Publications Internes de l'IRISA
ISSN : 2102-6327
PI 1992 – May 2012

Emptiness Of Alternating Parity Tree Automata Using Games With Imperfect Information

Sophie Pinchinat* , Olivier Serre**
Sophie.Pinchinat@irisa.fr, Olivier.Serre@liafa.univ-paris-diderot.fr

Abstract: We focus on the emptiness problem for alternating parity tree automata. The usual technique to tackle this problem first removes alternation, going to non-determinism, and then checks emptiness by reduction to a two-player perfect-information parity game. In this note, we give an alternative roadmap to this problem by providing a direct reduction to the emptiness problem to solving an *imperfect-information* two-player parity game.

Key-words: Alternating Tree Automata; Emptiness; Imperfect Information Games; Positional Determinacy

---

Le problème du vide pour les automates d'arbres alternant à parité via des jeux à information imparfaite

Résumé : *Nous considérons le problème du test du vide pour les automates d'arbres alternants à parité. La méthode usuelle pour résoudre ce problème, commence par supprimer l'alternance ce qui conduit à un automate non-déterministe dont le vide est ensuite testé par réduction à un jeux de parité à information parfaite. Dans cette note, nous proposons une approche alternative pour ce problème, en proposant une réduction directe du problème du vide à un jeu de parité à information* imparfaite.

Mots clés : *Automates d'arbres alternants; test du vide; jeux à information imparfaite; déterminaison positionnelle*

---

* IRISA Université de Rennes 1
** LIAFA (Université Paris Diderot – Paris 7 & CNRS)

1 Introduction

Tree automata [16, 10] are a powerful tool to handle sets of infinite trees which are widely needed in verification, since they provide a natural representation of branching-time system executions. It is well known that by equipping tree automata with the parity condition, one captures all ω-regular tree languages [11]. Additionally, tree automata may employ *alternation* [7], which makes their complementation an extremely simple task. In particular, combining alternation with the parity condition yields the automata-theoretic counterpart of the propositional μ-calculus, where the translation from one to the other can be done in linear time [2, 11]. Hence, the model-checking and the satisfiability/validity of logical formulas amount to respectively verify membership and non-emptyness/universality on their corresponding tree automata.

The membership problem for alternating tree automata has a fairly simple algorithm: one compiles the input tree and the automaton into a polyomial size perfect information parity game and solves it. On the contrary, the usual roadmap to check emptiness of an alternating tree automaton is more involved. First one builds an equivalent non-deterministic automaton thanks to the Simulation Theorem [14], and then one checks emptiness of this latter automaton by solving an associated two-player perfect information game.

The contribution of this note is to propose a decision procedure for emptiness of alternating parity tree automata that goes directly to solve a two-player game, but with *imperfect* information. The proposed construction extends the notion of *blindfold games* given by Reif in his seminal paper [15] used to check universality of non-deterministic automata on finite words (a similar idea was later considered in [17]).

This reduction of the non-emptiness of alternating parity automata to imperfect information games does not lead to a gain in complexity, due to intrinsic hardness. Nevertheless, we believe that its simplicity is of interest, and we also think that this approach generalizes to other classes of automata on infinite objects, as recently demonstrated for probabilistic qualitative tree languages [6].

2 Definitions

Let X be a (possibly infinite) alphabet. Then we denote by X^* (*resp.* X^ω) the set of finite (*resp.* infinite) words over X. We let ε be the empty word.

Let Σ be a *finite* alphabet. An *infinite Σ-labelled binary tree* (or simply a *tree* when Σ is clear from the context) is a map $t : \{0,1\}^* \to \Sigma$. In this setting, we shall refer to an element $n \in \{0,1\}^*$ as a *node* and to ε as the *root*. For a node n, we call $t(n)$ the *label* of n in t. A *language* of infinite Σ-labelled binary trees is a set of infinite Σ-labelled binary trees.

2.1 Two-Player Perfect Information Games

A *graph* is a pair $G = (V, E)$ where V is a set of *vertices* and $E \subseteq V \times V$ is a set of *edges*. The *size* of a graph is defined to be $|V| + |E|$. A *dead-end* is a vertex v such that there is no vertex v' with $(v, v') \in E$; in the rest of the paper, we only consider graphs that have no dead-end, hence this is implicit from now on.

An *arena* is a triple $\mathcal{G} = (G, V_{\mathrm{E}}, V_{\mathrm{A}})$ where $G = (V, E)$ is a graph and $V = V_{\mathrm{E}} \uplus V_{\mathrm{A}}$ is a partition of the vertices among two players, Éloïse and Abelard. A colouring function ρ is a mapping $\rho : V \to \mathrm{Col} \subset \mathbb{N}$ where Col is a *finite* set of colours. An *infinite two-player parity game* on an arena $\mathcal{G}$ is a triple $\mathbb{G} = (\mathcal{G}, \rho, v_0)$ where $v_0 \in V$ is some *initial* vertex.

Éloïse and Abelard play in $\mathbb{G}$ by moving a pebble along edges. A *play* starts from the initial vertex v_0 and proceeds as follows: the player owning v_0 (*i.e.* Éloïse if $v_0 \in V_{\mathrm{E}}$, Abelard otherwise) moves the pebble to a vertex v_1 such that $(v_0, v_1) \in E$. Then the player owning v_1 chooses a successor v_2 and so on. As we assumed that there is no dead-end, a play is an infinite word $v_0 v_1 v_2 \cdots \in V^\omega$ and it is won by Éloïse just in case $\liminf(\rho(v_i))_{i \geq 0}$ is even. A *partial play* is a prefix of a play.

A strategy for Éloïse is a function assigning, to every partial play ending in some vertex $v \in V_{\mathrm{E}}$, a vertex v' such that $(v, v') \in E$. Éloïse *respects a strategy* φ during a play $v_0 v_1 v_2 \cdots$ if $v_{i+1} = \varphi(v_0 \cdots v_i)$, for all $i \geq 0$ such that $v_i \in V_{\mathrm{E}}$. A strategy φ for Éloïse is *winning* if Éloïse wins every play where she respects φ.

Of special interest are strategies that do not require memory. A *positional strategy* is a strategy φ such that for any two partial plays of the form $\lambda.v$ and $\lambda'.v$, one has $\varphi(\lambda \cdot v) = \varphi(\lambda' \cdot v)$, *i.e.* φ only depends on the current vertex. It is a well known result that positional strategies suffice to win in parity games [18].

Theorem 1 (Positional determinacy)**.** *Let $\mathbb{G}$ be a parity game. Then for any vertex, either Éloïse or Abelard has a positional winning strategy.*

Hence, we now only consider positional strategies and we represent such a strategy as functions from V_{E} (or V_{A} depending on the player) into V.

2.2 Alternating Parity Tree Automata

An *alternating parity tree automaton* is a tuple $\mathcal{A} = \langle Q_\exists, Q_\forall, \Sigma, \Delta, q_{\rm in}, \rho\rangle$, where $Q_\exists$ is a set of existential states and $Q_\forall$ is a set of universal states such that $Q_\exists$ and $Q_\forall$ are disjoint (we let $Q = Q_\exists \uplus Q_\forall$), $q_{\rm in} \in Q$ is an initial state, Σ is a labelling alphabet, $\Delta \subseteq Q \times \Sigma \times Q \times Q$ is a transition relation and $\rho : Q \to \mathbb{N}$ is a colouring function.

In the following, we use alternating parity tree automata as acceptors for tree languages, and we define acceptance by means of a parity game. Let $\mathcal{A} = \langle Q_\exists, Q_\forall, \Sigma, \Delta, q_{\rm in}, \rho\rangle$ be an alternating parity tree automaton, and let t be a Σ-labelled tree. From $\mathcal{A}$ and t, we define a two-player perfect information parity game $\mathbb{G}_{\mathcal{A},t}$.

Intuitively, a play in $\mathbb{G}_{\mathcal{A},t}$ consists in moving a pebble along a branch of t in a top-down manner: the pebble is attached to a state and in some node n with state q Éloïse (if $q \in Q_\exists$) or Abelard (if $q \in Q_\forall$) picks a transition $(q, t(n), q_0, q_1) \in \Delta$, and then Abelard chooses to move down the pebble either to $n \cdot 0$ (and update the state to q_0) or to $n \cdot 1$ (and update the state to q_1).

More formally, $\mathbb{G}_{\mathcal{A},t} = (\mathcal{G}_{\mathcal{A},t}, \rho_{\mathcal{A},t}, (\varepsilon, q_{\rm in}))$ where:

- The arena $\mathcal{G}_{\mathcal{A},t}$ is $(G = (V_{\rm E} \cup V_{\rm A}, E), V_{\rm E}, V_{\rm A})$ is defined by:
  - $V_{\rm E} = \{(n, q) \mid n \in \{0,1\}^* \text{ and } q \in Q_\exists\}$. Vertices controlled by Éloïse are pairs made of a node together with a control state in $Q_\exists$.
  - $V_{\rm A} = \{(n, q) \mid n \in \{0,1\}^* \text{ and } q \in Q_\forall\} \cup \{(n, q, q_0, q_1) \mid n \in \{0,1\}^* \text{ and } (q, t(n), q_0, q_1) \in \Delta\}$. Vertices controlled by Abelard are pairs made of a node together with a control state in $Q_\exists$ as well as tuples consisting of a node and three states consistent with the transition relation of $\mathcal{A}$.
  - The set of edges E is equal to $\{((n, q), (n, q, q_0, q_1)) \mid (n, q, q_0, q_1) \in V\} \cup \{((n, q, q_0, q_1), (n \cdot x, q_x)) \mid x \in \{0, 1\} \text{ and } (n, q, q_0, q_1) \in V\}$.
- The colouring function mimics the one of the automaton $\mathcal{A}$: $\rho_{\mathcal{A},t}((n, q)) = \rho_{\mathcal{A},t}((n, q, q_0, q_1)) = \rho(q)$.

Note that this game is not symmetric as it is always Abelard who is in charge of choosing the direction.

A tree t is *accepted* by $\mathcal{A}$ if and only if Éloïse has a winning strategy in the game $\mathbb{G}_{\mathcal{A},t}$; if not the tree is rejected. Finally, we define the set $L(\mathcal{A})$ as those trees accepted by $\mathcal{A}$.

The *emptiness problem* for $\mathcal{A}$ consists in deciding whether $L(\mathcal{A}) = \emptyset$.

Remark 1. *There are several definitions of alternating tree automata, and another popular one is by not distinguishing between existential and universal states but replacing the transition relation by a map $\delta : Q \times \Sigma \to \mathcal{B}^+(Q \times \{0, 1\})$ where $\mathcal{B}^+(X)$ denotes the positive Boolean formulas over X (see* e.g. *[13]). Our model is easily seen to be equi-expressive with that one.*

Remark 2. *A positional strategy for Éloïse in $\mathcal{G}_{\mathcal{A},t}$ can be described as a function $\varphi : \{0,1\}^* \times Q_\exists \to Q \times Q$ that satisfies the following property: $\forall n \in \{0, 1\}^*$, if $\varphi(n, q) = (q_0, q_1)$ then $(q, t(n), q_0, q_1) \in \Delta$. Equivalently, if one let $\mathcal{T}$ be the set of functions from $Q_\exists$ into $Q \times Q$, a positional strategy can be described as a $\mathcal{T}$-labelled binary tree.*

A *non-deterministic* parity tree automaton is an alternating automaton in which all states are existential. In this setting it is well-known how to define a two-player perfect information parity game in which Éloïse wins iff the language of the automaton is non-empty (see *e.g.* [11, 1]). Note that this game is of polynomial size in the number of states of the automaton and uses the same set of colours.

It is a well-known (but technically involved) result that alternating and non-deterministic automata are equi-expressive [14].

Theorem 2 (Simulation Theorem)**.** *Let $\mathcal{A}$ be an alternating parity tree automaton with n states and using k colours. Then one can effectively construct a non-deterministic parity tree automaton $\mathcal{B}$ such that $L(\mathcal{A}) = L(\mathcal{B})$. The automaton $\mathcal{B}$ has $2^{\mathcal{O}(nk\log(nk))}$ and it uses $\mathcal{O}(nk)$ colours.*

Hence, the usual roadmap to check emptiness of an alternating tree automaton is as follows. First one builds an equivalent non-deterministic automaton thanks to Theorem 2, and then one checks emptiness of this latter automaton by solving the associated emptiness game.

2.3 Imperfect Information Games

In the following we introduce a quite restrictive class of games with imperfect information [8, 1].

A *game structure (of imperfect information)* is a tuple $\mathcal{G} = \langle S, s_{\rm in}, A, T, \sim\rangle$ where S is a finite set of states, $s_{\rm in} \in S$ is an initial state, A is a finite alphabet of actions, $T \subseteq S \times A \times S$ is a transition relation and $\sim$ is an equivalence relation over S. We additionally require that for all $(s,a) \in S \times A$ there is at least one $s' \in S$ such that $(s,a,s') \in T$. An imperfect information game is a pair $\mathbb{G} = \langle \mathcal{G}, \rho \rangle$ where $\mathcal{G}$ is a game structure with set of states S and $\rho : S \to \mathbb{N}$ is a colouring function.

A *play* starts from the initial state $s_0 = s_{\text{in}}$ and proceeds as follows: Éloïse plays an action $a \in A$ and then Abelard resolves the nondeterminism by choosing a state s_1 such that $(s_0, a, s_1) \in T$. Then Éloïse plays a new action, Abelard resolves the nondeterminism and so on forever. Hence a play is an infinite word $s_0a_0s_1a_1s_2a_2 \cdots \in (S \times A)^\omega$ and it is won by Éloïse just in case $\liminf(\rho(s_i))_{i\geq 0}$ is even. A *partial play* is a prefix of a play.

Imperfect information is important when defining strategies for Éloïse. Intuitively, she should not play differently in two indistinguishable plays, where the indistinguishability of Éloïse is based on *perfect recall* [9], that is: Éloïse cannot distinguish two plays $s_0a_0s_1a_1 \cdots s_\ell$ and $s'_0a'_0s'_1a'_1 \cdots s'_\ell$ with $s_i \sim s'_i$ for all $0 \leq i \leq \ell$ and $a_i = a'_i$ for all $0 \leq i < \ell$. Hence, a strategy for Éloïse is a function $\varphi : (S_{/\sim} \times A)^* \cdot (S_{/\sim}) \to A$ assigning an action to every set of indistinguishable plays (here $S_{/\sim}$ denotes the set of equivalence classes of $\sim$ in S, and for every $s \in S$, we shall write $[s]_\sim$ for its $\sim$-equivalence class). Éloïse *respects a strategy* φ during a play $\lambda = s_0a_0s_1a_1s_2a_2 \cdots$ if $a_{i+1} = \varphi([s_0]_\sim a_0 \cdots [s_i]_\sim)$, for all $i \geq 0$. A strategy φ for Éloïse is *winning* if Éloïse wins every play where she respects φ.

Remark 3. *Our model of imperfect information games belongs to a restrictive class compared to general ones as developed in [12, 3]. Indeed, Abelard is perfectly informed, and there is no stochastic transitions; moreover we only consider sure winning and non-randomized strategies (which is actually not a restriction for sure winning). However, our model turns out to be expressive enough for our purpose.*

Remark 4. *It is important to note that Éloïse may not observe the colour of the current state in general, as we do not require that $s \sim s' \Rightarrow \rho(s) = \rho(s')$. In particular, this has to be taken into account when eventually solving the game.*

3 An Emptiness Game For Alternating Parity Tree Automata

For the rest of this section, we fix an alternating parity tree automaton $\mathcal{A} = \langle Q_\exists, Q_\forall, \Sigma, \Delta, q_{\text{in}}, \rho \rangle$. We define a game structure of imperfect information $\mathcal{G}_\mathcal{A}$ that intuitively works as follows. Éloïse describes both a tree t and a positional strategy φ for her in the game $\mathcal{G}_{\mathcal{A},t}$; the strategy φ is described as a $\mathcal{T}$-labeled tree (where $\mathcal{T}$ is the set of functions from $Q_\exists$ into $Q \times Q$, see Remark 2). As the plays are of ω-length, she actually does not fully describe t and φ but only a branch: this branch is chosen by Abelard, who also takes care of computing the sequence of states along it (either by updating an existential state accordingly to φ or, when the state is universal, by choosing an arbitrary valid transition of the automaton). In this game, Éloïse observes the directions, but not the actual control state of the automaton.

Formally, we let $\mathcal{G}_\mathcal{A} = \langle S, s_{\text{in}}, A, T, \sim \rangle$ where $S = (Q \times \{0,1\}) \cup \{(q_{\text{in}}, \varepsilon)\}$ and $s_{\text{in}} = (q_{\text{in}}, \varepsilon)$; $A \subseteq \Sigma \times \mathcal{T}$ is the set of pairs (a, τ) such that for all $q \in Q_\exists$ we have that $(q, a, q_0, q_1) \in \Delta$ where $\tau(q) = (q_0, q_1)$ (recall that $\mathcal{T}$ is the set of functions from $Q_\exists$ into $Q \times Q$);

$$T = \{((q,i),(a,\tau),(q_0,0)),((q,i),(a,\tau),(q_1,1)) \mid q \in Q_\exists \text{ and } \tau(q) = (q_0,q_1)\} \\ \cup \ \{((q,i),(a,\tau),(q_0,0)),((q,i),(a,\tau),(q_1,1)) \mid q \in Q_\forall \text{ and } (q,a,q_0,q_1) \in \Delta\}$$

and $(q,i) \sim (q',i)$ for all $q, q' \in Q$ and $i \in \{0,1\}$.

Finally we let $\mathbb{G}_\mathcal{A} = \langle \mathcal{G}_\mathcal{A}, \rho_\mathcal{A} \rangle$ be the parity game obtained by letting $\rho_\mathcal{A}(q,i) = \rho(q)$ for any $(q,i) \in S$.

We then have the following result.

Theorem 3. *Éloïse has a winning strategy in $\mathbb{G}_\mathcal{A}$ iff $L(\mathcal{A}) \neq \emptyset$.*

Proof. Due to how $\sim$ is defined, a strategy for Éloïse in $\mathbb{G}_\mathcal{A}$ can also be viewed as a map $\varphi : \{0,1\}^* \to A$. As $A \subseteq \Sigma \times \mathcal{T}$, one can see φ as a pair (t, φ_t) where t is an infinite Σ-labelled binary tree, and φ_t is a strategy for Éloïse in the acceptance game $\mathbb{G}_{\mathcal{A},t}$ of t by $\mathcal{A}$. Now, once such a strategy φ is fixed, the set of plays in $\mathbb{G}_\mathcal{A}$ where Éloïse respects φ is in one-to-one correspondence with the set of plays in $\mathbb{G}_{\mathcal{A},t}$ where she respects φ_t, and this correspondence preserves the property of being winning. Therefore, $\varphi = (t, \varphi_t)$ is winning iff φ_t is a winning strategy in $\mathbb{G}_{\mathcal{A},t}$ iff $t \in L(\mathcal{A})$. Therefore, Éloïse has a winning strategy iff there exists some tree $t \in L(\mathcal{A})$. □

Remark 5. *From the proof of Theorem 3, one can also conclude that if $L(\mathcal{A}) \neq \emptyset$ then $L(\mathcal{A})$ contains a regular tree (*i.e.* the unfolding of a finite graph). Indeed, this is a direct consequence of the fact that if Éloïse has a winning strategy in $\mathbb{G}_\mathcal{A}$, then she has one that uses finite memory [8].*

Theorem 3 provides a reduction of the emptiness problem to deciding the existence of a winning strategy in a game with imperfect information. We prove a converse result.

Theorem 4. *Let $\mathbb{G}$ be an imperfect information game. Then one can construct an alternating parity tree automaton $\mathcal{A}_{\mathbb{G}}$ such that Éloïse has a winning strategy in $\mathbb{G}$ iff $L(\mathcal{A}_{\mathbb{G}}) \neq \emptyset$. Moreover in $\mathcal{A}_{\mathbb{G}}$ all states are universal.*

Proof. Let $\mathcal{G} = \langle S, s_{\text{in}}, A, T, \sim\rangle$ be the underlying game structure of $\mathbb{G} = \langle \mathcal{G}, \rho\rangle$ and let $C_1, \cdots, C_k$ be the equivalence classes of $\sim$. We design an alternating automaton $\mathcal{A}_{\mathbb{G}} = \langle Q_\exists = \emptyset, Q_\forall = S \cup \{\top\}, A, \Delta, s_{\text{in}}, \rho'\rangle$, working on infinite A-labelled trees of *arity* k, that are maps $t : \{1, \ldots, k\}^* \to A$. The only difference with the definition of Section 2.2 when considering automata handling such trees is that the transition relation Δ is a subset of $Q \times A \times Q^k$. We let $\rho'(\top) = 0$ and $\rho'(s) = \rho(s)$ for all $s \in S$ and $\Delta = \{(s, a, s_1, \ldots, s_k) \mid \forall 1 \leq i \leq k,\ (s_i \in C_i \text{ and } (s, a, s_i) \in T) \text{ or } s_i = \top\}\} \cup \{(\top, a, \top, \ldots, \top) \mid a \in A\}$

Note that A-labelled trees of arity k are trivially in bijection with strategies of Éloïse in $\mathbb{G}$. Indeed with a tree t of arity k we associate a strategy φ_t by letting $\varphi_t(C_{j_0} a_0 \cdots C_{j_i}) = t(j_0 \cdots j_i)$. Then it is rather immediate to check that the set of plays where Éloïse respects φ_t in $\mathbb{G}$ is isomorphic to the set of plays in the acceptance game $\mathbb{G}_{\mathcal{A}_{\mathbb{G}}, t}$ (note that in this game all vertices are controlled by Abelard as all states in $\mathcal{A}_{\mathbb{G}}$ are universal); moreover winning plays are preserved. By noting that one can always encode trees of arity k into binary trees (and modify $\mathcal{A}$ accordingly) one concludes the proof. □

4 Conclusion

In this note we presented a method for checking emptiness of alternating parity tree automata that does not rely on the Simulation Theorem [14]. It is however important to stress that the two ingredients in the proof of the Simulation Theorem did not vanish: we make crucial use of positional determinacy of games in our reduction, and determinization of automata on infinite word is implicitly needed when solving the imperfect information games. Indeed, since Éloïse does not observe the colour, one has to use a standard trick that consists in embedding in the subset construction of [8] a deterministic parity word automaton that checks that all plays consistent with the observations fit the parity condition; this requires to determinize a non-deterministic Büchi word automaton. Somehow, our construction helps separating the two techniques.

We also think that going to imperfect information games may lead to efficient implementation for emptiness checking, as many proper techniques (*e.g.* antichains) have recently been developed to deal with those games [4, 5].

References

[1] K. R. Apt and E. Grädel. *Lectures in Game Theory for Computer Scientists*. Cambridge University Press, 2011.

[2] A. Arnold and D. Niwiński. *Rudiments of μ-calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 2001.

[3] N. Bertrand, B. Genest, and H. Gimbert. Qualitative determinacy and decidability of stochastic games with signals. In *Proceedings of Logic in Computer Science (LiCS'09)*, pages 319–328. IEEE Computer Society Press, 2009.

[4] D. Berwanger, K. Chatterjee, M. D. Wulf, L. Doyen, and T. A. Henzinger. Alpaga: A tool for solving parity games with imperfect information. In *Proceedings of Tools and Algorithms for the Construction and Analysis of Systems (TACAS'09)*, volume 5505 of *Lecture Notes in Computer Science*, pages 58–61. Springer, 2009.

[5] D. Berwanger, K. Chatterjee, M. D. Wulf, L. Doyen, and T. A. Henzinger. Strategy construction for parity games with imperfect information. *Information and Computation*, 208(10):1206–1220, 2010.

[6] A. Carayol, A. Haddad, and O. Serre. Qualitative tree languages. In *Proceedings of Logic in Computer Science (LiCS'11)*, pages 13–22. IEEE Computer Society Press, 2011.

[7] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the Association of Computing Machinery*, 28(1):114–133, Jan. 1981.

[8] K. Chatterjee, L. Doyen, T. Henzinger, and J.-F. Raskin. Algorithms for omega-regular games with imperfect information. *Logical Methods in Computer Science*, 3(3), 2007.

[9] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[10] J. Flum, E. Grädel, and T. Wilke. *Logic and Automata: History and Perspectives*. Amsterdam University Press, 2007.

[11] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[12] V. Gripon and O. Serre. Qualitative concurrent stochastic games with imperfect information. In *Automata, Languages and Programming, 36th International Colloquium (ICALP'09)*, volume 5556 of *Lecture Notes in Computer Science*, pages 200–211. Springer, 2009.

[13] O. Kupferman, M. Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the Association of Computing Machinery*, 47(2):312–360, 2000.

[14] D. E. Muller and P. E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141(1&2):69–107, 1995.

[15] J. Reif. Universal games of incomplete information. In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing (STOC'79)*, pages 288–308. ACM, 1979.

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

[17] M. D. Wulf, L. Doyen, T. A. Henzinger, and J.-F. Raskin. Antichains: A new algorithm for checking universality of finite automata. In *Proceedings of Computer Aided Verification (CAV'06)*, volume 4144 of *Lecture Notes in Computer Science*, pages 17–30. Springer, 2006.

[18] W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998.