Qualitative Tree Languages

Arnaud Carayol
LIGM (Univ. Paris Est & CNRS)
Arnaud.Carayol@univ-mlv.fr

Axel Haddad
LIAFA & LIGM
Axel.Haddad@liafa.jussieu.fr

Olivier Serre
LIAFA (Univ. Paris 7 & CNRS)
Olivier.Serre@liafa.jussieu.fr

***Abstract*—We study finite automata running over infinite binary trees and we relax the notion of accepting run by allowing a negligible set (in the sense of measure theory) of non-accepting branches. In this qualitative setting, a tree is accepted by the automaton if there exists a run over this tree in which *almost every* branch is accepting. This leads to a new class of tree languages, called the *qualitative tree languages* that enjoys many properties.**

Then, we replace the existential quantification — a tree is accepted if there exists some accepting run over the input tree — by a probabilistic quantification — a tree is accepted if almost every run over the input tree is accepting. Together with the qualitative acceptance and the Büchi condition, we obtain a class of probabilistic tree automata with a decidable emptiness problem. To our knowledge, this is the first positive result for a class of probabilistic automaton over infinite trees.

I. Introduction

Finite automata on infinite trees were originally introduced by Rabin in [16] to prove the decidability of the monadic second order logic over the full binary tree as well as to solve the Church's synthesis problem [7]. Since then, automata on infinite trees and their variants have been intensively studied and found many applications, in particular in logic. Connections between automata on infinite trees and logic are discussed in the excellent surveys [17], [18].

Roughly speaking a tree automaton is a finite memory machine that takes as input an infinite node-labelled binary tree and processes it in a top-down fashion as follows. It starts at the root of the tree in its initial state, and picks (possibly nondeterministically) two successor states, one per son, according to the current control state, the letter at the current node and the transition relation. Then the computation proceeds in parallel from both sons, and so on. Hence a run of the automaton on an input tree is a labelling of this tree by control states of the automaton, that should satisfy the local constrains imposed by the transition relation. A branch in a run is accepting if the ω-word obtained by reading the states along the branch satisfies some acceptance condition (typically an ω-regular condition such as a Büchi or a parity condition). Finally, a tree is accepted by the automaton if there exists a run over this tree in which *every* branch is accepting. An ω-regular tree language is a tree language accepted by some tree automaton equipped with a parity condition. A fundamental result of Rabin is that ω-regular tree languages form a Boolean algebra [16]. The hard part in this proof is the complementation, and since the publication of this result in 1969, it has been a challenging problem to simplify the proof. A much simpler one was obtained in [11] making use of two-player games on graphs. Beyond this result, the connection between automata and games is one of the main tool in automata theory (see *e.g.* [10]).

In this article, we consider variations of the classical model of tree automata over infinite trees. Theses variations involve probabilities and preserve the fruitful connection with game theory.

In the first part of this paper, we consider a relaxed notion of accepting run. While the usual definition requires that all branches satisfy the acceptance condition, we allow a negligible set (in the sense of measure theory) of non-accepting branches. In this qualitative setting, a tree is accepted by the automaton if there exists a run over this tree in which *almost every* branch is accepting. With the parity condition, this leads to a new class of tree languages, *qualitative tree languages*. We show that this class enjoys many desirable properties: closure under union and intersection (but not under complement), emptiness is decidable in polynomial time (note that no polynomial algorithm is known for the emptiness test of parity tree automata). We also prove that there exists a strong connection between automata accepting qualitative tree languages and Markov decision processes, which play here a similar role as two-player games for usual tree automata.

The idea of allowing a certain amount of rejecting branches in a run was already considered in [4], where it was required that the number of accepting branches in a run belong to a specified set of cardinals Γ. In particular, they proved that if Γ consists of all cardinals greater than some γ, then one obtains a regular tree language. Qualitative tree languages as defined in this article are not captured by the work of [4]. Indeed, they are incomparable with regular tree languages.

In the second part of this paper, we investigate probabilistic automata on infinite trees. Acceptance by an automaton is based on existential quantification: an input is accepted if *there exists* an accepting run over it. Probabilistic automata are an alternative way to define acceptance. On finite words they have been introduced by Rabin in [15]. Compared with the standard setting, the non-deterministic guesses are replaced by random choices. Hence the set of transitions is replaced by a probability distribution over the set of all transitions which induces a probability measure on the set of runs of the automaton. The acceptation is defined using a threshold $0 < \lambda < 1$ on the probability of a run to be accepting. In contrast to the non-deterministic setting, the emptiness problem for probabilistic automata on finite words is undecidable [13].

This probabilistic model was recently extended to infinite words [2]. In addition to the threshold acceptation, two additional acceptations were considered : *almost-sure* and *positive* which respectively corresponds to a probability 1 or >0 for a run to be accepting[1]. Surprisingly the class of languages defined by Büchi automata with the positive acceptation is closed under complement, which implies that it coincides with the class of languages defined by co-Büchi automata with the almost-sure acceptation[2]. The emptiness problem for Büchi automata with the almost-sure acceptation are decidable. However the emptiness problem for Büchi automata with the positive acceptation as well as for co-Büchi automata with the almost-sure acceptation are undecidable. Of course, emptiness is undecidable when considering a threshold condition. See [1] for more details on this topic.

We consider probabilistic automata on infinite trees which accept a tree if almost every run over the input tree is accepting. For the qualitative definition of accepting runs, we prove that there exists a strong connection with partial observation Markov decision processes. In particular, for the Büchi condition, these probabilistic automata on infinite trees enjoy a decidable emptiness problem. To our knowledge, this is the first positive result for a class of probabilistic automaton over infinite trees.

In the last part, we discuss alternative definitions for qualitative acceptance as well as acceptance by probabilistic automata.

II. Definitions

A. Words and Trees

An *alphabet* A is a finite set of letters. In the sequel A^* denotes the set of finite words over A, and A^ω the set of infinite words over A. The *empty word* is written ε; the length of a word u is denoted by $|u|$. Let u be a finite word and v be a (possibly infinite) word. Then $u \cdot v$ denotes the *concatenation* of u and v; the word u is a *prefix* of v, denoted $u \sqsubseteq v$, iff there exists a word w such that $v = u \cdot w$. We denote by $u \sqsubset v$ the fact that u is a strict prefix of v (*i.e.* $u \sqsubseteq v$ and $u \neq v$). For some word u and some integer $k \geqslant 0$, we denote by u^k the word obtained by concatenating k copies of u (with the convention that $u^0 = \varepsilon$).

In this paper we consider full binary node-labelled trees. An A-*labelled tree* t is a mapping from $\{0,1\}^*$ to A. In this context, an element $u \in \{0,1\}^*$ is called a *node*, and the node $u \cdot 0$ (*resp.* $u \cdot 1$) is the *left son* (*resp. right son*) of u. The node ε is called the *root*. We shall refer to $|u|$ as the *depth* of u. The letter $t(u)$ is called the *label* of u in t.

A *branch* is an infinite word $\pi \in \{0,1\}^\omega$. We write $\mathrm{Br} = \{0,1\}^\omega$ for the set of all branches. A node u belongs to a branch π if u is a prefix of π. For an A-labelled tree t and a branch $\pi = \pi_0\pi_1\cdots$ we define the *label* of π as the ω-word $t(\pi) = t(\varepsilon)t(\pi_0)t(\pi_0\pi_1)t(\pi_0\pi_1\pi_2)\cdots$. The *cone* going through a node u is the set $\mathrm{Cone}(u) = u \cdot \{0,1\}^\omega$. A *sub-cone* of a cone $\mathrm{Cone}(u)$ is a cone $\mathrm{Cone}(v)$ with $u \sqsubseteq v$.

Given a tree t and a node u, the *subtree of t rooted at u* denoted $t[u]$ is the tree defined by $t[u](v) = t(u \cdot v)$. A tree t is said to be *regular* if it contains only finitely many different subtrees, *i.e.* the set $\{t[u] \mid u \in \{0,1\}^*\}$ is finite.

Let $\mathcal{F}_{\mathrm{Br}}$ be the σ-algebra generated by the set of cones (*i.e.* the smallest set of subsets of $\{0,1\}^\omega$ containing the cones and closed under countable union and complementation). Let μ be the unique probability measure on $\mathcal{F}_{\mathrm{Br}}$ such that for all $u \in \{0,1\}^*$, $\mu(\mathrm{Cone}(u)) = 2^{-|u|}$. The existence and unicity of μ are guaranteed by Carathéorody's extension theorem [3]. For all $0 < p < 1$, a probability measure μ_p is similarly defined by taking $\mu_p(\mathrm{Cone}(u)) = p^{|u|_0}(1-p)^{|u|_1}$ where $|u|_0$ and $|u|_1$ respectively designate the number of occurrences of 0 and 1 in u. In particular, the measure μ corresponds to $\mu_{1/2}$.

