Stochastic languages

Florent Garnier

April 14, 2009

Probabilistic automaton

Probabilistic automaton model was defined by M. Rabin [Rabin63].

- A varient of NFA.
- Express probabilistic transitions between states.
- Define stochastics languages.
- Variant are used in various domain, from learning, pattern recognition, signal processing etc ...

Probablistic automaton

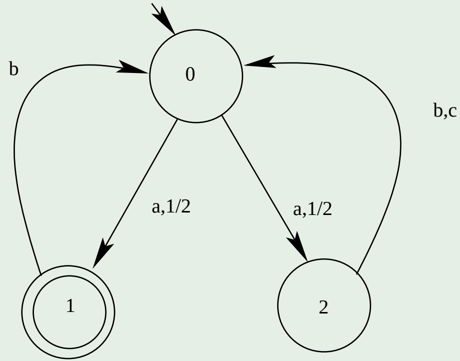
Definition

A probabilitic automaton is a tuple $(Q, \Sigma, q_0, F, \delta)$ with:

- Q is a finite set of state.
- Σ is an alphabet.
- q_0 is the initial state.
- F is the set of accepting state.
- δ is a function from $Q \times \Sigma \rightarrow Dist(Q)$.

Probablistic automaton

Example



Language of a probabilistic automaton (Rabin)

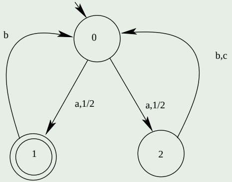
Definition (Run of a word)

Given a word $w = w_1 \dots w_n$ of Σ^*, a run of w on $\mathcal{A}$ is a sequence $q_1 \dots q_n$ of Q such that:

- q_0 is the initial state of $\mathcal{A}$.
- For all $0 \leq i < n$, $\delta(q_i, w_i, q_{i+1}) > 0$.
- If $q_n \in F$ the run is accepting.

Runs of a p-automaton

Example



Runs for the word *aba* are:

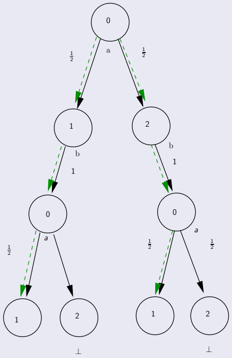
$$\{0 \xrightarrow{a} 1 \xrightarrow{b} 0 \xrightarrow{a} 1, 0 \xrightarrow{a} 2 \xrightarrow{b} 0 \xrightarrow{a} 1, 0 \xrightarrow{a} 2 \xrightarrow{b} 0 \xrightarrow{a} 2\}.$$

The two first are accepting, whereas the latter is not.

Measure of a word

Property

A p-automaton, associates to each word of Σ^* a probability:



Measure of a word

This value corresponds to the weighted ration between the accepting runs over all the run associated to a run. This value is set by zero if a word has no associated execution.

Definition (Probability of acceptance)

Given a word $w = w_1 \dots w_n$, we note by:

$$P(w) := \frac{\sum_{\rho \in Accept(w)} \prod_{i=1}^{n} \delta(\rho_i, w_i, \rho_{i+1})}{\sum_{\rho \in Run(w)} \prod_{i=1}^{n} \delta(\rho_i, w_i, \rho_{i+1})}.$$

Stochastic languages

Definition (Stochastic languages)

The set of recognized by probabilistic automata are called stochastic languages.

Definition (Languages recognized by a P-automaton)

Given a real number $\eta < 1$, a Probabilistic automaton $\mathcal{A}$, the language $\mathcal{L}_\eta(\mathcal{A})$ is defined as:

$$\mathcal{L}_\eta(\mathcal{A}) := \{w \in \Sigma^* | P(w) \geq \eta\}$$

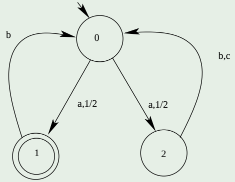
Stochastic languages

Property

- *If* $w \in \mathcal{L}_\eta(\mathcal{A})$*, then* $\mathcal{A}$ *recognize the word* w *with probability at least* η*.*
- *The set* $\mathcal{L}_\eta(\mathcal{A})$ *depends on the thresold* η*.*

Example

Example



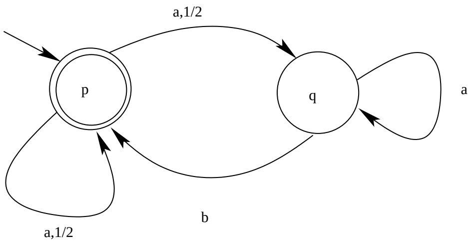
- $aca \in \mathcal{L}_{\frac{1}{2}}(\mathcal{A})$.
- $aca \notin \mathcal{L}_{\eta}(\mathcal{A})$ for all $\eta < \frac{1}{2}$.

Properties of stochastic languages

Property

1. *S.L. contains the set of regular languages.*
2. *There exists Non regular stochastic languages.*

A non regular but stochastic language:



Example

PBA for $L = \{a^{k_1} b a^{k_2} b \ldots a^{k_n} b \ldots s.t \prod_{i=1}^{n} (1 - (1/2)^{k_i}) > \eta\}$

Properties on cut-points

Definition (Isolated cut-point)

Let be $\mathcal{A}$ a probabilistic automaton. A cut-point η is called *isolated* cut-point if there exists $\delta > 0$ such that:

$$|P(w) - \eta| \geq \delta, \ \forall w \in \Sigma^*.$$

Property

If η is an isolated cutpoint, then $\mathcal{L}_\eta(\mathcal{A})$ is a regular language.

p-adic languages

Definition (p-adic languages)

A p-adic language is the set of strings in the alphabet $\{0, \ldots, (p-1)\}$, such that:

$$\mathcal{L}_\eta(p) := \{0.n_1 n_2 n_3 \ldots | 0 \leq n_k < p \wedge 0.n_1 n_2 n_3 \ldots > \eta\}$$

Property

- *$\mathcal{L}_\eta(p)$ is rational iff $\eta \in \mathbb{Q}$.*
- *The number of stochastic languages is not countable.*

A really powerfull model of automaton

Property

- *If* $w \in \mathcal{L}_\eta(\mathcal{P})$, *then* $P(A(w)) \geq \eta$.
- *If* $w \notin \mathcal{L}_\eta(\mathcal{P})$, *then* $P(A(w)) = 0$.

Stochastic languages are in the RP-complexity class.

Question: How to test whether a word w belongs to this language with high probability ?

The price to pay

Theorem

The following problem is undecidable: Given $\mathcal{A}$ a PFA, and $0 < \eta < 1$, does it exist $w \in \Sigma^*$, such that $P(w) > \eta$?

Remark

Deciding language emptiness is mandatory for dedicing:

- *Universality.*
- *Equality of two languages.*
- *Language inclusion.*

How to define a probabilistic bisimulation

Definition

Let $\mathcal{R}$ be an equivalence relation over the set X. Two probability spaces $(\Omega_1, \mathcal{F}_1, P_1)$ and $(\Omega_2, \mathcal{F}_2, P_2)$ of $Probs(X)$ are $\mathcal{R}-$equivalent, written $(\Omega_2, \mathcal{F}_2, P_2) \equiv_{\mathcal{R}} (\Omega_2, \mathcal{F}_2, P_2)$, iff for every class C of $X/\mathcal{R}$

$$\sum_{x \in \Omega_1 \cap C} P_1[x] = \sum_{x \in \Omega_1 \cap C} P_1[x]$$

In other words, $(\Omega_1, \mathcal{F}_1, P_1)$ and $(\Omega_2, \mathcal{F}_2, P_2)$ are $\mathcal{R}-$equivalent if they assign the same probability mesure to each equivalence class of $\mathcal{R}$.

Strong bisimulation

Definition (Strong bisimulation)

A strong bisimulation between two simple probabilistic automata M_1 and M_2 is an equivalence relation $\mathcal{R}$ over $states(M_1) \cup states(M_2)$ such that

(1) each start state of M_1 is related to at least one start state of M_2 and vice versa,

(3) for each $s_1 \mathcal{R} s_2$ and each transition $s_1 \xrightarrow{a} \mathcal{P}_1$ of either M_1, M_2, there exists a transition $s_2 \xrightarrow{a} \mathcal{P}_2$ of either M_1, M_2 such that $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$. We write $M_1 \simeq M_2$ whenever $acts(M_1) = acts(M_2)$ and there is a strong bisimulation between M_1 and M_2.

Strong simulation

Definition (Strong bisimulation)

A strong bisimulation between two simple probabilistic automata M_1 and M_2 is an equivalence relation $\mathcal{R}$ over $states(M_1) \cup states(M_2)$ such that

(1) each start state of M_1 is related to at least one start state of M_2.

(2) for each $s_1 \mathcal{R} s_2$ and each transition $s_1 \xrightarrow{a} \mathcal{P}_1$ of eitherM_1 there exists a transition $s_2 \xrightarrow{a} \mathcal{P}_2$ of either M_2 such that $\mathcal{P}_1 \equiv_{\mathcal{R}} \mathcal{P}_2$.We write $M_1 \sqsubseteq M_2$ whenever $acts(M_1) = acts(M_2)$ and there is a strong bisimulation between M_1 and M_2.

Model Checking techniques in Short

Model Checking in short consists in :

- Modeling a system with a formal language.
- Expressing specification or system propreties.
- Proving that every run of the system complies with the specification.

Question: Is it possible to use randomness to improve verification ?

Modeling a system

Low level modeling formalism Transition system.

Definition (Transition system)

A transition system is a tuple $\mathcal{K} = (S, I, \mathcal{A}, \delta)$ where :

S	system sates
$I \subseteq S$	initial states
$\mathcal{A}$	set of actions
$\delta \subseteq S \times \mathcal{A} \times S$	transition relation

For every $s \in S$ there exists $(a, t) \in S \times \mathcal{A}$ such that $(s, a, t) \in \delta$.

Semantic of a transition system

Definition (Run of a transition system)

A run of $\mathcal{K}$ is an infinite sequence $\rho = s_0 \ldots s_n \ldots$ of states $s_i \in S$ such that :

- $s_0 \in I$
- For all $i \in \mathbb{N}$ $(s_i, a_i, s_{i+1}) \in \delta$ for some $a_i \in \mathcal{A}$

High(er) level modeling formalism

- Kripke Structures
- Process Algebra
- Petri Nets
- Comunicating agents
- Automata.
- etc ...

Specifying properties using logics

- Linear logic : PTL, LTL.
  - Fairness, mutual exclusion, eventual acces to critical section etc ...
- Branching Logic : CTL
  - Combine temporal modalities and quantification over paths.
- Logic subsuming Linear and Branching logic : CTL*, μ-calcul.

The Linear Temporal Logic

The LTL logic express time dependent properties of system runs. Evaluated over infinite sequences of labels.

Type	Formula	$\rho \models \phi$ iff ...
Atomic	$p \in AP$	p holds for $\rho\|_0$
Boolean	$\neg\phi$	$\rho \not\models \phi$
	$\phi \vee \psi$	$\rho \models \phi$ or $\rho \models \psi$
Temporal	$X\phi$	$\rho\|_1 \models \phi$
	$F\phi$	$\rho\|_i \models \phi$ for some $i \in \mathbb{N}$
	$G\phi$	$\rho\|_i \models \phi$ for all $i \in \mathbb{N}$
	$\phi U \psi$	There is $i \in \mathbb{N}$ s.t. $\rho_i \models \psi$ and $\rho\|_j \models \phi$ for all $0 \leq j < i$
	$\phi W \psi$	$\rho \models \phi U \psi$ or $\rho \models G\phi$

LTL: Exemples

Exemple

- *invariants* GP

  $G\neg(crit_1 \wedge crit_2)$ *mutual exclusion*

  $G(preset_1 \vee \ldots \vee preset_n)$ *deadlock freedom*

- *Response, recurence* $G(P \Rightarrow FQ)$

  $G(try_1 \Rightarrow Fcrit_1)$ *eventual acces to critical section*

  $GF\neg crit_1$ *no starvation in critical section*

- *Reactivity, Steet* $GFP \Rightarrow GFQ$

  $GF(try_1 \wedge \neg crit2) \Rightarrow GFcrit_1$ *strong fairness*

- *Precedence* $G(P_1 W \ldots WP_n)$

  $G(try_1 \wedge \neg try_2 \Rightarrow \neg crit_2 W crit_2 W \neg crit_2 W crit_1)$

  1-bounded overtaking

How to check that $\mathcal{K} \models \phi$?

- LTL formulae can be represented by ω-automata.
- ω-automata are expressive enough to model transition systems.
- ω-automata are finite automata that accept infite length language, that can model program runs and runs constraints.

Let's now define and list ω-automata properties ...

Finite automata on ω-words

Definition (Büchi-automata)

A ω-automaton is a tuple $\mathcal{B} = (Q, I, \delta, F)$

Q	finite set of states
$I \subseteq Q$	initial states
$\delta \subseteq Q \times \Sigma \times Q$	transition relation
$F \subseteq Q$	accepting states

Definition (Run of $\mathcal{B}$ on ω-words $a_0 a_1 \ldots \in \Sigma^\omega$)

sequence	$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \ldots$
initialisation	$q_0 \in I$
consecution	$(q_i, a_i, q_{i+1}) \in \delta$ for all $i \in \mathbb{N}$
accepting	$q_i \in F$ for infinitely many $i \in \mathbb{N}$

ω-regular languages

Definition (ω-language defined by $\mathcal{B}$)

$\mathcal{L} = \{w \in \Sigma^\omega | \mathcal{B}$ has some accepting run on w $\}$

Definition

ω-regular languages class of ω languages definable by Büchi automata.

Büchi autoamata: basic properties

Property (Decidability emptiness problem)

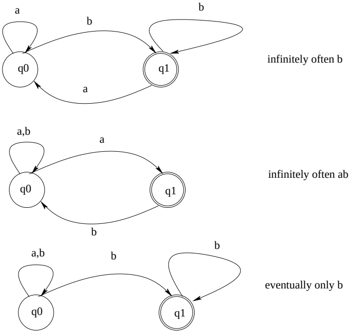
$\mathcal{L} \neq \emptyset$ *iff exists* $q_0 \in I, q \in F$ *such that* $q_0 \xrightarrow{\Sigma^*} q \xrightarrow{\Sigma^+} q$.

Complexity linear in $|Q|$.

Property (Closure properties)

- *union*
- *intersection*
- *complement, difficult construction* $O(2^{n \log n})$ *states*
- *projection*

Exemple of Büchi automata



Property

Deterministic Büchi automata are strictly less expressive than Non deterministic Büchi automata.

From LTL to Büchi automata

Basic insight

- Given ϕ a LTL formula, let note $\mathcal{L}$ the set of sequences of labels satisfying ϕ.
- Construct automaton ϕ that accepts $\mathcal{L}(\phi)$.

Idea of construction

- states: sets of "subformulas of ϕ promised to be true
- initial states: states containing ϕ
- transition relation: Ensures satisfaction of non-temporal formulas using recursion laws :

$$
\begin{array}{rcl}
G\phi & \equiv & \phi \wedge XG\phi \\
F\phi & \equiv & \phi \vee XF\phi \\
\phi U\psi & \equiv & \psi \vee (\phi \wedge X(\phi U\psi))
\end{array}
$$

- accepting states: defined from $F\phi$ ou $\phi U\psi$.

Model Checking

Problem: Given $\mathcal{K}$ and ϕ, decide whether $\mathcal{K} \models \phi$

Automata-theoretic solution

- Consider $\mathcal{K}$ as a $\omega-$automaton with all states final
- Define $(L)(\mathcal{K})$ =set of computaions of $\mathcal{K}$
- The following assetions are equivalent :

$$\mathcal{K} \models \phi$$
$$\mathcal{L}(\mathcal{K}) \subseteq \mathcal{L}(\phi)$$
$$\mathcal{L}(\mathcal{K}) \cap \mathcal{L}(\neg\phi) = \emptyset$$
$$\mathcal{L}(\mathcal{K}) \times \mathcal{B}_{\neg\phi}) = \emptyset$$

Complexity $O(|\mathcal{K}| \times |B_{\neg\phi}|)$=$O(|\mathcal{K}| \times 2^{|\phi|})$

Using probabilities to reduce the size of automata

- The paper [BG] decribes a model of PBA –resp. uPBA that accepts a subset of ω-regular language –resp. ω-regular languages.
- The size of uPBA can be exponetially more concise that NSA (Strong fairness).
- Emptiness is not decidable for PBA.
- Emptiness is decidable for uPBA

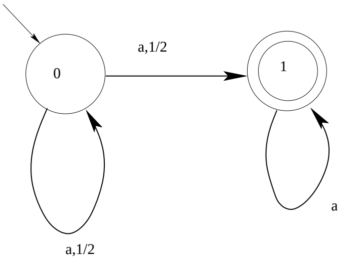
A model of probabilistic automata

Definition (Probabilistic Büchi Automata (PBA))

A PBA over the alphabet Σ is a tuple $\mathcal{B} = (Q, \delta, \mu, F)$ where :

Q	finite set of state
$\delta : Q \times \Sigma \times Q \to [0, 1]$	transition probabilities function
μ	inital distribution, $\mu(Q) = 1$
$F \subseteq Q$	Set of accepting states.

