CS 121: Lecture 9
Limits of Finite Automata

Adam Hesterberg

https://madhu.seas.Harvard.edu/courses/Fall2020

Book: https://introtcs.org

How to contact us
- The whole staff (faster response): CS 121 Piazza
- Only the course heads (slower): cs121.fall2020.course.heads@gmail.com

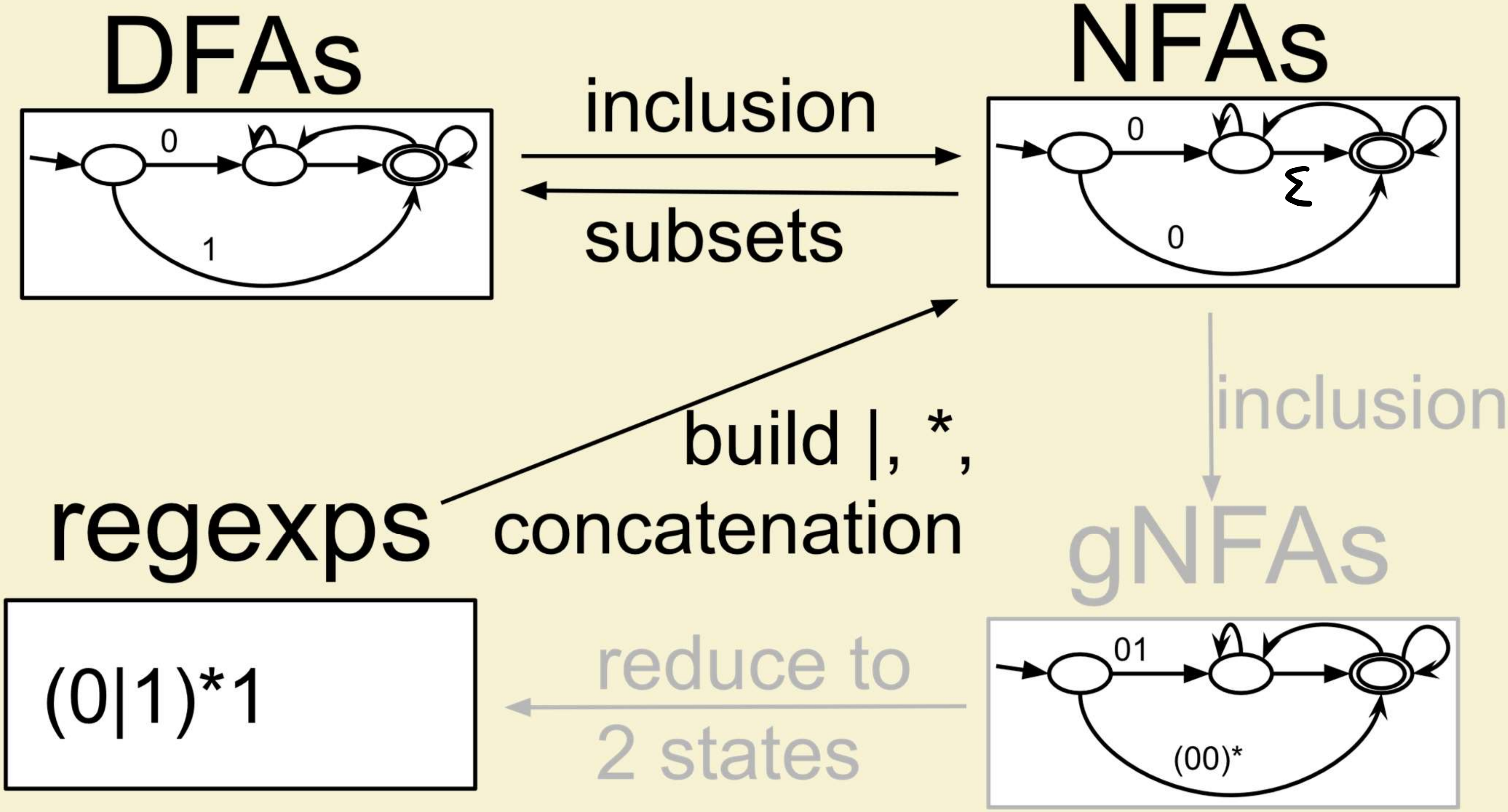
Reminders

- 121.5 at 4:30: Ben Edelman on Probably Approximately Correct learning
- Section 4 cycle begins today
- Problem set 2 due tonight (midnight ET)
- Problem set 3 out ~~tonight~~
- Midterm on 2020-10-13 (1.5 weeks), covering material through today

