Feb 3

Thm 1.54

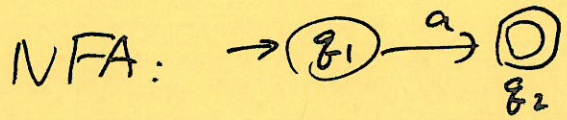
A language is regular iff some regular expression describes it.

"If part"

Lem 1.55 If a language is described by a regular expression then it is regular.

Proof: Let R be a regular expression, we will show how to convert R into an equivalent NFA.

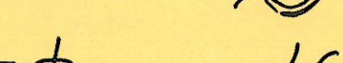
1. $R = a$, $a \in \Sigma$. $L(R) = \{a\}$

NFA: 

$N = (\{q_1, q_2\}, \Sigma, \delta, q_1, \{q_2\})$

$\delta(q_1, a) = \{q_2\}$.

2. $R = \varepsilon$, $L(R) = \{\varepsilon\}$.

NFA: 

3. $R = \phi$, $L(R) = \phi$

NFA: 

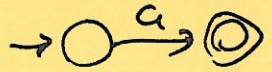
4. $R = R_1 \cup R_2$

5. $R = R_1 \circ R_2$

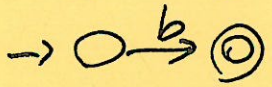
6. $R = R_1^*$

Ex. $(ab \cup a)^*$ $\Rightarrow$ NFA

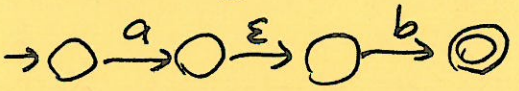
a



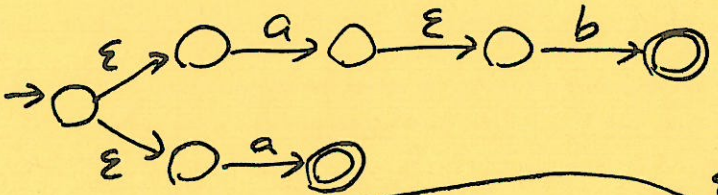
b



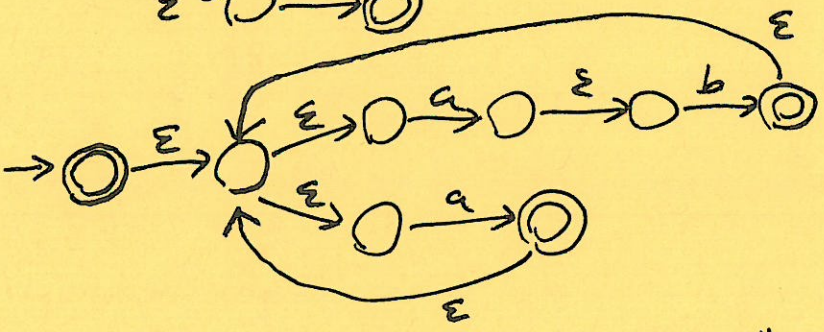
ab



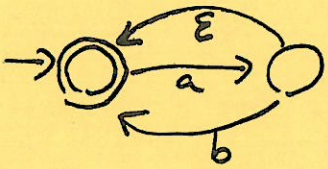
$ab \cup a$



$(ab \cup a)^*$



Can you design an NFA recognizing $(ab \cup a)^*$ with fewer states?



"only if" part

Lem 1.60 If a language is regular, then it is described by a regular expression.