B. Tree Automata and Regular Tree Languages

A *tree automaton* $\mathcal{A}$ is a tuple $\langle A, Q, q_{ini}, \Delta, \mathrm{Acc}\rangle$ where A is the *input alphabet*, Q is the finite *set of states*, $q_{\mathrm{ini}} \in Q$ is the *initial state*, $\Delta \subseteq Q \times A \times (Q \times Q)$ is the *transition relation* and $\mathrm{Acc} \subseteq Q^\omega$ is the *acceptance condition*. In the following, we use the notation $q \xrightarrow{a} (q_0, q_1)$ as a shorthand for $(q, a, (q_0, q_1)) \in \Delta$. An automaton is *deterministic* iff $q \xrightarrow{a} (q_0, q_1)$ and $q \xrightarrow{a} (q'_0, q'_1)$ implies $q_0 = q'_0$ and $q'_1 = q'_1$. An automaton is *complete* iff, for all $q \in Q$ and $a \in A$ there is at least one pair $(q_0, q_1) \in Q^2$ such that $q \xrightarrow{a} (q_0, q_1)$.

Given an A-labelled tree t, a *run* of $\mathcal{A}$ over t is a Q-labelled tree ρ such that

- the root is labelled by the initial state, *i.e.* $\rho(\varepsilon) = q_{\mathrm{ini}}$;
- for all nodes u, $(\rho(u), t(u), \rho(u \cdot 0), \rho(u \cdot 1)) \in \Delta$.

A branch $\pi \in \{0,1\}^\omega$ is *accepting* in the run ρ iff $\rho(\pi) \in \mathrm{Acc}$. A run ρ is *accepting* if all its branches are accepting. Finally, a tree t is *accepted* if there exists an accepting run of $\mathcal{A}$ over t. The set of all trees accepted by $\mathcal{A}$ is denoted $L(\mathcal{A})$.

We consider the following classical acceptance conditions:

[1] In the finite word case, the almost-sure and positive acceptation are trivial as the set of runs for a given word is finite.

[2] Indeed, let L be accepted by a Büchi automaton with the positive acceptation. As one can complement, there is a Büchi automaton with the positive acceptation $\mathcal{A}$ such that $\overline{L}$ is the language accepted by $\mathcal{A}$. If one sees $\mathcal{A}$ as a co-Buchi automaton $\mathcal{B}$ (final states become the forbidden ones) with a almost-sure acceptation, $\mathcal{B}$ accepts a word iff $\mathcal{A}$ does not. Hence, $\mathcal{B}$ recognises the complement of $\overline{L}$, namely L.

- A *reachability* condition is given by a subset $F \subseteq Q$ of final states by letting $Reach(F) = Q^*FQ^\omega$, *i.e.* a branch is accepting if it contains a final state.
- A *Büchi* condition is given by a subset $F \subseteq Q$ of final states by letting $Buchi(F) = (Q^*F)^\omega$, *i.e.* a branch is accepting if it contains infinitely many final states.
- A *co-Büchi* condition is given by a subset $F \subseteq Q$ of forbidden states by letting $coBuchi(F) = Q^*(Q\backslash F)^\omega$, *i.e.* a branch is accepting it contains finitely many forbidden states.
- A *parity* condition is given by a colouring mapping $\mathrm{Col} : Q \to \mathbb{N}$ by letting $Parity = \{q_0q_1q_2\cdots \mid \liminf(\mathrm{Col}(q_i))_i \text{ is even}\}$, *i.e.* a branch is accepting if the smallest colour appearing infinitely often is even.

All these conditions are examples of ω-regular acceptance condition, *i.e.* Acc is regular set of ω-words [14]. However, the parity condition is expressive enough to capture the general case of an arbitrary ω-regular condition. First, one considers a *deterministic* parity word automaton recognising Acc, and then takes the synchronised product of this automaton with the tree automaton. This leads to a parity tree automaton accepting the same language [14].

When it is clear from the context, we may replace, in the description of $\mathcal{A}$, Acc by F (for a reachability, Büchi or co-Büchi condition) or Col (for a parity condition), and we shall refer to the automaton as a reachability (*resp.* Büchi, co-Büchi, parity) tree automaton. A set L of trees is a *regular* language if there exists a parity tree automaton $\mathcal{A}$ such that $L = L(\mathcal{A})$. The class of regular tree languages is robust, as illustrated by the following statement.

***Theorem* 1.** *[16],[9] The class of regular tree languages is a Boolean algebra.*

A regular tree languages is non-empty iff it contains a regular tree. Testing the emptiness of a regular tree language (defined by a given parity automaton) is in $\mathrm{NP} \cap \mathrm{co}-\mathrm{NP}$.

C. Markov Decision Process

1) Perfect information setting: A *probability distribution* over a countable set X is a mapping $d : X \to [0,1]$ such that $\sum_{x\in X} d(x) = 1$. In the sequel we denote by $\mathcal{D}(X)$ the set of probability distributions over X.

An *arena* is a tuple $\mathcal{G} = \langle S, s_{\mathrm{ini}}, \Sigma, \zeta\rangle$ where S is a countable set of states, s_{ini} is an initial state, Σ is a finite set of *actions* and $\zeta : S \times \Sigma \to \mathcal{D}(S)$ is the transition (total) function.

A play in such an arena proceeds as follows. It starts in state s_{ini} and Éloïse picks an action σ, and a successor state is chosen according to the probability distribution $\zeta(s_{\mathrm{ini}}, \sigma)$. Then Éloïse chooses a new action and the state is updated and so on forever. Hence a *play* is an infinite sequence $s_0s_1s_2\cdots \in S^\omega$ such that $s_0 = s_{\mathrm{ini}}$ and for every $i \geqslant 0$, there exists a $\sigma \in \Sigma$ with $\zeta(s_i, \sigma)(s_{i+1}) > 0$. In the sequel we refer to a prefix of a play as a *partial play* and we denote by Plays the set of all plays.

A *(pure) strategy*[3] for Éloïse is a function $\varphi : S^* \to \Sigma$ assigning to every partial play an action. Of special interest are those strategies that does not require memory: a strategy φ is *memoryless* iff $\varphi(\lambda \cdot s) = \varphi(\lambda' \cdot s)$ for all partial play λ, λ' and all states s (*i.e.* φ only depends on the current state). A play $\lambda = s_0s_1s_2\cdots$ is *consistent* with a strategy φ if $\zeta(s_i, \varphi(v_0\cdots v_i))(s_{i+1}) > 0$, for all $i \geqslant 0$.

Now, for any partial play λ, the *cylinder* for λ is the set $\mathrm{Cyl}(\lambda) = \lambda S^\omega$. Let $\mathcal{F}_P$ be the σ-algebra generated by the set of cylinders. Then, $(\mathrm{Plays}, \mathcal{F}_P)$ is a measurable space.

A strategy φ induces a probability space over $(\mathrm{Plays}, \mathcal{F}_P)$. Indeed, one defines a measure μ_φ on cylinders and then uniquely extends it to a probability measure on $\mathcal{F}_P$ using the Carathéodory's unique extension theorem. For this, we first define inductively μ_φ on cylinders:

- as all plays start from s_{ini}, we let $\mu_\varphi(\mathrm{Cyl}(s_{\mathrm{ini}})) = 1$;
- for any partial play λ ending in some state s, we let $\mu_\varphi(\mathrm{Cyl}(\lambda\cdot s')) = \mu_\varphi(\mathrm{Cyl}(\lambda)) \cdot \zeta(s, \varphi(\lambda))(s')$.

We also denote by μ_φ the unique extension of μ_φ to a probability measure on $\mathcal{F}$. Then $(\mathrm{Plays}, \mathcal{F}_P, \mu_\varphi)$ is a probability space.

An *objective* is a measurable set $\mathcal{O} \subseteq$ Plays: a play is winning if it belongs to $\mathcal{O}$. A *Markov decision process (MDP, aka* one-and-half-player game*)* is a pair $(\mathcal{G}, \mathcal{O})$ where $\mathcal{G}$ is an arena and $\mathcal{O}$ is an objective. In the sequel we should focus on ω-regular objectives (which are easily seen to be measurable), whose definitions are the same as for the acceptance condition on tree automata (the only difference is that we may have an infinite set of states).

A strategy φ is *almost-surely winning* if $\mu_\varphi(\mathcal{O}) = 1$. If such a strategy exists, we say that Éloïse *almost-surely wins* $\mathbb{G}$. The *value* of $\mathbb{G}$ is defined as $\mathrm{Val}(\mathbb{G}) = \sup_\varphi \mu_\varphi(\mathcal{O})$, and a strategy φ is *optimal* iff $\mathrm{Val}(\mathbb{G}) = \mu_\varphi(\mathcal{O})$.

When the set of actions Σ is reduced to one element, the MDP $(\mathcal{G}, \mathcal{O})$ is called a *Markov chain* and we omit the unique action from all the definitions. The set of Plays is called the set of *traces* of the Markov chain and is denoted Traces. We write $\mu_{\mathcal{G}}$ the probability measure associated with the unique strategy. We say that the Markov chain *almost-surely* fulfils its objective if $\mu_{\mathcal{G}}(\mathcal{O}) = 1$.

