Unit-4: Regular properties

B. Srivathsan

Chennai Mathematical Institute

NPTEL-course

July - November 2015

Module 3:

Simple properties of finite automata

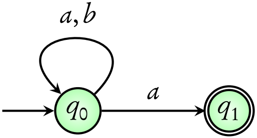
Determinization

Product construction

Emptiness

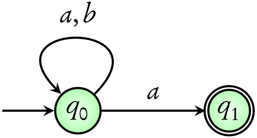
Complementation

Union



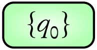
Non-deterministic automaton

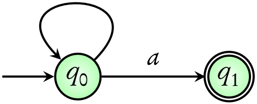
$\Sigma^* a$: words ending with an a



Non-deterministic automaton

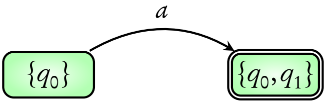
$\Sigma^* a$: words ending with an a

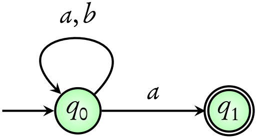




Non-deterministic automaton

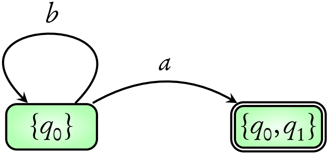
$\Sigma^* a$: words ending with an a

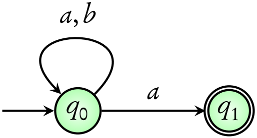




Non-deterministic automaton

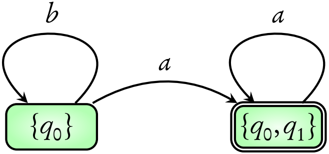
$\Sigma^* a$: words ending with an a

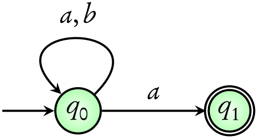




Non-deterministic automaton

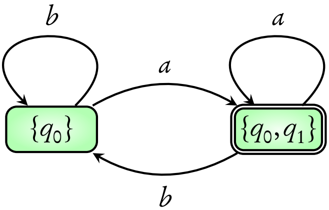
$\Sigma^* a$: words ending with an a

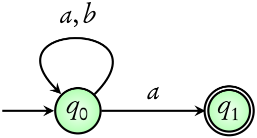




Non-deterministic automaton

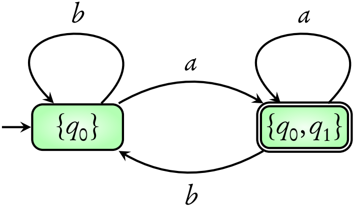
$\Sigma^* a$: words ending with an a

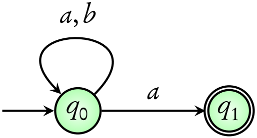




Non-deterministic automaton

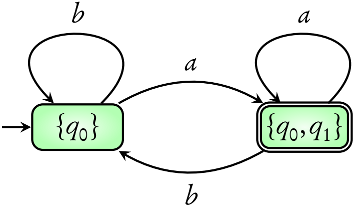
$\Sigma^* a$: words ending with an a



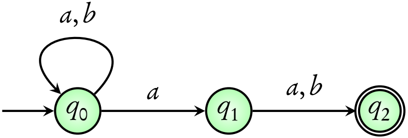


Non-deterministic automaton

$\Sigma^* a$: words ending with an a

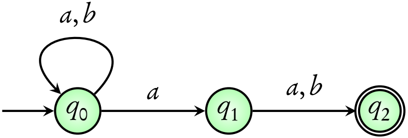


Deterministic automaton



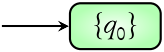
NFA

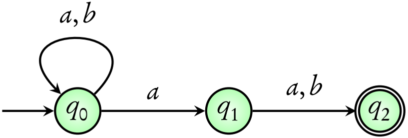
$\Sigma^* a \, \Sigma$: words where the second last letter is a



NFA

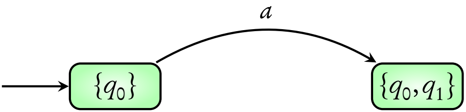
$\Sigma^* a\ \Sigma$: words where the second last letter is a

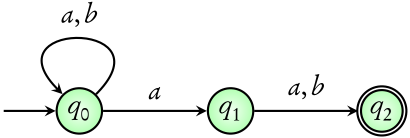




NFA

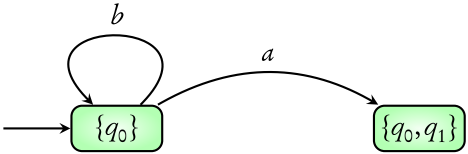
$\Sigma^* a \, \Sigma$: words where the second last letter is a

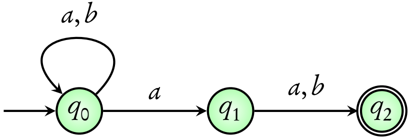




NFA

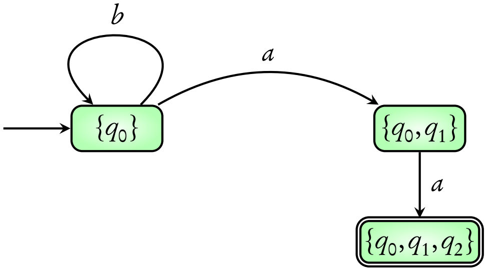
$\Sigma^* a \, \Sigma$: words where the second last letter is a

