Math 322

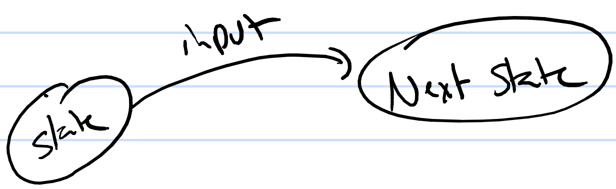
Foundations: ch 3 - 5

HW

3.1 (1acd)
3.2 (1ad, 2ac)
3.3 (nothing)
3.4 (1ab, 2ab)

3.5 (1, 2)
4.1 (3ac)
4.2 - 4.5 (nothing)
4.6 (1)

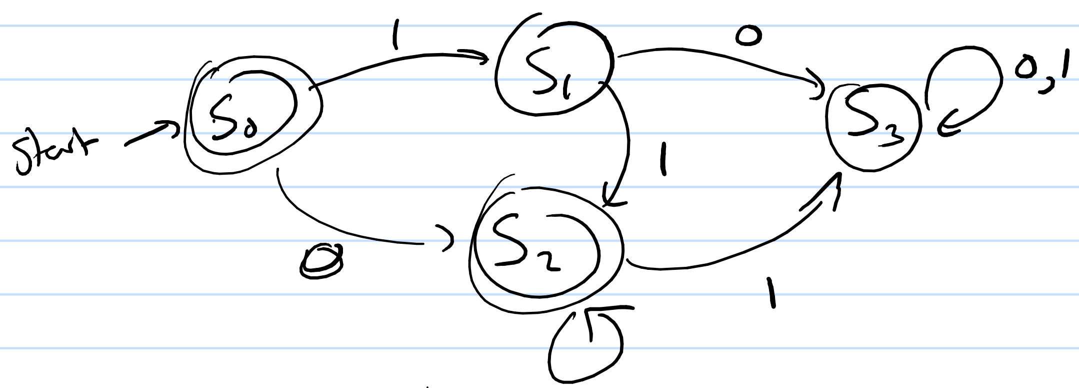
F.S.A.



inputs = $\{0,1\}$

transition function $S \times A \to S$

(ex)

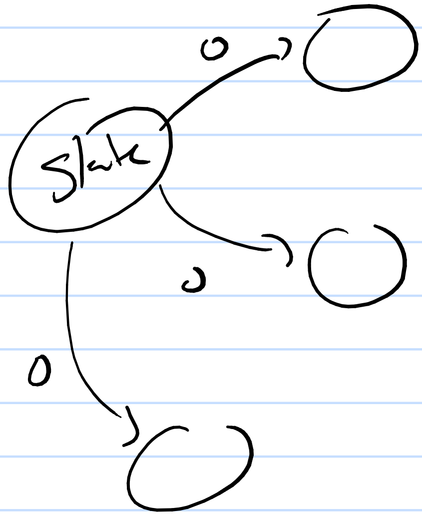


$L(M)$:

final states	strings to final state
S_0	ε
S_2	$(0\|11)0^*$

$L(M)$ is ε, $(0|11)0^*$

Nondet.



$(S \times A) \to S$

Not a function

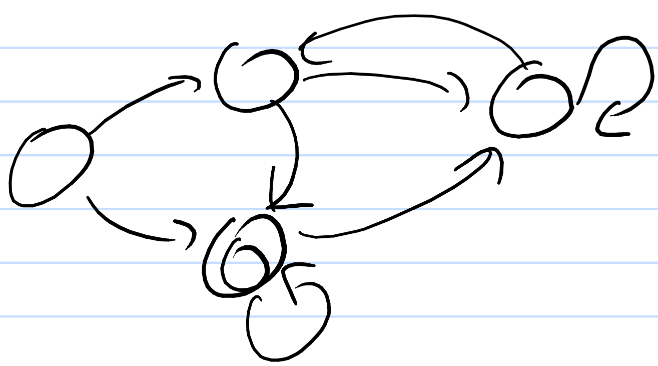
(state, 0) → goes to 3 places

For Non-Determinism we modify transition function to $(S \times A) \to P(S)$

Facts: ① if any state of $\boxed{\text{element of } P(S)}$ (subset of S) is final → you are @ final.