MDPs over finite graphs enjoys many good properties.

***Theorem* 2.** *[8], [6] Let* $\mathbb{G}$ *be an MDP over a finite arena with a parity objective. Then, one can decide in polynomial time whether Éloïse almost-surely wins. Moreover, Éloïse always has an optimal memoryless strategy.*

2) Imperfect information setting: Now we consider the case where Éloïse has imperfect information about the current state. For this, we consider an equivalence relation $\sim$ over S. We let $[s]_\sim$ be the equivalence class of s for $\sim$ and $S/_\sim$ be the set of equivalence classes of $\sim$ over S.

[3] We do not consider here randomised strategies as in the setting of this paper they are useless. Note that for finite MDP, optimal strategies — when exists — can always be chosen to be pure.

The intuitive meaning of $\sim$ is that two states $s_1 \sim s_2$ cannot be distinguished by Éloïse. We easily extend $\sim$ to partial plays: $s_0 s_1 \cdots s_n \sim s'_0 s'_1 \cdots s'_n$ iff $s_i \sim s'_i$ for all $i = 0, \cdots, n$. If $\lambda_1 \sim \lambda_2$ Éloïse cannot distinguish λ_1 from λ_2: therefore she should behave the same in both of them.

Hence, we should only consider so-called observation-based strategies. An *observation-based (pure) strategy* is a function $\varphi : (S/_\sim)^* \to \Sigma$, *i.e.*, to choose her next action, Éloïse considers the sequence of observations she got so far[4]. In particular, an observation-based strategy φ is such that $\varphi(\lambda) = \varphi(\lambda')$ whenever $\lambda \sim \lambda'$. In this context, a *memoryless strategy* is a function from $S/_\sim \to \mathcal{D}(\Sigma)$, *i.e.* it only depend on the current equivalence class.

A *partial observation Markov decision process (POMDP, aka* one-and-half-player imperfect information game*)* is a triple $(\mathcal{G}, \sim, \mathcal{O})$ where $\mathcal{G}$ is an arena, $\sim$ is an equivalence relation over states and $\mathcal{O}$ is an objective. We say that Éloïse *almost-surely wins* $\mathbb{G}$ if she has an almost-surely winning *observation-based* strategy. Finally the *value* of $\mathbb{G}$ is defined as $\mathrm{Val}(\mathbb{G}) = \sup_\varphi \mu_\varphi(\mathcal{O})$ where φ ranges over observation-based strategies; optimality is defined as previously.

The following decidability results are known for POMDP:

***Theorem* 3.** *[1] In a POMDP with a Büchi objective, deciding whether Éloïse almost-surely wins is ExpTime-complete. Moreover if Éloïse has an almost-surely winning strategy, she has an almost-surely winning strategy with finite memory. In a POMDP with a co-Büchi objective, it is undecidable whether Éloïse almost-surely wins.*

***Remark* 1.** *The results in Theorem 2 and 3 do not depend on the encoding of probability distributions, as the only relevant information is which probabilities are non zero.*

III. Qualitative Tree Languages

A. Definition

In the classical definition, a run of a tree automaton $\mathcal{A}$ is accepting if all its branches satisfy the acceptance condition. In this article, we introduce a more relaxed notion of acceptation: a run is qualitatively accepting if almost every (in the sense of the measure μ) branch in it is accepting. More formally, consider a tree automaton $\mathcal{A}$ with an ω-regular acceptance condition. A run ρ of $\mathcal{A}$ is *qualitatively accepting* if the set $\mathrm{AccBr}(\rho) = \{\pi \in \{0,1\}^\omega \mid \rho(\pi) \in \mathrm{Acc}\}$ has measure 1, *i.e.* $\mu(\mathrm{AccBr}(\rho)) = 1$. Note that, thanks to Proposition 1 below, the set $\mathrm{AccBr}(\rho)$ is indeed measurable. A tree t is *qualitatively accepted* if there exists a qualitatively accepting run of $\mathcal{A}$ over t and the set of all trees qualitatively accepted by $\mathcal{A}$ is denoted $L_{\mathrm{Qual}}(\mathcal{A})$. Finally, a *qualitative tree language* is a set L of trees such that there is a parity automaton $\mathcal{A}$ such that $L_{\mathrm{Qual}}(\mathcal{A}) = L$.

***Proposition* 1.** *Let $\mathcal{A}$ be a tree automaton equipped with an ω-regular acceptance condition, and let ρ be a run of $\mathcal{A}$. Then the set $\mathrm{AccBr}(\rho)$ is measurable.*

***Example* 1.** *Let $\mathcal{L}_{\mathtt{a}}$ be the language of $\{\mathtt{a}, \mathtt{b}\}$-labelled trees whose set of branches containing at least one $\mathtt{a}$ has measure 1. This language is recognised by the following reachability deterministic automaton $\mathcal{A} = \langle \{\mathtt{a}, \mathtt{b}\}, \{q_{\mathrm{ini}}, q_f\}, q_{\mathrm{ini}}, \Delta, \{q_f\} \rangle$ where: $\Delta = \{q_{\mathrm{ini}} \xrightarrow{b} (q_{\mathrm{ini}}, q_{\mathrm{ini}}), q_{\mathrm{ini}} \xrightarrow{a} (q_f, q_f), q_f \xrightarrow{a} (q_f, q_f), q_f \xrightarrow{b} (q_{\mathrm{ini}}, q_{\mathrm{ini}})\}$.*

If one considers $\mathcal{A}$ as a Büchi *automaton, the accepted language consists of those trees whose set of branches containing infinitely many $\mathtt{a}$ has measure 1.*

***Example* 2.** *Let $\mathcal{L}_1$ be the language of trees t such that in almost every branch, there is a node u labelled by $\mathtt{a}$ such that the subtree $t[u]$ has only $\mathtt{a}$ on its leftmost branch. This language is recognised by the non-deterministic reachability automaton $\mathcal{A} = \langle A, Q, q_w, \Delta, \{q_{acc}\} \rangle$ with $A = \{\mathtt{a}, \mathtt{b}\}$, $Q = \{q_w, q_l, q_{acc}, q_{rej}\}$, and Δ contains the following transitions: $q_w \xrightarrow{\_} (q_w, q_w)$, $q_w \xrightarrow{\_} (q_l, q_{acc})$, $q_l \xrightarrow{a} (q_l, q_{acc})$, $q_l \xrightarrow{b} (q_{rej}, q_{rej})$, $q_{acc} \xrightarrow{\_} (q_{acc}, q_{acc})$, $q_{rej} \xrightarrow{\_} (q_{rej}, q_{rej})$ (here _ is a shorthand for an arbitrary letter). Intuitively, the automaton can wait in state q_w as long as it wants. It can at some node u use the second transition: this leads to accept (all branches in) the cone going through the right son, as well as those subtrees rooted at the right of the left-most branch going through u if this branch does not contain a node labelled $\mathtt{b}$ (note that the left-most branch going through u will be rejecting, but this does not affect the measure as there are only countably many such branches). If the left-most branch going through u contains a $\mathtt{b}$ at node v then the cone of branches going through v is rejecting.*

***Remark* 2.** *The choice of the measure μ though natural is arbitrary. Considering the measure μ_p for some $0 < p < \frac{1}{2}$ would not affect the results obtained in this article (provided that definitions of the games are modified accordingly). However note that changing the measure does change the accepted language for a given automaton. For instance, consider the deterministic Büchi automaton $\mathcal{A}$ of Example 1 and the $\{\mathtt{a}, \mathtt{b}\}$-labeled tree such that for all $u \in \{0,1\}^*$ by $t(u) = \mathtt{a}$ if and only if $\mu_p(\mathrm{Cone}(u)) > \mu(\mathrm{Cone}(u))$.*

A consequence of the proof of Kakutani's theorem in [12] is that $\mu_p(\mathrm{AccBr}(\rho_0)) = 1$ and $\mu(\mathrm{AccBr}(\rho_0)) = 0$ where ρ_0 designates the unique run of $\mathcal{A}$ over t.

Remark 3. *A more general definition is to associate with any letter $\mathtt{a}$ in the alphabet a pair $(p^0_{\mathtt{a}}, p^1_{\mathtt{a}}) \in [0,1]^2$ with $p^0_{\mathtt{a}} + p^1_{\mathtt{a}} = 1$ and then to define the measure of a cone in a tree t by letting $\mu(\mathrm{Cone}(u_1 \cdots u_n)) = p^{u_1}_{t(\varepsilon)} p^{u_2}_{t(u_1)} \cdots p^{u_n}_{t(u_1 \cdots u_{n-1})}$. Intuitively, the node label determines the respective weights of the left and right sons in the definition of the measure. In particular the measure μ_p is the one obtained by letting $(p^0_{\mathtt{a}}, p^1_{\mathtt{a}}) = (p, 1-p)$ for all letters $\mathtt{a}$ in the alphabet.*

[4] By abuse of notation, we shall write $\varphi(s_0 \cdots s_n)$ to mean $\varphi([s_0]_\sim \cdots [s_n]_\sim)$

Again, with such a measure the results obtained in this article (provided that definitions of the games are modified accordingly) remains correct.

