Review for Midterm 1

Thursday, 25 February, 2021 10:53

- Strings, languages, ops on languages

  Σ finite alphabet

  string
  - ε is a string
  - ax $a \in \Sigma$ x is a string

  $L \cup L'$, $L \cap L'$, $L \setminus L'$ ... { languages are sets of strings }

  $L \circ L'$, L^* specific to languages.

- induction (esp on strings) incl proving about languages (HW0!)

- regular languages via recursive def

  reg lang
  - $\emptyset$ is reg
  - $\{w\}$ (w is a string) is reg ($\{\varepsilon\}$ is reg, $\{a\}$ ($a \in \Sigma$) is reg)
  - $L \circ L'$ (L, L' reg) is reg
  - $L \cup L'$ (L, L' reg) is reg
  - L^* (L is reg) is reg.

  (HW2 P2(b) went through this)

  e.g. Σ^* is a regular language

- reg expressions (notation)

reg lang	reg ex
$\emptyset$	$\emptyset$
$\{w\}$	w
$L \circ L'$	$r \cdot r'$
$L \cup L'$	$r + r'$
L^*	r^*

(done a lot in lab, HW)

- DFAs
  - pictures & notation
  - δ vs. δ^* ([illegible])

- δ vs. δ^* (HW1 P3)

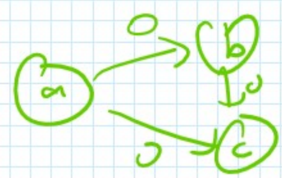
δ: $Q \times \Sigma \to Q$
state, char → state

δ^*: $Q \times \Sigma^* \to Q$
state, string → state

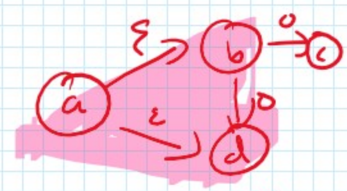
- DFA ⟷ english desc of langs (a lot in HW/lab)
- product construction (entire lab on this)
  - ↳ prove closure properties via product construction! (HW 1 P3)

— NFAs
- pictures & notation
- δ vs δ^* (ε reach)

δ: $Q \times \Sigma \to P(Q)$
$P(Q)$: set of states



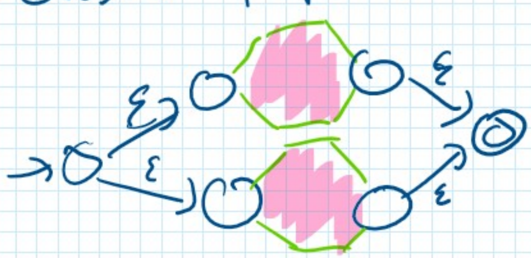
$\delta(a,0) = \{b,c\}$
$\delta(a,1) = \emptyset$



ε-reach(a)

- (incremental) power set/subset construction for NFAs → DFAs (HW2 P1, lab)
  k-states → $\leq 2^k$-states

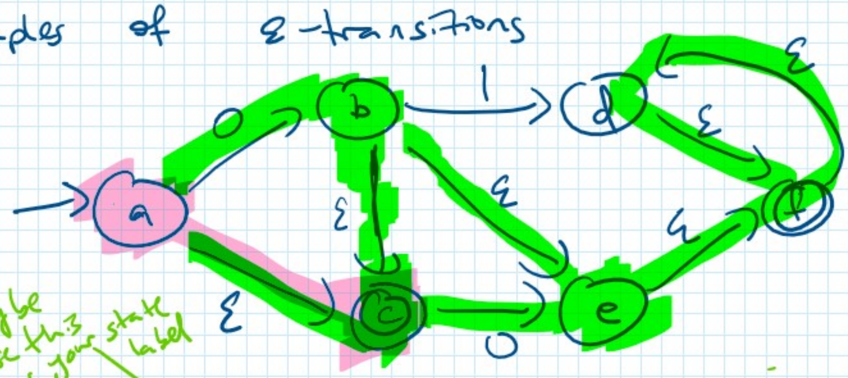
- Closure properties of NFAs aka Thompson's alg for regex → NFAs



union (lab)

