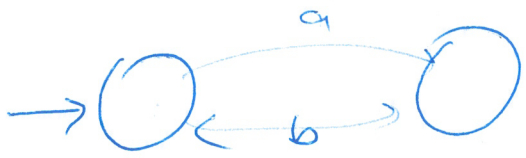
Last time: DFAs



$a^i b^i$

DFA

machines to process regular expressions

state based abstractions

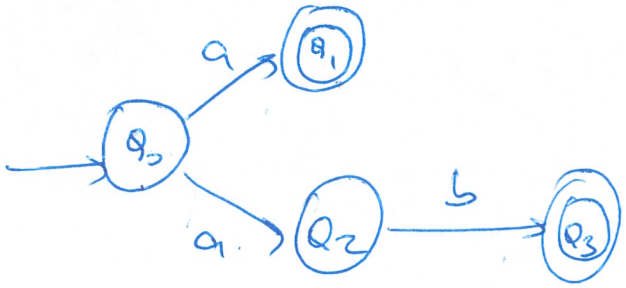
a ... | ab

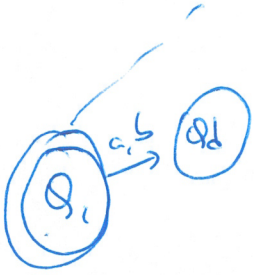
(ab | ab)

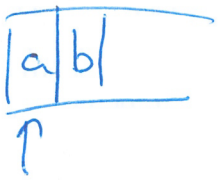
shopping

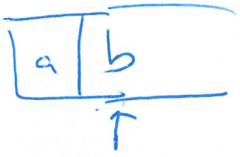
checkout

Nondeterministic finite automata
NFA

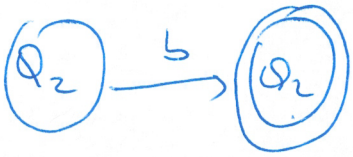


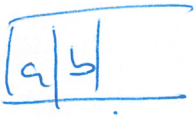






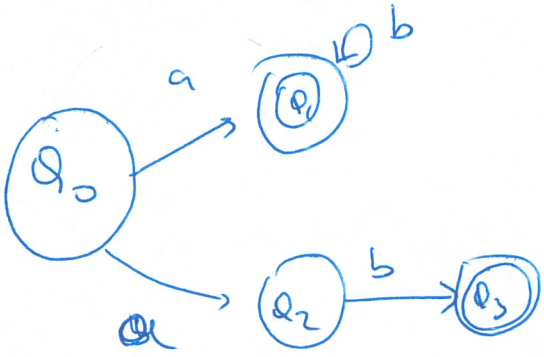
rejects





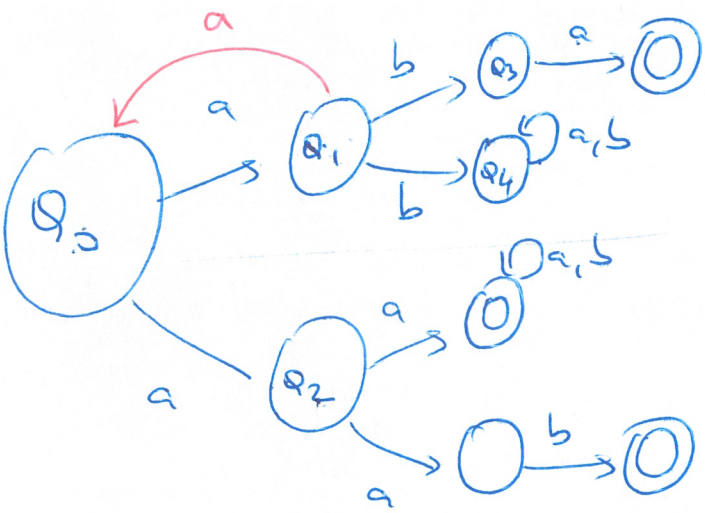
accepts

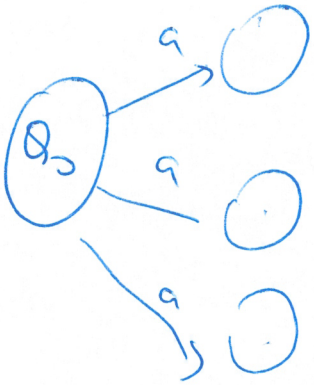
If a "path" leads to an accepting state then the NFA accepts.
Still the entire input has to be read.



