Kleene's Theorem: where multiple models meet

1/25/21 10:30 AM

Administrivia.

- Remember to upload your worksheets!
- HW2 is out. due Friday (1/29)
- Midterm 1 on the second week of Feb. (maybe 2/9 Tue?)

Nondeterministic Finite Automata (NFA).

definitions are not sacred.

- Q.
- S multiple starting states
- A multiple accepting states
- Σ_ε w/ ε-transition
- $\delta : 2^Q \times \Sigma_\varepsilon \to 2^Q$.

$$\delta^*(P, w) := \begin{cases} \varepsilon\text{-Reach}(P) & \text{if } w = \varepsilon \\ \delta^*(\delta(\varepsilon\text{-Reach}(P), a), x) & \text{if } w = ax \end{cases}$$

Looks deterministic to me... if we record all fingers.

Thm. For any NFA N, there's a DFA M accepting the same language.

$q \xrightarrow{a} q'$ if $q \overset{\varepsilon}{\leadsto} q'' \xrightarrow{a} q'$

Pf. 1. Construct NFA N' w/o ε-transitions.

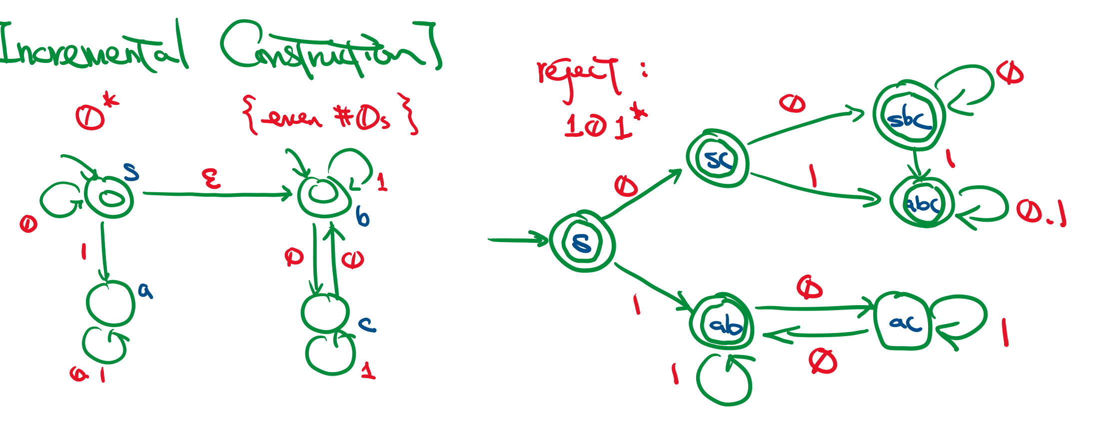
- Add $q \xrightarrow{a} q'$ if $\exists\, q \overset{\varepsilon}{\leadsto} q'' \xrightarrow{a} q'$ i.e. $q \in \varepsilon\text{-Reach}^{-1}(q'')$ for some $\delta(q'', a) = q'$
- $A_{N'} := \varepsilon\text{-Reach}^{-1}(A_N)$ $\quad \circ \overset{\varepsilon}{\leadsto} ⊚$

2. Construct DFA M emulating NFA N':

- $Q_M := 2^{Q_{N'}}$
- $S_M := S_{N'}$
- $A_M := \{P \in Q_M = 2^{Q_{N'}} : P \cap A_{N'} \neq \emptyset\}$
- $\delta_M(P, a) := \delta_{N'}(P, a)$

↖ type: subset of $Q_{N'}$, element in $2^{Q_{N'}} = Q_M$

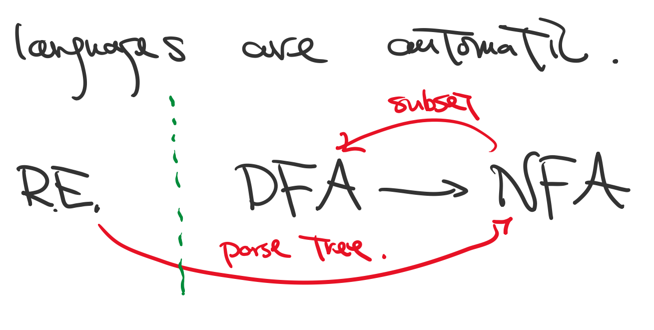
example. [Incremental Construction]



P	ε-Reach(P)	∈ A_M?	$\delta_M(P, 0)$	$\delta_M(P, 1)$
s	sb	✓	sc	ab
sc	sbc	✓	sbc	abc
ab	ab	✓	ac	ab
sbc	sbc	✓	sbc	abc
abc	abc	✓	abc	abc
ac	ac	✗	ab	ac

Cor. A language is automatic iff some NFA accepts it.

Cor. Regular languages are automatic.



Question. Are $O(1)$-memory programs ~~better~~ than no-memory ones?

No! [Kleene]

Kleene Thm. [1951]

Every automatic language is regular. i.e. every language accepted by some DFA has a reg. expression.

[Han-Wood '05]

Pf. (sketch) generalize NFAs even further.

$q \xrightarrow{R} q'$: take transition after reading $x \in R$.