For the same reasons as for regular tree languages, the parity condition is expressive enough to capture any ω-regular conditions: for any automaton $\mathcal{A}$ with an ω-regular acceptance condition, there exists a parity automaton $\mathcal{B}$ such that $L_{\mathrm{Qual}}(\mathcal{A}) = L_{\mathrm{Qual}}(\mathcal{B})$. Thanks to the following proposition, we can only focus on complete automata.

***Proposition* 2.** *For any tree automaton $\mathcal{A}$ with an ω-regular acceptance condition, there exists a* complete *tree automaton $\mathcal{B}$ with the same acceptance condition and such that $L_{\mathrm{Qual}}(\mathcal{A}) = L_{\mathrm{Qual}}(\mathcal{B})$.*

Unsurprisingly determinism is a restriction.

***Proposition* 3.** *There is a qualitative tree language that cannot be qualitatively accepted by any deterministic automaton.*

B. Pumping Lemma

Let t be a tree and $u \in \{0,1\}^*$ be a node. A pair $\Delta = (t, u)$ is called a *pointed tree*. With a pointed tree $\Delta_1 = (t_1, u_1)$ and a tree t_2, we associate a new tree, $\Delta_1 \cdot t_2$, by plugging t_2 in t_1 instead of the subtree rooted at u_1. Formally, $\Delta_1 \cdot t_2(u) = t_1(u)$ if u_1 is not a prefix of u and $\Delta_1 \cdot t_2(u) = t_2(u')$ if $u = u_1 u'$ for some $u' \in \{0,1\}^*$. We can also define the product of two pointed trees $\Delta_1 = (t_1, u_1)$ and $\Delta_2 = (t_2, u_2)$ by letting $\Delta_1 \cdot \Delta_2 = (\Delta_1 \cdot t_2, u_1 \cdot u_2)$. Finally, with a pointed tree $\Delta = (t, u)$, we associate a tree Δ^ω by taking an ω-iteration of the product: $\Delta^\omega(v) = t(v')$ where v' is the shortest word s.t. $v = u^k v'$ for some $k \geqslant 0$.

Qualitative tree languages enjoy a pumping lemma, which contrasts with regular tree languages.

***Lemma* 1.** *Let $\mathcal{A}$ be an n-states parity automaton, t be a tree in $L_{\mathrm{Qual}}(\mathcal{A})$ and u be a node of depth greater that n. Then there exists three pointed trees Δ_1, Δ_2 and Δ_3 such that $t = \Delta_1 \cdot \Delta_2 \cdot \Delta_3 \cdot t[u]$ and $\Delta_1 \cdot \Delta_2^\omega \in L_{\mathrm{Qual}}(\mathcal{A})$.*

C. Closure Properties

We now investigate the closure properties of qualitative tree languages under Boolean operations.

***Proposition* 4.** *Qualitative tree languages are closed under union and intersection.*

Unsurprisingly, qualitative tree languages are not closed under complement. This is a simple consequence of Lemma 1

***Proposition* 5.** *Qualitative tree languages are not closed under complement.*

Proof (sketch): One shows that the complement $\overline{\mathcal{L}_{\mathrm{a}}}$ of the language $\mathcal{L}_{\mathrm{a}}$ of Example 1 does not satisfy the pumping lemma. For this, consider a tree that contains only $\mathtt{a}$'s except on a subtree rooted deep enough where all nodes are labelled by $\mathtt{b}$. This tree belongs to $\overline{\mathcal{L}_{\mathrm{a}}}$, but if one pumps and removes the subtrees made of $\mathtt{b}$'s one gets a tree in $\mathcal{L}_{\mathrm{a}}$. ■

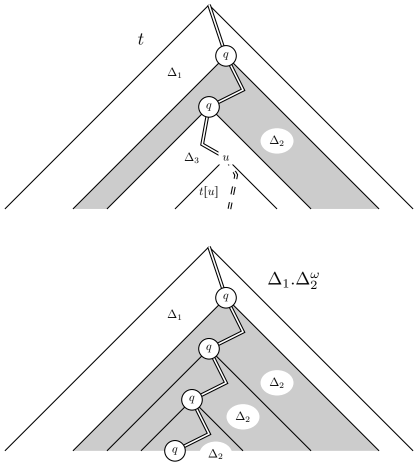


Figure 1. Pumping Lemma

D. Emptiness Problem

It is well known that tree automata (as acceptors of regular languages) and two-player (perfect information) game are closely related [11], [10]. In particular, the emptiness problem for regular tree languages and the problem of deciding the winner in a parity game on a finite graph are polynomially equivalent. From the proof of this result also follows that a regular tree language is non-empty iff it contains a regular tree.

We show that a similar connection exists between tree automata as acceptors of qualitative tree languages and MDPs. For this, fix a parity tree automaton $\mathcal{A} = \langle A, Q, q_{\mathrm{ini}}, \Delta, \mathrm{Col}\rangle$ and a tree t. Consider the arena $G_{\mathcal{A},t} = \langle S, s_{\mathrm{ini}}, \Sigma, \zeta\rangle$ where $S = Q \times \{0,1\}^* \cup \{\perp\}$, $s_{\mathrm{ini}} = (q_{\mathrm{ini}}, \varepsilon)$, $\Sigma = \Delta$ and ζ is defined as follows. First we let $d_\perp$ be the distribution defined by $d_\perp(s) = 1$ if $s = \perp$ and $d_\perp(s) = 0$ otherwise, and, for all $q_0, q_1 \in Q$ and $u \in \{0,1\}^*$, we let $d_{q_0,q_1,u}$ be the distribution such that $d_{q_0,q_1,u}(q_0, u0) = d_{q_0,q_1,u}(q_1, u1) = 1/2$ and $d_{q_0,q_1,u}(s) = 0$ for all other $s \in S$. Then we let $\zeta((q,u),(q',a,q_0,q_1)) = d_\perp$ if $q \neq q'$ or $a \neq t(u)$, $\zeta((q,u),(q,t(u),q_0,q_1)) = d_{q_0,q_1,u}$ and $\zeta(\perp,\sigma) = d_\perp$ for all $\sigma \in \Delta$. Finally, we define a colouring function ρ by letting $\rho((q,u)) = \mathrm{Col}(q)$ and $\rho(\perp) = 1$, and we call $\mathbb{G}_{\mathcal{A},t} = (\mathcal{G}_{\mathcal{A},t}, \mathcal{O}_\rho)$ the MDP equipped with the parity objective $\mathcal{O}_\rho$ defined by ρ. Then, the following holds:

***Theorem* 4.** *The tree t belongs to $L_{\mathrm{Qual}}(\mathcal{A})$ iff Éloïse almost-surely wins in $\mathbb{G}_{\mathcal{A},t}$.*

Proof (sketch): The key idea here is to note that strategies for Éloïse in $\mathbb{G}_{\mathcal{A},t}$ are in bijection with runs of $\mathcal{A}$ over t, and that this map preserves the measure. In particular, the set of winning plays when Éloïse follows a strategy has the same measure than the set of accepting branches in the corresponding run. ■

Consider the (finite) arena $G_{\mathcal{A}} = \langle S, s_{\text{ini}}, \Sigma, \zeta \rangle$ where $S = Q \times \{0,1\} \cup \{q_{\text{ini}}, \bot\}$, $s_{\text{ini}} = q_{\text{ini}}$, $\Sigma = \Delta$ and ζ is defined as follows. First we let $d_\bot$ be the distribution defined by $d_\bot(s) = 1$ if $s = \bot$ and $d_\bot(s) = 0$ otherwise, and, for all $q_0, q_1 \in Q$, we let d_{q_0,q_1} be the distribution such that $d_{q_0,q_1}((q_0,0)) = d_{q_0,q_1}((q_1,1)) = 1/2$ and $d_{q_0,q_1}(s) = 0$ for all other $s \in S$. Then we let $\zeta((q,i),(q,a,q_0,q_1)) = d_{q_0,q_1}$, $\zeta((q,i),(q',a,q_0,q_1)) = d_\bot$ if $q \neq q'$, $\zeta(q_{\text{ini}},(q_{\text{ini}},a,q_0,q_1)) = d_{q_0,q_1}$, $\zeta(q_{\text{ini}},(q,a,q_0,q_1)) = d_\bot$ if $q \neq q_{\text{ini}}$, and $\zeta(\bot,\sigma) = d_\bot$ for all $\sigma \in \Delta$. Finally, we define a colouring function ρ by letting $\rho((q,i)) = \text{Col}(q)$ and $\rho(\bot) = 1$, and we call $\mathbb{G}_{\mathcal{A}} = (\mathcal{G}_{\mathcal{A}}, \mathcal{O}_\rho)$ the MDP equipped with the parity objective $\mathcal{O}_\rho$ defined by ρ. Then, the following hold:

***Theorem* 5.** *The language $L_{\text{Qual}}(\mathcal{A})$ is non empty iff Éloïse almost-surely wins in $\mathbb{G}_{\mathcal{A}}$ from q_{ini}.*

Proof (sketch): The idea is the same as for the proof of Theorem 4 except that now strategies are seen as pairs made of an A-labeled tree and a run of $\mathcal{A}$ over it. Again, the value of the strategy is the same as the measure of the set of accepting branches in the run, which allows to conclude. ■

***Corollary* 1.** *Let $\mathcal{A}$ be a parity tree automaton. Then one can decide whether $L_{\text{Qual}}(\mathcal{A}) = \emptyset$ in polynomial time. Moreover, if $L_{\text{Qual}}(\mathcal{A}) \neq \emptyset$, it contains a regular tree, and such a tree can be constructed in polynomial time.*

Proof (sketch): Complexity follows from theorems 5 and 2. Regular trees are associated with memoryless strategy: as those are sufficient to play optimally in finite MDP, it permits to conclude. ■

E. Regular Tree Languages and Qualitative Tree Languages are Incomparable

In this section, we prove that regular tree languages and qualitative tree languages are incomparable.

***Proposition* 6.** *There is a regular tree language that is not qualitative.*

Proof (sketch): The regular language of those $\{\mathtt{a},\mathtt{b}\}$-labelled trees containing at least one node labeled by $\mathtt{b}$, does not satisfies Lemma 1, hence it is not qualitative. ■

***Theorem* 6.** *There is a qualitative tree language that is not regular.*

Proof: Let $\mathcal{L}_{\mathtt{a}}$ be the language of trees whose set of branches containing at least one $\mathtt{a}$ has measure 1. This language is qualitative as noticed in Example 1. In the sequel, we prove that $\mathcal{L}_{\mathtt{a}}$ is not regular.

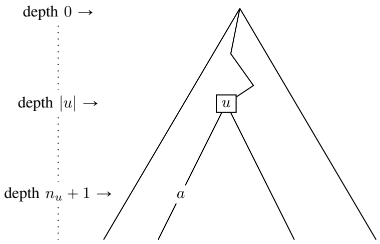


Figure 2. The tree t_0

We first prove that, for any regular tree t, if there is no cone in t whose branches only contain the letter $\mathtt{b}$, then $t \in \mathcal{L}_{\mathtt{a}}$. Let t be a regular tree, we can assume w.l.o.g. that if there is a node labelled by $\mathtt{a}$ then all its descendants are labeled by $\mathtt{a}$. Then the property "there is no cone in t whose branches only contain $\mathtt{b}$" is the same as "every subtree contains a subtree made only of $\mathtt{a}$". Let $X_1, \ldots, X_n$ be the n different subtrees of t, and for all i, let μ_i be the measure of the set of branches containing $\mathtt{a}$ in X_i (we call it the value of X_i). We can assume that $\forall i\ \mu_1 \leqslant \mu_i$. If X_{i_1} and X_{i_2} are the two sons of X_1, we know that $\mu_1 = \frac{\mu_{i_1} + \mu_{i_2}}{2}$. Since $\mu_1 \leqslant \mu_i$ for $i = i_1, i_2$, $\mu_{i_1} = \mu_{i_2} = \mu_1$. Hence we can prove by induction that for all X_i of minimal value, all the subtrees of X_i have minimal value too. Since there is a subtree of $\mathtt{a}$ (of value 1) in X_1, $\mu_1 = 1$ hence for all i, $\mu_i = 1$, hence the value of t is 1, hence $t \in \mathcal{L}_{\mathtt{a}}$.

We assume by contradiction that $\mathcal{L}_{\mathtt{a}}$ is regular. The closure properties of regular tree languages implies that the following language $\mathcal{L}$ is also regular:

$$\mathcal{L} = \Big\{\ t \ \Big|\ t \notin \mathcal{L}_{\mathtt{a}} \wedge \textit{“there is no cone} \\ \textit{in } t \textit{ whose branches only contain } \mathtt{b}\textit{”}\ \Big\}$$

Using our previous characterisation of regular trees in $\mathcal{L}_{\mathtt{a}}$ it follows that $\mathcal{L}$ does not contain any regular tree, hence $\mathcal{L}$ is empty (Theorem 1). Then, to raise a contradiction, we build a (non-regular) tree $t \in \mathcal{L}$.

For every node $u \in \{0,1\}^*$, we let n_u be the integer whose binary representation is $1.u$. The tree t is defined as follows: let $v \in \{0,1\}^*$, if there exists some u such that $v = u.0^{n_u+1-|u|}$ then $t_0(v) = \mathtt{a}$, otherwise $t_0(v) = \mathtt{b}$ (see Figure 2). We now establish that $t \notin \mathcal{L}_{\mathtt{a}}$. First note that the set of branches in t that contains at least one $\mathtt{a}$ is obtained by taking the union of those cones $\mathcal{C}(u)$ such that $t(u) = \mathtt{a}$. Then remark that, for every level ℓ, there is one and only one node u of depth ℓ labelled by $\mathtt{a}$ (except for $\ell = 0, 1$ where there are no such u). Thus we can bound the measure μ of the set of branches in t that contains at least one $\mathtt{a}$:

$$\mu \ \leqslant\ \sum_{\ell=2}^{+\infty} 2^{-\ell} \ =\ \frac{1}{2}$$

This proves that $t \notin \mathcal{L}_{\mathtt{a}}$. Moreover, it follows from the definition, that for every node u, there is a branch (the leftmost one) in the cone $\mathcal{C}(u)$ that contains an $\mathtt{a}$, hence $t \in \mathcal{L}$, which contradict the fact that $\mathcal{L}$ is empty. ∎

F. The Value of a Tree May not be Reached

So far we defined qualitative acceptance of a tree by the existence of a run whose set of accepting branches has measure 1. We can refine this notion by defining the *value* of a tree as follows. For $\mathcal{A}$ a tree automaton, and t a tree we let

$$\mathrm{Val}_{\mathcal{A}}(t) = \sup_{\rho_t \text{ run of } \mathcal{A} \text{ over } t} \mu(\mathrm{AccBr}(\rho_t))$$

In particular $L_{\mathrm{Qual}}(\mathcal{A})$ is the set of trees t whose value is 1 and is reached for some run (*i.e.* the $\sup$ is a $\max$). The following result proves that the value may not be reached by some run.

***Theorem* 7.** *There is a reachability automaton $\mathcal{A}$ and a tree t such that $\mathrm{Val}_{\mathcal{A}}(t) = 1$ but $t \notin L_{\mathrm{Qual}}(\mathcal{A})$.*

Actually, the proof of Theorem 4 leads the following.

***Theorem* 8.** *Let $\mathcal{A}$ be a parity tree automaton and let t be a tree. Then $\mathrm{Val}_{\mathcal{A}}(t) = \mathrm{Val}(\mathbb{G}_{\mathcal{A},t})$.*

IV. Beyond non-deterministic automata: the probabilistic setting

Following [16] for finite words and [2], [1] for infinite words we investigate probabilistic automata on infinite trees. That is the set of transitions of an automaton is replaced by a probability distribution over the set of all transitions which induces a probability measure on the set of runs of the automaton. Now, a tree is accepted if almost every run over the input tree is accepting. For the run, we may use either the classical or the qualitative acceptance criterion.

A. Definitions

A *probabilistic tree automaton* $\mathcal{A}$ is a tuple $\langle A, Q, q_{ini}, \delta, \mathrm{Acc}\rangle$ where A is the *input alphabet*, Q is a finite *set of states*, $q_{\mathrm{ini}} \in Q$ is the *initial state*, $\mathrm{Acc} \subseteq Q^\omega$ is the *acceptance condition* and δ is a mapping from $Q \times A \times Q \times Q$ to $[0,1]$ such that for all $q \in Q$ and $\mathtt{a} \in A$, $\sum_{q_1,q_2 \in Q} \delta(q, \mathtt{a}, q_1, q_2) = 1$. Intuitively, the value $\delta(q, \mathtt{a}, q_1, q_2)$ is the probability for a transition $q \xrightarrow{\mathtt{a}} (q_1, q_2)$ to be used by the automaton when it is in state q and reads the symbol $\mathtt{a}$.

This probability distribution on the transitions induces a probability measure on the set of runs of $\mathcal{A}$. In this setting, a *run* of $\mathcal{A}$ is simply a Q-labeled tree whose root is labeled by the initial state q_{ini}. We denote by $\mathrm{Runs}(\mathcal{A})$ (or simply Runs if $\mathcal{A}$ is clear from the context) the set of all runs of $\mathcal{A}$. We denote by $\mathrm{AccRuns}(\mathcal{A})$ the set of accepting runs of $\mathcal{A}$ and by $\mathrm{QualAccRuns}(\mathcal{A})$ the set of qualitatively accepting runs of $\mathcal{A}$.

