

# Regular Expressions

Last time:

- 1) What a reg expr is
- 2) What set of strings a reg expr stands for (aka what strings match a reg expr)
- 3) How to construct an NFA that accepts precisely the strings that match a given reg. expr.

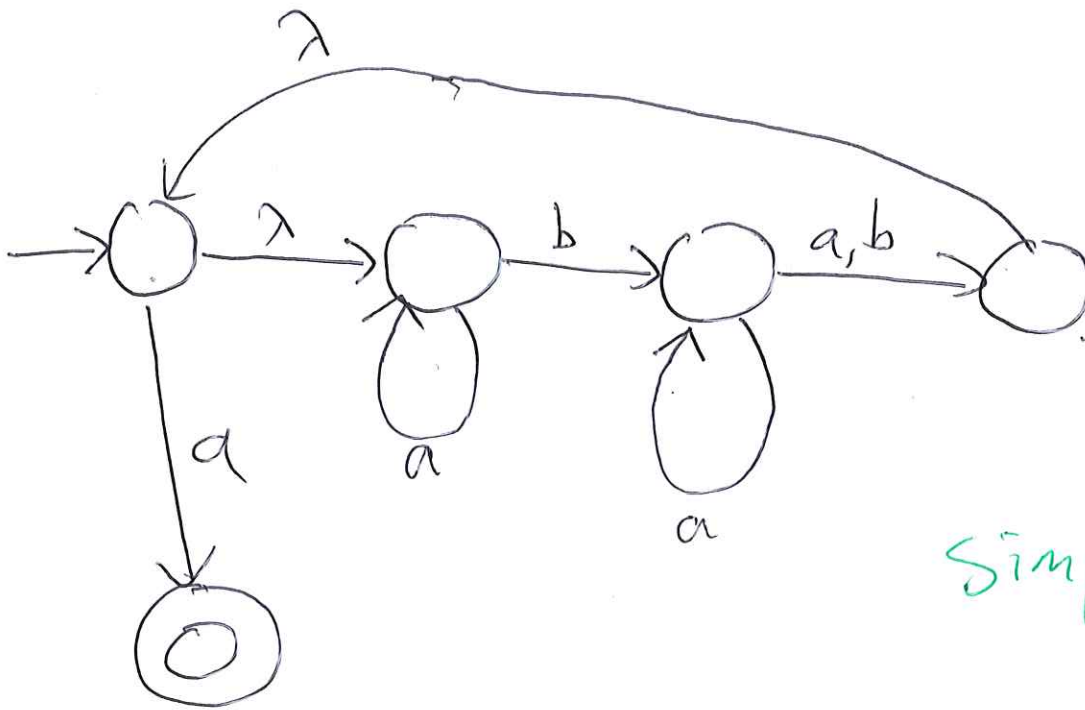
Today: How to ~~write~~ write a reg. expr. for the set of strings accepted by an NFA.

---

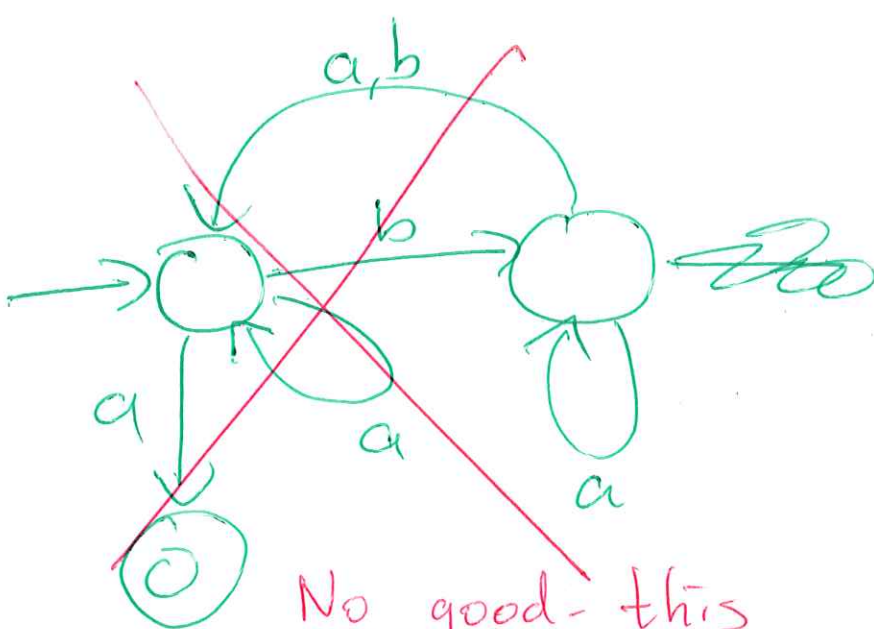
Review example

$(a^*ba^*(a+b))^*$  is a reg expr.