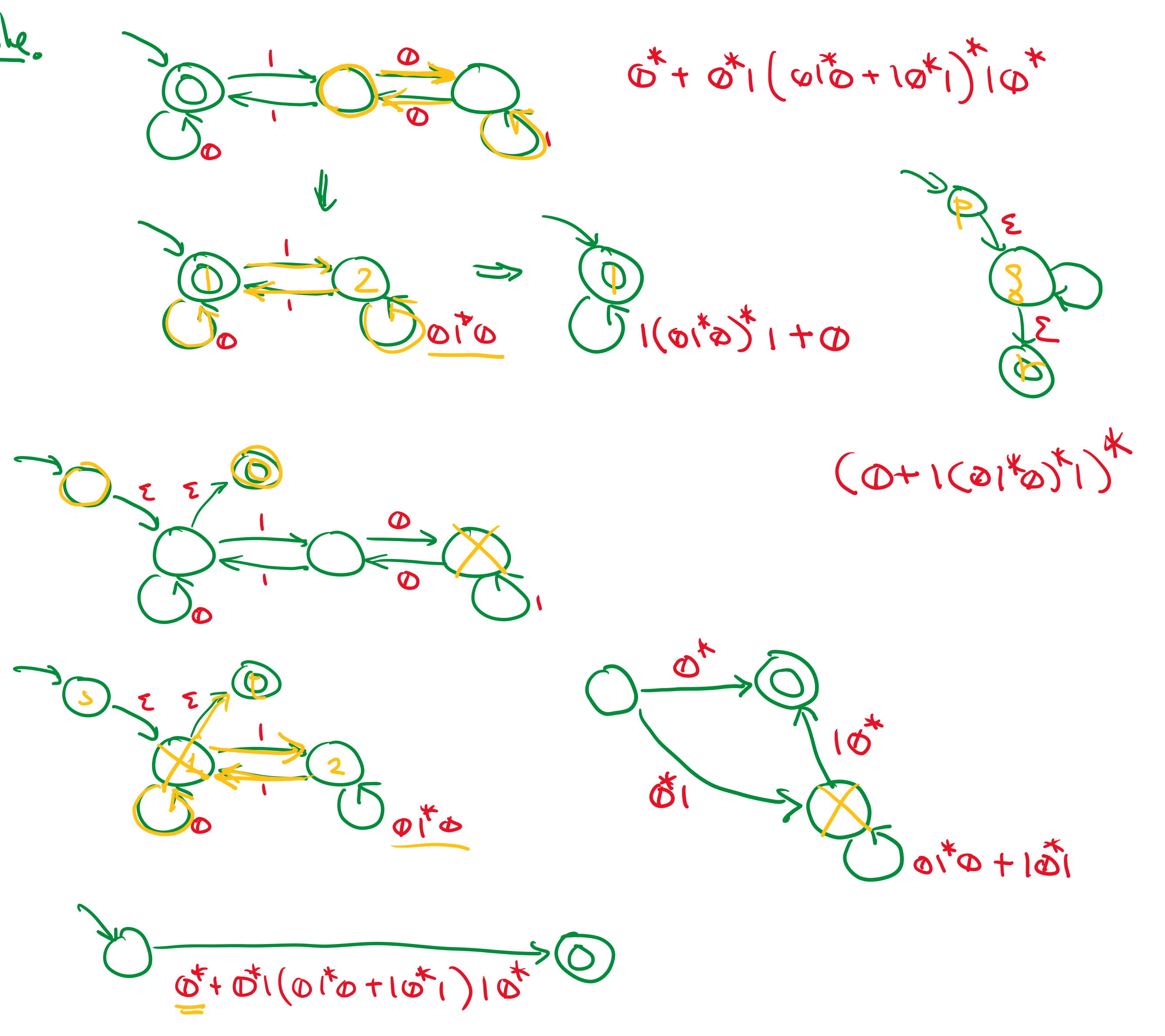
GNFA accepts w if $\exists\, s \xrightarrow{R_1} q_1 \xrightarrow{R_2} \cdots \xrightarrow{R_\ell} q_\ell$ (accepting), $w = x_1 \cdot x_2 \cdot \cdots \cdot x_\ell$, $x_i \in R_i$

(intuition: any decomposition of w matching any walk in GNFA)

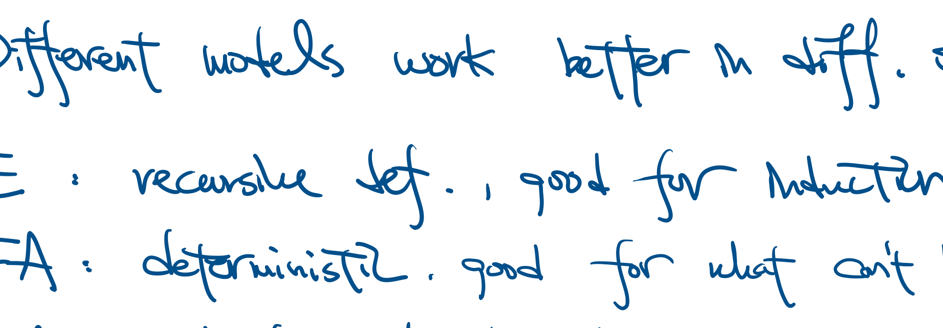
- Now, turn GNFA into RE.

To remove q:

example.



Cor. Regular languages can be modeled as:



Moral. Different models work better in diff. scenarios.

- RE : recursive def., good for induction.
- DFA : deterministic. good for what can't be done.
- NFA : good for algorithm design.
- GNFA : exist for the sake of reduction to RE. (middle-step object).

Concluding Question. DFAs (reg. lang.) are surprisingly powerful. What can't DFAs do?