18.404/6.840 Lecture 2

Last time:

- Finite automata, regular languages
- Regular operations
- Regular expressions
- Closure under

Today:

- Nondeterminism
- Closure under and
- Regular expressions → finite automata

Goal: Show finite automata equivalent to regular expressions

- This week's check-ins will not be counted
- TA office hours will be posted tomorrow
- Chat is restricted to TAs only.

Problem Sets

- 35% of overall grade

- Problems are hard! Leave time to think about them.

- Writeups need to be clear and understandable, handwritten ok.
  Level of detail in proofs comparable to lecture: focus on main ideas.
  Don't need to include minor details.

- Submit via gradescope (see Canvas) by 2:30pm Cambridge time.
  Late submission accepted (on gradescope) until 11:59pm following day:
    1 point (out of 10 points) per late problem penalty.
  After that solutions are posted so not accepted without S3 excuse.

- Optional problems:
    Don't count towards grade except for A+.
    Value to you (besides the challenge):
    Recommendations, employment (future grading, TA, UROP)

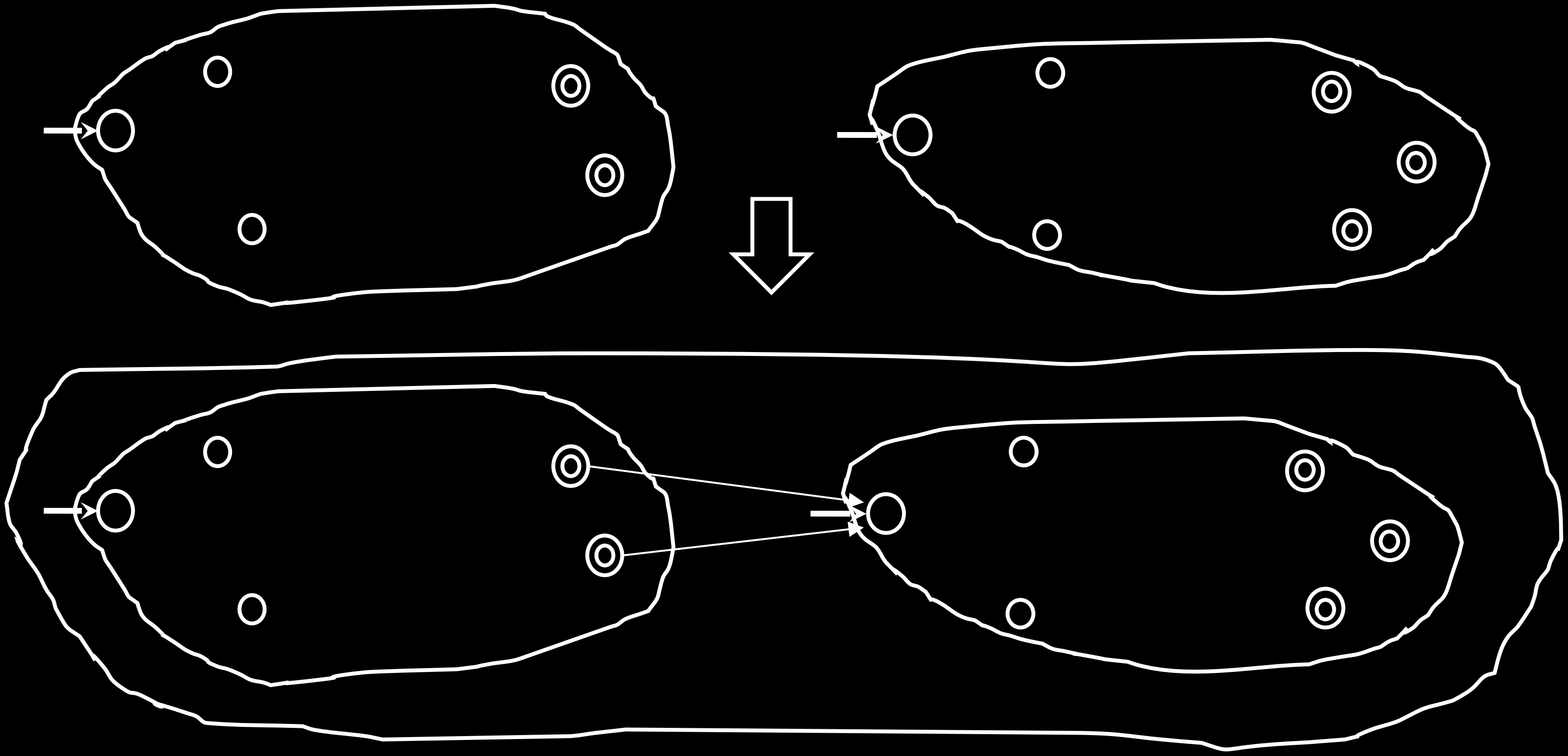
- Problem Set 1 is due in one week.