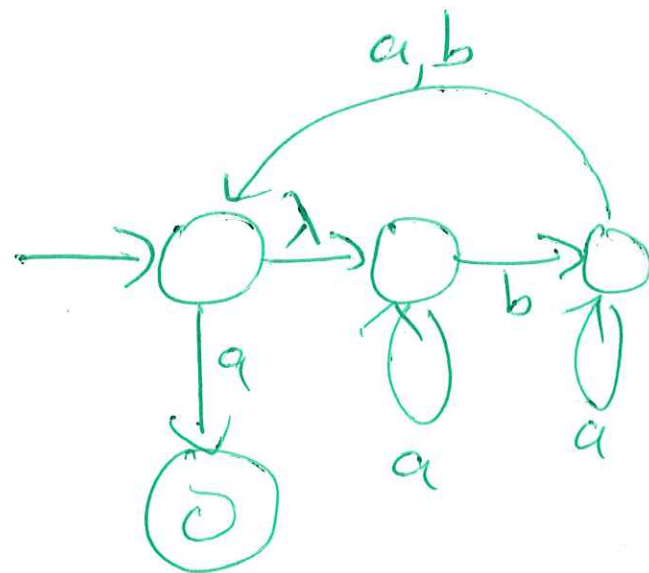
Construct an NFA that accepts this reg. expr.:



Simplify this:



No good - this would let me repeat a's and then use an a to go to final state.



What about  $\emptyset$  in reg. expr.?

$$((a+\emptyset)a^*(b+\emptyset))^* \equiv aa^*b$$

(since " $a+\emptyset$ " and " $a$ " are both matched only by  $a$ )

$$a\emptyset \equiv \emptyset \quad (\text{since nothing can be split up into } a \text{ and } \dots)$$

(but

$$a\lambda \equiv a)$$

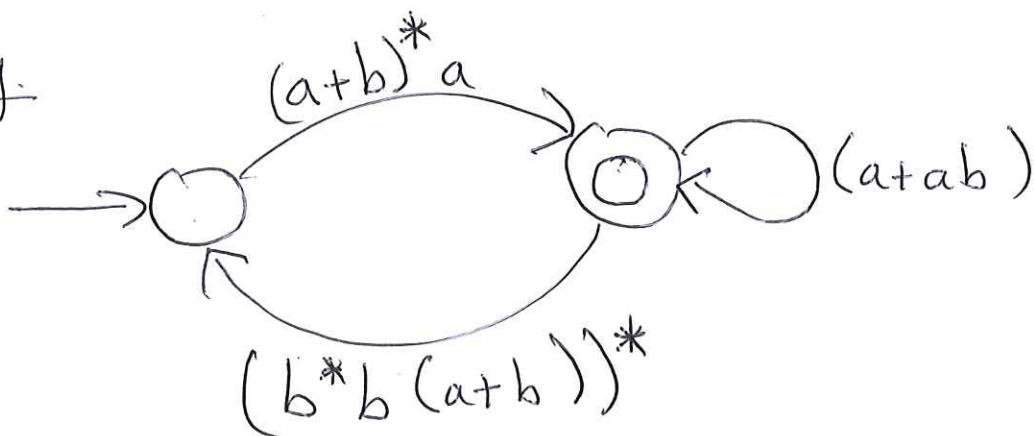
Actually, the  $\emptyset$  is mostly useless; anytime it is combined w/ something you can simplify the result to either just  $\emptyset$  or something w/o  $\emptyset$ . The only use is to specify the reg expr that matches nothing.

## Turning NFAs into reg. exprs:

Define a "super-NFA" (the book calls them "generalized transition graphs": to be a NFA where the arrows are labelled with reg. expr.

(To use a path, you pay any toll that matches the reg. expr.)