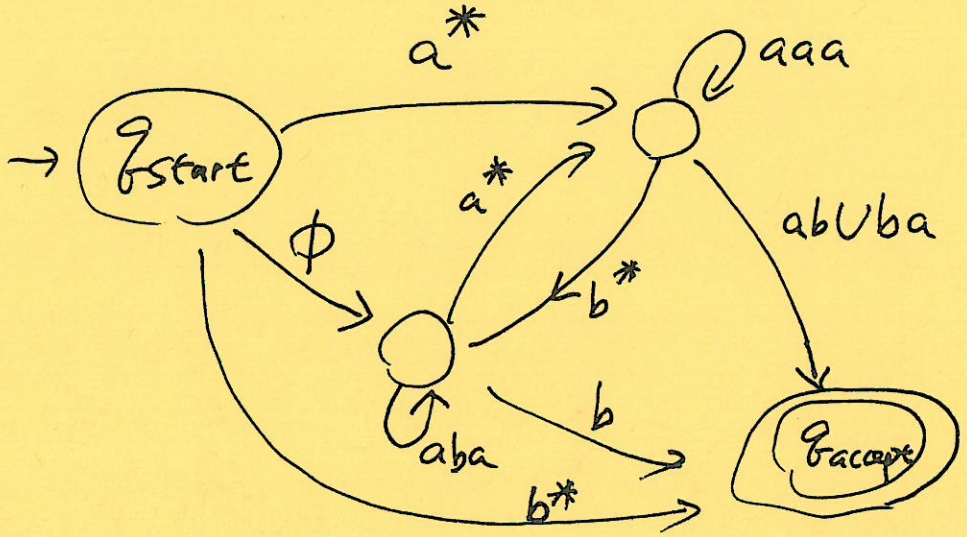
Proof: regular $\rightsquigarrow$ a DFA accepts it.

DFA $\rightsquigarrow$ expression.

Step 1: DFA $\rightarrow$ GNFA (generalized nondeterministic finite automaton)

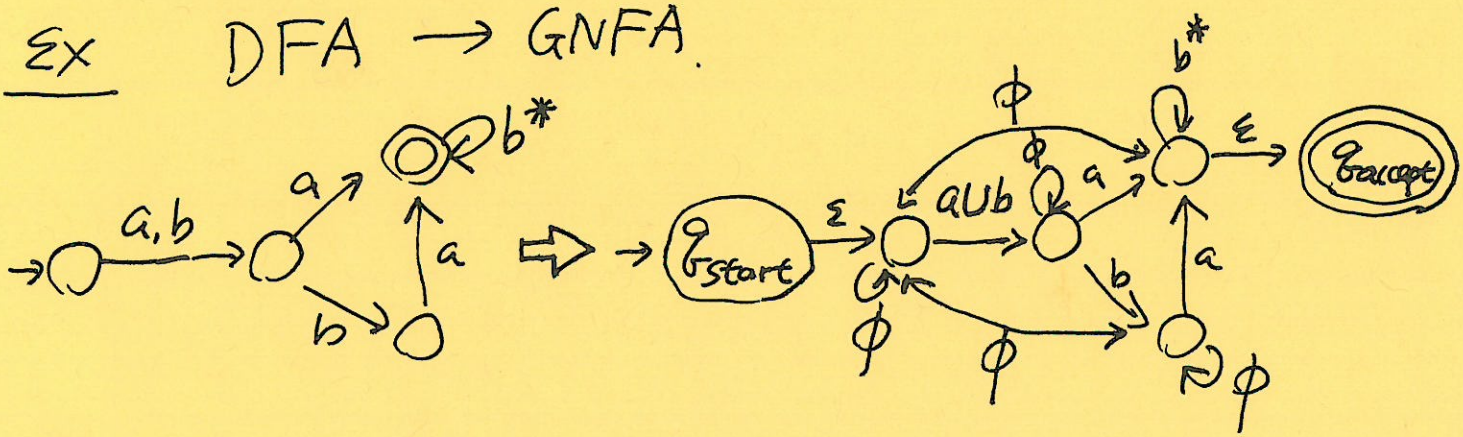
Step 2: GNFA $\rightarrow$ regular expression (with an algorithm).

What is a GNFA?