Let t be a tree. A *partial run* λ is a mapping from $\{0,1\}^n$ to Q for some $n \geqslant 0$ with $\lambda(\varepsilon) = q_{\mathrm{ini}}$. A partial run λ is a prefix of a run ρ if for all $w \in \mathrm{Dom}(\lambda)$, $\lambda(w) = \rho(w)$. The cylinder for a partial run λ, denoted $\mathrm{Cyl}_{\mathcal{A}}(\lambda)$, is the set of runs of $\mathcal{A}$ admitting λ as a prefix. Let $\mathcal{F}_R$ be the σ-algebra generated by the cylinders. By Carathéorody's extension theorem, there exists a unique probability measure μ_t on the measurable space $(\mathrm{Runs}, \mathcal{F}_R)$ such for all partial run $\lambda : \{0,1\}^n \to Q$, $\mu_t(\mathrm{Cyl}_{\mathcal{A}}(\lambda))$ is equal to $\prod_{w \in \{0,1\}^{n-1}} \delta(\lambda(w), t(w), \lambda(w0), \lambda(w1))$. Note that both μ_t and $(\mathrm{Runs}, \mathcal{F}_R)$ depend on t.

***Proposition* 7.** *For all probabilistic tree automata $\mathcal{A}$ with an ω-regular acceptance condition, the sets $\mathrm{AccRuns}(\mathcal{A})$ and $\mathrm{QualAccRuns}(\mathcal{A})$ are measurable.*

A tree t is (almost-surely) accepted by $\mathcal{A}$ with the classical semantic if almost all runs of $\mathcal{A}$ on t are accepting (*i.e.* $\mu_t(\mathrm{AccRuns}(\mathcal{A})) = 1$). We denote by $L^{=1}(\mathcal{A})$ the set of trees accepted by $\mathcal{A}$ with the classical semantics for runs. A tree t is (almost-surely) accepted by $\mathcal{A}$ with the qualitative semantic if almost all runs of $\mathcal{A}$ on t are accepting (*i.e.* $\mu_t(\mathrm{QualAccRuns}(\mathcal{A})) = 1$). We note $L^{=1}_{\mathrm{Qual}}(\mathcal{A})$ the set of trees accepted by $\mathcal{A}$ with the qualitative semantics for runs.

***Remark* 4.** *Our motivation for considering almost-sure acceptation and not positive acceptation is discussed in Section V.*

The almost-sure acceptation of a tree t by an automaton $\mathcal{A}$ for the qualitative semantic can be defined by integrating the mapping $f_{\mathcal{A}} : \mathrm{Runs} \times \mathrm{Br} \to [0,1]$ associating to a pair $(\rho, \pi) \in \mathrm{Runs} \times \mathrm{Br}$ the value 1 if $\rho(\pi)$ belongs to Acc and 0 otherwise.

***Proposition* 8.** *Let $\mathcal{A}$ be a probabilistic tree automaton with an ω-regular acceptance condition and let t be a tree. The mapping $f_{\mathcal{A}}$ is integrable in the product space $(\mathrm{Runs}(\mathcal{A}), \mathcal{F}_R, \mu_T) \otimes (\mathrm{Br}, \mathcal{F}_{\mathrm{Br}}, \mu)$ and we have:*

$$t \in L^{=1}_{\mathrm{Qual}}(\mathcal{A}) \Leftrightarrow \int f_{\mathcal{A}} d\mu_t \otimes \mu = 1$$

Proof sketch: First we show that $f_{\mathcal{A}}$ is measurable. Then, by Fubini's theorem [3, Thm 23.6 - p. 138], the mapping $g : \mathrm{Runs} \to [0,1]$ associating with a run $\rho \in \mathrm{Runs}$ the value $\int_{\mathrm{Br}} f_{\mathcal{A}}(\rho, \cdot) d\mu$ is measurable.

$$\begin{array}{lll} & t \text{ belongs to } L^{=1}_{\mathrm{Qual}}(\mathcal{A}) & \\ \text{iff} & \mu_t(g^{-1}(\{1\})) = 1 & \\ \text{iff} & \int_{\mathrm{Runs}} g \, d\mu_t = 1 & \\ \text{iff} & \int_{\mathrm{Runs}} \int_{\mathrm{Br}} f_{\mathcal{A}} d\mu d\mu_t = 1 & \text{by definition of } g \\ \text{iff} & \int_{\mathrm{Runs} \times \mathrm{Br}} f_{\mathcal{A}} d\mu_T \otimes \mu = 1 & \text{by Fubini's thm.} \end{array}$$

∎

We conclude this section with examples of languages accepted by probabilistic tree automata.

For an ω-word language $L \subseteq \{\mathtt{a}, \mathtt{b}\}^\omega$, we denote by $\mathrm{Path}^{=1}(L)$ the set of trees labeled by $\{\mathtt{a}, \mathtt{b}\}$ with almost all their branch labels in L (*i.e.* $\mu(\{\pi \in \mathrm{Br} \mid t(\pi) \in L\}) = 1$). It is easy to see that, for any ω-regular language L, the tree language $\mathrm{Path}^{=1}(L)$ is a qualitative tree language. More interesting, if L is almost-surely accepted by a probabilistic ω-word automaton[5] with an ω-regular acceptance condition, we can show that $\mathrm{Path}^{=1}(L)$ is accepted by a probabilistic tree automaton (with the qualitative semantic).

***Proposition* 9.** *Given a probabilistic ω-word automaton $\mathcal{B}$ with an ω-regular acceptance condition, there exists a probabilistic tree automaton $\mathcal{A}$ with the same acceptance condition such that $L^{=1}_{\mathrm{Qual}}(\mathcal{A})$ is equal to* $\mathrm{Path}^{=1}(L^{=1}(\mathcal{B}))$.

Proof (sketch): Let $\mathcal{B} = \langle A = \{\mathtt{a}, \mathtt{b}\}, Q, q_{ini}, \delta, \mathrm{Acc}\rangle$ be a probabilistic ω-word automaton with an ω-regular condition. Consider the probabilistic tree automaton $\mathcal{A}$ simulating $\mathcal{B}$ along all branches. Formally $\mathcal{A}$ is equal to $\langle A, Q, q_{ini}, \delta', F\rangle$ where, for all $p, q \in Q$ and $\mathtt{x} \in A$, $\delta'(p, \mathtt{x}, q, q) = \delta(p, \mathtt{x}, q)$ and otherwise is equal to 0.

Let t be a tree and let $g : \mathrm{Br} \to [0,1]$ be the mapping associating to a branch π the value $\int f_{\mathcal{A}}(\cdot, \pi)\, d\mu_t$. For any $\pi \in \mathrm{Br}$, $g(\pi) = 1$ if and only if $t(\pi)$ belongs to $L^{=1}(\mathcal{B})$.

$$\begin{array}{lll} & t \in L^{=1}_{\mathrm{Qual}}(\mathcal{A}) & \\ \Leftrightarrow & \int f_{\mathcal{A}} d\mu_t \otimes \mu = 1 & \text{by Proposition 8} \\ \Leftrightarrow & \int g d\mu = 1 & \text{by Fubini's thm} \\ \Leftrightarrow & g(\pi) = 1 \text{ almost everywhere} & \\ \Leftrightarrow & \mu(\{\pi \mid t(\pi) \in L^{=1}(\mathcal{B})\}) = 1 & \\ \Leftrightarrow & t \in \mathrm{Path}^{=1}(\mathcal{B}) & \end{array}$$

■

B. Acceptance games for qualitative probabilistic tree automata

Fix a probabilistic tree automaton $\mathcal{A} = \langle A, Q, q_{\mathrm{ini}}, \delta, \mathrm{Acc}\rangle$ and a tree t. We define a Markov chain $\mathcal{M}_{\mathcal{A},t} = (G_{\mathcal{A},t}, \mathcal{O}_{\mathrm{Acc}})$ such that $\mathcal{M}_{\mathcal{A},t}$ almost-surely fulfils its objective iff t belongs to $L^{=1}_{\mathrm{Qual}}(\mathcal{A})$. Compared with the acceptance game for qualitative tree automata, the transition is no longer chosen by Éloïse: it is now randomly chosen with the probability distribution given by $\mathcal{A}$. Hence we simply obtain a Markov chain instead of an MDP.

The arena $G_{\mathcal{A},t}$ is equal to $\langle S, s_{\mathrm{ini}}, \zeta\rangle$ where $S = Q \times \{0,1\}^* \cup \Delta \times \{0,1\}^*$ with $\Delta = Q \times Q \times Q$, $s_{\mathrm{ini}} = (q_{\mathrm{ini}}, \varepsilon)$ and $\zeta : S \mapsto D(S)$ is defined as follows. For all $w \in \{0,1\}^*$ and all $q \in Q$, $\zeta((q, w), (q, q_0, q_1)) = \delta(q, t(w), q_0, q_1)$ for all $q_0, q_1 \in Q$ and is equal to 0 otherwise. For all $w \in \{0,1\}^*$ and $q_0, q_1 \in Q$, $\zeta(((q, q_0, q_1), w), (q_0, w0) = \zeta(((q, q_0, q_1), w), (q_1, w1)) = \frac{1}{2}$ and 0 otherwise. Recall that $\mu_{\mathcal{M}_{\mathcal{A},t}}$ denotes the probability measure associated to $\mathcal{M}_{\mathcal{A},t}$.

