Theoretical Computer Science

Tutorial Week 10

Prof. Andrey Frolov

INNOPOLIS UNIVERSITY

Agenda

Non-determinism:

- **Non-deterministic FSA (NDFSA)**
- Examples
- NDFSA to DFSA

Regular Expressions (RegExp)

- Definition
- RegExp to (N)FSA
- FSA to RegExp

Non-deterministic Finite State Automata (NDFSA)

Definition: NDFSA

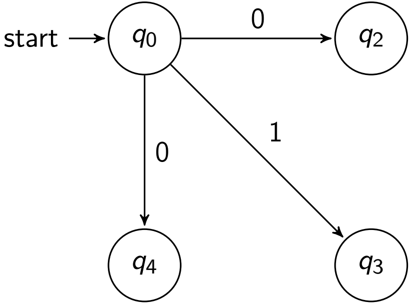
A NDFSA is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where Q, Σ, q_0, A are defined as in (D)FSA and the transition function is defined as

$$\delta : Q \times \Sigma \to \mathbb{P}(Q)$$

$\mathbb{P}$ is the powerset function (i.e., the set of all possible subsets)

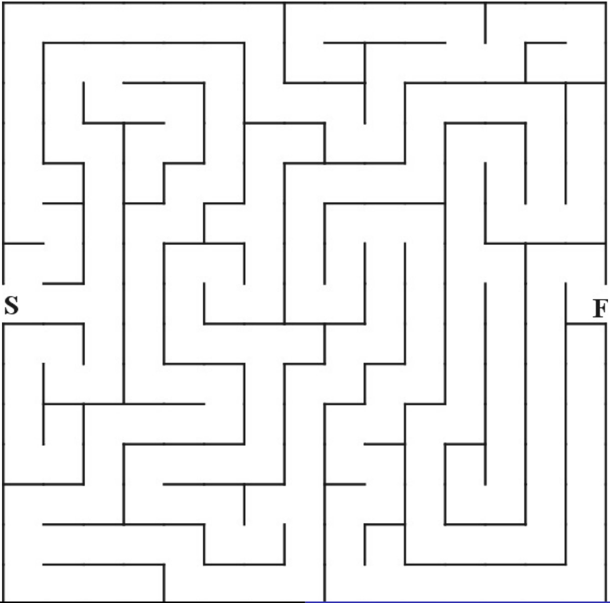
A NDFSA modifies the definition of a FSA to permit transitions at each stage to either zero, one, or more than one states.

Example



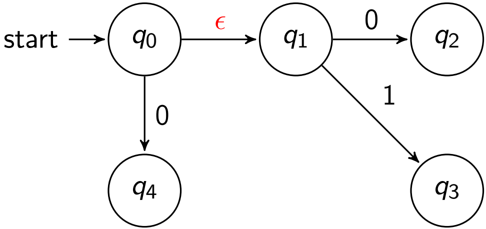
δ	0	1
$\rightarrow q_0$	$\{q_2, q_4\}$	q_3
q_1	$\emptyset$	$\emptyset$
q_2	$\emptyset$	$\emptyset$
q_3	$\emptyset$	$\emptyset$

Maze analogy



NDFSA with ϵ

What about ϵ-transition???



NDFSA with ϵ

Could we add ϵ?

Definition: NDFSA

A NDFSA is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where Q, Σ, q_0, A are defined as in (D)FSA and the transition function is defined as

$$\delta : Q \times \Sigma \cup \{\epsilon\}??? \rightarrow \mathbb{P}(Q)$$

$\mathbb{P}$ is the powerset function (i.e., the set of all possible subsets)

NDFSA with ϵ

Could we add ϵ?

Definition: NDFSA

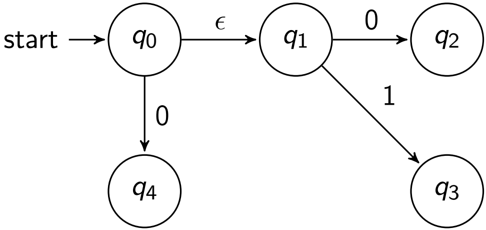
A NDFSA is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where Q, Σ, q_0, A are defined as in (D)FSA and the transition function is defined as

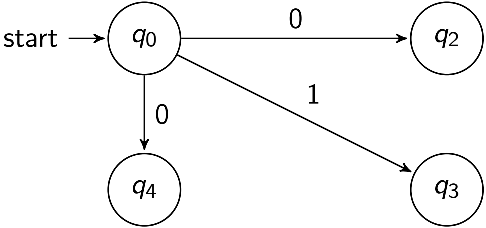
$$\delta : Q \times \Sigma \cup \{\epsilon\} ??? \rightarrow \mathbb{P}(Q)$$

$\mathbb{P}$ is the powerset function (i.e., the set of all possible subsets)

Yes, but it is not necessary!