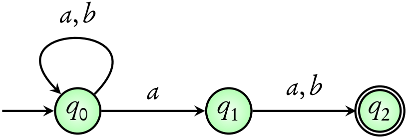




NFA

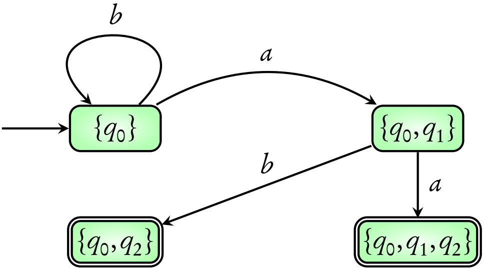
$\Sigma^* a \, \Sigma$: words where the second last letter is a

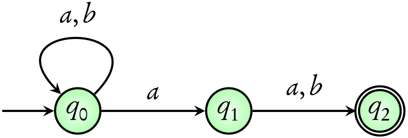




NFA

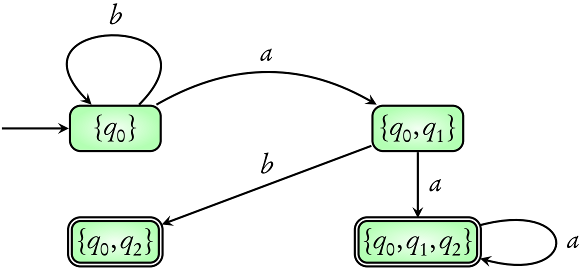
$\Sigma^* a\ \Sigma$: words where the second last letter is a

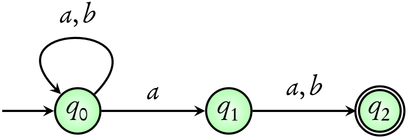




NFA

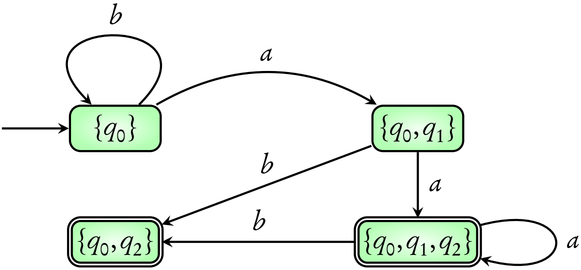
$\Sigma^* a\, \Sigma$: words where the second last letter is a

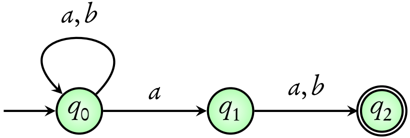




NFA

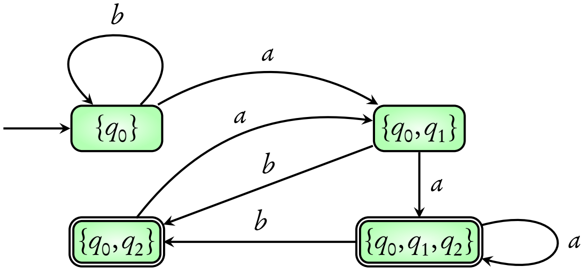
$\Sigma^* a \, \Sigma$: words where the second last letter is a

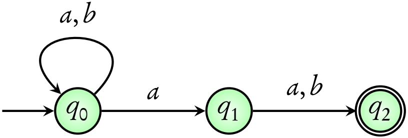




NFA

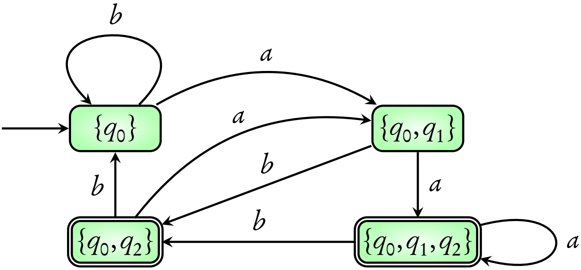
$\Sigma^* a\, \Sigma$: words where the second last letter is a

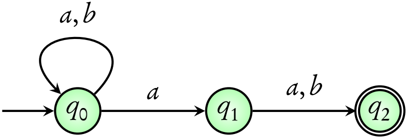




NFA

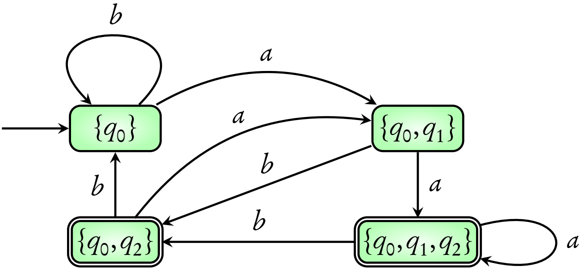
$\Sigma^* a\, \Sigma$: words where the second last letter is a



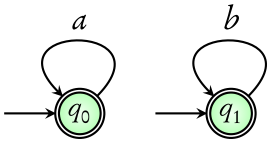


NFA

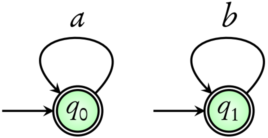
$\Sigma^* a\, \Sigma$: words where the second last letter is a



DFA

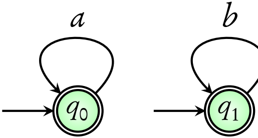


NFA



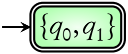
NFA

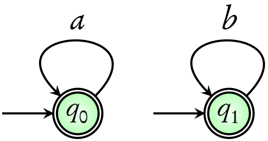
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ



NFA

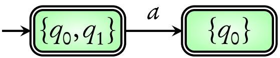
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ

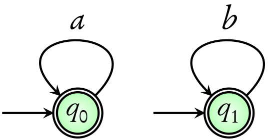




NFA

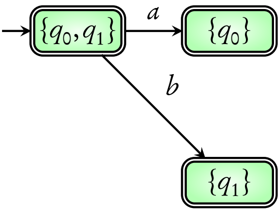
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ

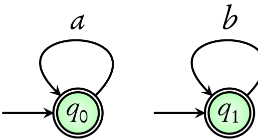




NFA

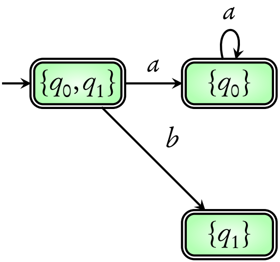
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ

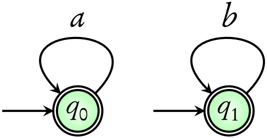




NFA

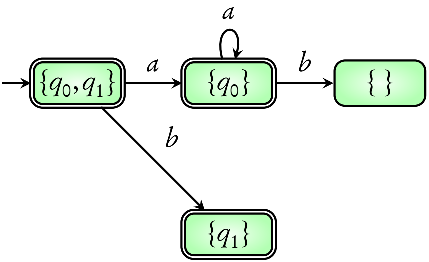
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ

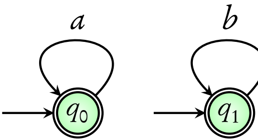




NFA

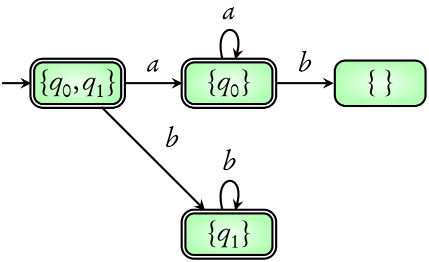
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ

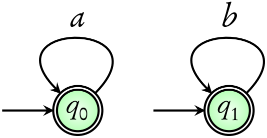




NFA

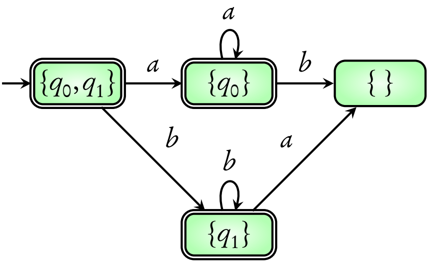
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ

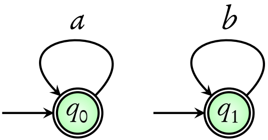




NFA

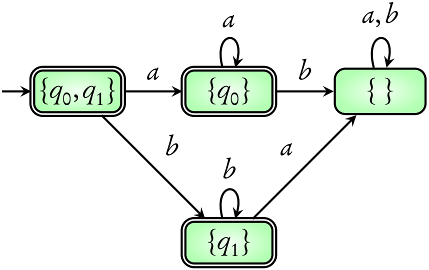
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ

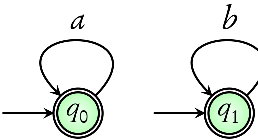




NFA

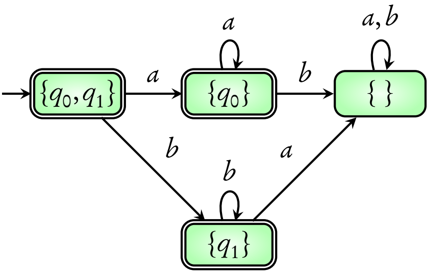
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ





NFA

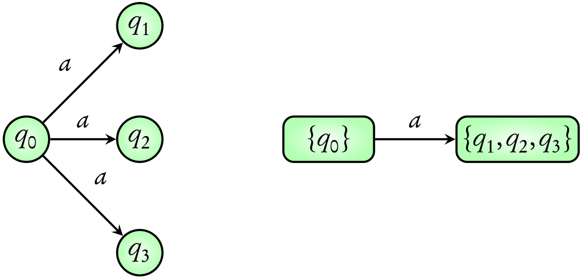
$a^* \cup b^*$: words of the form a^i, b^i, or ϵ