ab accepts

accepts





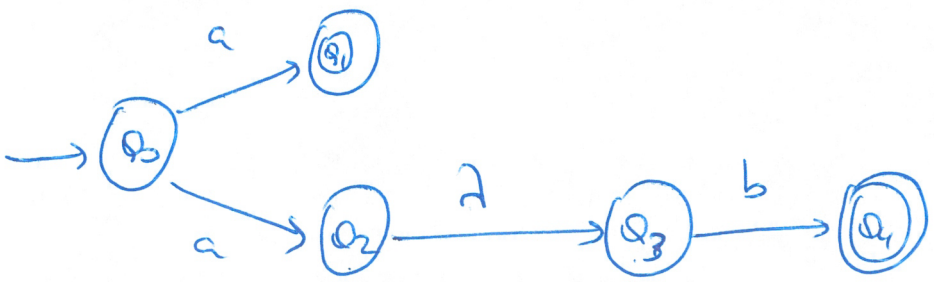
NFA $\longleftrightarrow$ DFA conversions.

A DFA is an NFA. (T) F

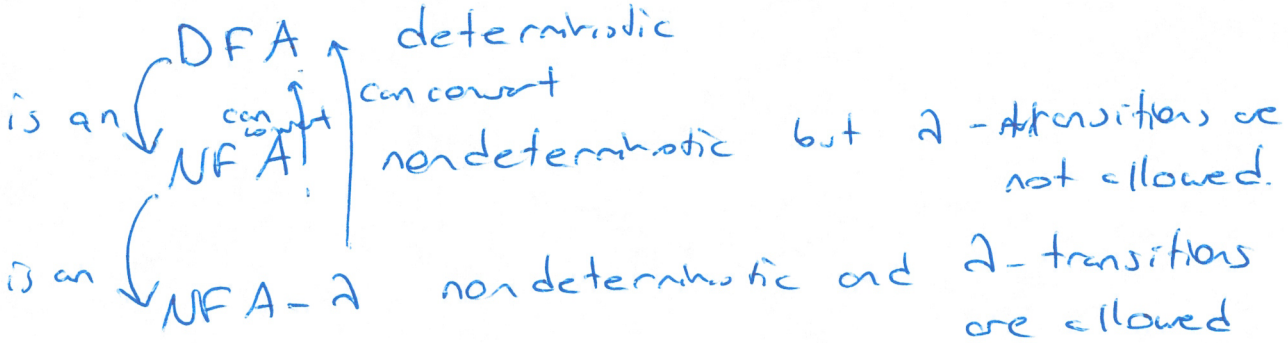
An NFA is a DFA : is not true in general.

