Relating REs, NFAs, DFAs, CFGs

CSE 311: Foundations of Computing I
Lecture 23

Announcements

- HW6 solutions are at the front, grades to be published soon
- HW7 due on Wednesday at 11:59 pm on Grin
  - No Late Days permitted
- Final Exam is this Thursday and Friday
  - Information posted on Exams page
  - Optional review session tomorrow from 3:00 – 4:20pm in DEM 104. Will be recorded on Panopto.

Recall: REs & CFGs

- Regular Languages are languages that can be matched by a Regular Expression.

  For Example, $(0 \cup 1)^*11(0 \cup 1)^*$ matches all binary strings containing 11.

- Context-Free Languages are languages that can be generated by a Context-Free Grammar.

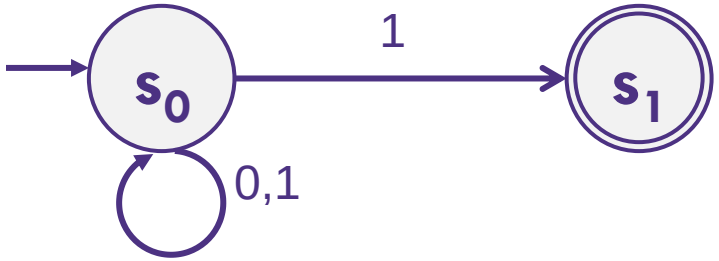
  For Example, $S \rightarrow 0S \mid S1 \mid \varepsilon$ generates $\{0^n 1^n : n \geq 0\}$

Recall: DFAs and NFAs

- Deterministic Finite Automata are one model of computation. DFAs require exactly one outgoing edge per state per character in Σ. Accepts a string iff there is a path to an accept state. For example:

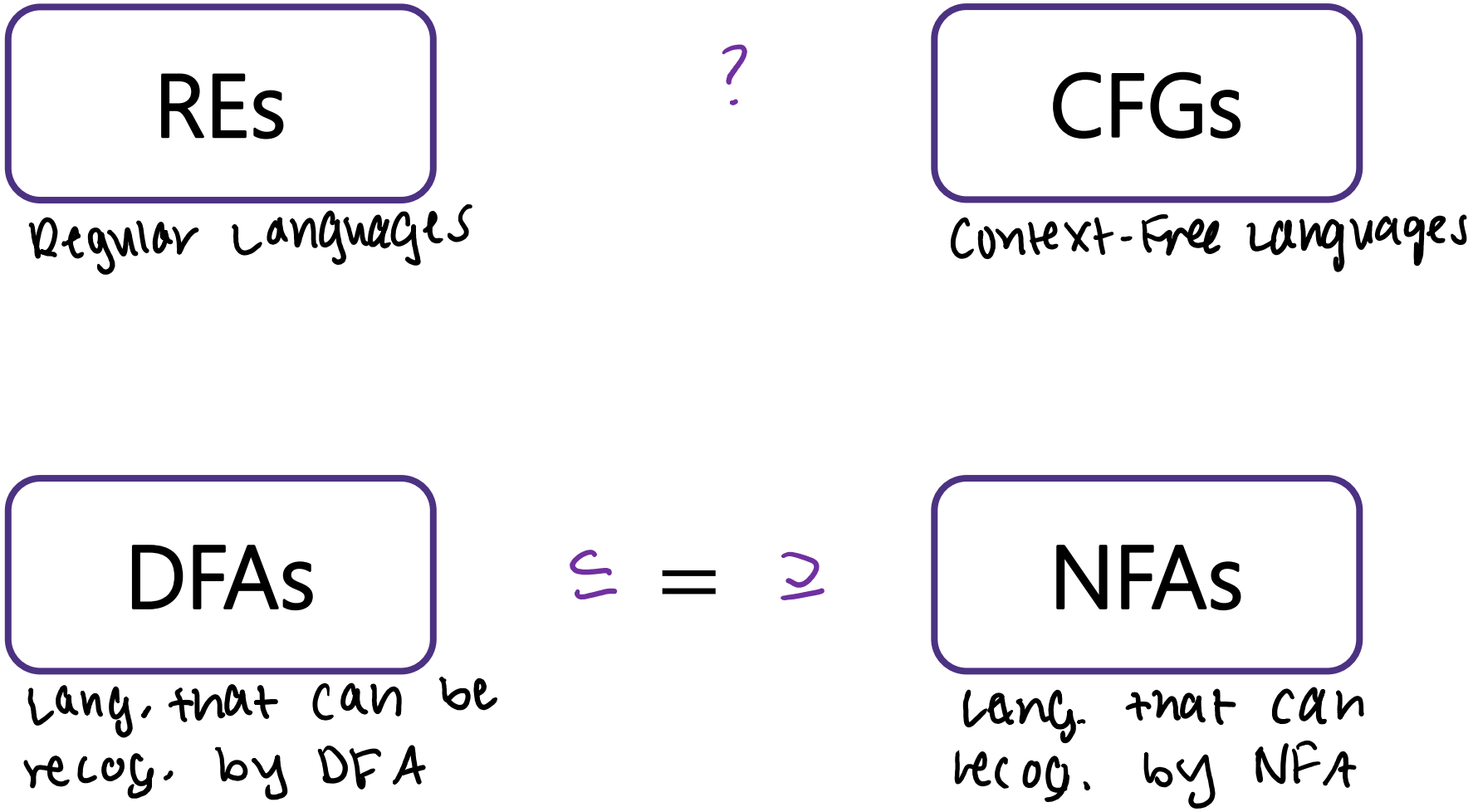
Recognizes binary strings of even length

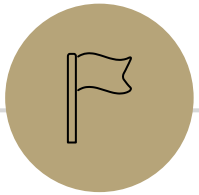
- Nondeterministic Finite Automata are another model of computation. NFAs allow for multiple or no edges per state per character in Σ. Accepts a string iff there is a path to an accept state. For example:



Recognizes binary strings ending in 1

The story so far...



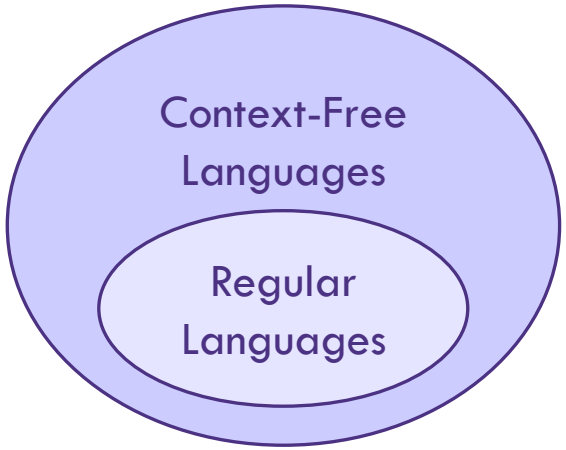


REs ⊆ CFGs

REs and CFGs

Theorem 1:

For any language A matched by a Regular Expression, there is a CFG that generates A.



Proof Idea: Structural Induction! Show that claim holds for all Basis Steps and Recursive Steps of REs.