DFA

Subset construction

Every NFA can be converted to an **equivalent** DFA



Determinization

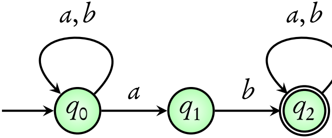
Subset construction

Product construction

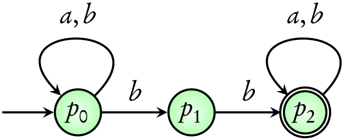
Emptiness

Complementation

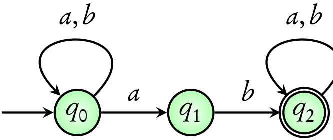
Union



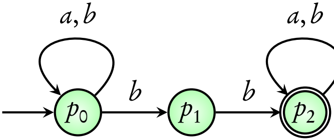
$\Sigma^* ab \Sigma^*$



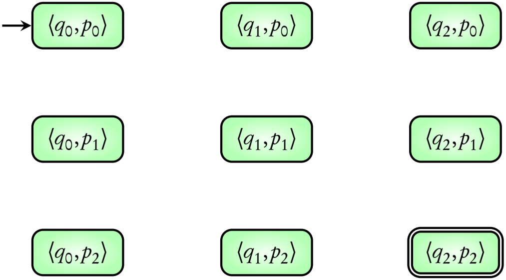
$\Sigma^* bb \Sigma^*$

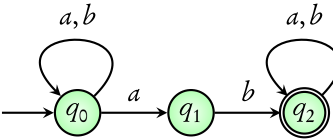


$\Sigma^* ab \Sigma^*$

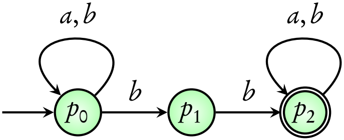


$\Sigma^* bb \Sigma^*$

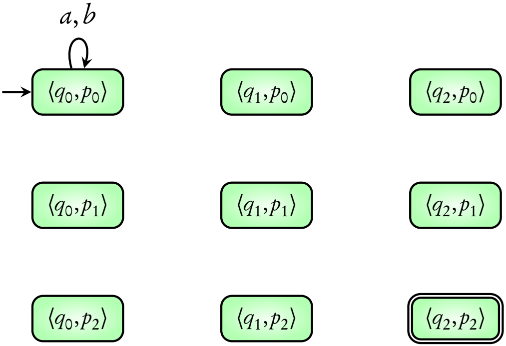


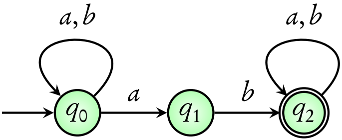


$\Sigma^* ab \Sigma^*$

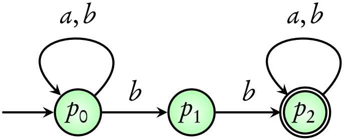


$\Sigma^* bb \Sigma^*$

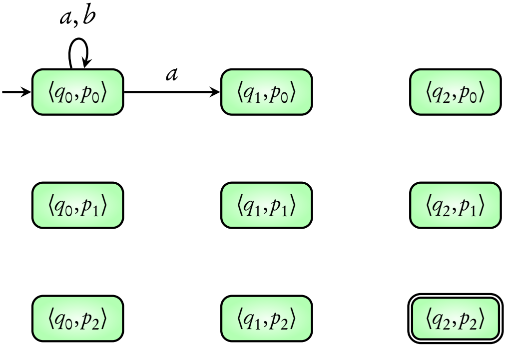


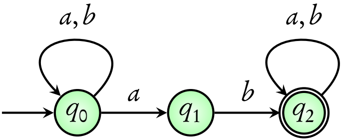


$\Sigma^* ab \Sigma^*$

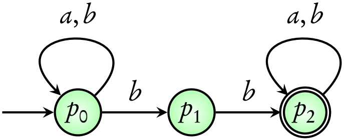


$\Sigma^* bb \Sigma^*$

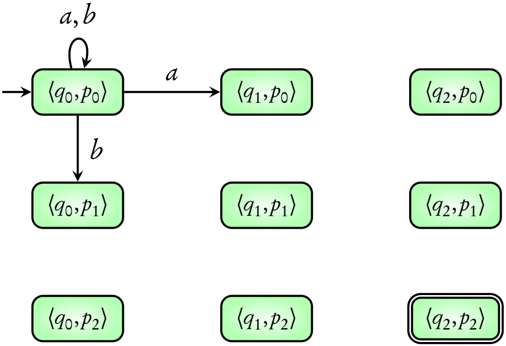


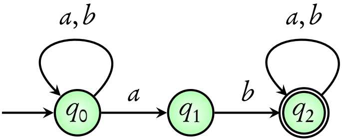


$\Sigma^* ab \Sigma^*$

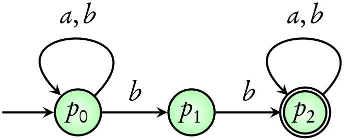


$\Sigma^* bb \Sigma^*$

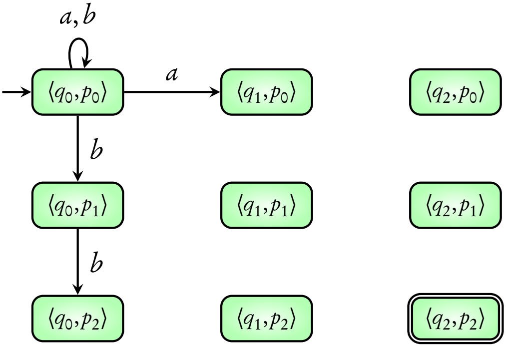


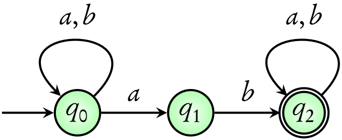


$\Sigma^* ab \Sigma^*$

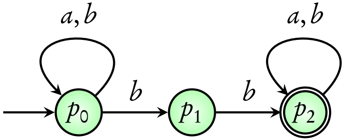


$\Sigma^* bb \Sigma^*$