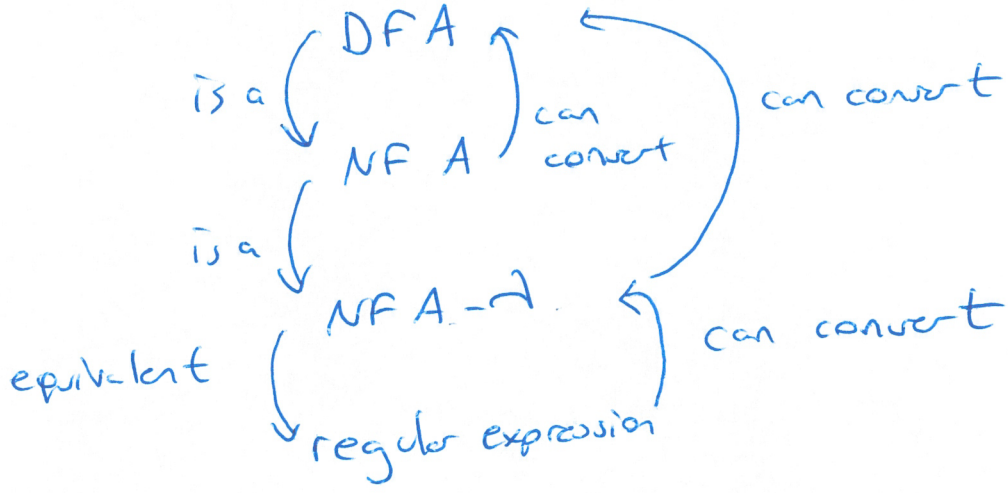
Can we convert an NFA into a DFA yes

wait for next week please

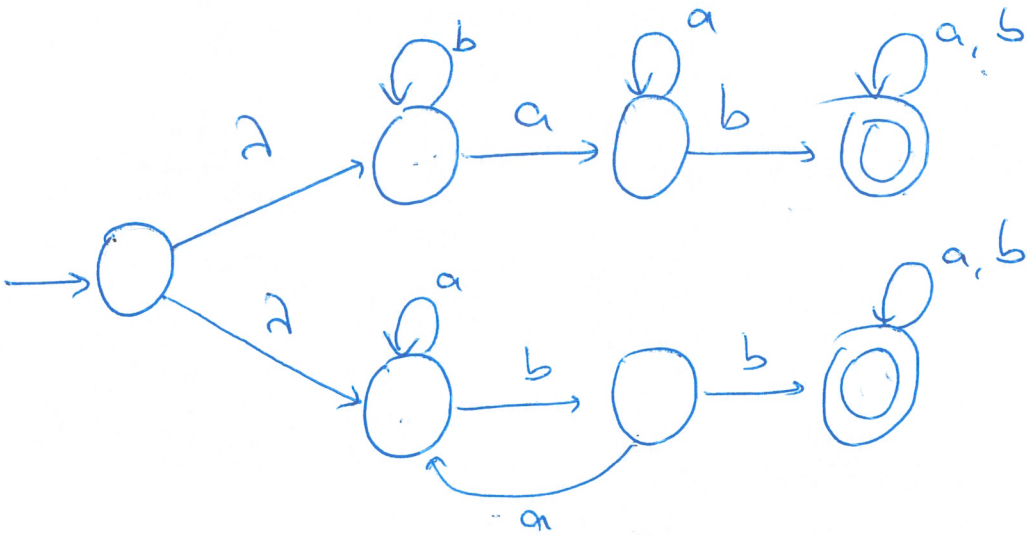

a
a
λ
b

NFA - λ


DFA
is an
can convert
NFA
deterministic
can convert
nondeterministic but λ-transitions are not allowed.
is an
NFA-λ
nondeterministic and λ-transitions are allowed


DFA
is a
NFA
can convert
can convert
is a
NFA-λ
equivalent
regular expression
can convert

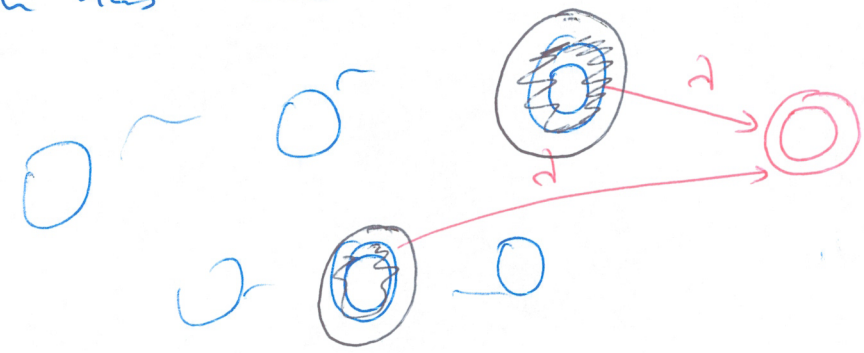
$L_1 = \{ w \mid w$ contains an "ab" or "bb" $\}$? (5)



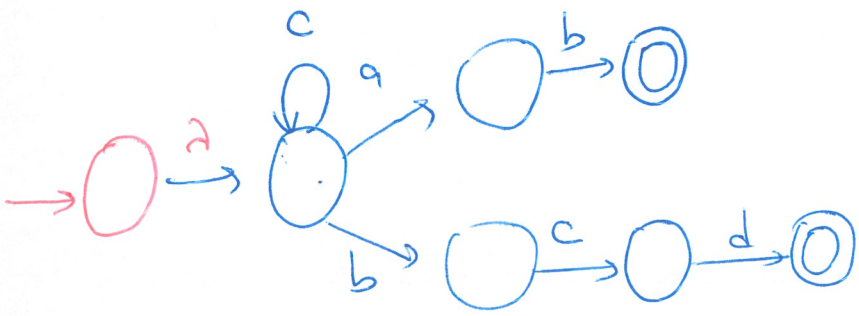
$$\left[(a \cup b)^* ab (a \cup b)^* \right] \cup \left[(a \cup b)^* bb (a \cup b)^* \right]$$

Consider an NFA-λ which has 2 accepting states.