Example with ϵ





Agenda

Non-determinism:

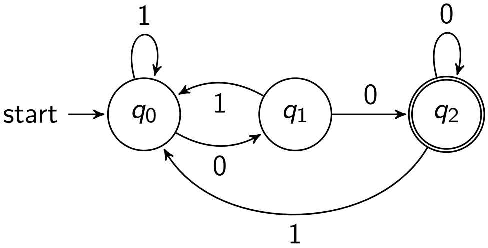
- Non-deterministic FSA (NDFSA)
- **Examples**
- NDFSA to DFSA

Regular Expressions (RegExp)

- Definition
- RegExp to (N)FSA
- FSA to RegExp

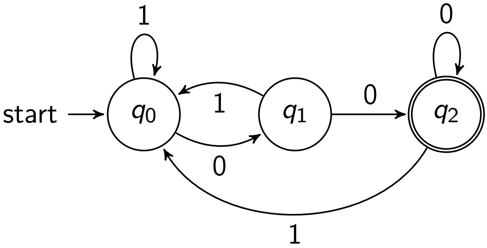
FSA vs NDFSA

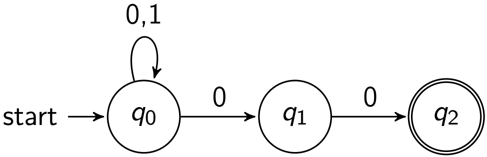
The FSA and NDFSA accepting strings ending with 00



FSA vs NDFSA

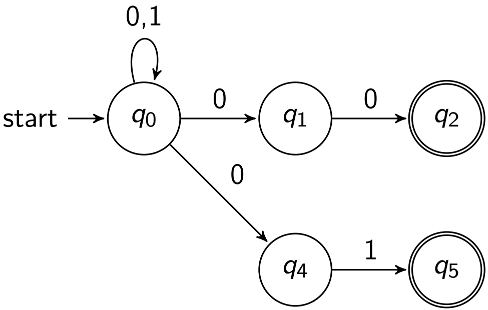
The FSA and NDFSA accepting strings ending with 00





Example

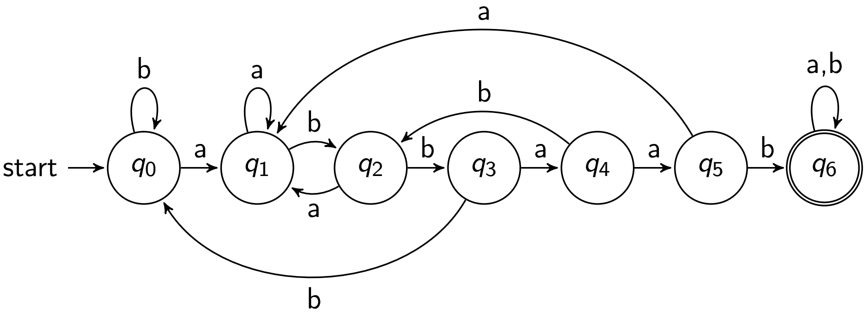
The NDFSA accepting strings ending with 00 or 01



FSA vs NDFSA

Let Σ be the alphabet $\Sigma = \{a, b\}$

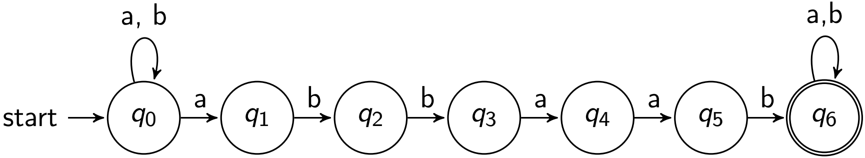
- $L = \{x \in \Sigma^* \mid \; x \text{ contains the substring } abbaab\}$;