To simplify the presentation, note that a trace in $\mathcal{M}_{\mathcal{A},t}$ can be uniquely represented by an infinite sequence $((p_0, q_0^0, q_0^1), a_0)((p_1, q_1^0, q_1^0), a_1)\ldots$ labeled by $\Delta \times \{0,1\}$ such that $p_0 = q_{\mathrm{ini}}$ and for all $i \geqslant 0$, $p_{i+1} = q_i^{a_i}$. The objective $\mathcal{O}_{\mathrm{Acc}}$ is the set of traces $((p_0, q_0^0, q_0^1), a_0)((p_1, q_1^0, q_1^0), a_1)\ldots$ such that $p_0 p_1 \ldots \in$ Acc.

***Proposition* 10.** *Let $\mathcal{A}$ be a probabilistic tree automaton with an ω-regular acceptance condition and let t be a tree. $t \in L^{=1}_{\mathrm{Qual}}(\mathcal{A})$ iff $\mathcal{M}_{\mathcal{A},t}$ almost-surely fulfils its objective.*

Proof (sktech): Let $\mathrm{AccPairs}(\mathcal{A}) = \{(\rho, \pi) \mid \rho(\pi) \in \mathrm{Acc}\}$. By Proposition 8, to establish the desired equivalence, it is sufficient to show that $\mu_t \otimes \mu(\mathrm{AccPairs}(\mathcal{A})) = \mu_{\mathcal{M}_{\mathcal{A},t}}(\mathcal{O}_{\mathrm{Acc}})$.

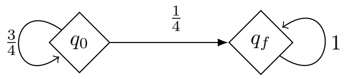
Consider the mapping $\psi : \mathrm{Runs} \times \mathrm{Br} \mapsto \mathrm{Traces}$ associating with any pair $(\rho, a_0 a_1 \ldots)$ the trace $(\rho(\varepsilon), \rho(0), \rho(1))a_0(\rho(a_0), \rho(a_0 0), \rho(a_0 1))a_1 \ldots$ of the Markov chain. It is clear that $\mathrm{AccPairs}(\mathcal{A}) = \psi^{-1}(\mathcal{O}_{\mathrm{Acc}})$. The mapping ψ is measurable and $\mu_{\mathcal{M}_{\mathcal{A},t}}$ is the image of $\mu_t \otimes \mu$ under ψ. In particular it implies that $\mu_{\mathcal{M}_{\mathcal{A},t}}(\mathcal{O}_{\mathrm{Acc}}) = \mu_t \otimes \mu(\mathrm{AccPairs}(\mathcal{A}))$. ■

Let us now consider the case of probabilistic tree automata with the usual semantic for runs (*i.e.* all branches must be accepting). In this setting, a naive definition of an acceptance game consists, when choosing the successor, in replacing random by a second player, Abelard, However this game does not faithfully reflect the acceptation of the automaton.

Consider the reachability probabilistic tree automaton $\mathcal{A} = \langle\{\mathtt{a}\}, \{q_0, q_f\}, q_0, \delta, \{q_f\}\rangle$ with $\delta(q_0, \mathtt{a}, q_0, q_0) = \frac{3}{4}$, $\delta(q_0, \mathtt{a}, q_f, q_f) = \frac{1}{4}$, $\delta(q_f, \mathtt{a}, q_f, q_f) = 1$ and 0 otherwise. Consider the tree $t_{\mathtt{a}}$ where all nodes are labeled by $\mathtt{a}$. It can be shown that the set of accepting runs of $\mathcal{A}$ over t_a has measure $\frac{1}{3}$. Hence the tree $t_{\mathtt{a}}$ does not belong to $L^{=1}(\mathcal{A})$.

Now consider the naive acceptance game for $\mathcal{A}$ on $t_{\mathtt{a}}$. Intuitively in this game, player random chooses a transition (p, q_0, q_1) and Abelard chooses to proceed either to q_0 or to q_1. The set of states is $\{0,1\}^* \times \{\theta_0, \theta_1, \theta_f\}$ with $\theta_0 = (q_0, q_0, q_0)$, $\theta_1 = (q_0, q_f, q_f)$ and $\theta_f = (q_f, q_f, q_f)$, the initial state is (ε, θ_0) and the actions of Abelard are in $\{0,1\}$. For $x \in \{0,1\}$, the transition function is such that for all $w \in \{0,1\}^*$ we have: $\zeta((w, \theta_0), x)$ is the probability distribution assigning $\frac{3}{4}$ to (wx, θ_0) and $\frac{1}{4}$ to (wx, θ_1), $\zeta((w, \theta_1), x) = \zeta((w, \theta_f), x)$ is the probability distribution assigning 1 to (wx, θ_f). The objective $\mathcal{O}$ is the set of plays containing θ_f.

It is easy to check that the strategy of Abelard has no influence on the value of the game. In fact for any fixed strategy, the game is equivalent to the Markov chain depicted below which fulfils its objective with probability 1.



C. Decidability results

In this section, we show that the emptiness problem for probabilistic Büchi tree automata is decidable for the qualitative semantics for runs. This result is by reduction to

[5] In the context of this article, probabilistic ω-word automata are simply probabilistic tree automata running over unary trees. For such an automaton $\mathcal{B}$, we denote by $L^{=1}(\mathcal{B})$ the language almost-surely accepted by $\mathcal{B}$.

deciding almost-surely winning in a POMDP, and the reduction works for any ω-regular acceptance condition. However, the decision problem on POMDPs is only decidable for the Büchi condition.

Let $\mathcal{A} = \langle A, Q, q_{\mathrm{ini}}, \delta, \mathrm{Acc}\rangle$ be a probabilistic automaton with an ω-regular acceptance condition and $\Delta = Q \times Q \times Q$.

We consider the POMDP $\mathbb{G}^{=1}_{\mathcal{A}} = (\mathcal{G}, \sim, \mathcal{O})$. The arena $\mathcal{G}$ is equal to $\langle S, s_{\mathrm{ini}}, \Sigma, \zeta\rangle$ where $S = Q \times \{0, 1, \bot\} \times (\Delta \cup \{\bot\})$, $s_{\mathrm{ini}} = (q_0, \bot, \bot)$, $\Sigma = A$ and ζ is defined as follows. For all $a \in A$ and $(p, x, t) \in S$, $\zeta((p, x, t), a)$ is the distribution that assigns $\frac{1}{2}\delta(p, a, q_0, q_1)$ to $(q_y, y, (p, q_0, q_1))$ where $y = 0, 1$ and 0 to all other state. The objective $\mathcal{O}$ is the set of plays for which the sequence of states obtained when projecting on the first component belongs to Acc. The equivalence $\sim$ is defined by $(q, x, t) \sim (q', x', t')$ iff $x = x'$.

***Theorem* 9.** *Let $\mathcal{A}$ be a probabilistic tree automaton with an ω-regular acceptance condition. The language $L^{=1}_{\mathrm{Qual}}(\mathcal{A})$ is non-empty if and only if Éloïse almost-surely wins in $\mathbb{G}_{\mathcal{A}}$.*

Proof: From the definitions, we easily have that $\{[\lambda]_\sim \mid \lambda \in \mathrm{Plays} = [\bot]_\sim\{[0]_\sim, [1]_\sim\}^*$. Hence it is sufficient to consider strategies from $[\bot]_\sim\{[0]_\sim, [1]_\sim\}^*$ to A, that can also be seen as A-labeled trees. Once such a strategy φ_t (seen as a tree t) is fixed, the resulting Markov chain is, up to renaming, $\mathbb{G}_{\mathcal{A},t}$, meaning that the value of φ_t is the value of $\mathbb{G}_{\mathcal{A},t}$. In particular, Éloïse almost-surely wins in $\mathbb{G}_{\mathcal{A}}$ iff there is some t such that $\mathrm{Val}(\mathbb{G}_{\mathcal{A},t}) = 1$ iff $t \in L^{=1}_{\mathrm{Qual}}(\mathcal{A})$ (thanks to Proposition 10). ■

***Corollary* 2.** *Let $\mathcal{A}$ be a probabilistic Büchi tree automaton. Deciding whether $L^{=1}_{\mathrm{Qual}}(\mathcal{A}) = \emptyset$ is ExpTime-complete. Moreover, if $L^{=1}_{\mathrm{Qual}}(\mathcal{A}) \neq \emptyset$, it contains a regular tree.*

Proof: The Exptime upper-bound follows from the polynomial time reduction to deciding almost-surely winning in a Büchi POMDP. Existence of finite memory optimal strategies in POMDP implies the existence of a regular tree when $L^{=1}_{\mathrm{Qual}}(\mathcal{A}) \neq \emptyset$. The lower bound follows from Proposition 9: emptiness of probabilistic Büchi ω-word automata with the almost-sure acceptation (which is Exptime-complete [1]) can be reduced to our problem. ■

We show that the emptiness problem for probabilistic co-Büchi tree automata is undecidable for both the classical and qualitative semantics for runs. These results are obtained by reduction to the undecidability of the emptiness problem for co-Büchi ω-word automata with the almost-sure acceptation [1].

***Theorem* 10.** *The following problems are undecidable :*

1) *given a probabilistic co-Büchi tree automaton $\mathcal{A}$, decide if $L^{=1}(\mathcal{A}) = \emptyset$,*
2) *given a probabilistic co-Büchi tree automaton $\mathcal{A}$, decide if $L^{=1}_{\mathrm{Qual}}(\mathcal{A}) = \emptyset$.*

Proof: Both undecidability results are shown by reduction to the undecidability of the emptiness problem for co-Büchi ω-word automata (with the almost-sure acceptation). The case of qualitative semantic directly follows from Proposition 9. It remains to treat the case of the classical semantic for runs.

Let $\mathcal{B} = \langle A = \{\mathtt{a}, \mathtt{b}\}, Q, q_{ini}, \delta, F\rangle$ be a probabilistic co-Büchi ω-word automaton. We construct a probabilistic co-Büchi automaton $\mathcal{A}$ which simulates $\mathcal{B}$ on the left-most branch of the tree and checks that all other branches contain only as. Formally $\mathcal{A}$ is equal to $\langle A, Q \cup \{q_\bot\}, q_{\mathrm{ini}}, \delta', F \cup \{q_\bot\}\rangle$. The probability distribution δ' is given by:

$$\begin{cases} \delta'(q_a, \mathtt{a}, q_a, q_a) & = & 1 & \\ \delta'(q_a, \mathtt{b}, q_\bot, q_\bot) & = & 1 & \\ \delta'(q_\bot, x, q_\bot, q_\bot) & = & 1 & \text{for } x \in A \\ \delta'(p, x, q, q_a) & = & \delta(p, x, q) & \text{for } x \in A \text{ and } q \in Q \end{cases}$$

In all other cases, δ' is equal to 0.

A tree t belongs to $L^{=1}(\mathcal{A})$ if and only if for all $u \notin 0^*$, $t(u) = a$ and $t(0^\omega) \in L^{=1}(\mathcal{B})$. In particular $L^{=1}(\mathcal{A})$ is empty if and only if $L^{=1}(\mathcal{B})$ is empty. ■

D. Comparison with qualitative languages

In this section, we give an example of a tree language that is accepted by a co-Büchi probabilistic automaton but that is not a qualitative tree language.

For this we consider the ω-word language P_λ over $\{\mathtt{a}, \mathtt{b}\}$ defined, for all $0 < \lambda < 1$, by:

$$L_\lambda = \{a^{k_1}ba^{k_2}b\ldots \mid k_1, k_2, \ldots > 0 \text{ s.t.} \prod_{i=1}^{\infty}(1 - \lambda^{k_i}) > 0\}$$

In [1], L_λ is shown to be almost-surely accepted by a co-Büchi probabilistic automaton[6]. Therefore, by Proposition 9, $\mathrm{Path}^{=1}(L_\lambda)$ is a co-Büchi probabilistic qualitative tree language.

***Proposition* 11.** *For all $0 < \lambda < 1$, $\mathrm{Path}^{=1}(L_\lambda)$ is not a qualitative tree language.*

Proof (sketch): For all $0 < \lambda < 1$, $\mathrm{Path}^{=1}(L_\lambda)$ is nonempty and does not contain any regular tree. Hence according to Corollary 1, it is not a qualitative tree language. ■

***Remark* 5.** *Using the correspondence with POMDP introduced in Theorem 9, any co-Büchi automaton accepting $\mathrm{Path}^{=1}(L_\lambda)$ gives rise to an example of a co-Büchi POMDP $\mathbb{G}_{\mathcal{A}}$ in which Éloïse needs infinite memory to almost-surely win.*

V. Discussion

Throughout this paper, we favoured the almost-sure condition (*i.e.* requiring the measure to be equal to 1) over the positive one (*i.e.* requiring the measure to be strictly positive). However, the decidability results on MDPs stated

[6] They show that L_λ is positively accepted by a Büchi automaton. But as previously remarked in the introduction, the two classes coincide.

in Theorem 2 still hold if we replace the almost-sure acceptance by the positive acceptance [8], [6]. Similarly the decidability and undecidability results on POMDP stated in Theorem 3 transfer to positive acceptance provided that one replaces Büchi by co-Büchi and *vice versa* [1]. In this discussion, we summarise the impact when considering positive acceptance instead of almost-sure acceptance, and motivate our choices.

A run is >0-qualitively accepting if the measure of its set of accepting branches is strictly positive. The class of >0-qualitative tree languages is defined similarly to the class of qualitative tree languages by consider >0-qualitatively accepting run instead of qualitatively accepting run. The two classes are incomparable. However, >0-qualitative languages enjoy most of the good properties of qualitative languages presented in Section III-C. In particular, the decidability results stated in Theorem 4, Theorem 5 and Corollary 1 transfer to >0-qualitative languages (of course almost-sure acceptance needs to be replaced by positive acceptance). In addition, the >0-qualitative languages are also incomparable with regular tree languages. The most notable difference between the two classes is that >0-qualitative languages are not closed under intersection. In addition, the pumping lemma (Lemma 1) and the restriction to complete automata (Proposition 2) no longer hold.

Moving to the probabilistic automata setting, we now have to consider two possible acceptance conditions on the set of runs – almost-sure $(=1)$ and positive (>0) – and two possible definitions for a run to be accepting – qualitative $(=1)$ and positive (>0). By combining these conditions, we obtain four semantics for probabilistic tree automata denoted by $(>0,>0)$, $(>0,=1)$, $(=1,>0)$ and $(=1,=1)$ where the first component corresponds to the requirement on the set of accepting runs and the second to the requirement on the set of accepting branches of a run. In Section IV, we mainly dealt with $(=1,=1)$-probabilistic automata which have a tight link with POMDP for the almost-sure winning condition (cf. Theorem 9). It can be shown that $(>0,>0)$-probabilistic automata share the same connection with POMDP with the positive winning condition. It implies that the emptiness problem for the $(>0,>0)$-probabilistic automata with the co-Büchi acceptance condition is ExpTime-complete.

When the two conditions are not of the same nature (as for the $(>0,=1)$ and $(=1,>0)$ semantics), we were unable to define a proper acceptance game — see for instance a similar discussion at the end of Subsection IV-B for the semantic $(=1,\forall)$.

Acknowledgment. This work was supported by the following projects: AMIS (ANR 2010 JCJC 0203 01 AMIS) and FREC (ANR 2010 BLAN 0202 02 FREC).

References

[1] Christel Baier, Nathalie Bertrand, and Marcus Grösser. On decision problems for probabilistic Büchi automata. In *Proc of FoSSaCS'08*, volume 4962 of *Lecture Note in Computer Science*, pages 287–301. Springer, 2008.

[2] Christel Baier and Marcus Grösser. Recognizing omega-regular languages with probabilistic automata. In *Proc. of LICS'05)*, pages 137–146. IEEE, 2005.

[3] Heinz Bauer. *Measure and Integration Theory*. Walter de Gruyter, 2001.

[4] Danièle Beauquier and Damian Niwiński. Automata on infinite trees with path counting constraints. *Inf. Comput.*, 120(1):117 – 125, 1995.

[5] Krishnendu Chatterjee, Marcin Jurdziński, and Thomas A. Henzinger. Quantitative stochastic parity games. In *Proc. of SODA'04*, pages 121–130. SIAM, 2004.

[6] Alonzo Church. Logic, arithmetic and automata. In *Proceedings of the International Congress of Mathematicians*, pages 23–35, 1962.

[7] Costas Courcoubetis and Mihalis Yannakakis. Markov decision processes and regular events. In *Proc. of ICALP'90*, volume 443 of *Lecture Note in Computer Science*, pages 336–349. Springer, 1990.

[8] E. Allen Emerson, Charanjit S. Jutla, and A. Prasad Sistla. On model-checking for fragments of μ-calculus. In *Proc. of the CAV'93*, volume 697 of *Lecture Note in Computer Science*, pages 385–396. Springer, 1993.

[9] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *Lecture Note in Computer Science*. Springer, 2002.

[10] Yuri Gurevich and Leo Harrington. Trees, automata, and games. In *Procof STOC'82*, pages 60–65. ACM, 1982.

[11] Andrei A. Muchnik, Alexei L. Semenov, and Vladimir A. Uspensky. Mathematical metaphysics of randomness. *Theor. Comput. Sci.*, 207(2):263–317, 1998.

[12] A. Paz. *Introduction to probabilistic automata*. Academic Press New York, 1971.

[13] Dominique Perrin and Jean-Éric Pin. *Infinite Words*. Academic Press, 2004.

[14] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[15] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transaction of the American Mathematical Society*, 141:1–35, 1969.

[16] Wolfgang Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Language Theory*, volume III, pages 389–455. 1997.

[17] Moshe Y. Vardi and Thomas Wilke. Automata: from logics to algorithms. In *Logic and Automata: History and Perspectives*, pages 629–736. Amsterdam University Press, 2007.