p-Automata and Obligation Games

(Invited Paper)

Nir Piterman
Department of Computer Science
University of Leicester
Leicester, UK
nir.piterman@le.ac.uk

***Abstract*—We present our automata-based approach to probabilistic verification. This new approach adapts notions and techniques from alternating tree automata to the realm of Markov chains. The resulting p-automata determine languages of Markov chains. In order to determine acceptance of Markov chains by p-automata we develop a new notion of games, which we call *obligation games*. Intuitively, one player commits to achieving a certain probability of winning in the interaction.**

We survey the initial results regarding obligation games and p-automata. These include algorithms for solving obligation parity games, initial results about the expressive power of p-automata, and the relation between p-automata and pCTL model checking. In particular, these initial foundations show that p-automata enable abstraction-based probabilistic model checking for probabilistic specifications that subsume Markov chains, and LTL and CTL* like logics. Many interesting questions remain open. For example, further algorithmic studies of obligation games, the theory of p-automata, and the usage in practice of p-automata as an abstraction framework for Markov chains.

***Keywords*-Markov chains, model checking, pCTL, automata, games**

I. Introduction

Markov chains are a central concept in many aspects of engineering and science. Recent years have seen an increase in the interest in applications of model checking to Markov chains and other probabilistic systems (cf. [1], [2]). Unfortunately, these applications are severely hindered by the inability to apply abstraction (cf. [3]), *the* technique that enabled model checking to scale to realistic application in the non-probabilistic world. Reasoning abstractly about Markov chains (and probabilistic systems in general), is a very active field of research. Recent years have seen various suggestion of more elaborate forms of probabilistic systems whose aim is to enable abstraction in the probabilistic context (cf. [4], [5], [6], [7]).

In the non-probabilistic setting, the automata-theoretic approach to verification provides a unifying framework for reasoning about systems. In particular, the underlying concepts of abstraction can be formulated in the context of automata (see, e.g., [8]). The automata framework that supports branching-time temporal reasoning is that of alternating tree automata [9]. Recently, Dams and Namjoshi showed that alternating tree automata are a complete framework for abstraction with respect to branching-time logic [10], [11]. Motivated by their work, we introduced p-automata as a corresponding unifying automata framework for reasoning about Markov chains [12].

Much like tree automata read trees, p-automata read entire Markov chains. They combine pCTL [13] and alternating tree automata, namely, they have a rich combinational structure and the ability to quantify the probabilities of sets of paths. Acceptance of tree automata is determined by solving two-player games (cf. [9]). We would like to reduce acceptance by p-automata to some form of stochastic games. However, existing game formalisms are too weak to enable a similar reduction for p-automata.

We extend the winning conditions of two-player games by a new structural restriction, which we call *obligations* [14]. Intuitively, one of the players adds commitments as to the value of the game as she goes along playing. Interestingly, the resulting games are not covered by the classical determinacy results of two-player games [15], [16]. We show that the resulting games are indeed determined and that the definition of p-automata acceptance based on these games is well structured.

Building upon this determinacy result we show the following:

- The values of obligation games can be determined in NEXPTIME. The values of uniform obligation games, which satisfy some structural restrictions, can be computed in EXPTIME. These complexity results apply also to the appropriate kinds of p-automata.
- Languages of p-automata are closed under bisimulation. The set of languages accepted by p-automata is closed under Boolean operations. p-Automata can express bisimulation classes of Markov chains and pCTL formulas. Complexity of acceptance for p-automata that result from pCTL formulas matches that of pCTL model checking.
- Simulation of p-automata approximates language containment.
- p-automata are the first complete abstraction framework for PCTL model checking on Markov chains.

The latter means that if an infinite-state Markov chain satisfies a PCTL formula, there is a finite p-automaton that abstracts (i.e. simulates) this Markov chain and whose language is contained in that of the formula.

We mention a few interesting open problems that arise.

First, our algorithms for solving obligation games summarize a first attempt at such algorithms. They show that these games can be analyzed algorithmically. We believe that the complexity of solving these games can be considerably improved and algorithms made more practical. Furthermore, questions that have been answered regarding existing notions of games will have to be repeated and answered anew for obligation games.

Second, we have merely touched upon the well definedness of the automata theory of p-automata. Major issues such as what are nondeterministic p-automata and how to convert alternating p-automata to nondeterministic p-automata are left open. Algorithmic aspects of p-automata, such as solving emptiness, could lead to a solution of the satisfiability problem of pCTL, which has been open since the early 80s.

Finally, as mentioned, p-automata are the first complete abstraction framework for model checking pCTL over Markov chains. This crucially depends on the ability to combine probabilistic quantifications on regular sets of *paths*, which is completely lacking from other approaches towards abstraction of Markov chains. We need to study the usage of p-automata in a model checking and abstraction framework.

In this paper we give the basic definitions and results relating to p-automata. Definitions are very terse and are intended only to give the flavor of these formalisms. The interested reader is referred to [12] and [14].

II. Obligation Parity Games

Obligation parity games introduce a new "structural" winning paradigm to infinite duration two-player games [14]. A win can no longer be decided based only on linear plays. It is evaluated over the entire Markov chain resulting from unwinding the strategies of the players. This is done by adding obligations, promises by one of the players to achieve a certain value in some configurations. Intuitively, in order to win, Player 0 has to make sure that all plays are in the target set (as usual) and, in addition, that all obligations are met. The value for Player 0 at an obligation configuration, where the obligation is met, is 1. Otherwise, it is 0. In particular, it is independent of the strategy and probability choices made following the visit to that configuration.

An obligation parity game (OPG) is $G = ((V, E), (V_0, V_1, V_p), \kappa, (\alpha, O))$, where (V, E) is a finite directed graph with a finite set of configurations V, (V_0, V_1, V_p) is a partition of V to Player 0, Player 1, and probabilistic configurations, respectively, $\kappa : V_p \to \mathcal{D}(V)$ associates with every probabilistic configuration a distribution with finite support over V such that $\kappa(v)(u) > 0$ iff $(v, u) \in E$, α is a parity condition over V, and $O : V \to (\{\geq, >\} \times [0, 1]) \cup \{\bot\}$. The obligation function O associates with some configurations the value $\bot$ saying that there is no special obligation associated with this configuration. With other configurations O associates an obligation $>v$ or $\geq v$ stating that Player 0 can use this configuration (and it is going to have the value 1 for her) only if she can ensure that the value she can get from this configuration meets the obligation. It follows that, recursively, Player 0 has to adjust her strategy after every obligation configuration so that the values in all obligations are met using plays in φ. For configuration v, if $O(v) \neq \bot$ we call v an *obligation* configuration and if $O(v) = \bot$ we call v a *non-obligation* configuration. We call a sequence of configurations a *play prefix* or just a *prefix*.

We define the notion of value of an obligation parity game using the well known notions of value in stochastic games for obligations of the form $\mathsf{reach}(S)$, $\mathsf{safe}(S)$, and parity, for a set S of configurations. Obligations are handled through the notion of a *choice set*, the set of obligations that can be met, which we introduce now.

Consider an OPG $G = ((V, E), \ldots, (\alpha, O))$. Let $\mathcal{O}$ denote the set of configurations $v \in V$ such that $O(v) \neq \bot$ and $\hat{\mathcal{O}}$ prefixes $w \cdot v \in V^+$ such that $v \in \mathcal{O}$. We denote by $O(w)$ the obligation $O(v)$, where $w = w' \cdot v$. That is, $\mathcal{O}$ is the set of configurations with a non-empty obligation and the set $\hat{\mathcal{O}}$ is the set of prefixes that end in a configuration in $\mathcal{O}$. Let $\mathcal{N} = S - \mathcal{O}$ denote the set of configurations that have no obligation and $\hat{\mathcal{N}}$ denote the set of prefixes $\hat{S} - \hat{\mathcal{O}}$. For a prefix w a *choice set* is $C_w \subseteq \hat{\mathcal{O}} \cap (\{w\} \cdot V \cdot V^*)$. That is, it is a set of extensions of w that have obligations. Given a prefix $w \in V^+$ and a choice set C_w, an infinite path $w \cdot y$ is *good* if either (a) $y = x \cdot z$, $x \in \mathcal{N}^* \cdot \mathcal{O}$, and $w \cdot x \in C$, or (b) $y \in \mathcal{N}^\omega$ and $w \cdot y \in \alpha$. That is, either the first visit to $\mathcal{O}$ after w is in C or $\mathcal{O}$ is never visited and the infinite path is in α. A choice set is *good* for strategy σ if the following two conditions hold:

- Every infinite path $\pi = s_0, s_1, \ldots$ in M such that π has infinitely many prefixes in C is in α.
- For every prefix $w \in C$ we have $\mathsf{val}(\mathsf{reach}(C) \vee (\alpha \wedge \mathsf{safe}(\hat{\mathcal{N}}))) \bowtie r$, where $O(w) = \bowtie r$.

That is, if infinitely many obligations on the same path are chosen by a choice set then that path needs to be winning. In addition, from every obligation configuration met along the way Player 0 must be able to ensure that the value of either (a) reaching the choice set or (b) completely avoiding new obligations and satisfying the parity condition is high enough to meet the obligation. Let $\mathcal{C}_\sigma$ denote the set of good choice sets for σ.

For a prefix w the pre-value of w is

$$\tilde{\mathsf{v}}(G, w) = \sup_{C \in \mathcal{C}_w} \mathsf{val}(\mathsf{reach}(C) \vee (\alpha \wedge \mathsf{safe}(\neg \hat{\mathcal{O}})))$$

Finally, we define the value of w. For a prefix w such that $O(w) \neq \bot$ we define $\mathsf{v}(G, w)$ to be 1 if $\tilde{\mathsf{v}}(G, w) \bowtie r$, where $O(w) = \bowtie r$, and $\mathsf{v}(G, w)$ is 0 otherwise. For a prefix w such that $O(w) = \bot$ we define $\mathsf{v}(G, w)$ to be $\tilde{\mathsf{v}}(G, w)$.

Lemma 1. ([14]) *For every obligation parity game G and every prefix w such that $O(w) \neq \bot$ we have* $\mathsf{val}_0(G, w) \in \{0, 1\}$.

We define the value of Player 1 through a definition of the dual game. Dualization of a game consists of changing the roles of the two players and switching the goal to the complement. Here, the complementation of the goal includes complementation of both the parity condition and the obligations. Consider a game $G = ((V, E), (V_0, V_1, V_p), (\alpha, O))$. The dual game is $\mathtt{dual}(G) = ((V, E), (V_1, V_0, V_p), (\mathtt{dual}(\alpha), \mathtt{dual}(O)))$, where $\mathtt{dual}(\alpha)$ is defined as usual and $\mathtt{dual}(O)$ is defined below.

$$\mathtt{dual}(O)(v) = \begin{cases} \bot & \text{If } O(v) = \bot \\ >1-r & \text{If } O(v) = \geq r \\ \geq 1-r & \text{If } O(v) = >r \end{cases}$$

Intuitively, if in G Player 0 has the obligation to achieve more than r with the set φ, then the dual player (Player 0 in $\mathtt{dual}(G)$) has the obligation to achieve at least $1 - r$ with the complementary parity goal. Syntactically, $\mathtt{dual}(\mathtt{dual}(G)) = G$. We define the value of Player 1 in G to be the value of Player 0 in $\mathtt{dual}(G)$.

Theorem 2. ([14]) *Forall prefixes w in an OPG G we have* $\mathsf{val}_0(G, w) + \mathsf{val}_1(G, w) = 1$.

Theorem 3. ([14]) *The value of an obligation parity game G can be computed in NEXPTIME.*

Consider an obligation parity game $G = ((V, E), \ldots, (\alpha, O))$. We say that a configuration v is *pure* if $v \in V_p$ and there is a unique configuration v' such that $(v, v') \in E$. We say that the game is *uniform* if the following holds. There is a partition $\{V_i\}_{i \in \mathbb{N}}$ of V such that for every i we have, either (a) for every $v \in V_i$ we have $O(v) = \bot$ or (b) for every $v \in V_i$ we have $O(v) \neq \bot$ or v is pure. We say that $V_i \leq V_{i'}$ if there are some $v \in V_i$, $v' \in V_{i'}$ such that $(v, v') \in E$. The partition also satisfies that every chain according to $\leq$ is finite.

Theorem 4. ([12], [14]) *The value of a uniform obligation parity game G can be computed in EXPTIME.*

III. p-Automata

We define a specialized version of p-automata for the purposes of this paper based on [12].

We assume familiarity with basic notions of trees and (alternating) tree automata. For set T, let $B^+(T)$ be the set of positive Boolean *formulas* generated from elements $t \in T$, constants tt and ff, and disjunctions and conjunctions:

$$\varphi, \psi ::= t \mid \mathsf{tt} \mid \mathsf{ff} \mid \varphi \vee \psi \mid \varphi \wedge \psi \qquad (1)$$

Formulas in $B^+(T)$ are finite even if T is not.

For set Q, the set of states of a p-automaton, we define *term* sets $[\![Q]\!]_>$ as follows.

$$[\![Q]\!]_> = \{[\![q]\!]_{\bowtie p} \mid q \in Q, \bowtie \in \{\geq, >\}, p \in [0, 1]\}$$

Intuitively, a state $q \in Q$ of a p-automaton and its transition structure model a probabilistic path set. So $[\![q]\!]_{\bowtie p}$ holds in location s if the measure of paths that begin in s and satisfy q is $\bowtie p$.

An element of $Q \cup [\![Q]\!]_>$ is therefore either a state of the p-automaton, or a term of the form $[\![q]\!]_{\bowtie p}$. Given $\varphi \in B^+(Q \cup [\![Q]\!]_>)$, its closure $\mathsf{cl}(\varphi)$ is the set of all subformulas of φ according to (1). For a set Φ of formulas, let $\mathsf{cl}(\Phi) = \bigcup_{\varphi \in \Phi} \mathsf{cl}(\varphi)$.

A p-automaton A is a tuple $\langle \Sigma, Q, \delta, \varphi^{\mathrm{in}}, \alpha \rangle$, where Σ is a finite input alphabet, Q a set of states (not necessarily finite), $\delta \colon Q \times \Sigma \to B^+(Q \cup [\![Q]\!]_>)$ the transition function, $\varphi^{\mathrm{in}} \in B^+([\![Q]\!]_>)$ the initial condition, and α a parity acceptance condition.

For every set of atomic propositions $\mathbb{AP}$, p-automata $A = \langle 2^{\mathbb{AP}}, Q, \delta, \varphi^{\mathrm{in}}, \alpha \rangle$ have $\mathsf{MC}_{\mathbb{AP}}$, the set of Markov chains labeled by proposition in $\mathbb{AP}$, as set of inputs. For $M = (S, P, L, s^{\mathrm{in}}) \in \mathsf{MC}_{\mathbb{AP}}$, we define whether A accepts M by a reduction to an obligation parity game. The language of A is $\mathcal{L}(A) = \{M \in \mathsf{MC}_{\mathbb{AP}} \mid A \text{ accepts } M\}$.

We construct a game $G_{M,A} = ((V, E), (V_0, V_1, V_p), \kappa, (\tilde{\alpha}, O))$. Configurations of $G_{M,A}$ correspond to a subformula appearing in the transition of A and a location in M. Configurations with a term of the form $[\![q]\!]_{\bowtie p}$ correspond to obligations. All other configurations have no obligations. The Markov chain is accepted if the configuration $(\varphi^{\mathrm{in}}, s^{\mathrm{in}})$ has value 1 in $G_{M,A}$.

Formally, let $G_{M,A} = ((V, E), (V_0, V_1), \kappa, \mathcal{G})$, where the components of $G_{M,A}$ are as follows.

- $V = S \times \mathsf{cl}(\delta(Q, \Sigma))$.
- $V_0 = \{(s, \psi_1 \vee \psi_2) \mid s \in S \text{ and } \psi_1 \vee \psi_2 \in \mathsf{cl}(\delta(Q, \Sigma))\}$.
- $V_1 = \{(s, \psi_1 \wedge \psi_2) \mid s \in S \text{ and } \psi_2 \wedge \psi_2 \in \mathsf{cl}(\delta(Q, \Sigma))\}$.
- $V_p = S \times (Q \cup [\![Q]\!]_>)$.
- The set of edges E is defined as follows.

$$\begin{aligned} E = \ & \{((s, \varphi_1 \wedge \varphi_2), (s, \varphi_i)) \mid i \in \{1, 2\}\} \cup \\ & \{((s, \varphi_1 \vee \varphi_2), (s, \varphi_i)) \mid i \in \{1, 2\}\} \cup \\ & \{((s, q), (s', \delta(q, L(s))) \mid s' \in \mathsf{succ}(s)\} \cup \\ & \{((s, [\![q]\!]_{\bowtie p}), (s', \delta(q, L(s))) \mid s' \in \mathsf{succ}(s)\} \end{aligned}$$

- $\kappa((s, q), (s', \delta(q, L(s)))) =$ $\kappa((s, [\![q]\!]_{\bowtie p}), (s', \delta(q, L(s)))) = P(s, s')$.
- For $q \in Q$ and $p \in [0, 1]$ we have $\tilde{\alpha}(s, q) = \alpha(q)$, $\tilde{\alpha}(s, [\![q]\!]_{\bowtie p}) = \alpha(q)$. For every other configuration c we set $\tilde{\alpha}(c)$ to the maximal possible priority.
- For $q \in Q$ and $p \in [0, 1]$ we have $O(s, [\![q]\!]_{\bowtie p}) = \bowtie p$. For every other configuration c, we have $O(c) = \bot$.

Theorem 5. ([12], [14]) *Given a p-automaton $A = \langle 2^{\mathbb{AP}}, \ldots \rangle$, its language $\mathcal{L}(A)$ is well defined.*

Theorem 6. ([12], [14]) *Given a p-automaton A and a finite Markov chain M, we can decide whether $M \in \mathcal{L}(A)$ in NEXPTIME.*

In [12] we introduce also a simulation relation between p-automata that over approximates language inclusion. Decision of this simulation can be also reduced to obligation games. We do not include this definition or reduction here. For two automata A and B, we write $A \preceq B$ for B simulates A.

Theorem 7. ([12]) *Given p-automata A and B we have $A \preceq B$ implies $\mathcal{L}(A) \subseteq \mathcal{L}(B)$.*

IV. pCTL Model Checking

We mention several results concerning p-automata. The combination of these results shows that p-automata are a complete abstraction framework for infinite Markov chains with respect to pCTL model checking.

In this context, a complete abstraction framework is such that if some infinite state Markov chain M satisfies a pCTL formula ϕ there is a finite state p-automaton A such that $M \in \mathcal{L}(A)$ and A is simulated by the automaton for the language for ϕ.

Theorem 8. ([12]) *For every Markov chain $M \in \mathsf{MC}_{\mathbb{AP}}$, there is a p-automaton A_M such that the language $\mathcal{L}(A_M)$ is the bisimulation equivalence class of M.*

Lemma 9. ([12]) *For every pCTL formula ϕ over $\mathbb{AP}$, there is a p-automaton A_ϕ such that $M \models \phi$ iff $M \in \mathcal{L}(A_\phi)$.*

Theorem 10. ([12]) *For $M \in \mathsf{MC}_{\mathbb{AP}}$ and PCTL formula ϕ over $\mathbb{AP}$, $M \models \phi$ iff $M \in \mathcal{L}(A_\phi)$. Deciding $M \in \mathcal{L}(A_\phi)$ is polynomial in the size of M and linear in the size of ϕ.*

Finally, from all these results, completeness of abstraction for pCTL model checking follows.

Corollary 11. ([12]) *For every infinite Markov chain M and pCTL formula ϕ, we have $M \models \phi$ iff there is a* finite *p-automaton A with $M \in \mathcal{L}(A)$ and $A \preceq A_\phi$.*

Acknowledgment

The contents of this short paper is based on [12] and [14]. I am grateful to Michael Huth, Daniel Wagner, and Krishnendu Chatterjee for the great pleasure in working with them on these results.

References

[1] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker, "Prism: A tool for automatic verification of probabilistic systems," in *12th TACAS*, ser. Lecture Notes in Computer Science, vol. 3920. Springer, 2006, pp. 441–444.

[2] F. Ciesinski and C. Baier, "Liquor: A tool for qualitative and quantitative linear time analysis of reactive systems," in *3rd QEST*. IEEE Computer Society, 2006, pp. 131–132.

[3] E. Clarke, O. Grumberg, and D. Long, "Verification tools for finite-state concurrent systems," in *Decade of Concurrency – Reflections and Perspectives*, ser. Lecture Notes in Computer Science, vol. 803. Springer, 1993, pp. 124–175.

[4] B. Jonsson and K. Larsen, "Specification and refinement of probabilistic processes," in *6th LICS*. IEEE Computer Society, 1991, pp. 266–277.

[5] H. Fecher, M. Leucker, and V. Wolf, "*Don't Know* in probabilistic systems," in *13th SPIN*, ser. Lecture Notes in Computer Science, vol. 3925. Springer, 2006, pp. 71–88.

[6] B. Caillaud, B. Delahaye, K. Larsen, A. Legay, M. Pedersen, and A. Wasowski, "Compositional design methodology with constraint markov chains," in *7th QEST*. IEEE Computer Society, 2010, pp. 123–132.

[7] B. Delahaye, J.-P. Katoen, K. Larsen, A. Legay, M. Pedersen, F. Sher, and A. Wasowski, "Abstract probabilistic automata," in *12th VMCAI*, ser. Lecture Notes in Computer Science, vol. 6538. Springer, 2011, pp. 324–339.

[8] T. Henzinger, O. Kupferman, and S. Rajamani, "Fair simulation," *Inf. and Comp.*, vol. 173, no. 1, pp. 64–81, 2002.

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

[10] D. Dams and K. Namjoshi, "The existence of finite abstractions for branching time model checking," in *19th LICS*, 2004, pp. 335–344.

[11] ——, "Automata as abstractions," in *6th VMCAI*, ser. Lecture Notes in Computer Science, vol. 3385. Springer, 2005, pp. 216–232.

[12] M. Huth, N. Piterman, and D. Wagner, "Weak p-automata: Acceptors of Markov chains," in *7th QEST*. IEEE Computer Society Press, 2010.

[13] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal Aspects of Computing*, vol. 6, no. 5, pp. 512–535, 1994.

[14] K. Chatterjee and N. Piterman, "Obligation blackwell games and p-automata," 2011, in preparation.

[15] D. Martin, "Borel determinacy," *Annals of Mathematics*, vol. 65, pp. 363–371, 1975.

[16] ——, "The determinacy of Blackwell games," *The Journal of Symbolic Logic*, vol. 63, no. 4, pp. 1565–1581, 1998.