FSA vs NDFSA

Let Σ be the alphabet $\Sigma = \{a, b\}$

- $L = \{x \in \Sigma^* \mid \; x \text{ contains the substring } abbaab\}$;



Agenda

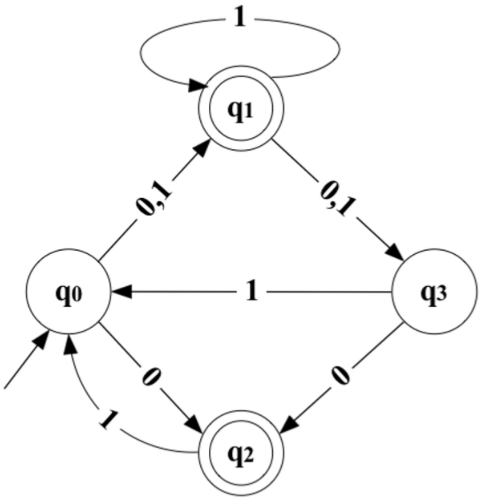
Non-determinism:

- Non-deterministic FSA (NDFSA)
- Examples
- **NDFSA to DFSA**

Regular Expressions (RegExp)

- Definition
- RegExp to (N)FSA
- FSA to RegExp

NDFSA to FSA: Example



NDFSA to FSA: Example

First, we build the transition table of the NDFSA:

δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$
$^*q_1$	$\{q_3\}$	$\{q_1, q_3\}$
$^*q_2$	$\emptyset$	$\{q_0\}$
q_3	$\{q_2\}$	$\{q_0\}$

NDFSA to FSA: Example

δ	0	1

NDFSA to FSA: Example

δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$

NDFSA to FSA: Example

δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$
${}^*\{q_1, q_2\}$	$\{q_3\}$	$\{q_0, q_1, q_3\}$

NDFSA to FSA: Example

δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$
$^*\{q_1, q_2\}$	$\{q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1\}$	$\{q_3\}$	$\{q_1, q_3\}$

NDFSA to FSA: Example

δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$
$^*\{q_1, q_2\}$	$\{q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1\}$	$\{q_3\}$	$\{q_1, q_3\}$
$\{q_3\}$	$\{q_2\}$	$\{q_0\}$

NDFSA to FSA: Example

δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$
$^*\{q_1, q_2\}$	$\{q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1\}$	$\{q_3\}$	$\{q_1, q_3\}$
$\{q_3\}$	$\{q_2\}$	$\{q_0\}$
$^*\{q_0, q_1, q_3\}$	$\{q_1, q_2, q_3\}$	$\{q_0, q_1, q_3\}$

NDFSA to FSA: Example

δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$
$^*\{q_1, q_2\}$	$\{q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1\}$	$\{q_3\}$	$\{q_1, q_3\}$
$\{q_3\}$	$\{q_2\}$	$\{q_0\}$
$^*\{q_0, q_1, q_3\}$	$\{q_1, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1, q_3\}$	$\{q_2, q_3\}$	$\{q_0, q_1, q_3\}$

NDFSA to FSA: Example

δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$
$^*\{q_1, q_2\}$	$\{q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1\}$	$\{q_3\}$	$\{q_1, q_3\}$
$\{q_3\}$	$\{q_2\}$	$\{q_0\}$
$^*\{q_0, q_1, q_3\}$	$\{q_1, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1, q_3\}$	$\{q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_2\}$	$\emptyset$	$\{q_0\}$

NDFSA to FSA: Example

δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$
$^*\{q_1, q_2\}$	$\{q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1\}$	$\{q_3\}$	$\{q_1, q_3\}$
$\{q_3\}$	$\{q_2\}$	$\{q_0\}$
$^*\{q_0, q_1, q_3\}$	$\{q_1, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1, q_3\}$	$\{q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_2\}$	$\emptyset$	$\{q_0\}$
$^*\{q_1, q_2, q_3\}$	$\{q_2, q_3\}$	$\{q_0, q_1, q_3\}$