Recall: Regular Expressions

Basis Step:

- ε is a regular expression
- a is a regular expression for any $a \in \Sigma$

Recursive Step: If A and B are regular expressions, then…

- $A \cup B$ is a regular expression
- AB is a regular expression
- A^* is a regular expression

Proof of Theorem 1

Basis Steps:
ε is a regular expression
a is a regular expression for any $a \in \Sigma$

1. Let $\mathrm{P}(R)$ be "For a Regular Expression R, there exists a CFG that generates the same language". We prove $\mathrm{P}(R)$ for all REs R by structural induction.

2. Base Cases:
   $R = \varepsilon$ ε is generated by the CFG $S \to \varepsilon$.

   $R = a$ a is generated by the CFG $S \to a$.

Proof of Theorem 1

Recursive Step: If A, B are REs…
- $A \cup B$ is a RE
- AB is a RE
- A^* is a RE

3. IH: Let A, B be arbitrary REs. Suppose there exists a CFG with start symbol S_1 that generates the language matched by A. Suppose there exists a CFG with start symbol S_2 that generates the language matched by B.

4. IS:

   $R = A \cup B$ $\quad S \to S_1 \mid S_2$ $\quad$ so $\quad P(A \cup B)$ holds

   $R = AB$ $\quad S \to S_1 S_2$ $\quad$ so $\quad P(AB)$ holds

   $R = A^*$ $\quad S \to SS_1 \mid \varepsilon$ $\quad$ so $\quad P(A^*)$ holds

5. Conclusion: (omitted)

Example of RE to CFG Construction

The proof of Theorem 1 gives us a process to write every RE as a CFG.
For example, write the RE $(a \cup b)^* c$ as a CFG using the proof construction.

1. To write a, we can use the CFG
$$\mathrm{A} \to a$$

2. To write b, we can use the CFG
$$\mathrm{B} \to b$$

3. To write c, we can use the CFG
$$\mathrm{C} \to c$$

4. To write $a \cup b$, we can use the CFG
$$\mathrm{D} \to \mathrm{A} \mid \mathrm{B}$$
$$\mathrm{A} \to a$$
$$\mathrm{B} \to b$$

5. To write $(a \cup b)^*$, we can use the CFG
$$\mathrm{E} \to \mathrm{DE} \mid \varepsilon$$
$$\mathrm{D} \to \mathrm{A} \mid \mathrm{B}$$
$$\mathrm{A} \to a$$
$$\mathrm{B} \to b$$

6. To write $(a \cup b)^* c$, we can use the CFG
$$\mathrm{F} \to \mathrm{EC}$$
$$\mathrm{E} \to \mathrm{DE} \mid \varepsilon$$
$$\mathrm{D} \to \mathrm{A} \mid \mathrm{B}$$
$$\mathrm{A} \to a$$
$$\mathrm{B} \to b$$
$$\mathrm{C} \to c$$

$(a \cup b)^* c$
$(a \cup b)^*$
c

Example of RE to CFG Construction

The proof of Theorem 1 gives us a process to write every RE as a CFG.
For example, write the RE $(a \cup b)^* c$ as a CFG using the proof construction.

$\mathrm{F} \rightarrow \mathrm{EC}$
$\mathrm{E} \rightarrow \mathrm{DE} \mid \varepsilon$
$\mathrm{D} \rightarrow \mathrm{A} \mid \mathrm{B}$
$\mathrm{A} \rightarrow a$
$\mathrm{B} \rightarrow b$
$\mathrm{C} \rightarrow c$

Example of RE to CFG Construction

Note: This construction will not necessarily give us the simplest CFG.

For example, for $(a \cup b)^* c$, the construction gave:

$\mathrm{F} \rightarrow \mathrm{EC}$

$\mathrm{E} \rightarrow \mathrm{DE} \mid \varepsilon$

$\mathrm{D} \rightarrow \mathrm{A} \mid \mathrm{B}$

$\mathrm{A} \rightarrow a$

$\mathrm{B} \rightarrow b$

$\mathrm{C} \rightarrow c$

$S \Rightarrow AS \Rightarrow AAS \Rightarrow AAc$

$\Rightarrow aAc$

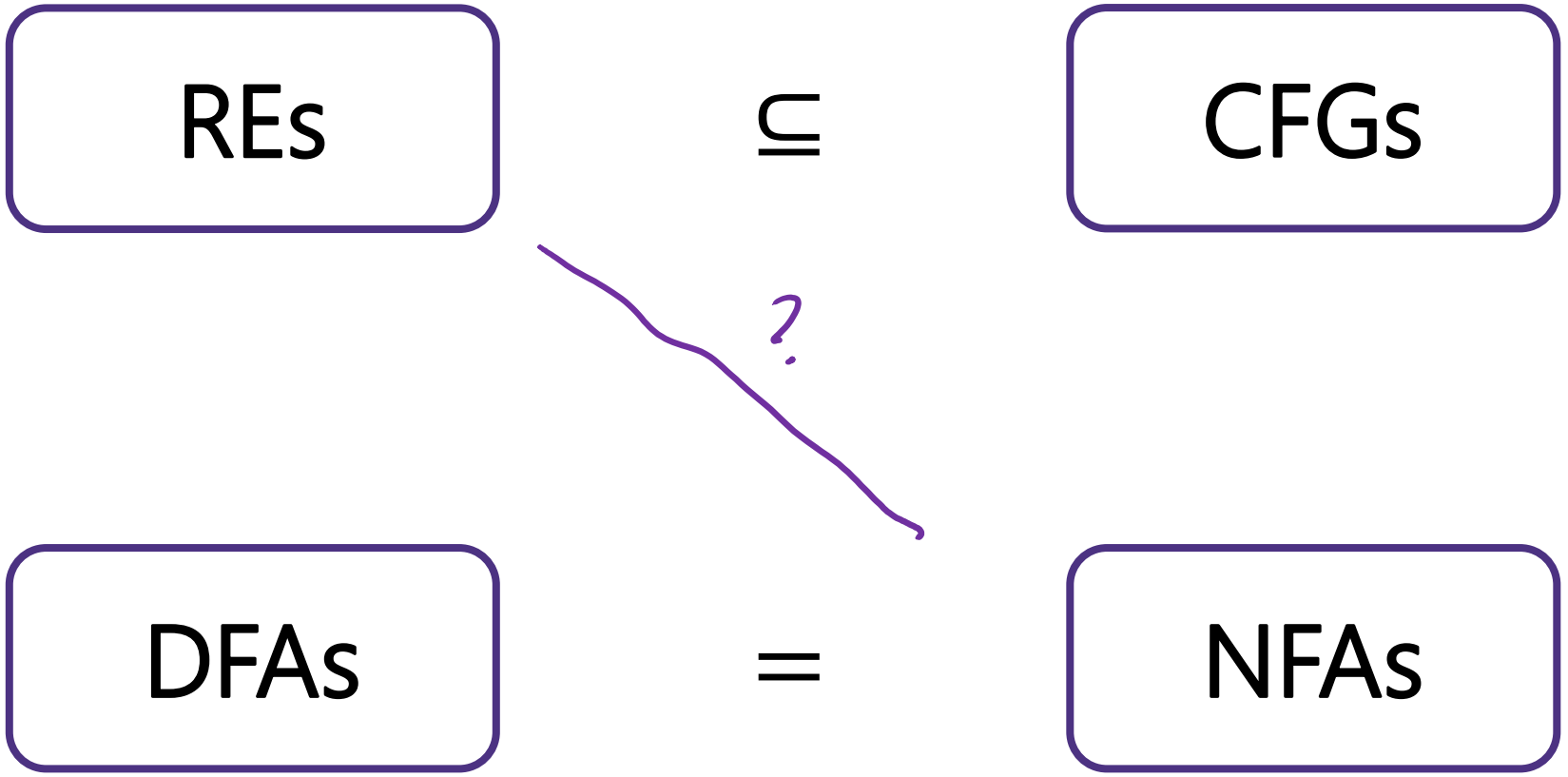
$\Rightarrow abc$

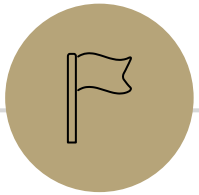
But this CFG would also work:

$\mathrm{S} \rightarrow \mathrm{AS} \mid c$

$\mathrm{A} \rightarrow a \mid b$

The story so far…

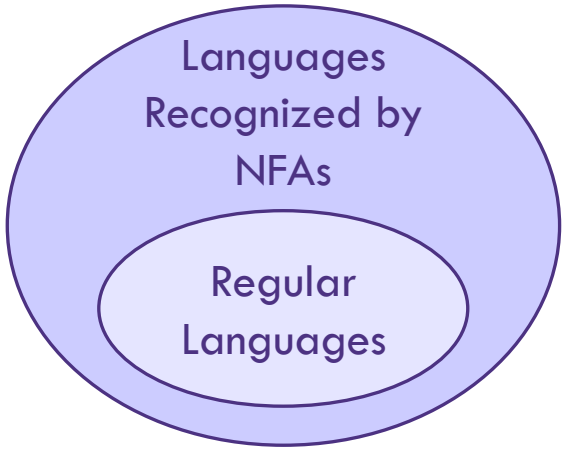




REs ⊆ NFAs

REs and NFAs

Theorem 2:
For any language A matched by a Regular Expression, there is an NFA that recognizes A.



Proof Idea: Structural Induction again! Show that claim holds for all Basis Steps and Recursive Steps of REs.

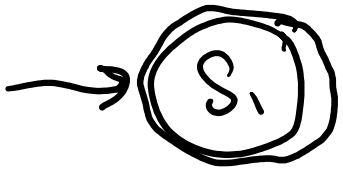
Proof Sketch of Theorem 2

Basis Steps:
ε is a regular expression
a is a regular expression for any $a \in \Sigma$

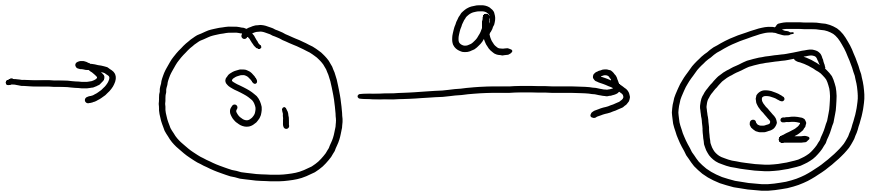
1. Let $\mathrm{P}(R)$ be "For a Regular Expression R, there exists an NFA that recognizes the same language". We prove $\mathrm{P}(R)$ for all REs R by structural induction.

2. Base Cases:

   $R = \varepsilon$



   $R = a$

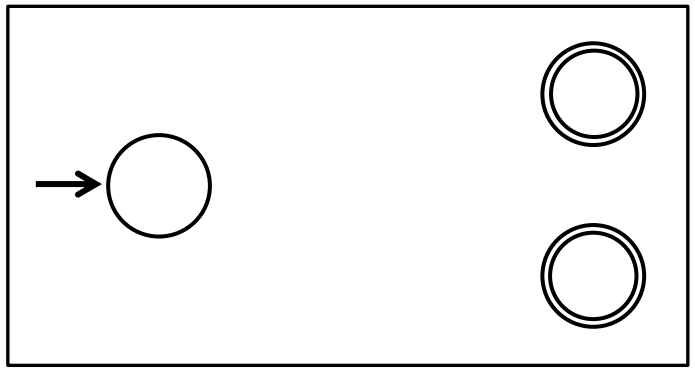


Proof Sketch of Theorem 2

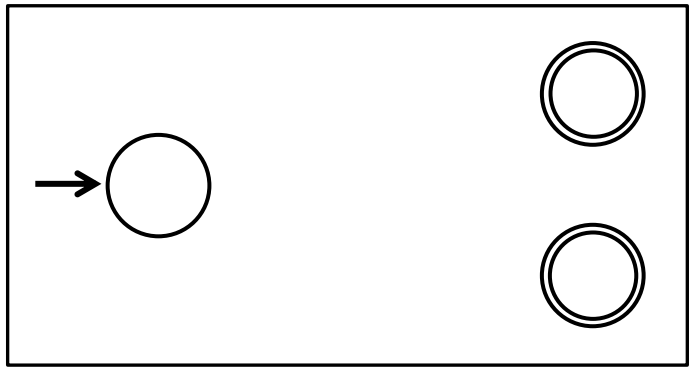
Recursive Step: If A, B are REs…

- $A \cup B$ is a RE
- AB is a RE
- A^* is a RE

3. IH: Suppose there for some arbitrary REs A, B there exists NFAs N_A and N_B that recognize the language matched by A and B respectively.