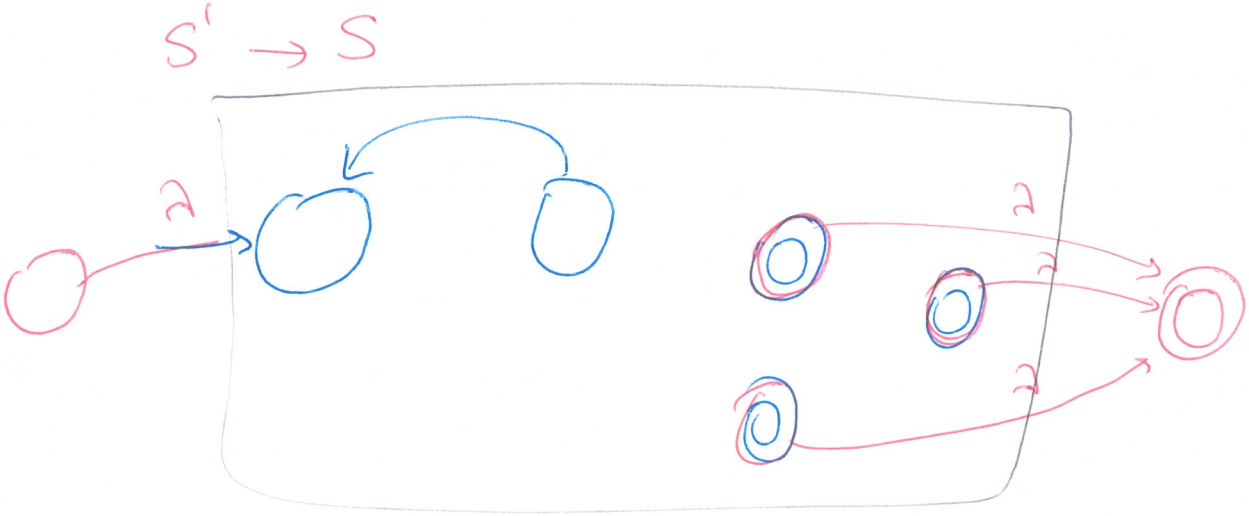
Can we convert this machine into an NFA-λ which has one accepting state?



maybe
yes
no
TGIF, idk, idc



We don't want incoming transitions to the start state.



Regular expression

a

λ]

ϕ]