NDFSA to FSA: Example

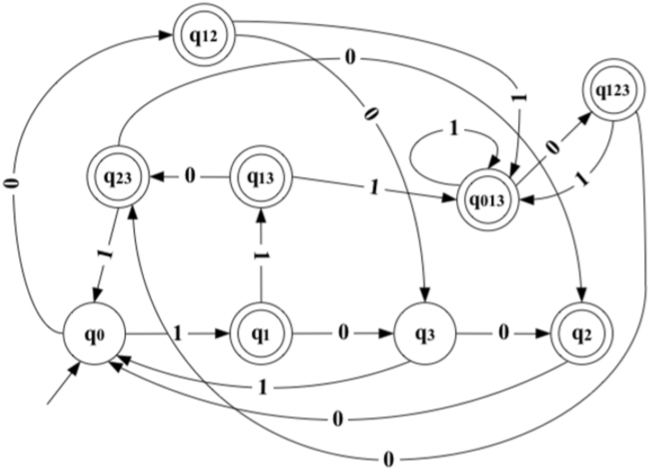
δ	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_1\}$
$^*\{q_1, q_2\}$	$\{q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1\}$	$\{q_3\}$	$\{q_1, q_3\}$
$\{q_3\}$	$\{q_2\}$	$\{q_0\}$
$^*\{q_0, q_1, q_3\}$	$\{q_1, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_1, q_3\}$	$\{q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_2\}$	$\emptyset$	$\{q_0\}$
$^*\{q_1, q_2, q_3\}$	$\{q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$^*\{q_2, q_3\}$	$\{q_2\}$	$\{q_0\}$

NDFSA to FSA: Example

δ	0	1
$\rightarrow$q0	q12	q1
*q12	q3	q013
*q1	q3	q13
q3	q2	q0
*q013	q123	q013
*q13	q23	q013
*q2	$\emptyset$	q0
*q123	q23	q013
*q23	q2	q0

NDFSA to FSA: Example

Finally, we can build the resulting DFSA:



Algorithm for NDFSA to DFSA

1. Create state table from the given NDFA
2. Create a blank state table under possible input alphabets for the equivalent DFA
3. Mark the start state of the DFA by $\{q_0\}$ (the same as the NDFA)
4. Find out the combination of States $q_0, q_1, ..., q_n$ for each possible input alphabet
5. Each time we generate a new DFA state under the input alphabet columns, we have to apply step 4 again, otherwise go to step 6
6. The states which contain any of the accepting states of the NDFA are the accepting states of the equivalent DFA

Agenda

Non-determinism:

- Non-deterministic FSA (NDFSA)
- NDFSA to DFSA

Regular Expressions (RegExp)

- **Definition**
- RegExp to (N)FSA
- FSA to RegExp

Regular Expressions (RegExp): Definition

Singletons

$$\epsilon = \{\epsilon\}$$

$$0 = \{0\}$$

$$1 = \{1\}$$

Regular Expressions (RegExp): Definition

Union

$$S_1 \mid S_2 = S_1 \cup S_2 = \{s \mid s \in S_1 \ \lor \ s \in S_2\}$$

Examples

$\epsilon \mid a = \{\epsilon, a\}$

$0 \mid 1 = \{0, 1\}$

Regular Expressions (RegExp): Definition

Concatenation

$$S_1 \cdot S_2 = \{s_1.s_2 \mid s_1 \in S_1 \;\&\; s_2 \in S_2\}$$

Examples

$\epsilon.a = \{a\}$

$0.1 = \{01\}$

$(0 \mid 1).(\epsilon \mid 0) = \{0, 1\}.\{\epsilon, 0\} = \{0, 1, 00, 10\}$

Regular Expressions (RegExp): Definition

Kleene star

$$S^* = \{s_1.s_2.\cdots.s_n \mid s_i \in S \,\&\, n \in \mathbb{N}\}$$

Example

$$\{00, 11\}^* =$$

$$= \{\epsilon, 00, 11, 0000, 0011, 1100, 1111, 000000, 000011, 001100, \ldots\}$$

Regular Expressions (RegExp): Definition

Inductive definition of RegExps over an alphabet A:

Basis.

- $\emptyset$ is a regular expression;
- The empty string $\{\epsilon\}$ is a RegExp;
- Each symbol $a \in A$ is a RegExp.

Induction. Let r and s be two RegExps, then

- $(r.s)$ is a RegExp;
- $(r|s)$ is a RegExp;
- $(r)^*$ is a RegExp.