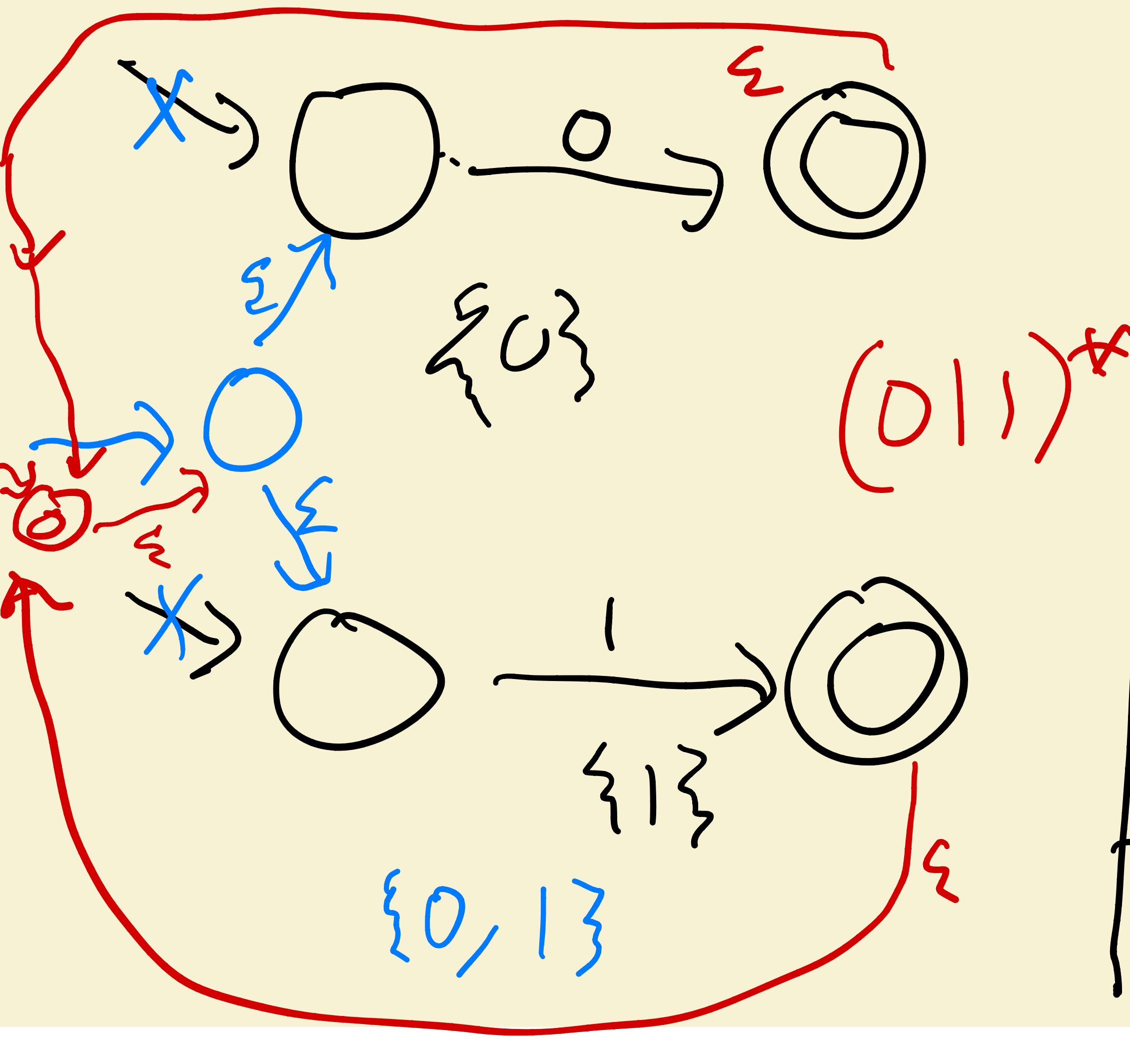
Today:

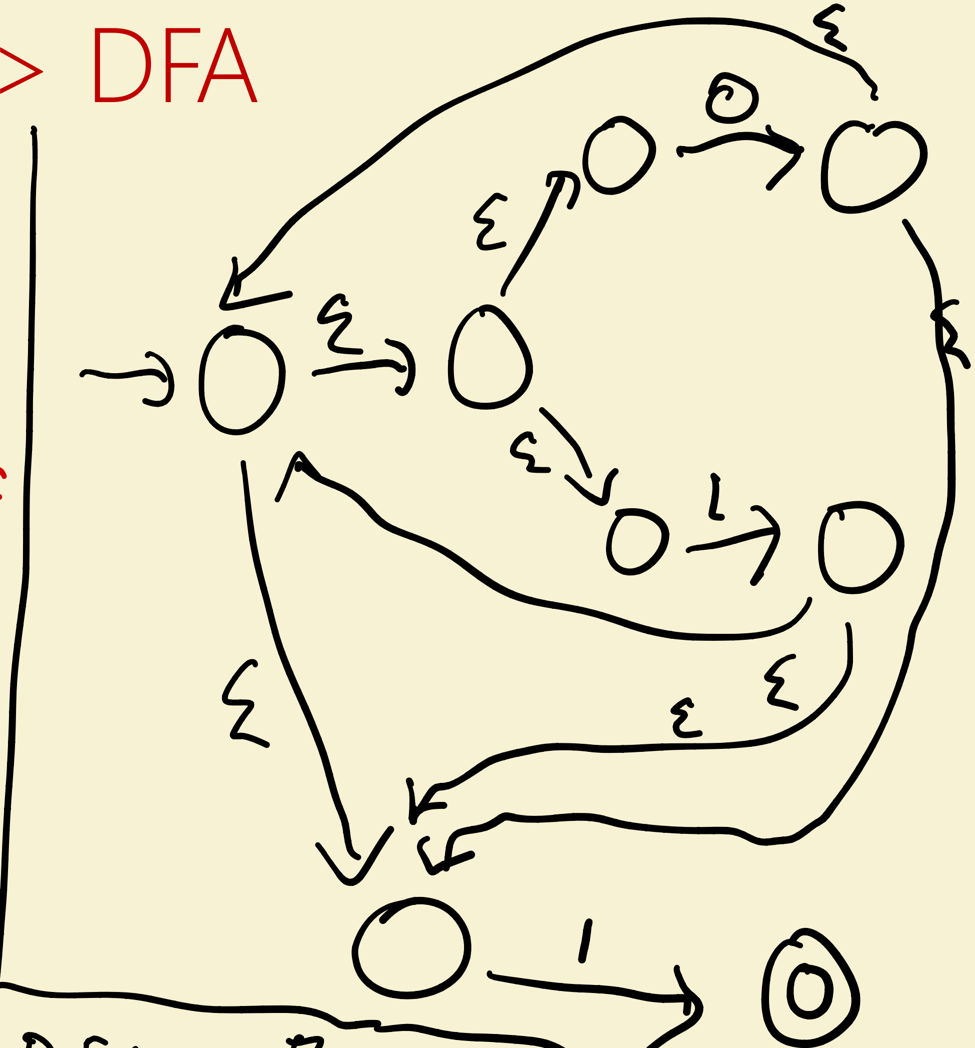
- Recap of DFA-regexp equivalence
- Break 1: convert a regular expression to a DFA
- Limits of DFA
    - All functions computed by DFAs take O(n) time.
    - Some functions are not computed by any DFA.
- Break 2: Regular or not?
- Summary of nonregularity: the “Pumping Lemma”

