- automatic door 1-way

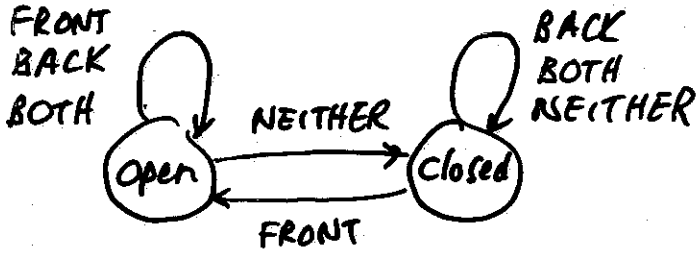
2 states: open closed

input: FRONT (somebody at the front)

BACK (someone at back)

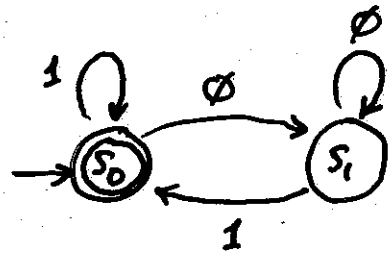
NEITHER (no one at front or back)

BOTH (someone at front + someone at back)



state transitions

-



S_0 – start state →○

accept states ◎

What "language" does this machine recognize?

Σ - set of alphabets

accept

ØØ end up in ● S_1 (not a final state)

∴ not a valid sentence

1Ø ends up in S_1 ∴ not valid

Ø1 " " " S_0 ∴ valid

accepts sentences that end in 1 (plus empty sentence)

λ - empty string

- write a function that squeezes spaces

eg. input: 1␣␣␣23␣4

output: 1␣23␣4

state change function ν | output function ω

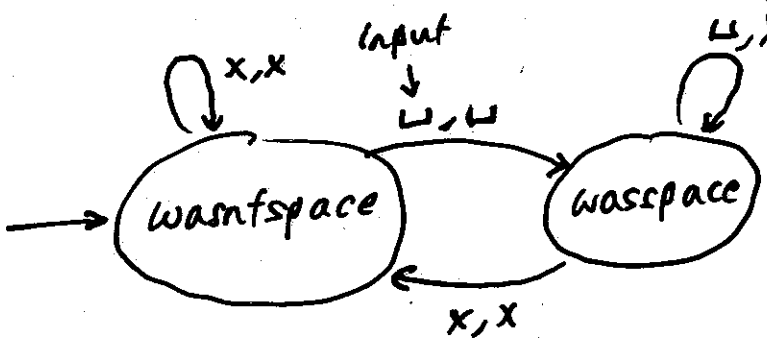
input: nonspace space

states: wasspace wasntspace

states \ input	nonspace	space
wasspace	① print	② print nothing
wasntspace	③ print	④ print

1␣␣␣23␣4

④ ② ① ⑤



empty string

␣,λ ␣123

␣ space

x - any character that is not a space

ω(wasspace, a) = a

a: character a

$\omega : S \times \Sigma \longrightarrow O$

S: set of internal states; Σ: input; O: output

$\nu : S \times \Sigma \longrightarrow S$

ν(wasspace, a) = wasntspace

$M =$

A finite state machine is a 5-tuple $(S, \mathcal{I}, \mathcal{O}, \nu, \omega)$

where S = set of internal states

$\mathcal{I}$ = input alphabet

$\mathcal{O}$ = output alphabet

ν = next state function, $\nu: S \times \mathcal{I} \longrightarrow S$

ω = output function, $\omega: S \times \mathcal{I} \to \mathcal{O}$

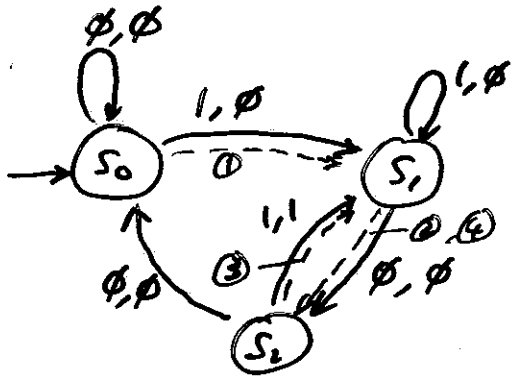
eg. p. 321 ex. 6.17

	ν		ω	
	0	1	0	1
s_0	s_0	s_1	0	0
s_1	s_2	s_1	0	0
s_2	s_0	s_2	0	1

s_0 – starting state

What's the output when the input is 1ø1ø?

øø1ø



eg. p. 323 ex. 6.19

carry

2 states: s_0 & s_1

carry = ø carry = 1

$x =$ 0 0 1 1 1

$+ \quad y =$ 0 1 1 0 1

1 0 1 0 0

1 1 1 1 1

input: øø øl lø ll

both digits ø

10, 1
00, 0
01, 1

01, 0
10, 0
11, 1

11, 0

ø

00, 1

	ν				ω			
	00	01	10	11	00	01	10	11
s_0	s_0	s_0	s_0	s_1	0	1	1	0
s_1	s_0	s_1	s_1	s_1	1	0	0	1

p. 324 – 325 # 3, 5, 6