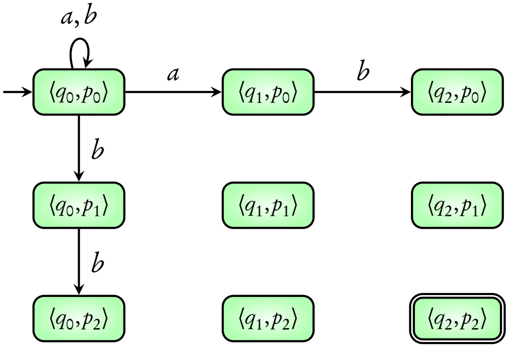


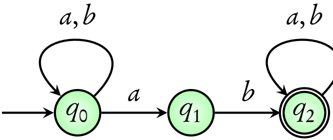


$\Sigma^* ab \Sigma^*$

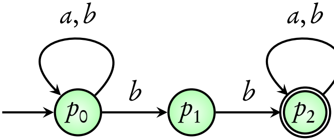


$\Sigma^* bb \Sigma^*$

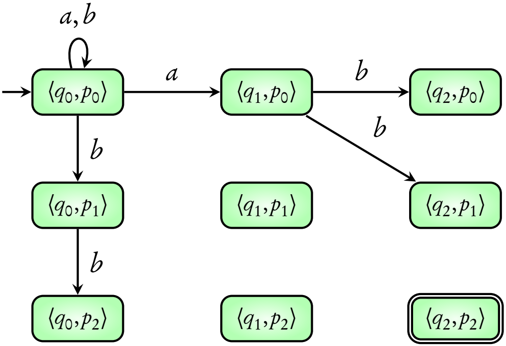


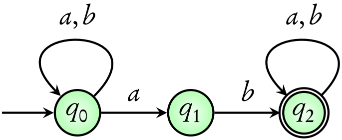


$\Sigma^* ab \Sigma^*$

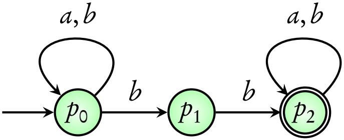


$\Sigma^* bb \Sigma^*$

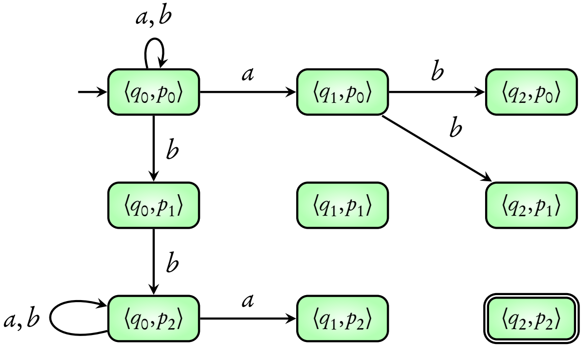


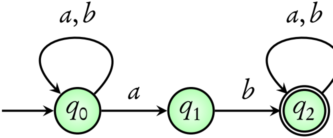


$\Sigma^* ab \Sigma^*$

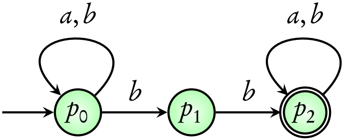


$\Sigma^* bb \Sigma^*$

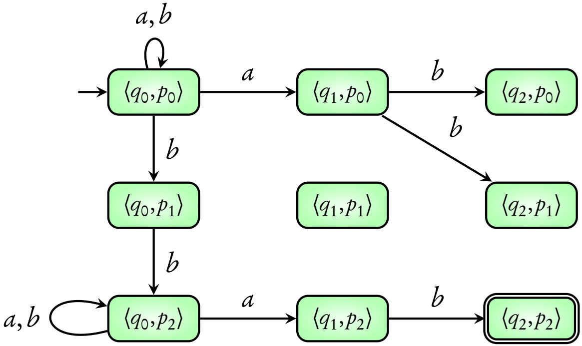


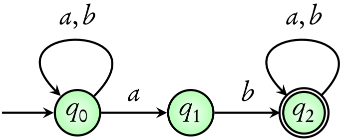


$\Sigma^* ab \Sigma^*$

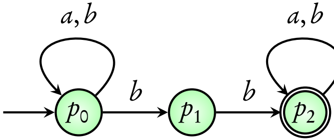


$\Sigma^* bb \Sigma^*$

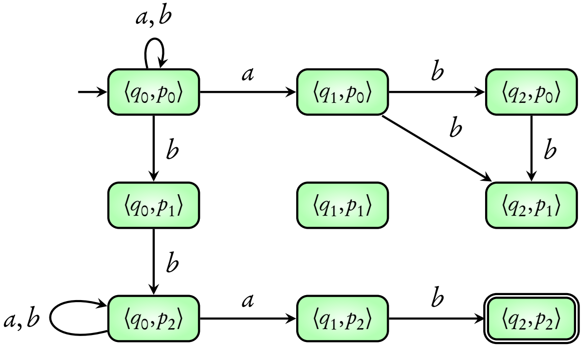


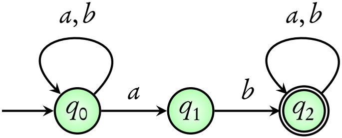


$\Sigma^* ab \Sigma^*$

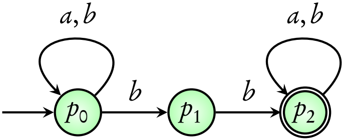


$\Sigma^* bb \Sigma^*$

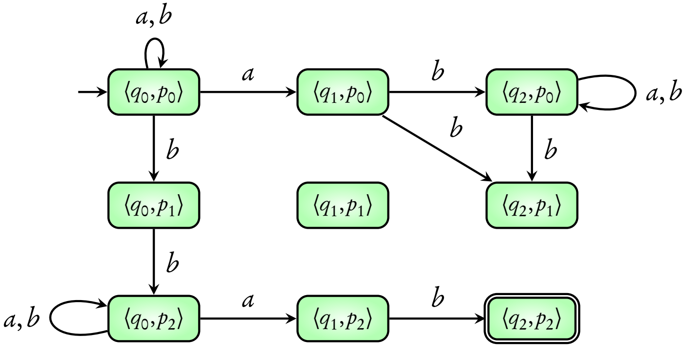


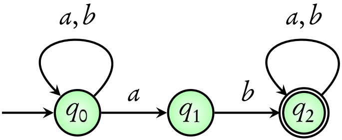


$\Sigma^* ab \Sigma^*$

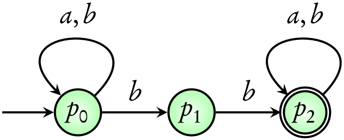


$\Sigma^* bb \Sigma^*$

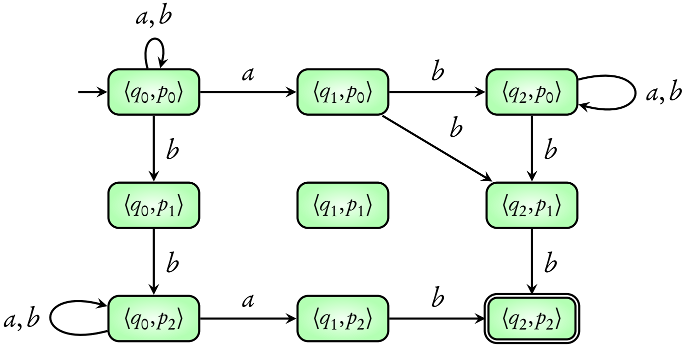


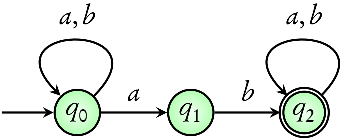


$\Sigma^* ab \Sigma^*$

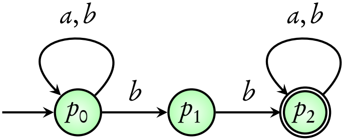


$\Sigma^* bb \Sigma^*$

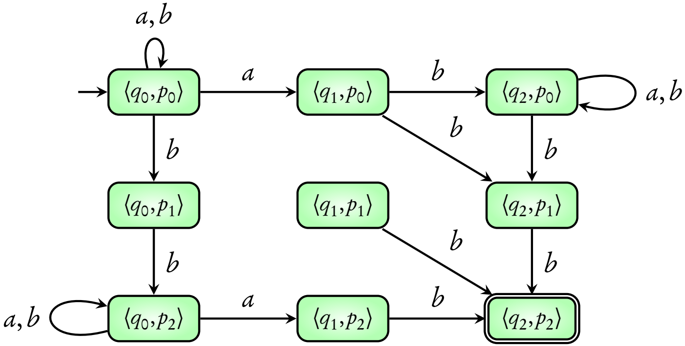