Equivalence of DFAs and regular expressions



Example: (0|1)*1 -> NFA -> DFA

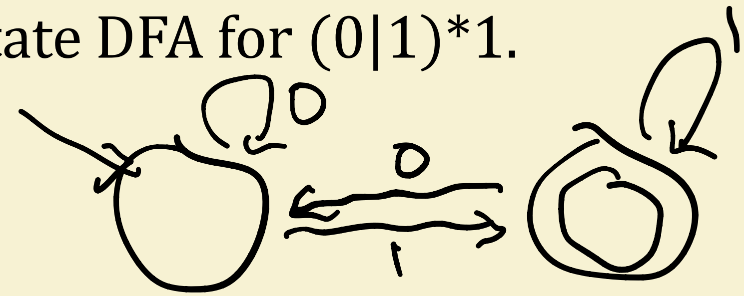




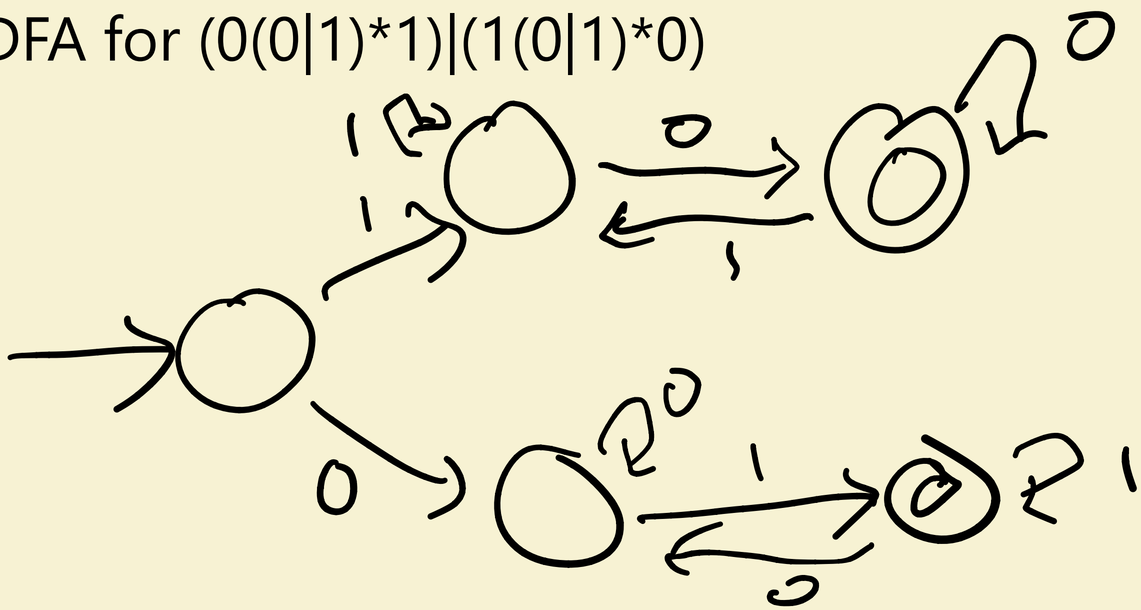
DFA: 2^8 states

Exercise Break 1:

1) Find a 2-state DFA for (0|1)*1.



2) Find a DFA for (0(0|1)*1)|(1(0|1)*0)



Theorem: Let e be a regular expression.
Then the function $\Phi_e : \{0,1\}^* \to \{0,1\}$ is computable.

Moreover $\exists$ algorithm computing $\Phi_e(x)$ for $x \in \{0,1\}^n$ in $O(n)$ time.