Example

$$((0.(0|1))^* \mid ((0|1)^*).0)$$

- It is a regular expression over the alphabet $\{0, 1\}$
  - Strings that start with 0 (left part)
  - Strings that end with 0 (right part)

Priority of operations

Priority of operations from higher to lower:

1. * (Kleene star)
2. . (Concatenation)
3. | (Union)

Example:

- $(\epsilon \mid a^*.b)$ is equivalent to $(\epsilon \mid ((a)^*.b))$

Agenda

Non-determinism:

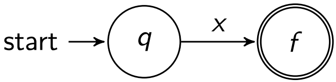
- Non-deterministic FSA (NDFSA)
- NDFSA to DFSA

Regular Expressions (RegExp)

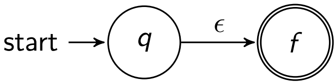
- Definition
- **RegExp to (N)FSA**
- FSA to RegExp

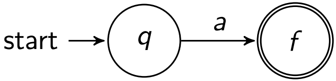
Rules

For $x \in A \cup \{\epsilon\}$,



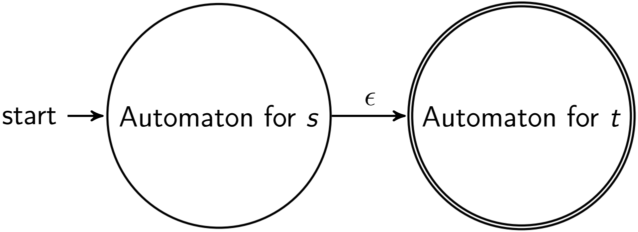
Examples



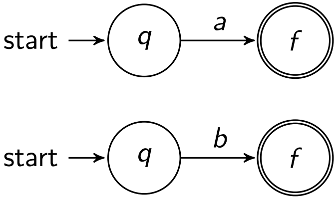


Rule: Concatenation Expression

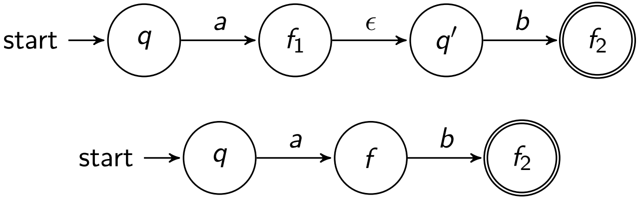
The concatenation expression $s.t$



Rules

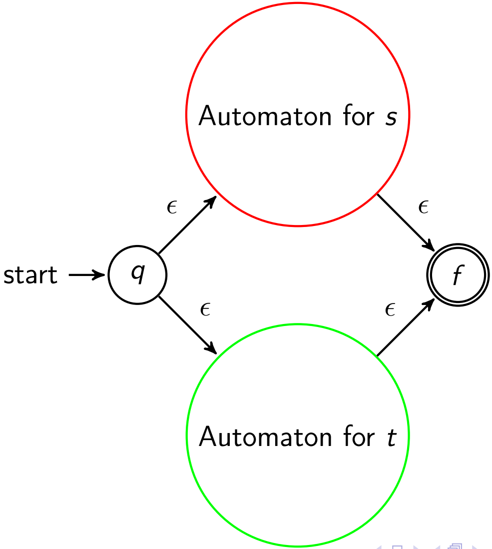


Example for $a.b$



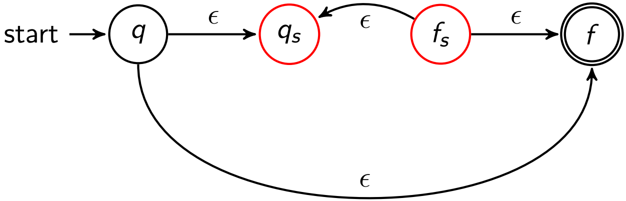
Rule: Union Expression

The union expression $s|t$



Rule: Kleene Star Expression

The Kleene star expression s^* is converted to



$N(s)$ is the (N)FSA of the subexpression s.