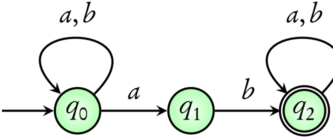


$\Sigma^* ab \Sigma^*$

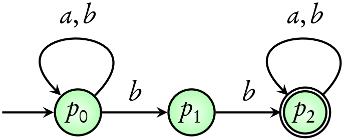


$\Sigma^* bb \Sigma^*$

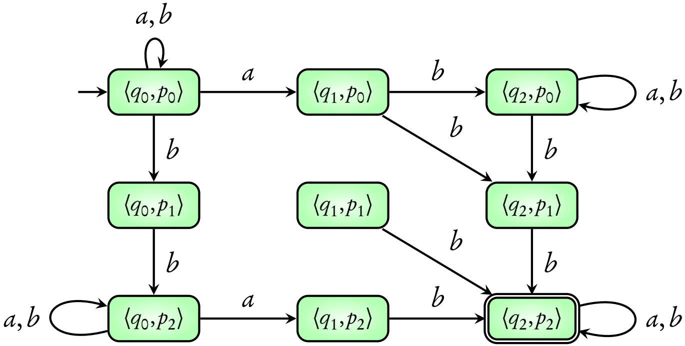


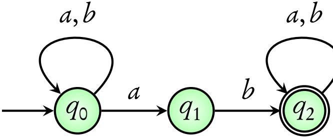


$\Sigma^* ab \Sigma^*$

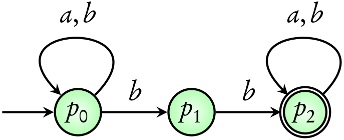


$\Sigma^* bb \Sigma^*$

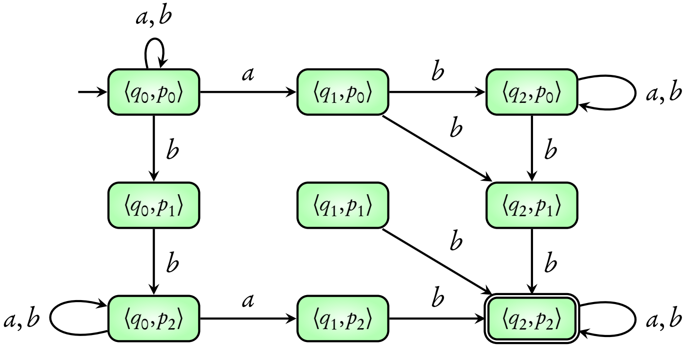




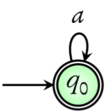
$\Sigma^* ab \Sigma^*$



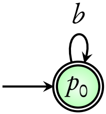
$\Sigma^* bb \Sigma^*$



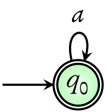
$\Sigma^* ab \Sigma^* \cap \Sigma^* bb \Sigma^*$: words containing both *ab* and *bb*



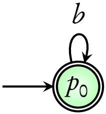
a^*



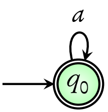
b^*



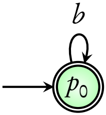
a^*



b^*



a^*



b^*

$$a^* \cap b^* = \{ \epsilon \}$$

Synchronous product

Gives the **intersection** of the two languages

Determinization

Subset construction

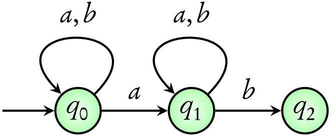
Product construction

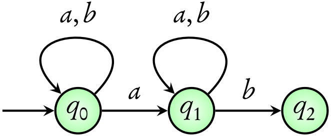
Intersection of languages

Emptiness

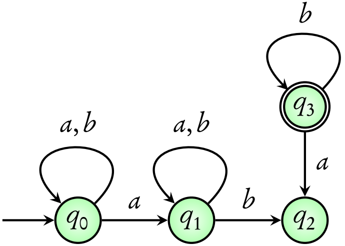
Complementation

Union

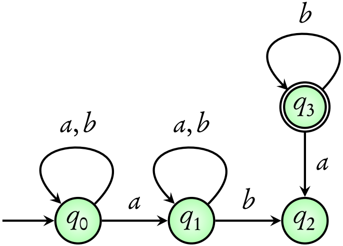

a, b
a, b
a
b
q0
q1
q2



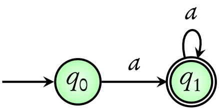
Language is empty as there is no accepting state