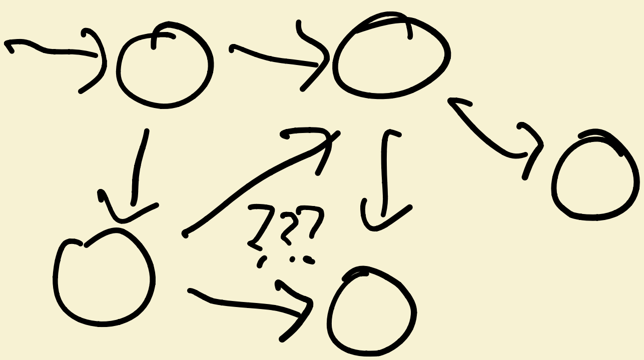
Linear in input size,
regardless of size of DFA

A non-regular language: $\{0^n 1 0^n\}$

Theorem: There's no DFA or regular expression that accepts exactly $\{0^n 1 0^n\} = \{1, 010, 00100, \dots\}$.

Proof 1:

Suppose, for contradiction, that there is such a DFA, say with q states…



Consider the input

$0^{q+1} 1 0^{q+1}$

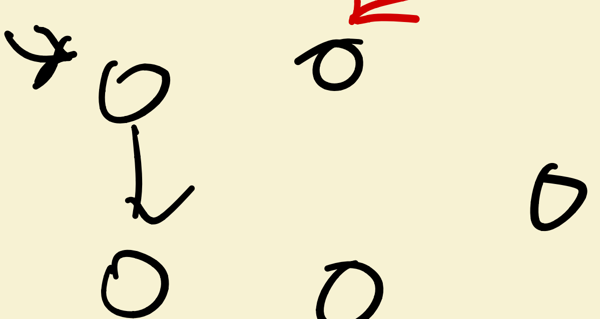
$\{0^n 1 0^n\}$

Define $f: [q+1] \to \{\text{states of DFA } D\}$

by $f(x) = $ state of DFA after x 0s

$|[q+1]| = q+1 > q = |\{\text{states of DFA}\}|$

By PHP, $\exists x \neq y : f(x) = f(y)$, i.e.



In same state after 0^x, 0^y.

From that state, $10^x \to$ accept.

So $0^y 1 0^x \to$ accept, contradiction.

A non-regular language: $\{0^n 1 0^n\}$

matches ≤4 bits matches ≤5 bits

Q: Let $e = (0000|111|0100)(020)^*(00111|11|00|11)$.

Prove that if $|x| > 100$ and $\Phi_e(x) = 1$ then x must contain the digit 2

Theorem: There's no DFA or regular expression that accepts exactly $\{0^n 1 0^n\} = \{1, 010, 00100, \dots\}$.

Proof 2:

Suppose, for contradiction, that there is such a regexp, say with q characters…

Proof 2 (that $\{0^n 1 0^n\}$ isn't regular):
Suppose for contradiction that $\exists$ regexp. with q characters matching $\{0^n 1 0^n\}$.
To match $0^{q+1} 1 0^{q+1}$, must use a star in first $q+1$ 0s, else too short.
Use same starred expression more: $\exists r > q$ s.t. regular expression matches $0^{r+1} 1 0^{q+1}$. $\cancel{R}$

Exercise Break 2:

For each of the following sets of strings, either describe a DFA or regular expression for it or prove that none exists.

1) Strings with the same number of 0s and 1s
2) Strings with the same number of 01s and 10s
3) Strings with at least 4 1s
4) Strings with at least ¼ 1s

Strings with the same number of 0's and 1's

If $\exists$ DFA D w/ q states, consider $(0^0), (0^1), \ldots, (0^q)$: Two of them leave D in the same state, say 0^x and 0^y. Then reading 1^x must leave us in accept for 0^x1^x and reject for 0^y1^x.

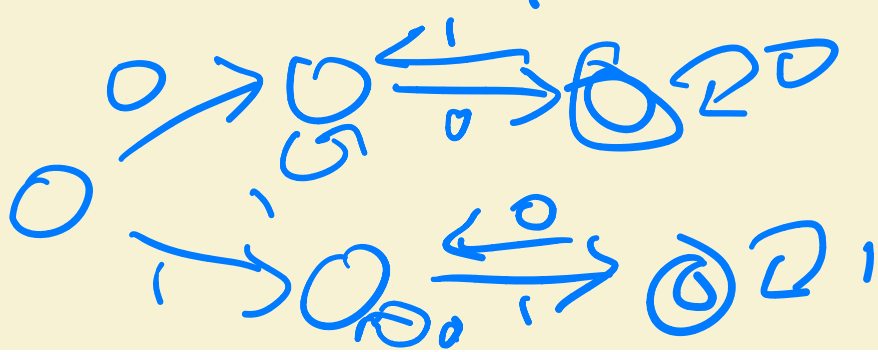
Strings with the same number of 01s and 10s

… is regular!