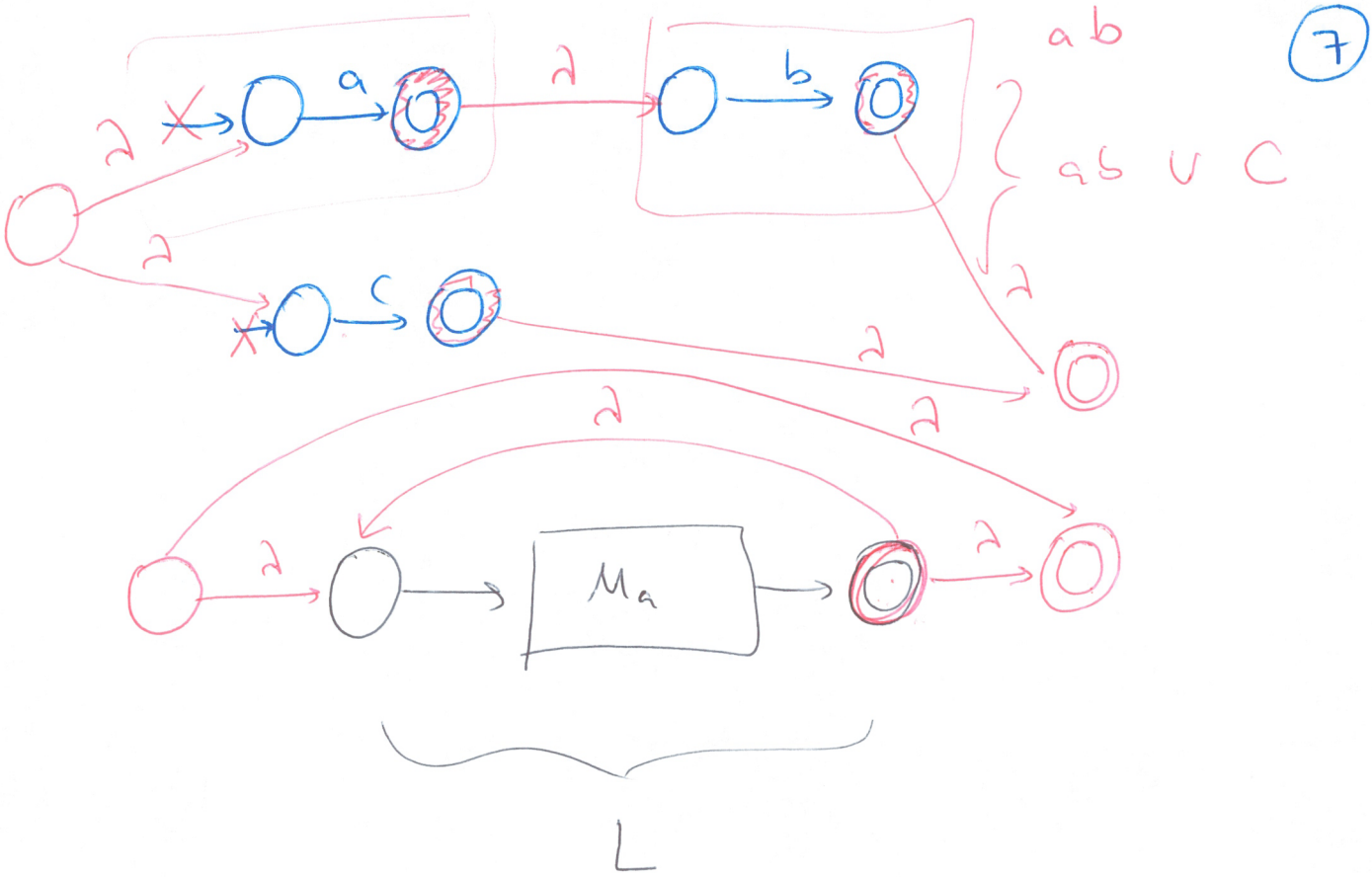
$r_1 \cup r_2$

$r_1 \cdot r_2$

r_1^*

$ab \cup c$

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

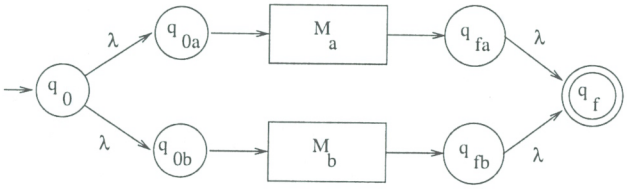

a
b
c
λ
λ
λ
λ
λ
λ
λ
λ
λ
M_a
ab
ab ∪ c
L

CS3311: How to convert a regular expression into an NFA-λ

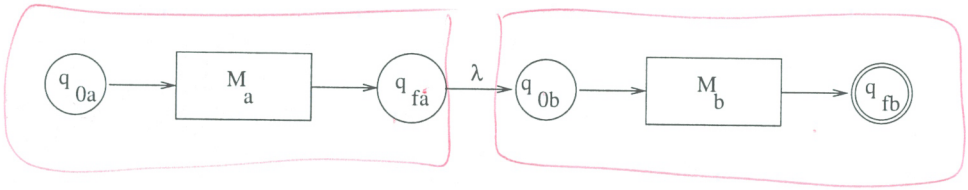
Theorem 5.5.3 Let M_1 and M_2 be two NFA-λs. There are NFA-λs that accept $L(M_1) \cup L(M_2)$, $L(M_1)L(M_2)$, and $L(M_1)^*$.

Consider two machines corresponding to two regular expressions:

The following machine represents the union of the two regular expressions:



The following machine represents the concatentation of the two regular expressions:



The following machine represents the Kleene star of a regular expression: