Formal Languages and Linguistics

Pascal Amsili

Sorbonne Nouvelle, Lattice (CNRS/ENS-PSL/SN)

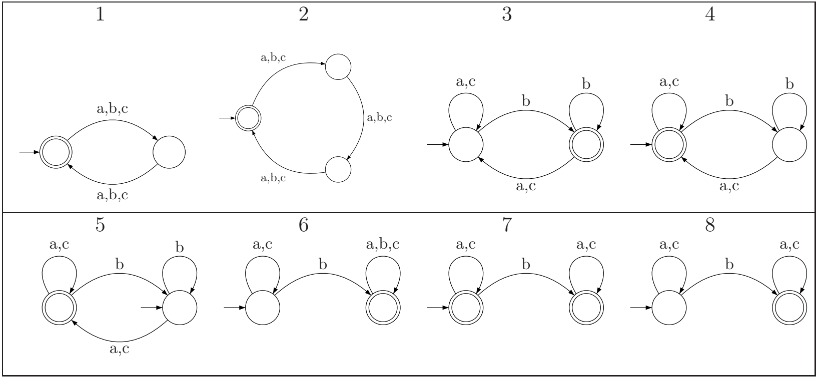
Cogmaster, september 2022

Exercices

Let $\Sigma = \{a, b, c\}$. Give deterministic finite state automata that accept the following languages:

1. The set of words with an even length.
2. The set of words where the number of occurrences of b is divisible by 3.
3. The set of words ending with a b.
4. The set of words not ending with a b.
5. The set of words non empty not ending with a b.
6. The set of words comprising at least a b.
7. The set of words comprising at most a b.
8. The set of words comprising exactly one b.

Answers


1
a,b,c
a,b,c
2
a,b,c
a,b,c
a,b,c
3
a,c
b
b
a,c
4
a,c
b
b
a,c
5
a,c
b
b
a,c
6
a,c
b
a,b,c
7
a,c
b
a,c
8
a,c
b
a,c

Overview

Ways of non-determinism

A word is recognized if there exists a path in the automaton. It is not excluded however that there be several paths for one word: in that case, the automaton is non deterministic.
What are the sources of non determinism?

- $\delta(a, S_1) = \{S_2, S_3\}$
- "spontaneous transition" = ε-transition

Equivalence theorems

For any non-deterministic automaton, it is possible to design a complete deterministic automaton that recognizes the same language.

Proofs: algorithms (constructive proofs)

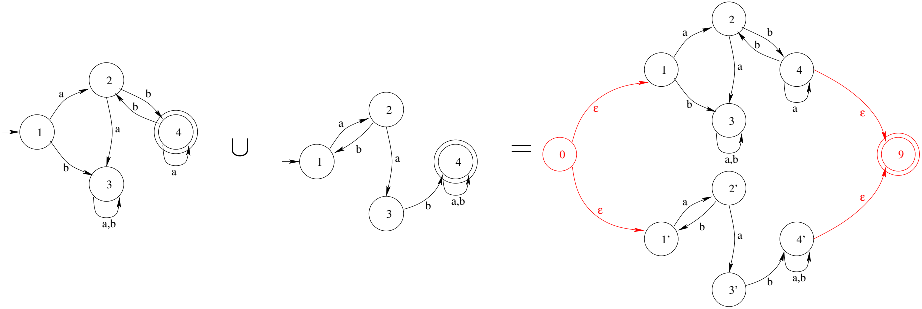
First "remove" ε-transitions, then "remove" multiple transitions.

Closure (1)

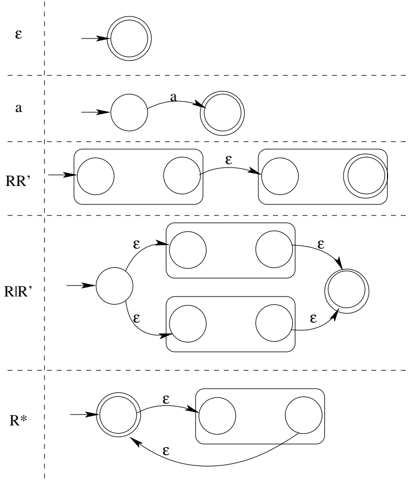
Regular languages are closed under various operations: if the languages L and L' are regular, so are:

- $L \cup L'$ (union); $L.L'$ (product); L^* (Kleene star)
  (rational operations)

Union of regular languages: an example



Rational operations



Closure (2)

Regular languages are closed under various operations: if the languages L and L' are regular, so are:

- $L \cup L'$ (union); $L.L'$ (product); L^* (Kleene star)
  (rational operations)
  $\rightarrow$ for every rational expression describing a language , there is a FSA that recognizes L

Closure (2)

Regular languages are closed under various operations: if the languages L and L' are regular, so are:

- $L \cup L'$ (union); $L.L'$ (product); L^* (Kleene star)
(rational operations)
$\rightarrow$ for every rational expression describing a language , there is a FSA that recognizes L and vice-versa

Closure (2)

Regular languages are closed under various operations: if the languages L and L' are regular, so are:

- $L \cup L'$ (union); $L.L'$ (product); L^* (Kleene star) *(rational operations)*
  $\rightarrow$ for every rational expression describing a language , there is a FSA that recognizes L and vice-versa
- $L \cap L'$ (intersection); $\overline{L}$ (complement)
- ...

Intersection of regular languages

Algorithmic proof
Deterministic complete automata

L_1	a	b
$\rightarrow$ 1	2	4
2	4	3
$\leftarrow$ 3	3	3
4	4	4

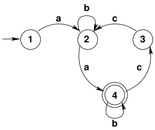
L_2	a	b
$\leftrightarrow$ 1	2	5
2	5	3
3	4	5
4	1	4
5	5	5

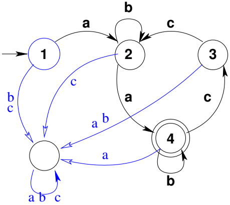
$L_1 \cap L_2$	a	b
$\rightarrow$ (1,1)	(2,2)	(4,5)
(2,2)	(4,5)	(3,3)
(4,5)	(4,5)	(4,5)
(3,3)	(3,4)	(3,5)
(3,4)	(3,1)	(3,4)
$\leftarrow$ (3,1)	(3,2)	(3,4)
(3,2)	(3,4)	(3,3)
(3,5)	(3,5)	(3,5)

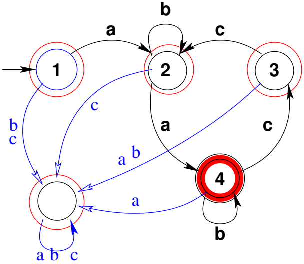
Complement of a regular language

Deterministic complete automata

completed complemented







Sorbonne Nouvelle

Pumping lemma: Intuition

Take an automaton with k states.

Pumping lemma: Intuition

Take an automaton with k states.
If the accepted language is infinite,
then some words have more than k letters.

Pumping lemma: Intuition

Take an automaton with k states.
If the accepted language is infinite,
then some words have more than k letters.
Therefore, at least one state has to be “gone through” several times.

Pumping lemma: Intuition

Take an automaton with k states.
If the accepted language is infinite,
then some words have more than k letters.
Therefore, at least one state has to be "gone through" several times.
That means there is a loop on that state.

Pumping lemma: Intuition

Take an automaton with k states.
If the accepted language is infinite,
then some words have more than k letters.
Therefore, at least one state has to be “gone through” several times.
That means there is a loop on that state.
Then making any number of loops will end up with a word in L.

$\Rightarrow$ Pumping lemma

Pumping lemma: definition

Def. 12 (Pumping Lemma)

Let L be an infinite regular language.
There exists an integer k such that:

$$\begin{array}{rl} \multicolumn{2}{l}{\forall x \in L,\ |x| > k,\ \exists u, v, w \ \text{ such that } x = uvw,\ \text{with:}} \\ (i) & |v| \geq 1 \\ (ii) & |uv| \leq k \\ (iii) & \forall i \geq 0,\ uv^i w \in L \end{array}$$

Pumping lemma: Illustration

Let's illustrate the lemma with a language which trivialy satisfies it: a^*bc.
Let $k = 3$, the work abc is long enough, and can be decomposed:

$$\underbrace{\varepsilon}_{u} \quad \underbrace{a}_{v} \quad \underbrace{b \quad c}_{w}$$

The three properties of the lemma are satisfied:

- $|v| \geq 1$ $(v = a)$
- $|uv| \leq k$ $(uv = a)$
- $\forall i \in \mathbb{N}$, $uv^iw(= a^ibc)$ belongs to the language by definition.

Pumping lemma: Consequences

The pumping lemma is a tool to prove that a language is **not** regular.

$\mathcal{L}$ regular	$\Rightarrow$	pumping lemma $(\forall i, uv^i w \in \mathcal{L})$
pumping lemma	$\not\Rightarrow$	$\mathcal{L}$ regular
NO pumping lemma	$\Rightarrow$	$\mathcal{L}$ **NOT** regular

Pumping lemma: Consequences

The pumping lemma is a tool to prove that a language is **not** regular.

$\mathcal{L}$ regular	$\Rightarrow$	pumping lemma $(\forall i, uv^i w \in \mathcal{L})$
pumping lemma	$\not\Rightarrow$	$\mathcal{L}$ regular
NO pumping lemma	$\Rightarrow$	$\mathcal{L}$ **NOT** regular

to prove that $\mathcal{L}$ is

regular: provide an automaton

not regular: show that the pumping lemma does not apply