Closure Properties for Regular Languages

Theorem: If are regular languages, so is (closure under)

Recall proof attempt: Let recognize
recognize

Construct recognizing

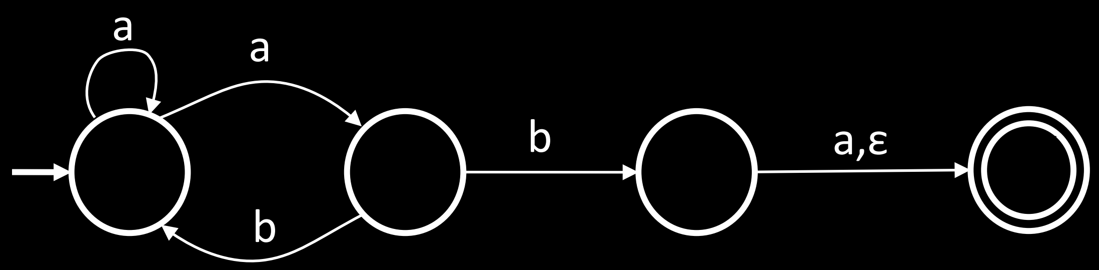


should accept input
if where
accepts and accepts .

Doesn’t work: Where to split ?

Hold off. Need new concept.

Nondeterministic Finite Automata



New features of nondeterminism:

- multiple paths possible (0, 1 or many at each step)
- ε-transition is a "free" move without reading input
- Accept input if <u>some</u> path leads to ◎ accept

Example inputs:

- ab
- aa
- aba
- abb

Check-in 2.1

What does do on input aab ?

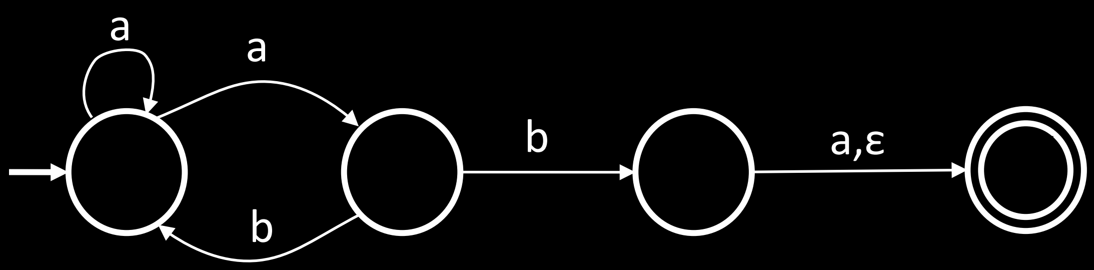
(a) Accept

(b) Reject

(c) Both Accept and Reject

Check-in 2.1

NFA – Formal Definition



Defn: A <u>nondeterministic finite automaton (NFA)</u> is a 5-tuple

states
alphabet
transition function
start state
accept states

- all same as before except
-

power set

- In the example:

Ways to think about nondeterminism:

<u>Computational:</u> Fork new parallel thread and accept if any thread leads to an accept state.