N_A

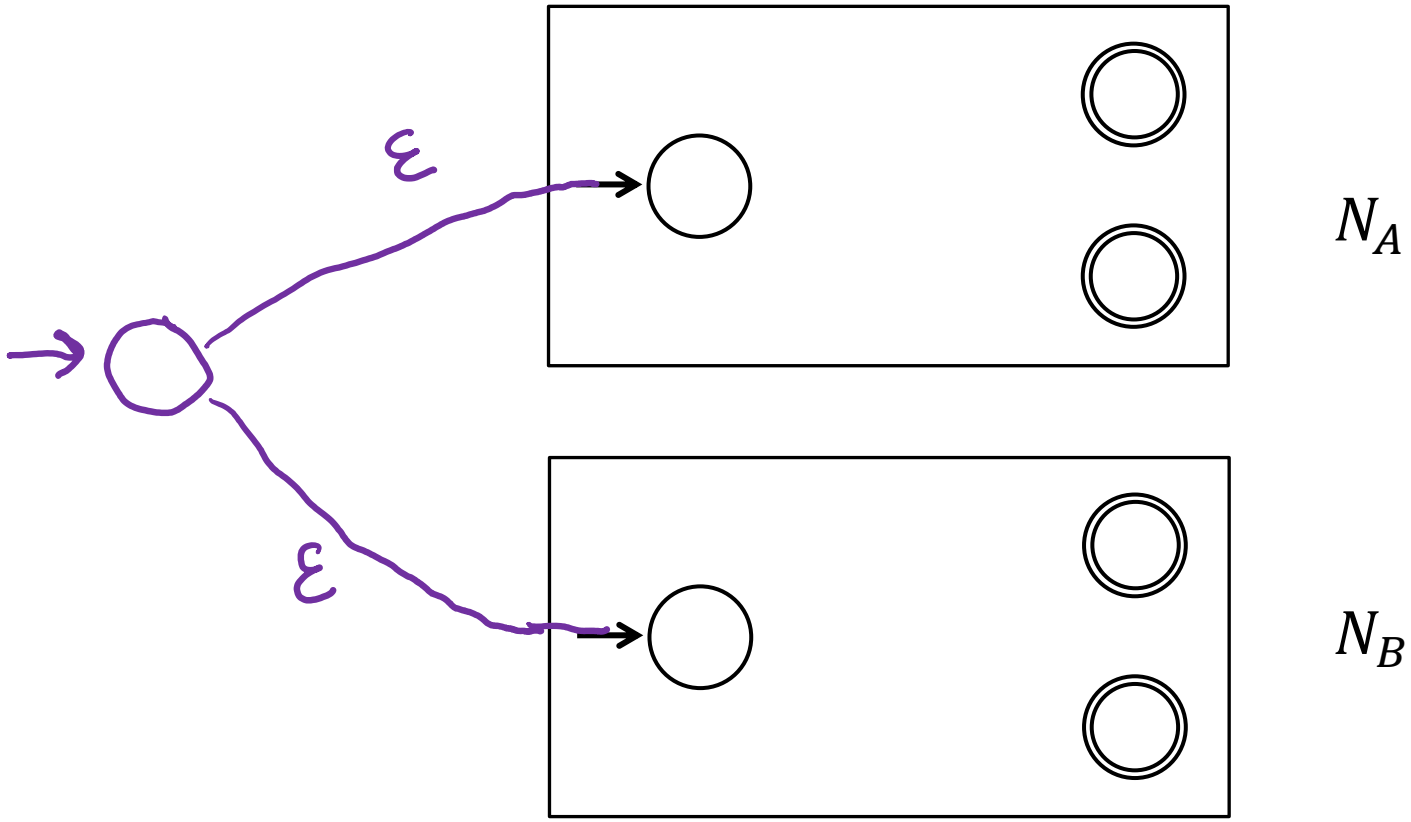


N_B

Proof Sketch of Theorem 2

Recursive Step: If A, B are REs…
- $A \cup B$ is a RE
- AB is a RE
- A^* is a RE

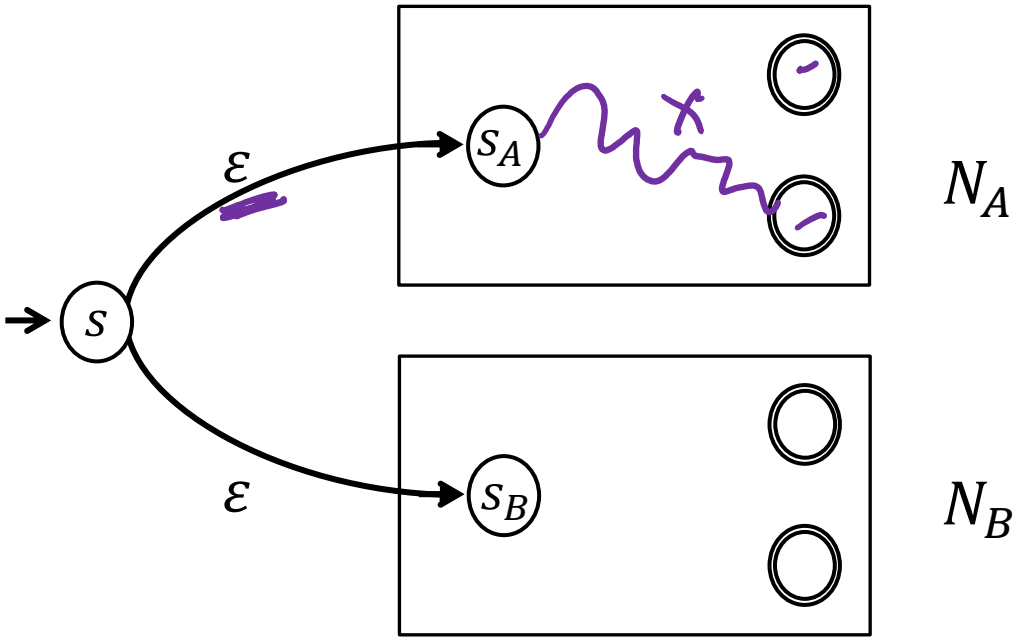
4. IS: The RE $A \cup B$ can be matched by the following NFA:



Proof Sketch of Theorem 2

Recursive Step: If A, B are REs…
- $A \cup B$ is a RE
- AB is a RE
- A^* is a RE

4. IS: The RE $A \cup B$ can be matched by the following NFA:



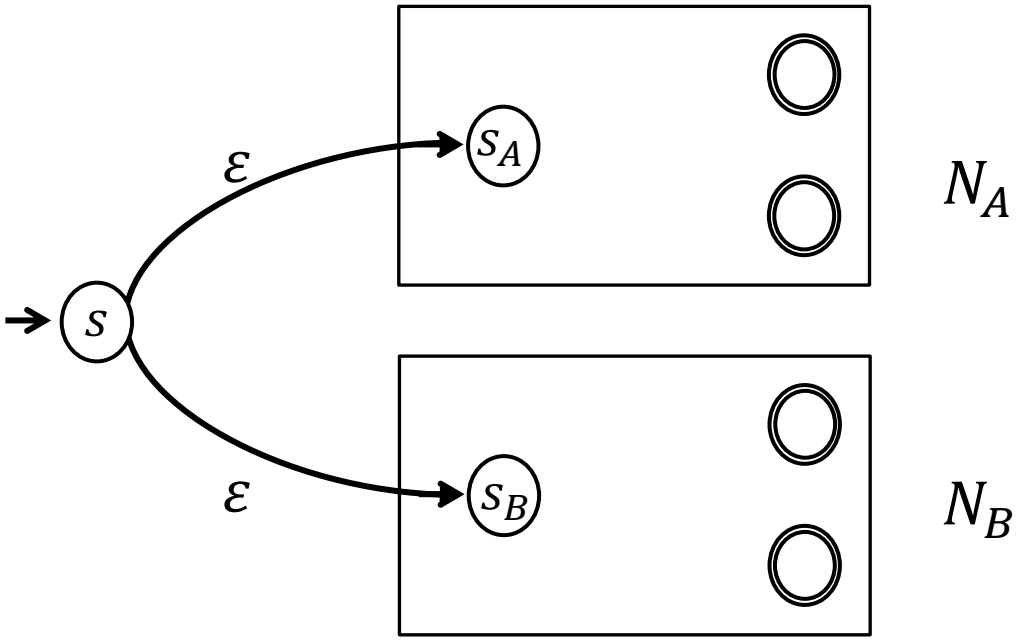
Any string matched by AUB is accepted by this NFA:

- Let x be arb that is matched by AUB
- Then x is matched by A or by B. Suppose wlog x is matched by A.
- Then there's a path from s_A to an accept state in N_A & that path is labelled x.
- Then there is a path from s to an accept state in the NFA labelled x.
- So x is recognized by the NFA.