② $|S| = n \qquad |P(S)| = 2^n$

ex

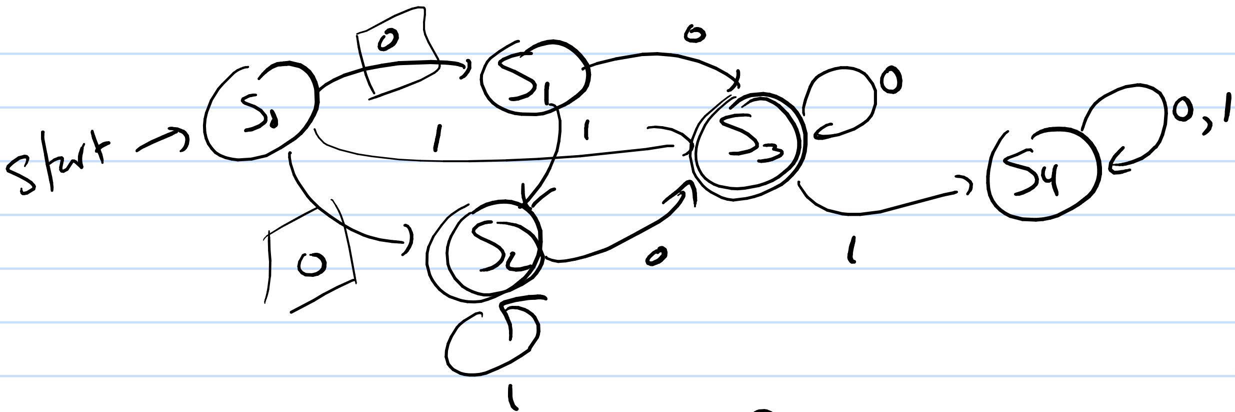


$|S| = 4$

$|P(S)| = 2^4 = 16$

Thm for any NFA and its language L(NFA) there exists a DFA such that $L(DFA) = L(NFA)$

(4)



L(NFA) : final states — strings to final states

S_2 — $(0|01)1^*$

S_3 — $(1|00|(0|01)1^*0)0^*$

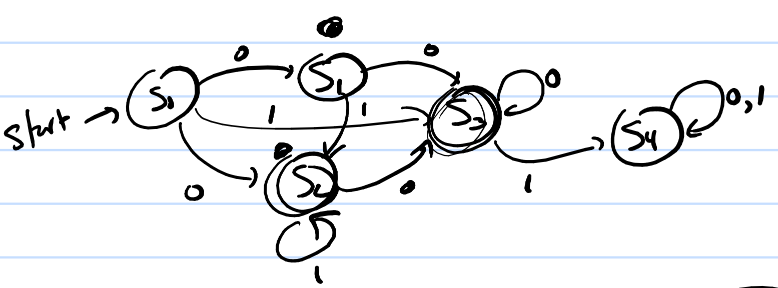
Make a DFA for L(NFA) is use $\underline{\underline{P(S)}}$

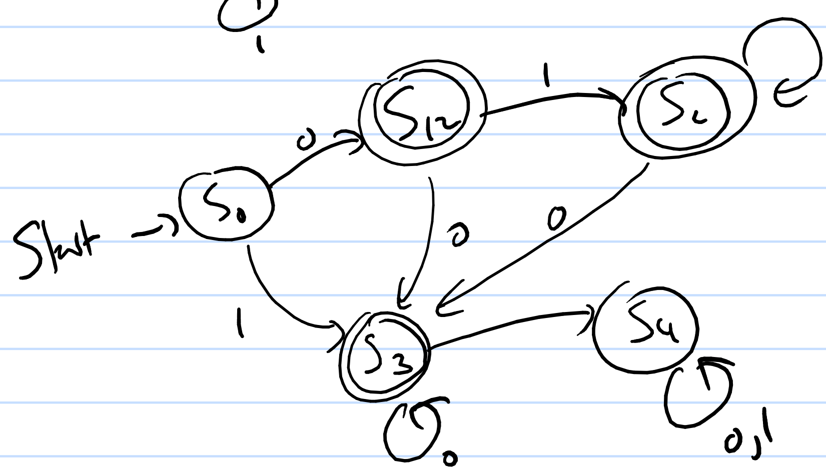
S: S_0, S_1, S_2, S_3, S_4

$P(S)$: $S_\emptyset$, S_0, S_1, S_2, S_3, S_4, S_{01}, S_{02}, S_{03}, S_{04}, ...