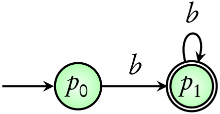
a, b
a, b
b
q_0
a
q_1
b
q_2
q_3
a



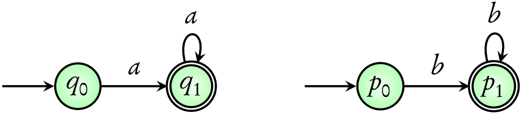
Language is empty as accepting state is **not reachable**



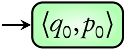
aa^*

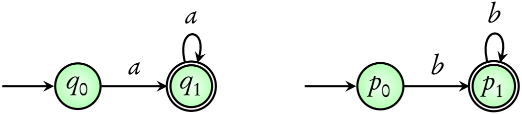


bb^*



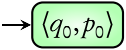
aa^* bb^*





aa^*

bb^*



Language is empty as there is **no accepting state**

Question: Given NFA $\mathscr{A}$, is language accepted by $\mathscr{A}$ empty?

Question: Given NFA $\mathscr{A}$, is language accepted by $\mathscr{A}$ empty?

Emptiness of NFA

Language of an NFA is empty if and only if it has
no reachable accepting states

Question: Given NFA $\mathscr{A}$, is language accepted by $\mathscr{A}$ empty?

Emptiness of NFA

Language of an NFA is empty if and only if it has **no reachable accepting states**

Algorithm

Run a **depth-first or breadth-first search** to find if there is a path to an accepting state

Determinization

Subset construction

Product construction

Intersection of languages

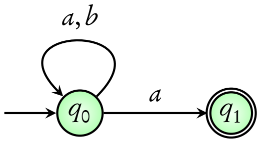
Emptiness

Algorithm for emptiness

Complementation

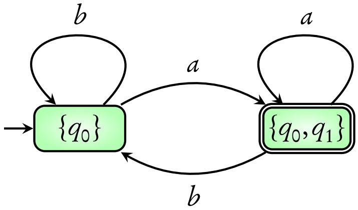
Union

NFA

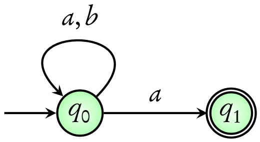


$\Sigma^* a$: words ending with an a

DFA

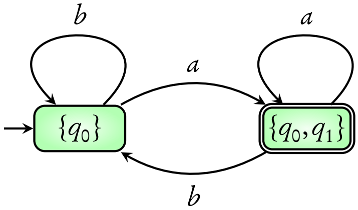


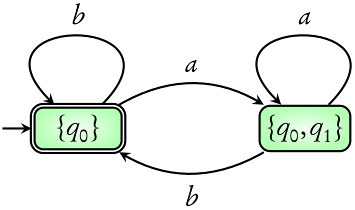
NFA



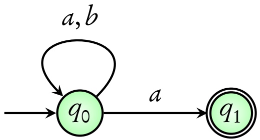
$\Sigma^* a$: words ending with an a

DFA



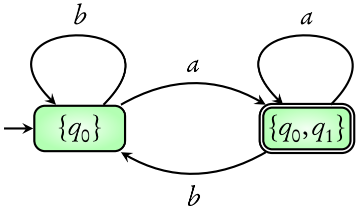


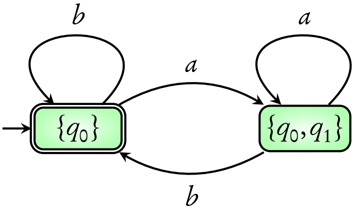
NFA



$\Sigma^* a$: words ending with an a

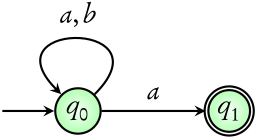
DFA

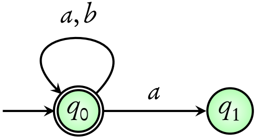




complement of $\Sigma^* a$

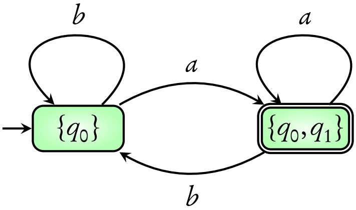
NFA

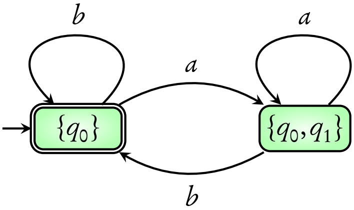




$\Sigma^* a$: words ending with an a

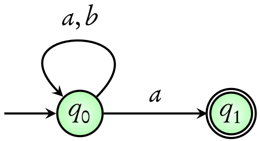
DFA

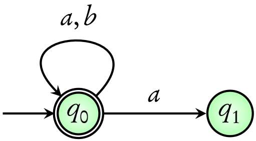




complement of $\Sigma^* a$

NFA

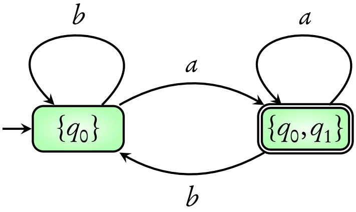


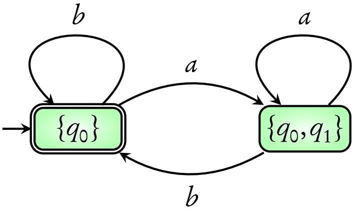


$\Sigma^* a$: words ending with an a

not the complement!