E.g.



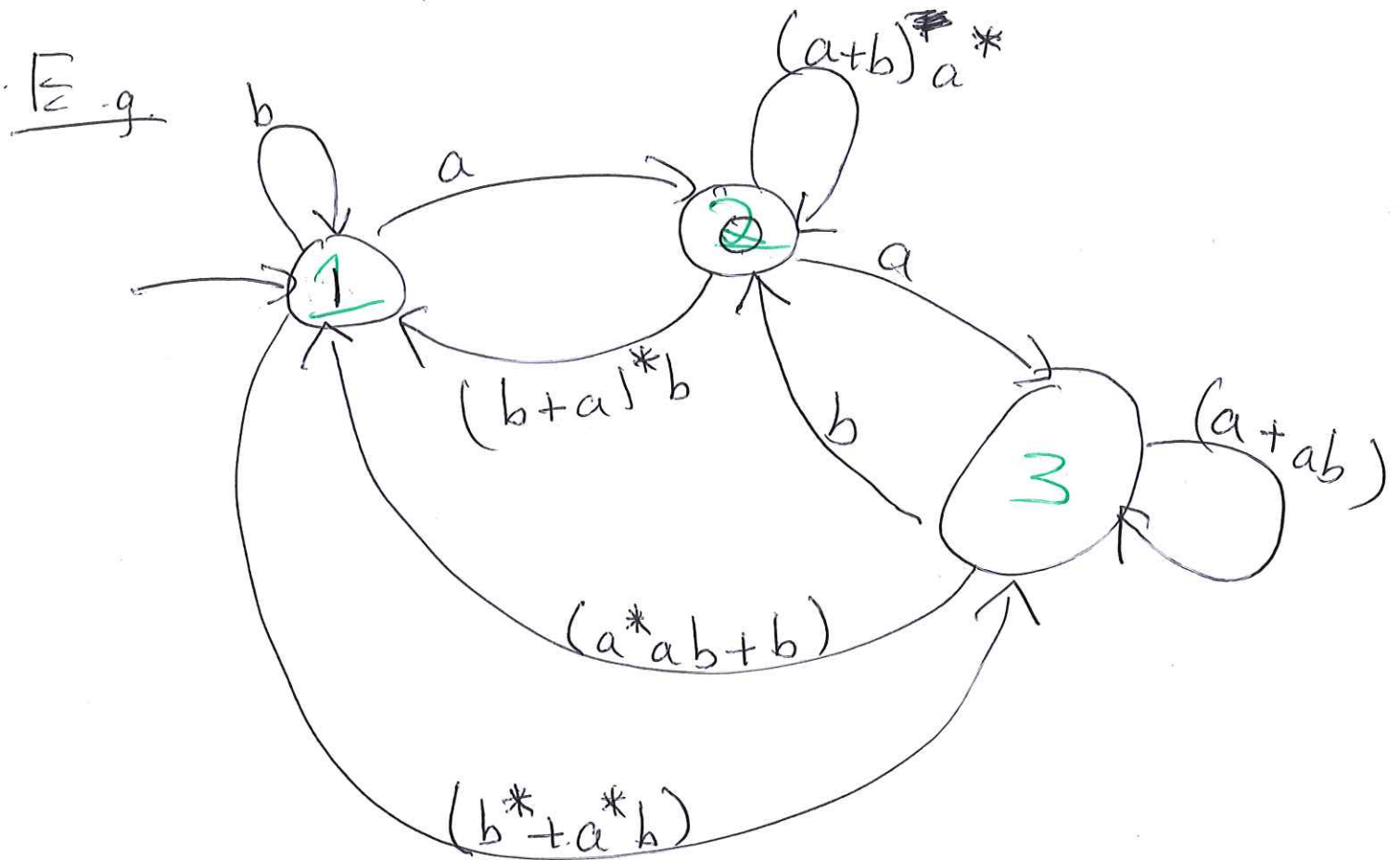
I can figure out what reg. expr this stands for:

$$(a+b)^*a \left[ (atab) + (b^*b(a+b))^*(a+b)^*a \right]^*$$

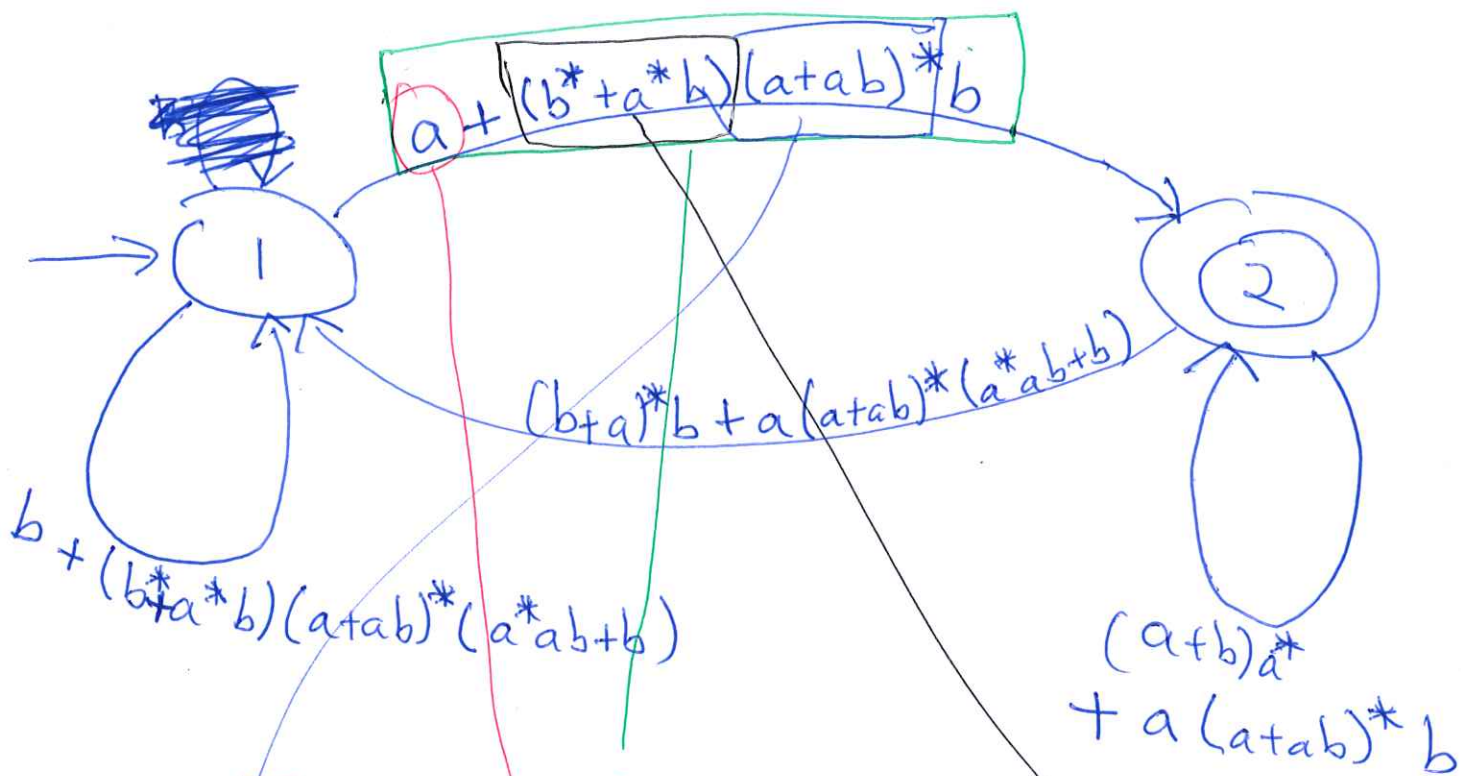
If I have just 2 states, <sup>in a super NFA</sup> 1 of which is final, I can figure out the reg. expr.

For a general super-NFA, I can

- 0) convert to have only one final state.
- 1) reduce the number of states by making my transitions more complicated:



If I want to get rid of  $\textcircled{3}$ ,  
 I need to add extra possibilities  
 to my transitions to account for  
 the transitions that used  $\textcircled{3}$



To get from 1 to 2, I can  
 either go directly or go  $1 \rightarrow 3$ ,  
loop around 3, and then go  $3 \rightarrow 2$ .

2) Keep reducing the number of states, getting rid of all states except the initial state & the final state.

3) With the 2 state super-NFA, write down the reg. expr.