<u>Mathematical:</u> Tree with branches.
Accept if any branch leads to an accept state.

<u>Magical:</u> Guess at each nondeterministic step which way to go. Machine always makes the right guess that leads to accepting, if possible.

Converting NFAs to DFAs

Theorem: If an NFA recognizes then is regular

Proof: Let NFA recognize
Construct DFA recognizing

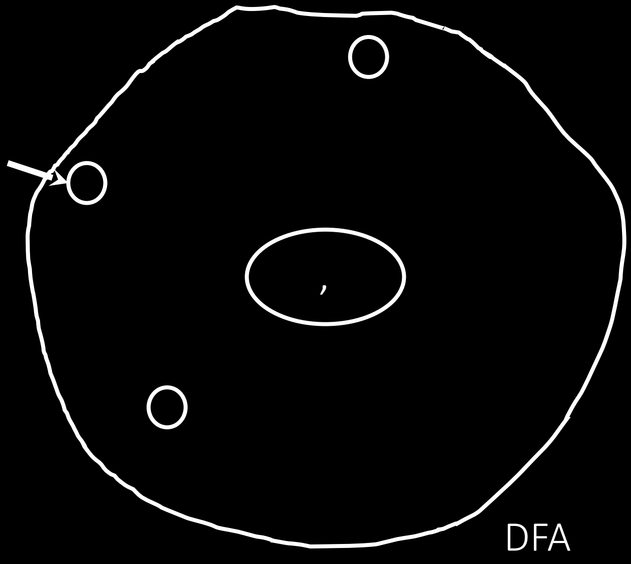
(Ignore the **ε-transitions, can easily modify to handle them)**

IDEA: DFA keeps track of the <u>subset of possible states</u> in NFA .

Check-in 2.2

If has states, how many states does have by this construction?

(a)
(b)
(c)



Construction of :

for some

intersects

Check-in 2.2

Coffee
Break

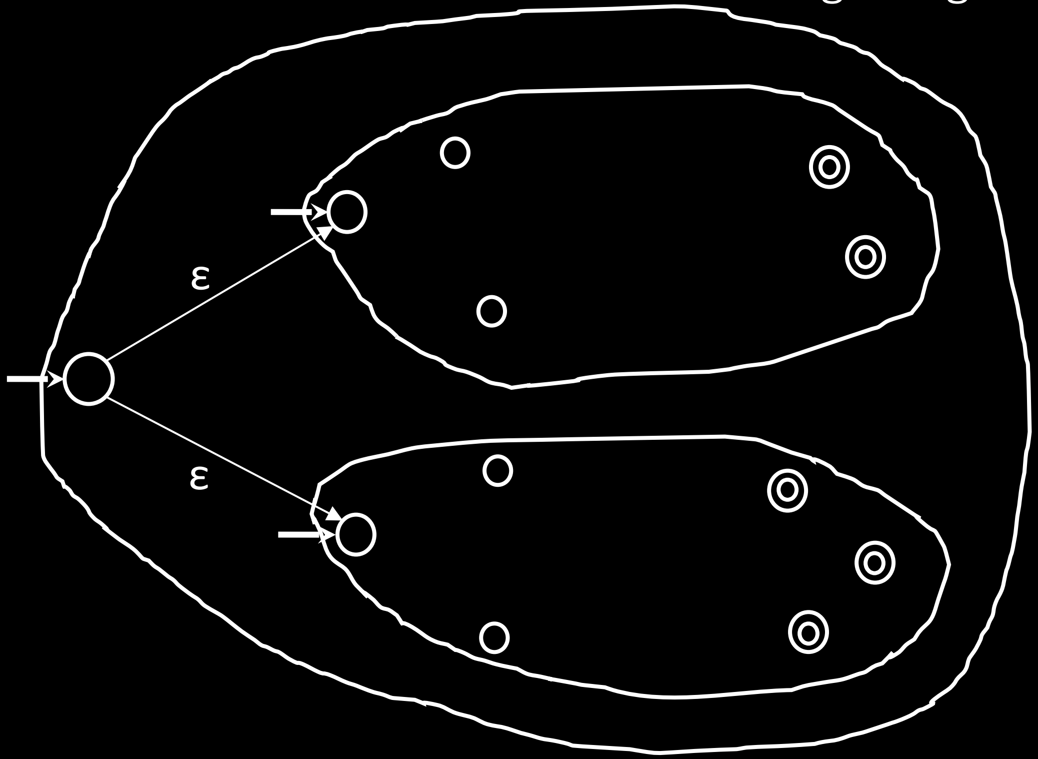
Return to Closure Properties

Recall Theorem: If are regular languages, so is

(The class of regular languages is closed under union)

New Proof (sketch): Given DFAs and recognizing and

Construct NFA recognizing



<u>Nondeterminism</u>

parallelism

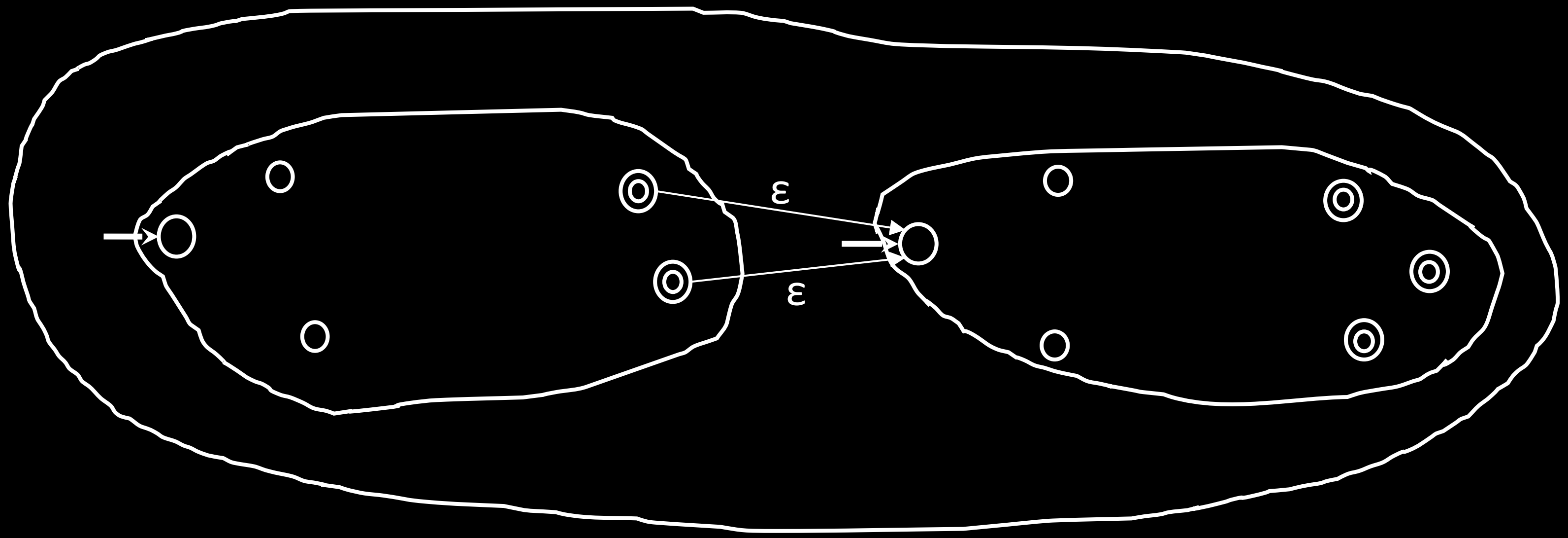
vs

guessing

Closure under (concatenation)

Theorem: If are regular languages, so is

Proof sketch: Given DFAs and recognizing and
Construct NFA recognizing



should accept input
if where
accepts and accepts .