S_{012}, S_{013}, ... , S_{01234}

@ $\{S_0, S_1, S_2, S_3 \wedge S_4\}$





$L(M) = 0, (1|00|01^*0)0^*, 011^*$

Th^n a language is generated by a regular expression

(1) iff it is generated by a regular grammar

(2) iff it is recognized by a <u>F.S.A.</u>

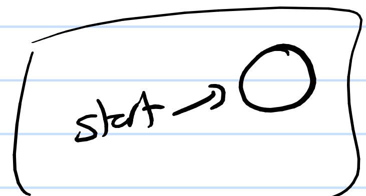
<u>Basis</u> of $\phi, \{\epsilon\}, \{a\}$

<u>Induction:</u> $s_1 s_2$, $s_1 | s_2$, s_1^*

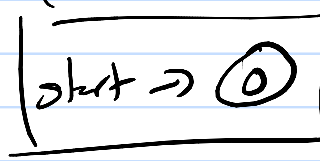
pf regular expressions. **NEED** F.S.A.

<u>Basis</u>

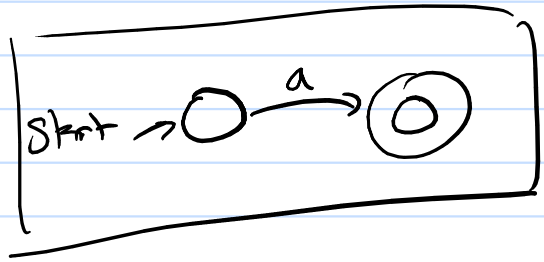
(1) ϕ is regular FSA is



(2) $\{\epsilon\}$ is regular FSA is

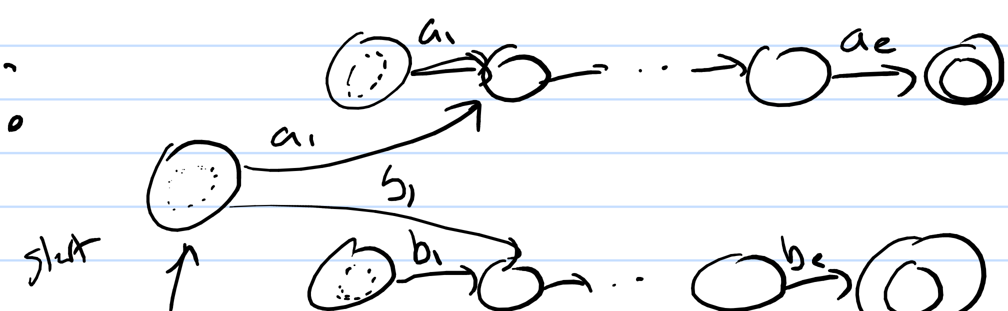


(3) $\{a\}$ is regular FSA is



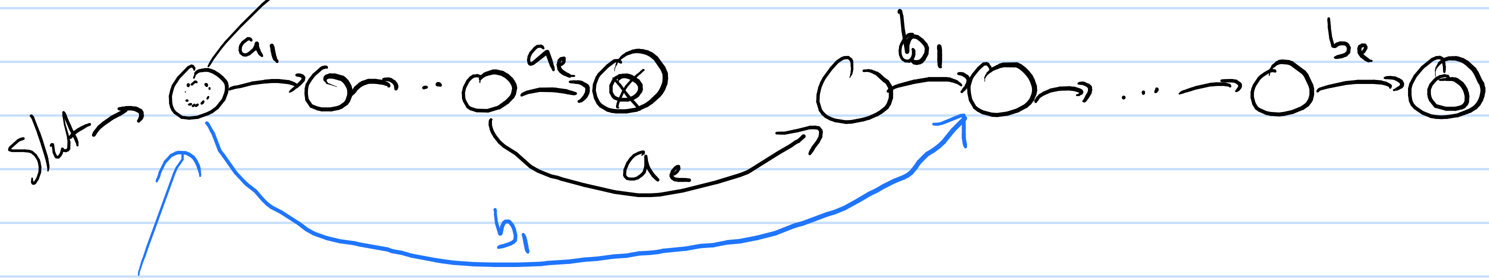
<u>Inductive:</u> Assume M_A, M_B are regular expressions A, B