Kleene star
concat

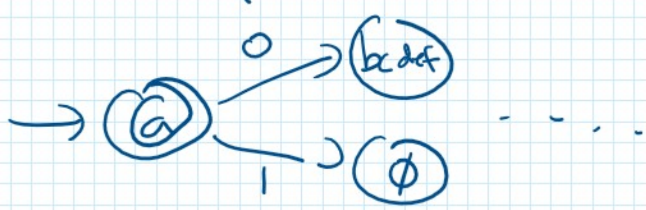
examples of ε-transitions

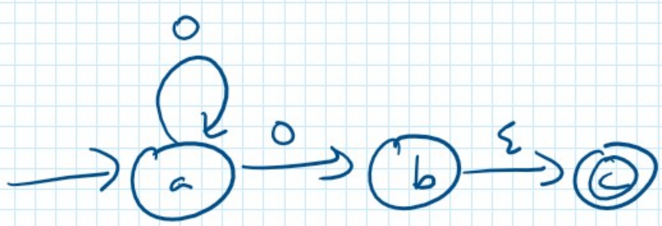


maybe use this as your state label

$\varepsilon\text{reach}(q') \cap A \neq \emptyset$

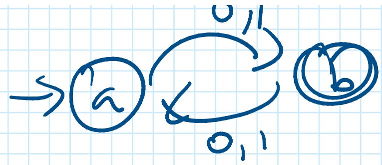
q'	ε reach	$\delta'(q',0)$	$\delta'(q',1)$	Acc?
a	ac	bcdef	∅	✓
bcdef	bcdef	def	df	✓
⋮	⋮	⋮	⋮	⋮





q'	ε reach	$\delta'(q',0)$	$\delta'(q',1)$	Acc?
a	a	abc	∅	✗
abc	abc	abc	∅	✓
∅	∅	∅	∅	✗

easiest entry



a	a	b	b	X
b	b	a	a	✓

— (advanced) closure properties of reg langs

in Patrick's opinion 2nd hardest topic for most students

- given a DFA for L
  construct NFA for $f(L)$ — HW2 P2, Hw3 P2

  prefix, cycle, flipodds, insert1, delete1, psuffix

- given regex for L
  construct (recursively) regex for $f(L)$ — HW2 P2

(see also lab)

— fooling sets

x, y distinguishable if

$\exists z \quad xz \in L \;\&\; yz \notin L$

OR $xz \notin L \;\&\; yz \in L$.

- fooling set F is a set of strings
  each pair $x, y \in F$ is distinguishable

- if x, y are distinguishable

$$\delta^*(s,x) \neq \delta^*(s,y)$$ (HW3P3)

starting state

starting state $\delta^*(s,x) \neq \delta^*(s,y)$ (HW3P5)

for any DFA for L

Fooling set $F \Rightarrow |Q| \geq |F|$.

- if $|F| = \infty$ <u>no DFA</u> (HW3P1a)

  (non-reg $\Rightarrow \exists$ infinite fooling set (but also finite ones))

- combining w/ closure properties (HW3P1b)

L, L' reg $\Rightarrow L \cup L'$ reg

contrapositive.

$L \cup L'$ not reg $\Rightarrow$ (L not reg) OR (L' not reg)

Q: - how to come up w/ fooling set?

- (one can prove) at most one string in F can be a non-prefix
- <u>control</u> the elements of F to make choosing z easier.

e.g. some pattern 1^n shows up twice

$F = \{ 1^n 0 \} \rightarrow$ control xz & yz:

xz 1^n shows up twice $\rightarrow xz \in L$

yz shows up once $\rightarrow yz \notin L$

Q: - how to come up w/ distinguishing suffix z?

- stare at x, z see how they differ

  case work! (wlog).

[Fooling set guide]

Folder 2.21: $0^n w 1^n$ for $n > 1$

$= 0 w' 1$ where $w' = 0^{n-1} w 1^{n-1} \in \Sigma^*$.

so $\{0^n w 1^n \mid n \geq 1 \;\; w \in \Sigma^+\}$

$= \{0 w' 1 \mid w' \in \Sigma^+\}$

$= L(0(0+1)(0+1)^*1)$.