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 $+++----+----++++---\ \ldots$

($+$ = on ; $-$ = off)

a server $rttRrtttRrtrtRttrtRRr\ldots$

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: $\{+,-\}$
- 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-Automaton

To solve this problem, we need a way to represent these language in a finite way

An **ω-automaton** is a tuple:

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

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

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

Name	Presentation	Condition
Büchi	$Q_A \subset Q$ Accepting states	$\mathrm{Inf}(x) \cap Q_A \neq \emptyset$ "an accepting state is visited infinitly often"
Parity	$\pi : Q \to \mathbb{N}$ 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"
Mull	2^Q	$\mathrm{Inf}(x) \in$

Muller $\mathcal{L}_A \subset \mathcal{L}$ $Inf(x) \in \mathcal{L}_A$

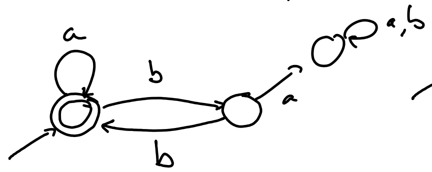
"The sets of infinitly often visited states is in $\mathcal{L}_A$"

The ω-language **accepted** by the automaton $\mathcal{A}$ is the set of ω-words u such that <u>there exists</u> a path (x_i, u_i) of $\mathcal{A}$ where $(x_i)_{i \in \omega}$ is accepted by $\mathbb{A}$. 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:




: initial state

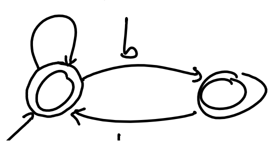
: accepting state

: transition

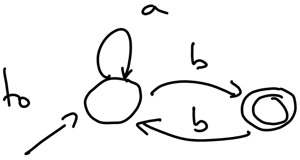
$(a + bb)^\omega$

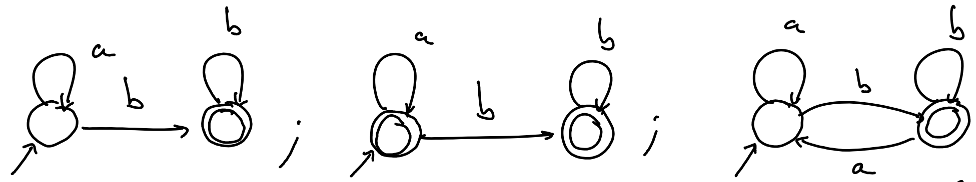
{ b comes by pairs }

This automaton is equivalent to

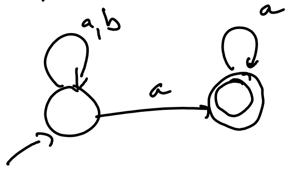


but not to



Ex: 

Deterministic ω- automata



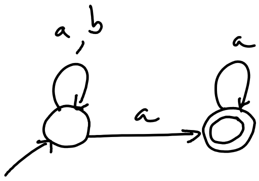
An ω- automaton $\mathcal{A}$ is **deterministic** if for all ω- word u there is at most one path of $\mathcal{A}$ labelled by u

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

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

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

Theorem The deterministic Büchi automata accept strictly less languages than their non-deterministic 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

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

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

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

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

$$\begin{array}{ccccccccc} \bigodot_{i \in \omega} u_i = & u_0 & u_1 & u_2 & u_3 & \dots & u_m & \dots \\ & \cap\!\!\!| & \cap\!\!\!| & \cap\!\!\!| & \cap\!\!\!| & \dots & \cap\!\!\!| & \dots \\ & A & A & A & A & & A & \end{array}$$

<u>Theorem</u> Let L be an ω-language

The three followings 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**.

<u>Corollary</u>

If $\mathbb{A}$ and $\mathbb{B}$ are ω-regular languages and C is a regular language then:

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

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