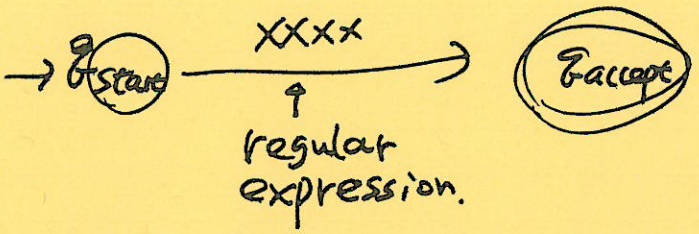


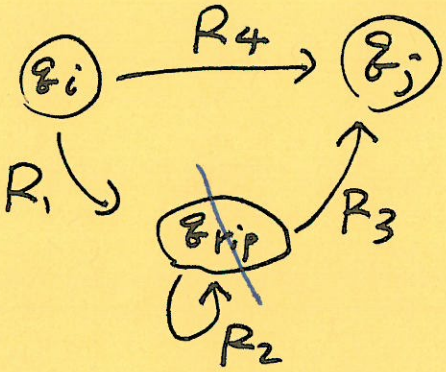
1. Arrows are described with strings instead of a letter.
2. The start state has arrows to all states, but not to itself.
3. The accept state has arrows coming from every other state but no arrows going to any other state.
4. $q_{start} \neq q_{accept}$
5. Except for q_{start}, q_{accept}, we have arrows between every pair of states.

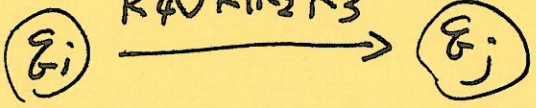
Ex DFA → GNFA.



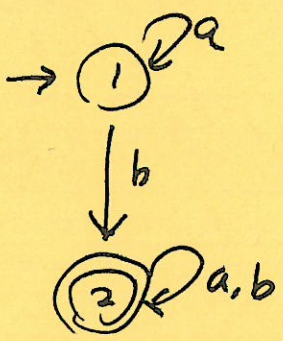
Step 2 GNFA → regular expression

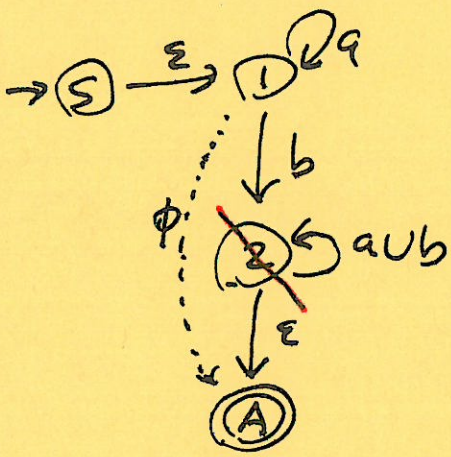


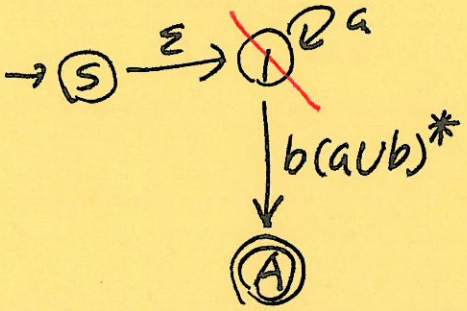


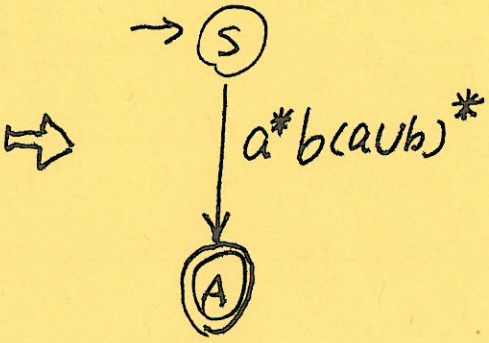


Example









$\phi \cup b(a \cup b)^* \varepsilon$

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