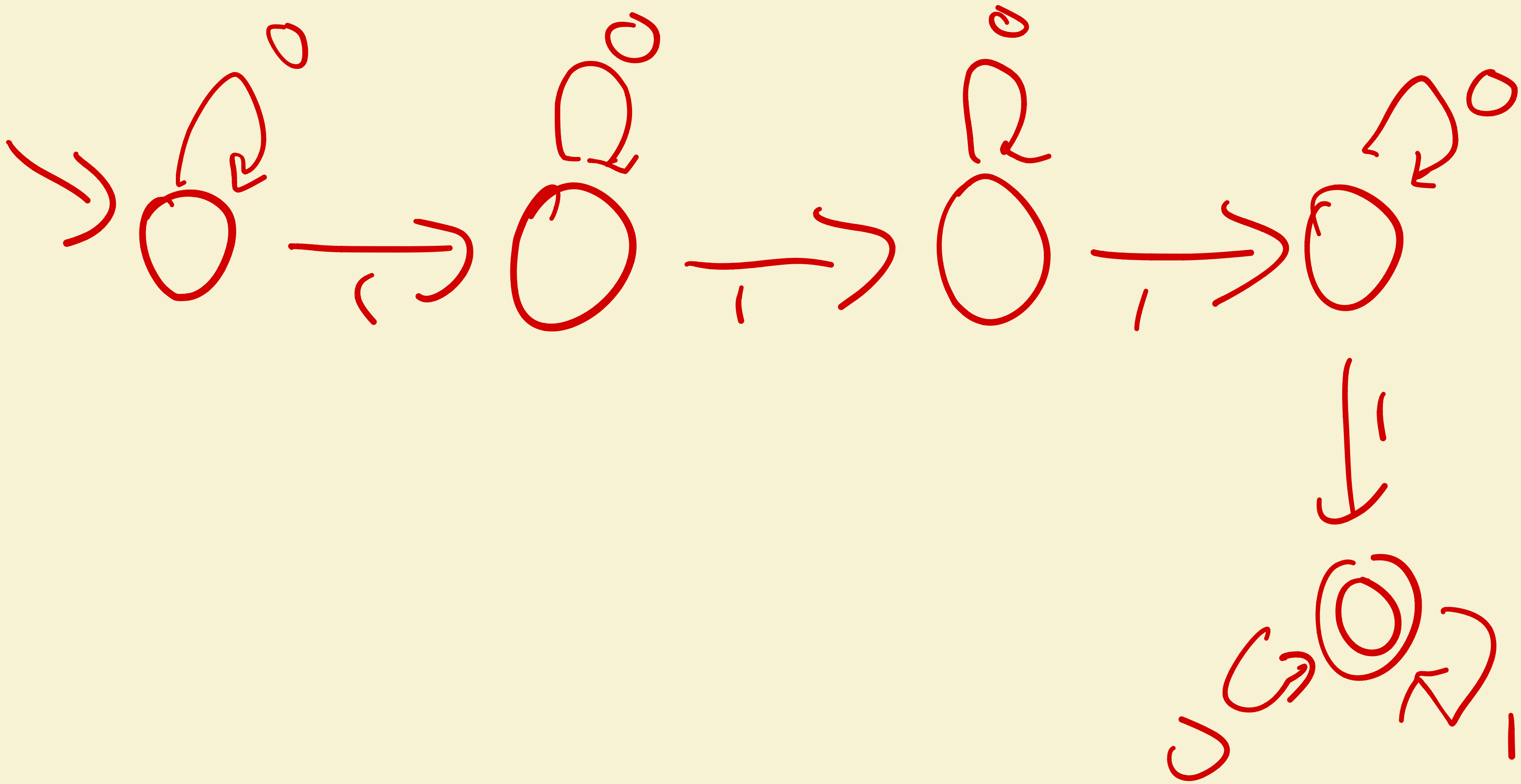
transition from 0s to 1s

transition back

strings that start & end w/same character.