Proof Sketch of Theorem 2

Recursive Step: If A, B are REs…
- $A \cup B$ is a RE
- AB is a RE
- A^* is a RE

4. IS: The RE $A \cup B$ can be matched by the following NFA:

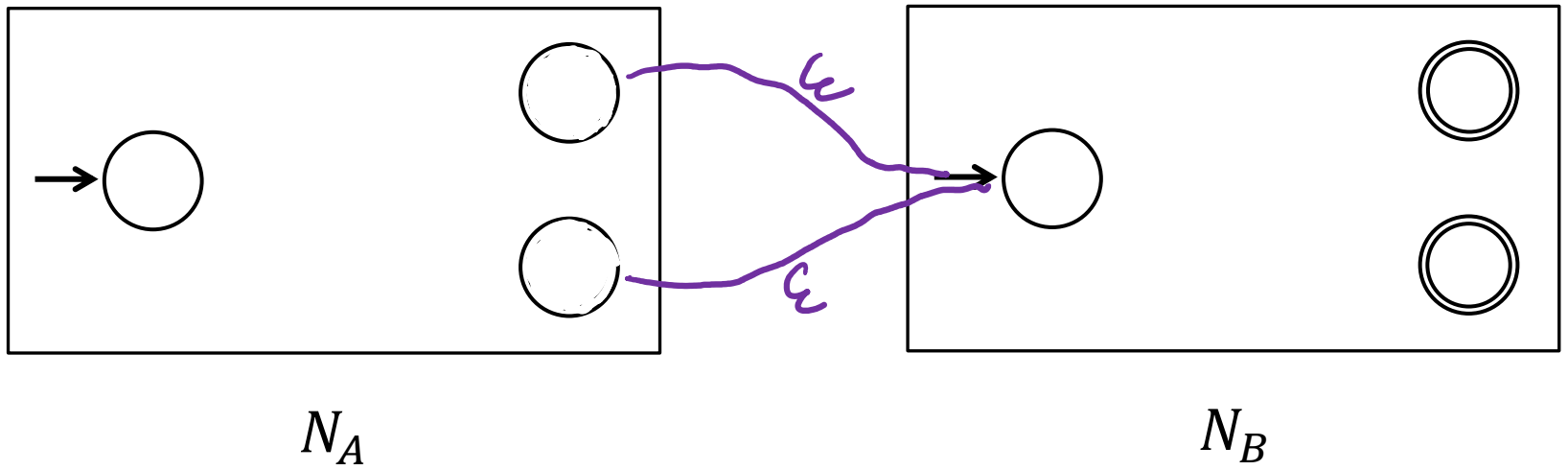


Any string recognized by the new NFA matches A∪B.

Proof Sketch of Theorem 2

Recursive Step: If A, B are REs…
- $A \cup B$ is a RE
- AB is a RE
- A^* is a RE

4. IS: The RE AB can be matched by the following NFA:

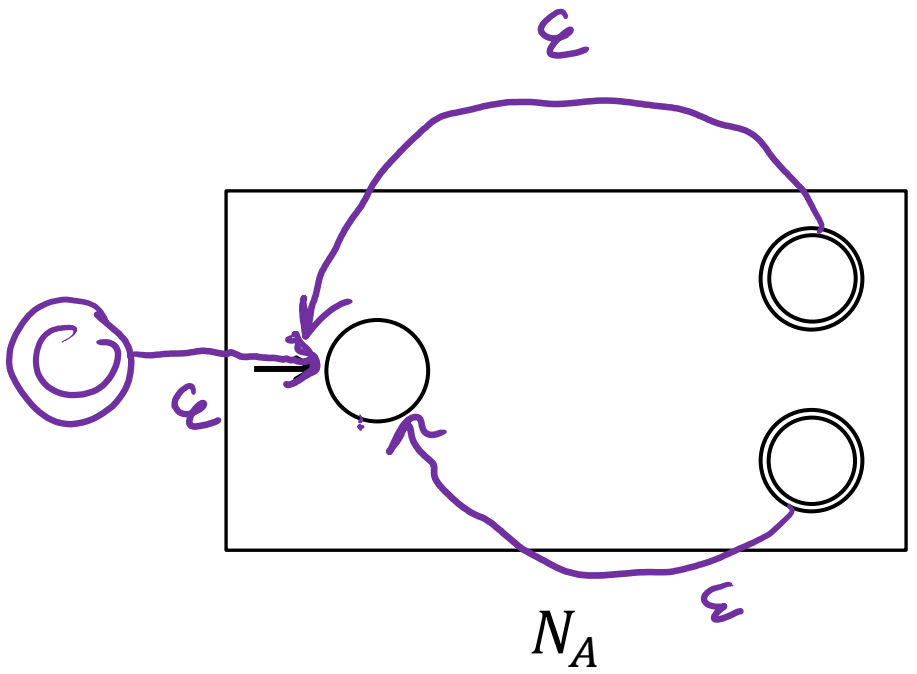


Proof Sketch of Theorem 2

Recursive Step: If A, B are REs…

- $A \cup B$ is a RE
- AB is a RE
- A^* is a RE

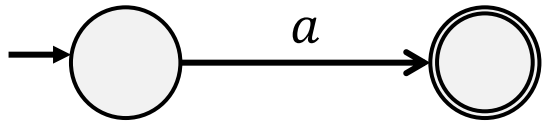
4. IS: The RE A^* can be matched by the following NFA:

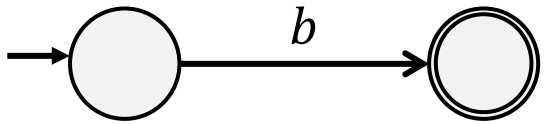


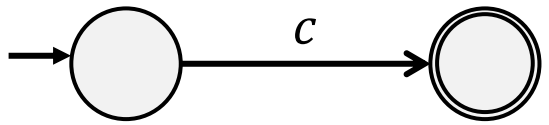
5. Conclusion: (omitted)

Example of RE to NFA Construction

The proof of Theorem 2 gives us a process to write every RE as an NFA.
For example, write the RE $(a \cup b)^* c$ as an NFA using the proof construction.

1. To write a, we can use the NFA 

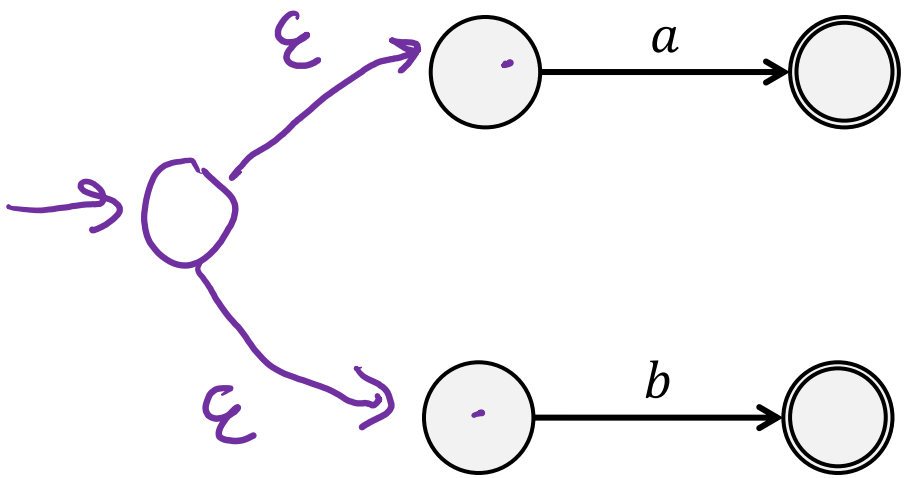
2. To write b, we can use the NFA 

3. To write c, we can use the NFA 

Example of RE to NFA Construction

The proof of Theorem 2 gives us a process to write every RE as an NFA.
For example, write the RE $(a \cup b)^* c$ as an NFA using the proof construction.