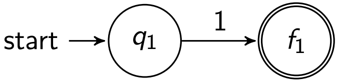
Example 1

Build a (N)FSA for $(1 \mid 01)^*$

Example 1

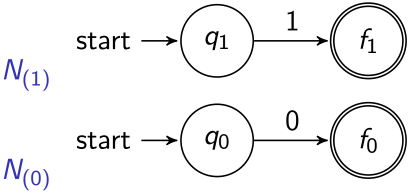
Build a (N)FSA for $(1 \mid 01)^*$

$N_{(1)}$



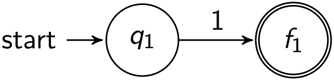
Example 1

Build a (N)FSA for $(1 \mid 01)^*$

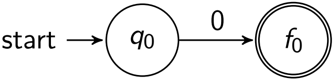


Example 1

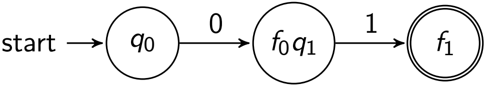
Build a (N)FSA for $(1 \mid 01)^*$



$N_{(1)}$



$N_{(0)}$



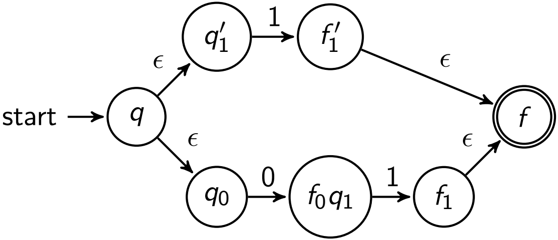
$N_{(01)}$

Example 1

Build a (N)FSA for $(1 \mid 01)^*$

Example 1

Build a (N)FSA for $(1 \mid 01)^*$

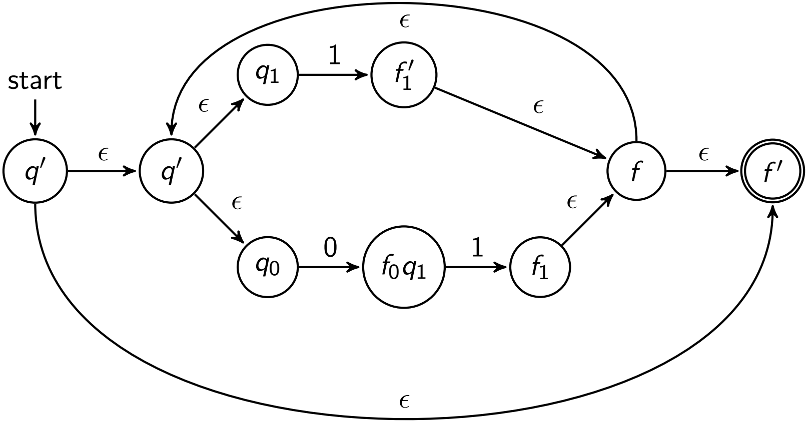


$N_{(1|01)}$

Example 1

Build a (N)FSA for $(1 \mid 01)^*$

$N_{(1|01)^*}$



Agenda

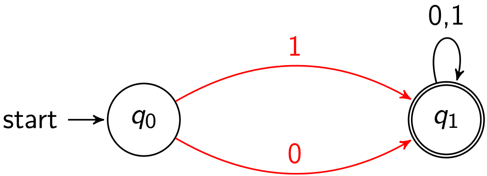
Non-determinism:

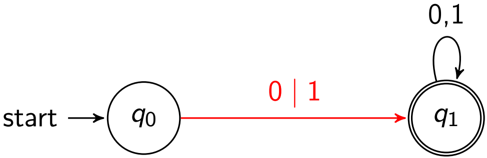
- Non-deterministic FSA (NDFSA)
- NDFSA to DFSA

Regular Expressions (RegExp)

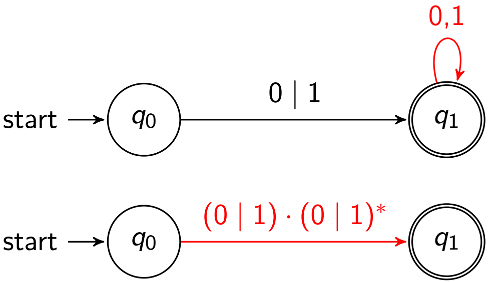
- Definition
- RegExp to (N)FSA
- **FSA to RegExp**