(1) $M_{A|B}$:



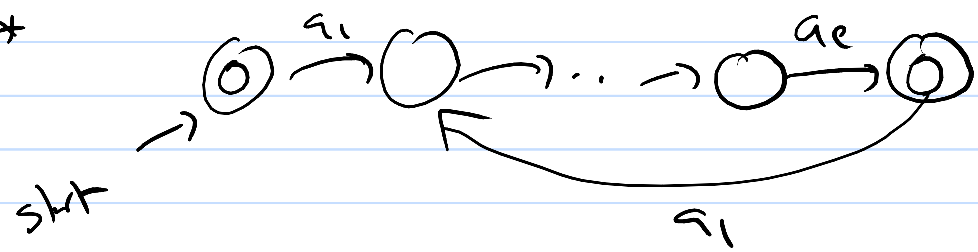
make final if either start state of M_A, M_B was final.

2 M_{AB} Keeps final only if $\boxed{\text{both}}$ M_A, M_B starts are final



if final? jump past all M_A

3 M_{A^*}



regular grammar?

Productions:

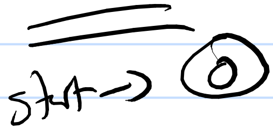
$S \to \varepsilon$

$A \to a$

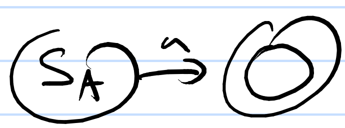
$A \to bB$

Proof.

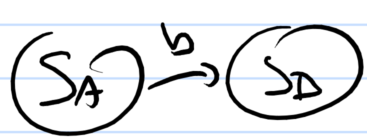
$S \to \varepsilon$



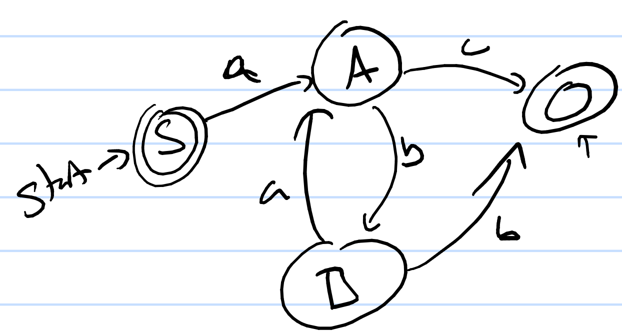
$A \to a$



$A \to bB$



DSA



4 $P = \{S \to \varepsilon,\ S \to aA,\ A \to bB,\ B \to aA,$
$A \to c,\ B \to b\}$

So what about context free grammars?

→ iff FSA with pushdown memory

---

context sensitive grammar

iff FSA with finite linear bounded memory.

---

any phrase structure grammar

iff FSA with infinite tape memory

$\boxed{\text{Turing Machine}}$

$$T = (Q, \Lambda, q_0, \delta)$$

$Q \equiv$ set of states (finite and has a special state, h, which means halt)

$\Lambda \equiv$ alphabet (has $\# =$ blank symbol)

$q_0 \equiv$ start state

$\delta \equiv$ transition function

$$\boxed{(Q \setminus \{h\}) \times \Lambda} \xrightarrow{\delta} \boxed{\Lambda, \{L, R\}, Q}$$

input pair is state, symbol — output triple symbol, move, state

set of 5-tuples

ex $(q_0, 1, 0, R, q_1)$