4. Construct the NFA recognizing $a \cup b$. Below we have NFAs for a and b.

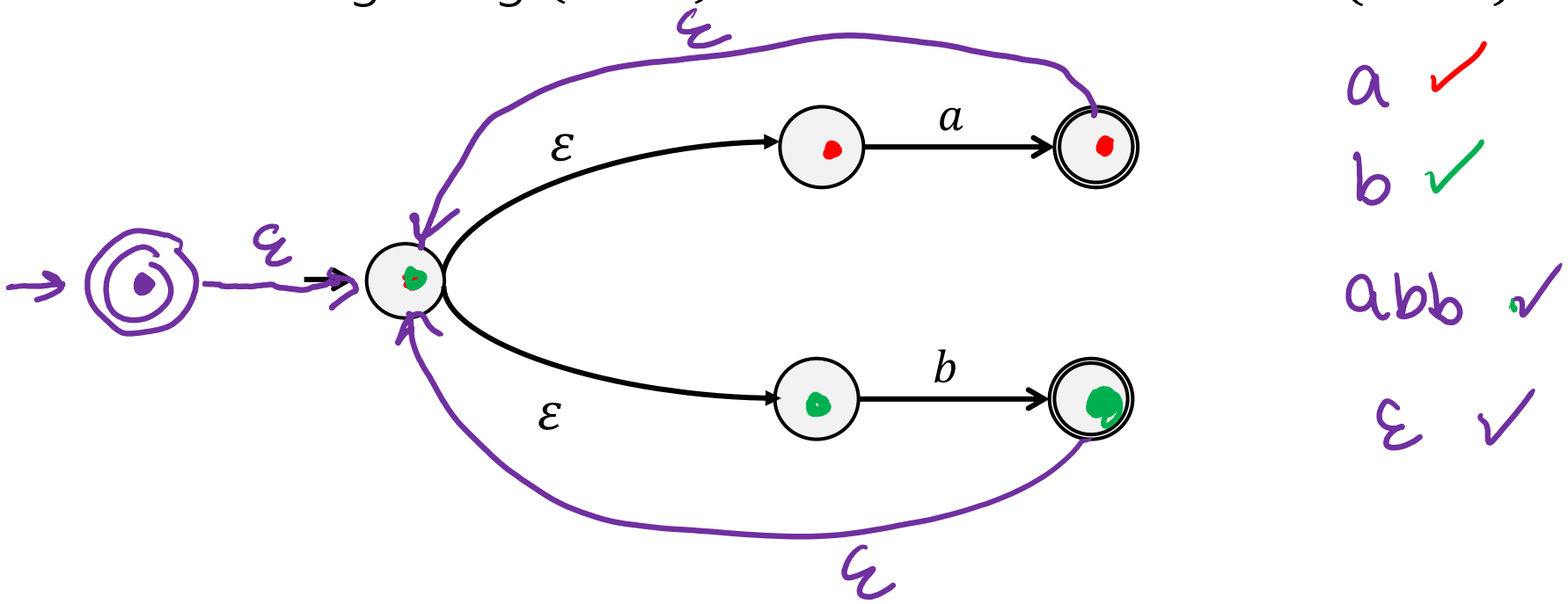


a ✓
b ✓
ab ✗
ε ✗

Example of RE to NFA Construction

The proof of Theorem 2 gives us a process to write every RE as an NFA.
For example, write the RE $(a \cup b)^* c$ as an NFA using the proof construction.

5. Construct the NFA recognizing $(a \cup b)^*$. Below we have NFA for $(a \cup b)$.



a ✓

b ✓

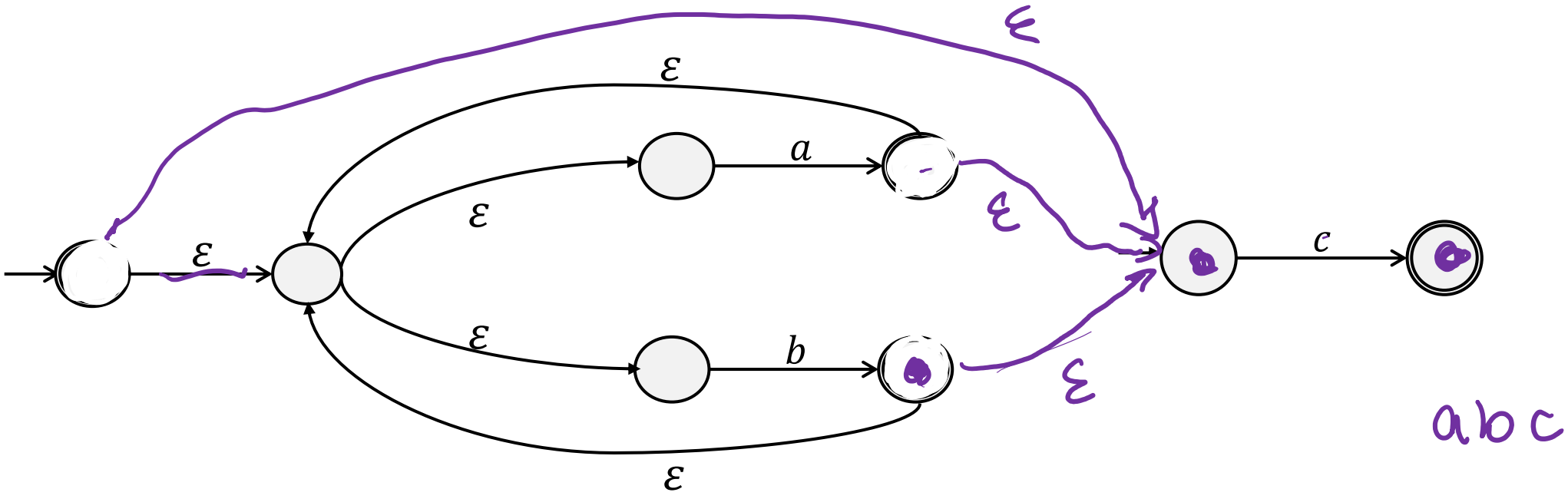
abb ✓

ε ✓

Example of RE to NFA Construction

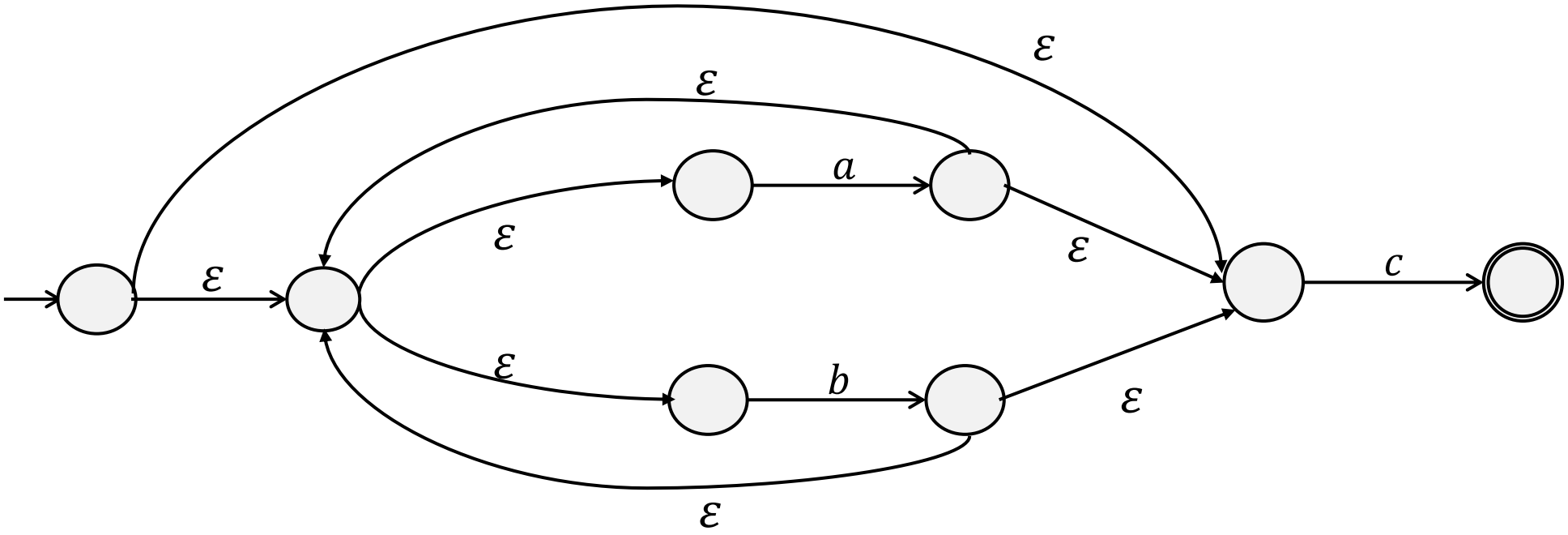
The proof of Theorem 2 gives us a process to write every RE as an NFA.
For example, write the RE $(a \cup b)^* c$ as an NFA using the proof construction.

6. Construct the NFA recognizing $(a \cup b)^* c$. Below we have NFAs for $(a \cup b)^*$ and c.



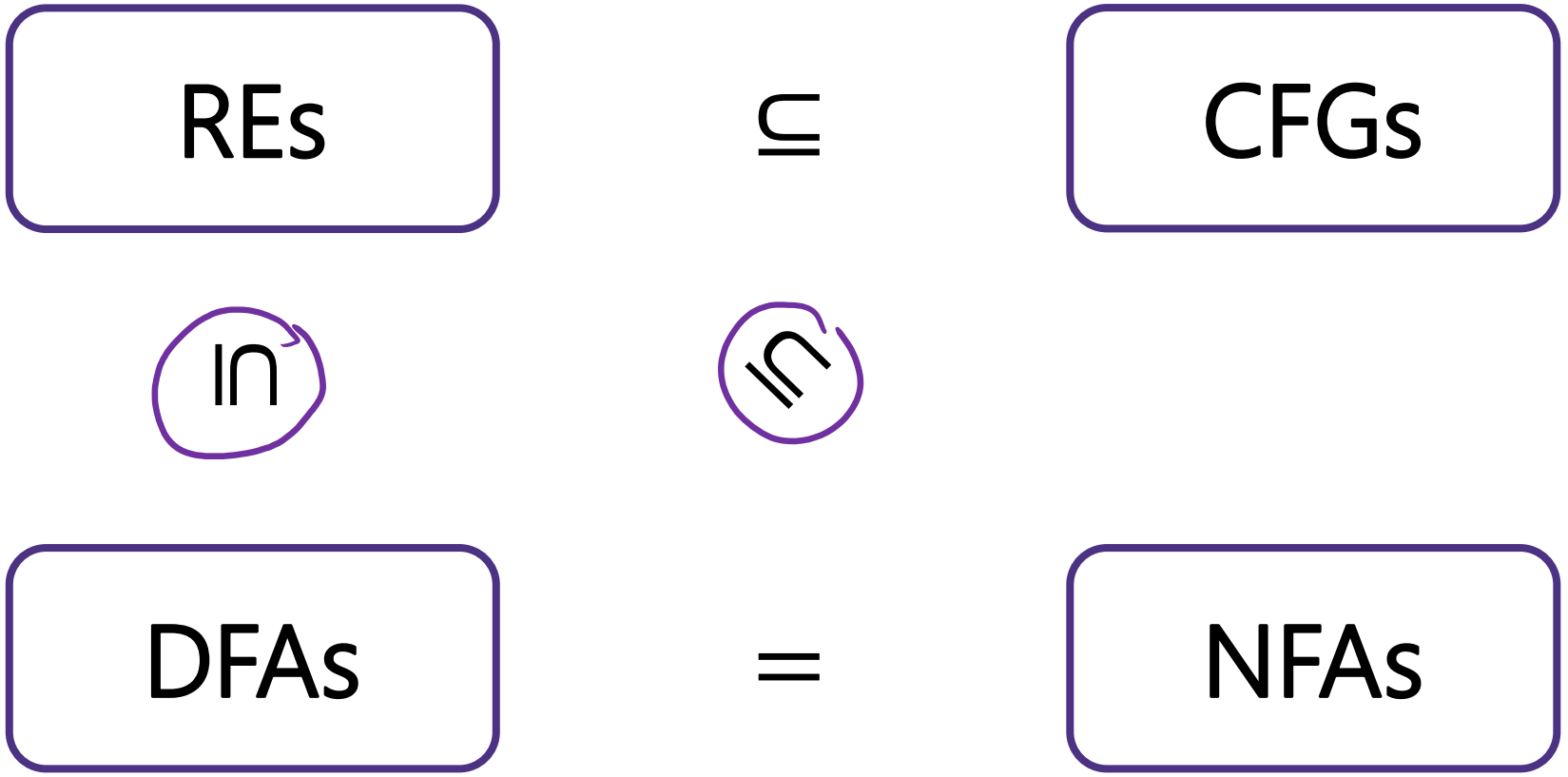
Example of RE to NFA Construction

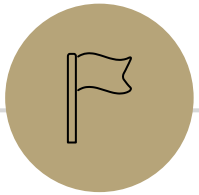
Final NFA construction for the RE $(a \cup b)^* c$:



Note: This construction will not necessarily give us the simplest NFA.

The story so far…





NFAs ⊆ REs

The story so far…

1) =
2) ⊂ ≠

REs ⊆ CFGs

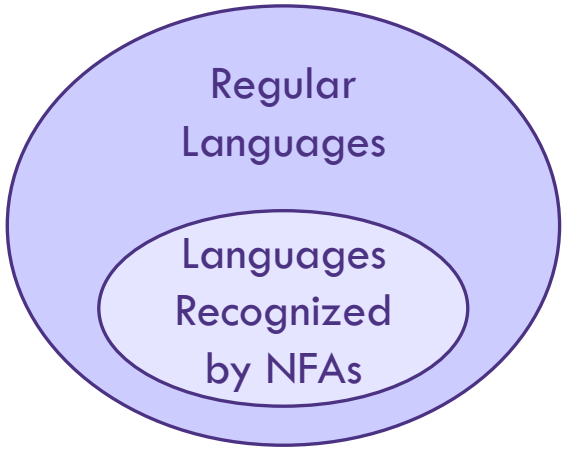
∩ ⊈

DFAs = NFAs

Are these ⊆ really "=" or "⊊"?

NFAs and REs

Theorem 3:
For any language A recognized by an NFA, there is a RE that matches A.



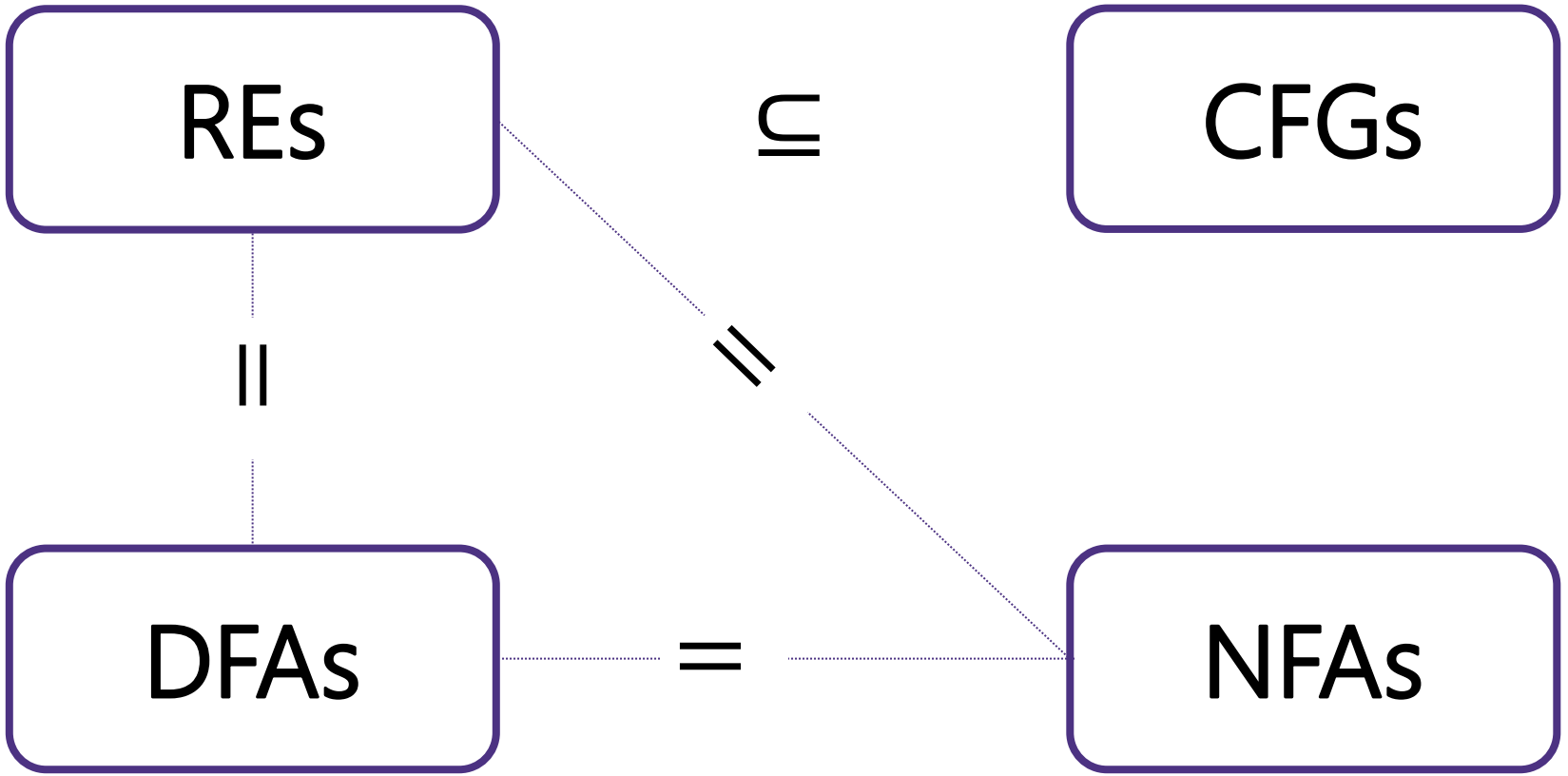
We won't cover this construction, but here are optional notes if you are interested.

Regular Languages

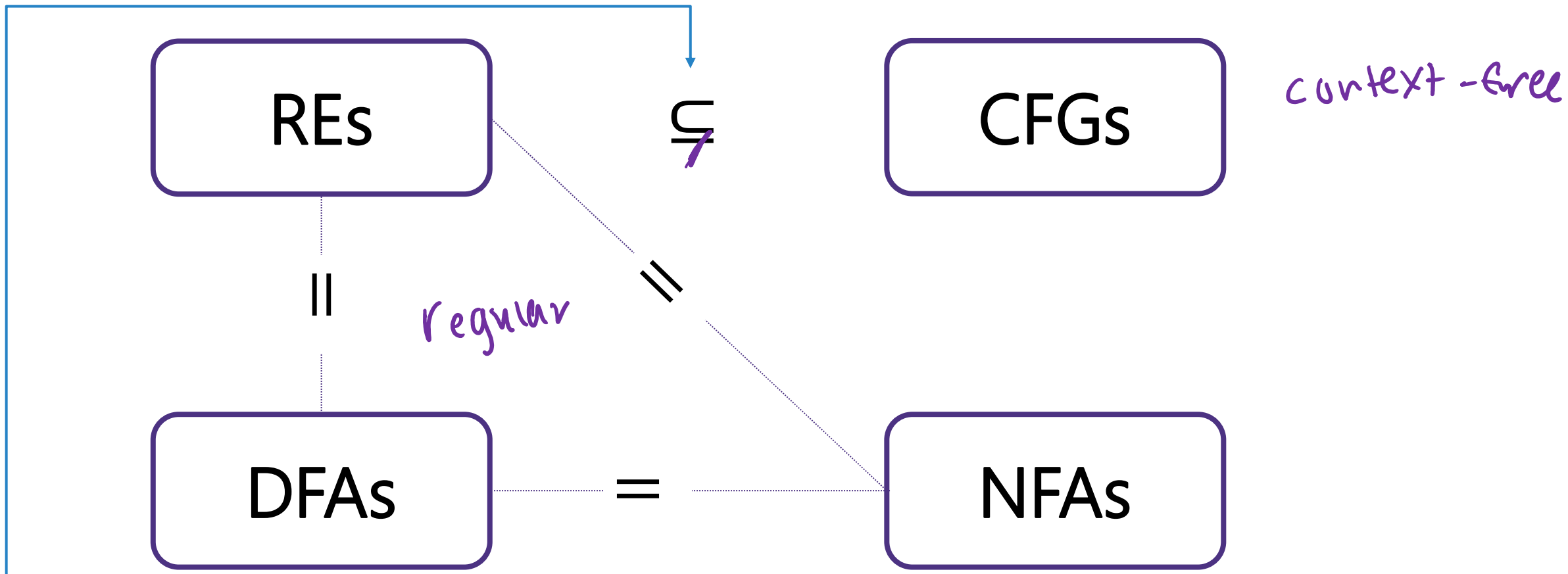
Corollary:

A language is recognized by a DFA iff it is recognized by an NFA, iff it is matched by a RE.

The story so far…



The story so far...



Next time: Proving that this is actually ⊊, & in general how to prove a language is irregular

What do I need to know for the Final?

- We won't ask you to replicate any of the constructions.
  - NFA to DFA (Friday's Class)
  - RE to CFG (Theorem 1)
  - RE to NFA (Theorem 2)
  - NFA to RE (Theorem 3)
- We will ask you True / False, Multiple Choice, or short-answer questions that assess your knowledge of the relationships, and the overall proof concepts.
  - E.g. "Is there an irregular language that can be recognized by a DFA?" No

Feedback: Course Evaluations

We'd **really** appreciate if you take the time right now to fill out the course evaluations! The link should be in your email.