Exemple of PBA



PBA for $(a+b)^* a^\omega$

Accepted language of a PBA

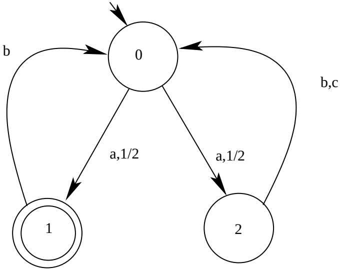
Definition (Accepting run)

An infinite run $\pi = s_0, \ldots s_n \ldots$ of $\mathcal{B}$ is accepting if $inf(\pi) \cap F \neq \emptyset$.

Definition (Accepted language of a PBA)

If one note $Pr(\rho) = P(\pi : \pi$ is an accepting run for $\rho)$,
$\mathcal{L}(\mathcal{B})$ consists of all words $\rho \in \Sigma^\omega$ s.t $Pr(\rho) > 0$

Exemple of PBA



PBA for $(ab + ac)^*(ab)^\omega$

Expressiveness of PBA

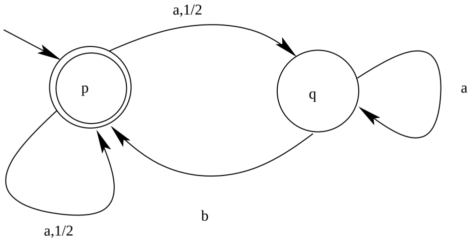
Theorem

The class of languages that can be accepted by a PBA strictly contains the class of ω-regular languages.

Lemma

There exists PBA that accepts non ω-regular languages.

Non ω-regular accepting PBA



PBA for $L = \{a^{k_1} b a^{k_2} b a^{k_3} b \dots s.t \prod_{i \geq 1}(1 - (1/2)^{k_i}) > 0\}$

Deciding emptyness of PBA

Theorem

The emptiness of PBA is undecidable [BBG08].

However, this problem is decidable for a subclass of PBA.

End components

Definition (End components)

Let $\mathcal{B} = (Q, \delta, \mu, F)$ be a PBA. An end component of $\mathcal{B}$ is a pair (P, A) where : $\begin{array}{l} \subseteq Q \\ A : P \longrightarrow 2^{\Sigma} \end{array}$

- $\sum_{q \in P} \delta(p, a, q) = 1$ for $a \in A(p)$
- The graph (P, A) is strongly conected.

(P, A) is accepting if $P \cap F \neq \emptyset$.

Uniform PBA

Definition (Uniform PBA)

A PBA $\mathcal{B}$ is called uniform if there exists $\theta > 0$ s.t for all (P, A) accepting end component, all $p \in P$ and all finite word $w \in \Sigma^*$ one of the following condition holds :

1. $Pr_{(P,A)}(p \xrightarrow{w}) \leq \theta$
2. $Pr_{(P,A)}(p \xrightarrow{w}) = 1 \wedge Pr_{(P,A)}(p \xrightarrow{w} q) \geq \theta$

Acceptance on uPBA

Definition

Let be $\mathcal{B}$ an uniform PBA and $\rho = a_0 \ldots \in \Sigma^\omega$. Then $\rho \in \mathcal{L}(\mathcal{B})$ iff there exists (P, A) accepting end component and a strongly prob fair run $\pi = q_0 \ldots$ for ρ s.t $inf(\pi) = (P, A)$ and q_I s.t

$$Pr_{(P,A)}(q_I \xrightarrow{w}) = 1$$

Properties of uPBA

Theorem

uPBA exactly accept the ω- regular languages.

Proof.

For any uPBA $\mathcal{P}$ there exists a NSA A such that $\mathcal{L}(P) = \mathcal{L}(A)$, with $|A| = O(exp(|P|))$.

$\square$

Properties of uPBA

Theorem

The emptiness problem for uPBA is decidable (NP-hard).

Proof.

The transforamtion uPBA$\rightarrow$ NSA is polynumial. Deciding emptyness for NSA is NP-hard. $\square$

uPBA can be exp better than NSA

There exists languages L_n which are accepted bu uPBA with 2n states, while any NSA for L_n has at least $\frac{2^n}{n}$