Regular Languages. Example 1



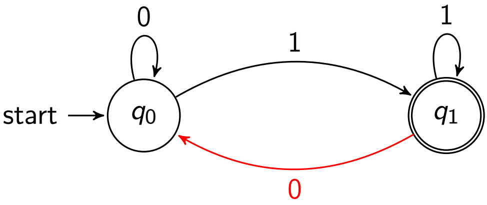


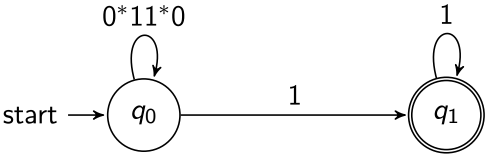
Regular Languages. Example 1



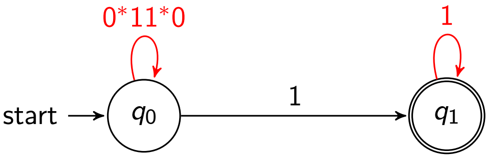
$$L = (0 \mid 1) \cdot (0 \mid 1)^* = \{(0 \mid 1) \cdot s \mid s \in \{0,1\}^*\} = \{s \in \{0,1\}^* \mid s \neq \epsilon\}$$

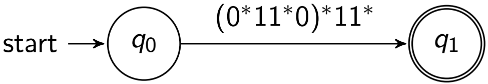
Regular Languages. Example 2





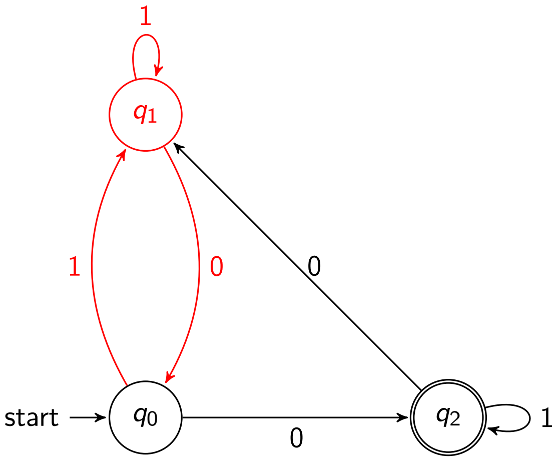
Regular Languages. Example 2



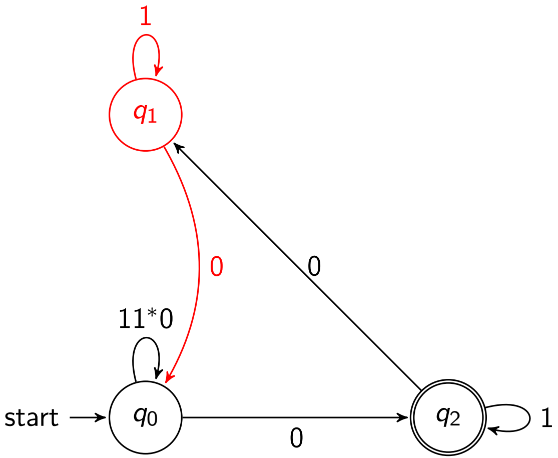


$$L = (0^*11^*0)^*11^* = \{s1 \mid s \in \{0,1\}^*\}$$

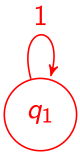
Regular Languages. Example 3

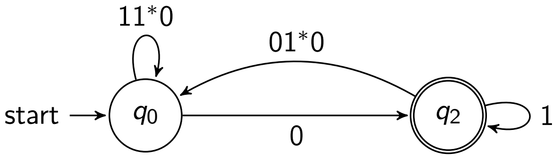


Regular Languages. Example 3

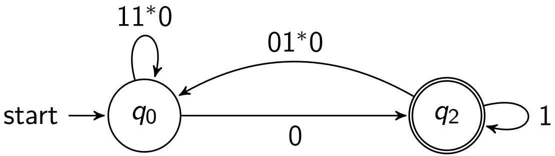


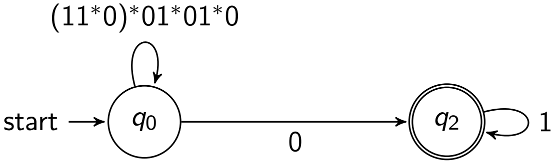
Regular Languages. Example 3

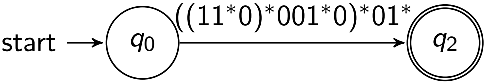




Regular Languages. Example 3







Thank you for your attention!