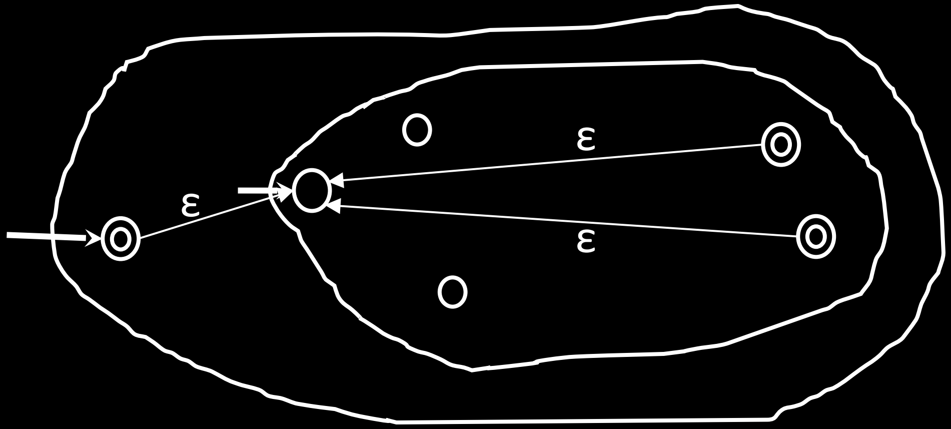
¿

Nondeterministic has <u>the option</u> to jump to when accepts.

Closure under (star)

Theorem: If is a regular language, so is

Proof sketch: Given DFA recognizing
Construct NFA recognizing



Make sure **accepts ε**

Check-in 2.3

If has states, how many states does have by this construction?
(a)
(b)
(c)

Check-in 2.3

Regular Expressions NFA

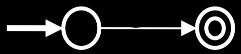
Theorem: If is a regular expr and then is regular

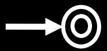
Proof: Convert to equivalent NFA :

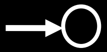
If is atomic:

for

Equivalent is:







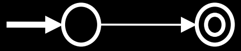
If is composite:

} Use closure constructions

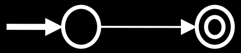
Example:

Convert to equivalent NFA

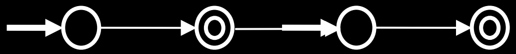
:



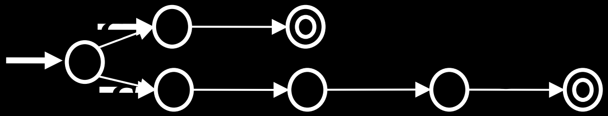
:



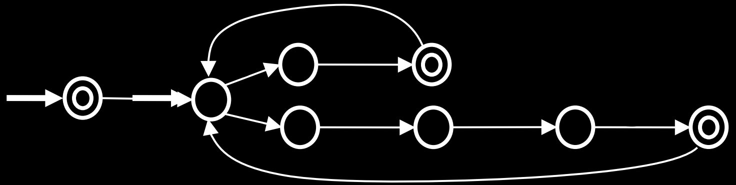
:



:



:



Quick review of today

1. Nondeterministic finite automata (NFA)
2. Proved: NFA and DFA are equivalent in power
3. Proved: Class of regular languages is closed under
4. Conversion of regular expressions to NFA

Check-in 2.4

Recitations start tomorrow online (same link as for lectures).
They are optional, unless you need more help.
You may attend any recitation(s).
Which do you think you'll attend? (you may check several)
(a) 10:00 (b) 11:00 (c) 12:00
(d) 1:00 (e) 2:00 (f) I prefer a different time (please post on piazza, but no promises)

Check-in 2.4