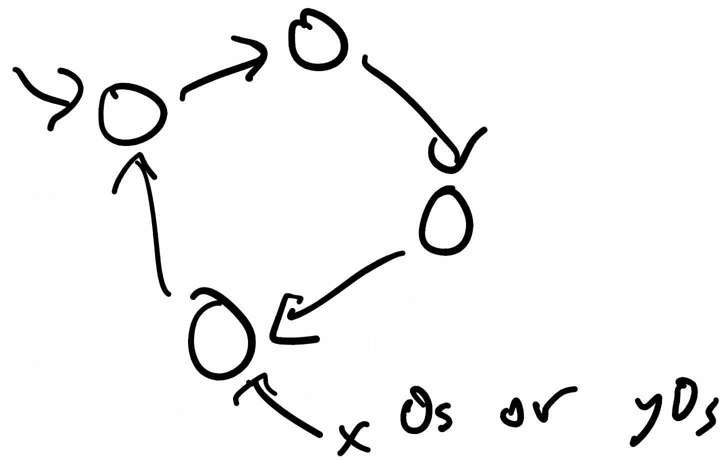


Strings with at least 4 1s



Strings with at least ¼ 1s

Pumping Lemma



Pumping Lemma: (Informal version). Let $F = \Phi_e$ for some e.
If $|w| > 2|e|$ and $\Phi_e(w) = 1$ then "we must use star" to match w.

Pumping Lemma: (formal version). Let $F = \Phi_e$ for some e and $n = 2|e|$.
If $|w| > n$ and $\Phi_e(w) = 1$ then $\exists x, y, z$ s.t. $w = xyz$, $|xy| \leq n$, $|y| \geq 1$ s.t.

$$\Phi_e(xy^k z) = 1$$

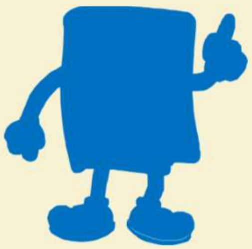
for every $k \in \mathbb{N}$

Proof:

Q: Let $F: \{0,1\}^* \to \{0,1\}$ defined such that $F(x) = 1$ iff $x = 0^n 1^n$ for $n \in \mathbb{N}$. Prove that F is not regular.

Blue Team: Student proving F is not regular

Red Team: Hypothetical "adversary" claiming F is regular



"F is computed by a regular expression exp"

"Is that so? Then what is the number whose existence is guaranteed by the pumping lemma?"

"Here is the number – you can call it n_0"

"In this case, let me choose $w = 0^{n_0} 1^{n_0}$. Notice that $F(w) = 1$. What is the partition $w = xyz$ from the pumping lemma?"

"Since $|xy| \leq n_0$ and $|y| \geq 1$, I guess I am forced to use $x = 0^a$, $y = 0^b$, $z = 0^{n_0 - a - b} 1^{n_0}$ for $b \geq 1$ and $a \leq n_0 - b$"

"In this case, since I can choose k as I want, let me set $k = 2$ and note that $xy^k z = 0^{n_0 + b} 1^{n_0}$ which contradicts the pumping lemma conclusion that $F(xy^k z) = 1$!"

Pumping Lemma: If exp computes F there exists n_0 such that for every w with $F(w) = 1$ and $|w| > n_0$ there exists partition $w = xyz$ with $|xy| \leq n_0$ and $|y| \geq 1$ such that for every $k \in \mathbb{N}$ it holds that $F(xy^k z) = 1$

Next lecture:

- Turing Machines
  - Like DFAs, can read and **write**, and can move right and **left** over input.
  - As powerful as any programming language.