Büchi automata

vendredi 23 octobre 2015 11:32

ω - langages

Let Σ be a finite alphabet

An ω-word is an element of Σ^ω $(\omega = \mathbb{N})$

↳ an infinite sequence of symbols of Σ

↳ we denote u_i the i^{th} letter of the word u

An ω-word could represent the sequence of states a process goes through

Ex: a light + + + − − − − + − − − − + + + + − − − ...

(+ = on ; − = off)

a server r t t R r t t t R r t r t R t t r t R R r ...

r : reception of a request

R : response to a request

t : internal treatment

An ω-language is a subset of Σ^ω

An ω-language could represent:

- the possible executions of a process
- the expected executions of a process

E : - language of the possible executions for the light: $\{+,-\}^\omega$

- expected language for the light

→ does not stay "on" more than 20 "ticks"

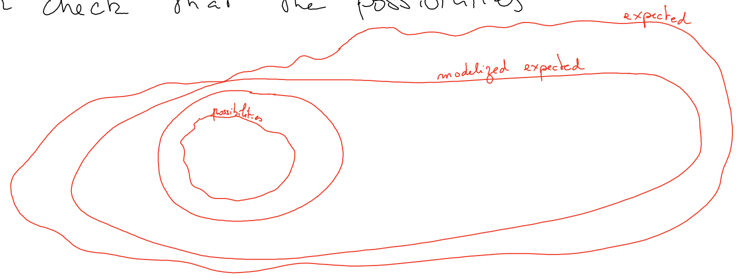
→ remains "off" 75% of the time (meaning?)

- language of the possible executions for the server:

→ the number of R so far is always less or equal to the number of r

- expected language for the server:
  → the R corresponding to each r is emitted at most 10 "ticks" after the arrival of the r

Verification problem: Prove or check that the possibilities language is included in the expected language



ω-Automata

Pour résoudre ce problème, il faut un moyen de décrire les ω-langages que l'on souhaite manipuler

An ω-automaton is a tuple:

$$\mathcal{A} = \langle \Sigma, Q, s, \mathbb{A}, R \rangle$$

Σ : finite alphabet

Q : finite set of states

$s \in Q$, the initial state

$\mathbb{A}$ is an accepting condition

$R \subseteq Q \times \Sigma \times Q$, is the transition relation

A path of $\mathcal{A}$ is a sequence:

$$(q_i, l_i)_{i \in \omega} \in (Q \times \Sigma)^\omega$$

such that

$$q_0 = s \quad \text{and} \quad \forall i \in \omega \quad (q_i, l_i, q_{i+1}) \in R$$

We say that this path is labelled by $(l_i)_{i \in \omega}$

The accepting condition is a condition on the sequence of states $(q_i)_{i \in \omega}$. It has to be expressible by a finite presentation

Usual conditions are expressed using, if $x \in Q^\omega$, the set

$$\mathrm{Inf}(x) = \{ q \in Q \mid q \text{ occurs infinitly often in } x \}$$
$$= \{ q \in Q \mid \forall i \in \omega\ \exists j > i\ \ x_j = q \}$$

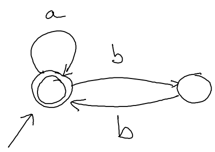
Name	Presentation	Condition
Büchi	$_A \subset Q$ ccepting states	$\mathrm{Inf}(x) \cap Q_A \neq \emptyset$ "an accepting state is visited infinitly often"
	$\pi : Q \to \mathbb{N}$ associates levels to states	$\min \pi(\mathrm{Inf}(x)) \in 2\mathbb{N}$ "The level of states, visited infinitly often, of lowest level is even"
Muller	$\mathcal{Q}_A \subset 2^Q$	$\mathrm{Inf}(x) \in \mathcal{Q}_A$ "The sets of infinitly often visited states is in $\mathcal{Q}_A$"

The ω-language accepted by the automaton $\mathcal{A}$ is the set of ω-words u such that there exists a path (x_i, u_i) of $\mathcal{A}$ where $(x_i)_{i \in \omega}$ is accepted by A. We denote this language $\mathcal{L}(\mathcal{A})$

<u>Theorem</u>: Any language accepted by an ω-automaton $\mathcal{A}$ whose accepting condition is expressed using only the Inf predicate is accepted by an ω-automaton using the Büchi accepting condition

An ω-automaton using the Büchi accepting condition is called a **Büchi automaton**

Büchi automata are represented the same way finite automata do:



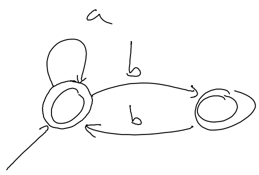
$(a+bb)^\omega$

{b comes by pairs}

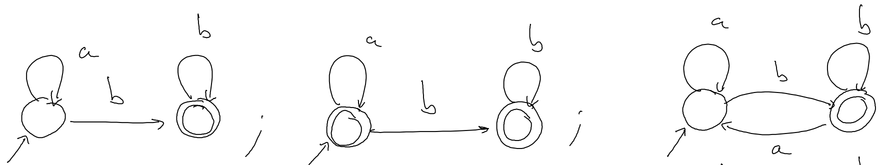
 : initial state

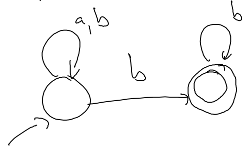
 : accepting state

 : transition

This automaton is equivalent to 

Ex:





Deterministic ω-automata

An ω-automaton $\mathcal{A}$ is **deterministic** if for all ω-word u there is <u>at most one</u> path of $\mathcal{A}$ labelled by u

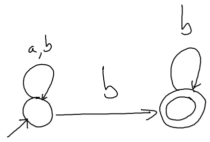
An ω-automaton $\mathcal{A}$ is deterministic if and only if R is a partial function:

$$\forall \ell \in \Sigma,\ \forall q \in Q,\quad \left|\{q' \in Q \mid (q,\ell,q') \in R\}\right| \leq 1$$

↳ for any letter ℓ, there is at most one arrow labelled by ℓ exiting each state

<u>Theorem</u> The deterministic Büchi automata accept strictly less languages than their non-deterministic counterparts.

counterparts.



ω-regular expressions

An ω-regular expression is recursively defined by

- A^ω where A is a regular expression $(\varepsilon \notin A$ and $A \neq \emptyset)$
- AB where A is a regular expression and B is an ω-regular expression
- $A + B$ where A and B are ω-regular expressions

$$AB = \{ uw \mid u \in A \text{ and } w \in B \}; \quad (uw)_i = \begin{cases} u_i & \text{if } i < |u| \\ w_{i-|u|} & \text{otherwise} \end{cases}$$

$$A + B = A \cup B$$

$$A^\omega = \{ w \mid \exists (u_i)_{i \in \omega},\ \forall i \in \omega\ u_i \in A \wedge w = \bigodot_{i \in \omega} u_i \};$$

$$\left(\bigodot_{i \in \omega} u_i \right)_j = (u_n)_k \text{ when } n \text{ is the largest such that } \sum_{i=0}^{n-1} |u_i| \leq j \text{ and } k = j - \sum_{i=0}^{n-1} |u_i|$$

$$\bigodot_{i \in \omega} u_i = \underset{\in A}{u_0}\ \underset{\in A}{u_1}\ \underset{\in A}{u_2}\ \underset{\in A}{u_3} \cdots \underset{\in A}{u_n} \cdots$$

Theorem Let L be an ω-language

The three following propositions are equivalent:

- L is accepted by a Büchi automaton

- L is accepted by a deterministic Muller or Parity automaton

- L is expressible by an ω-regular expression

We call such ω-language as ω-regular languages.

Corollary

If A and B are ω-regular languages and C is a regular language then:

$$\underbrace{A \cup B\ ;\ CA\ ;\ C^\omega}_{\text{using } \omega\text{-regular expressions}}\ ;\ \underbrace{\Sigma^\omega \setminus A}_{\text{using deterministic Muller automata}}\ ;\ \underbrace{A \cap B}_{\complement(\complement A \cup \complement B)}$$

(inefficient for the number of states: double exponential blow-up)