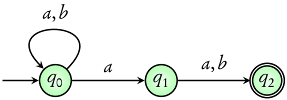
DFA





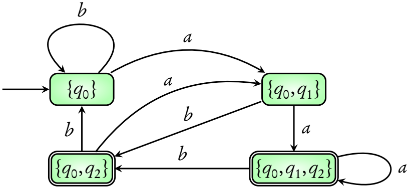
complement of $\Sigma^* a$

NFA

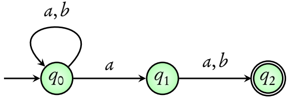


$\Sigma^* a \, \Sigma$: words where the second last letter is a

DFA

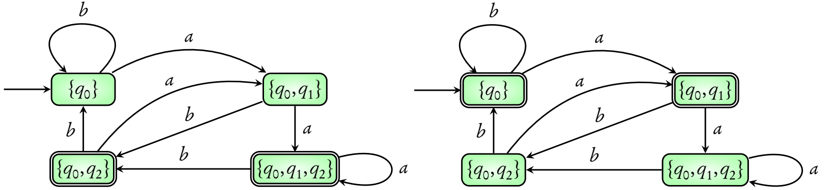


NFA

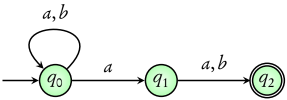


$\Sigma^* a \, \Sigma$: words where the second last letter is a

DFA

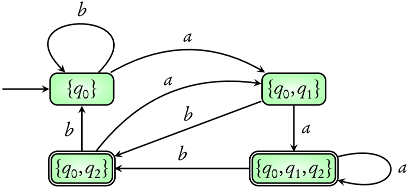


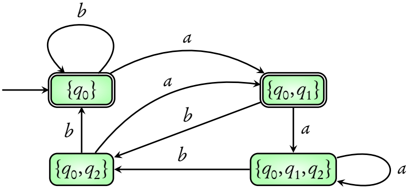
NFA



$\Sigma^* a \, \Sigma$: words where the second last letter is a

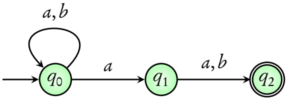
DFA

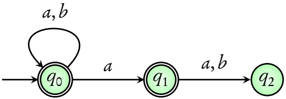




complement of $\Sigma^* a \Sigma$

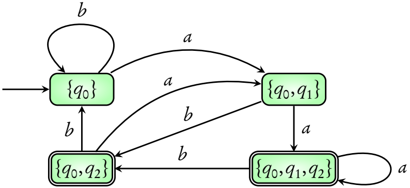
NFA

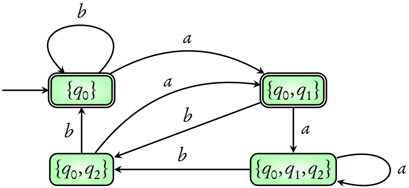




$\Sigma^* a\ \Sigma$: words where the second last letter is a

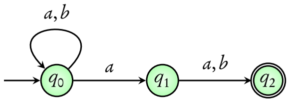
DFA

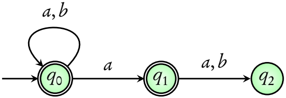




complement of $\Sigma^* a \Sigma$

NFA

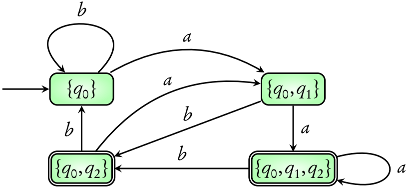


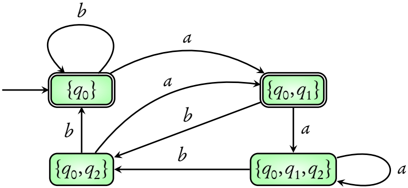


$\Sigma^* a \, \Sigma$: words where the second last letter is a

not the complement!

DFA





complement of $\Sigma^* a \Sigma$

Complementation

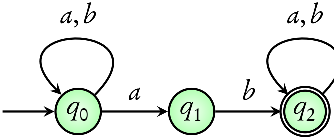
Interchange accepting and non-accepting states in a DFA

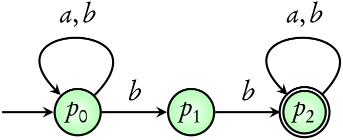
Complementation

Interchange accepting and non-accepting states in a DFA

Does not work in the case of NFA

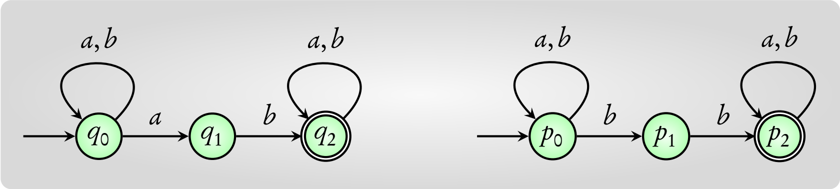
Coming next: Union of two regular languages



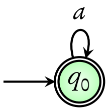


$\Sigma^* ab \Sigma^*$

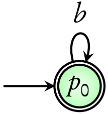
$\Sigma^* bb \Sigma^*$



$\Sigma^* ab \Sigma^*$ $\cup$ $\Sigma^* bb \Sigma^*$



a^*



b^*

a^* $\cup$ b^*

Union

Consider the two automata as a **single automaton**

Determinization

Subset construction

Product construction

Intersection of languages

Emptiness

Algorithm for emptiness

Complementation

Union