Probabilistic Contract Based Reasoning with Markov Decision Processes

Benoît Delahaye

IRISA / Université de Rennes 1, France, benoit.delahaye@irisa.fr

Supervisor(s): Benoît Caillaud and Albert Benveniste, IRISA / INRIA, France
Keywords: Reliability, Contracts, Markov Decision Processes

Abstract. In this paper we propose a probabilistic adaptation to the classical Assume/Guarantee contracts reasoning. This formalism relies on the notion of controllable Markov chains. We also propose an algorithm in order to compute probabilistic satisfaction and give possible definitions for probabilistic composition and dominance.

1 Introduction

Some conception methods using the composition of components that are described with contracts "Assumption on behavior of the environment", "Guarantee on the behavior of the component" allow a modular analysis of the functional properties of a system of reactive components [Dam05]. These techniques even permit the synthesis of adaptors allowing to reuse components described by "Assume/Guarantee" contracts [dAH05]. The aim of our current work is to allow reliability analysis of distributed systems of components described with Assume/Guarantee contracts.

In this paper, we describe a formalism based on Markov decision processes that should allow modular analysis of probabilistic properties such as, for example, reliability. Given a contract with probabilistic inputs and an implementation, we give an algorithm to compute the level of satisfaction of the contract by the implementation. This algorithm relies on the existence of pure stationary optimal strategies in Markov decision processes with mean-payoff functions [Gim07].

In section 2, we define controllable Markov chains and Markov decision Processes following the formalism of [Gim07]. In section 3, we give definitions of probabilistic contracts and satisfaction, and we propose an algorithm to compute the level of probabilistic satisfaction. We then give a possible adaptation of the classical composition and dominance operations to probabilistic contracts.

2 Background

This section introduces the concepts and definitions that will be used throughout the rest of this paper.

Definition 1. *A controllable Markov chain (CMC)* $\mathcal{A} = (S, A, (A(s))_{s\in S}, \mathbb{P})$ *consists of a finite set of states* S*, a finite set of actions* A*. To each state* s *of* S*, we associate* $A(s) \subseteq A$ *the set of actions available in* s*. For each* $s, t \in S$ *and* $a \in A(s)$*,* $\mathbb{P}(t|s, a)$ *is the conditional probability to go from* s *to* t *with the execution of the action* a*, with* $\forall s \in S$ *and* $a \in A(s)$*,* $\sum_{t\in S} \mathbb{P}(t|s, a) = 1$*.*

A *controllable Markov chain* is thus a transition system with a discrete evolution: at each state s, the controller chooses an available action $a \in A(s)$, and the state changes to state t with probability $\mathbb{P}(t|s, a)$.

A *finite history* is a finite sequence $h = s_0a_0s_1 \dots a_{n-1}s_n \in S(AS)^*$ such that for all i, $a_i \in A(s_i)$. The set of finite histories of $\mathcal{A}$ is denoted $\mathcal{H}^*_{\mathcal{A}}$, and $\mathcal{H}^*_{\mathcal{A},s}$ if we impose $s_0 = s$. Similarly, an *infinite history* is an infinite sequence $h = s_0a_0 \dots \in S(AS)^\omega$. The set of infinite histories is written $\mathcal{H}^\omega_{\mathcal{A}}$.

A strategy is a mapping $\sigma : \mathcal{H}^*_{\mathcal{A}} \to \mathcal{D}(A)$, where $\mathcal{D}(A)$ is the set of probability distributions on A such that, if h is a finite history with target t, $\sigma(h)$ puts non-zero probabilities only on actions available in t.

We consider *pure strategies*, that give a deterministic choice of action (i.e. for a given finite history h and an action a, either $\sigma(h)(a) = 0$ or $\sigma(h)(a) = 1$). A strategy is called stationary if for all finite histories with the same target t, the given distribution is the same. As a consequence, pure stationary strategies can be identified with mappings $\sigma : S \to A$.

Following [Gim07], a Markov decision process (MDP) $(\mathcal{A}, \phi)$ is the result of the association of a controllable Markov chain $\mathcal{A}$ and a payoff function ϕ. In our context, the aim is to choose the right strategy in order to optimize the expected payoff. Before giving the definitions of the expected payoff of a state s and the optimal strategies, we need to properly define the probabilities over infinite histories. If we consider a pure strategy σ, the probability of a finite history $h = s_0a_0 \dots s_n$ is $\mathbb{P}(s_1|s_0, a_0) \cdot \mathbb{P}(s_2|s_1, a_1) \dots \mathbb{P}(s_n|s_{n-1}, a_{n-1})$. This is extended as following for infinite histories: We equip the set of infinite histories with source s, $\mathcal{H}^\omega_{\mathcal{A},s}$, with the σ-field generated by the random variables S_n, $n \in \mathbb{N}$, such that $S_n(s_0a_0 \dots) = s_n$. As shown in [Gim07], there exists a unique probability measure $\mathbb{P}^\sigma_s$ on $\mathcal{H}^\omega_{\mathcal{A},s}$ such that for each finite history $s_0a_0 \dots a_{n-1}s_n$, $\mathbb{P}^\sigma_s(S_n = s_n \mid S_0A_0 \dots A_{n-1} = s_0 \dots a_{n-1}) = \mathbb{P}(s_n|s_{n-1}, a_{n-1})$

Definition 2. *A payoff function is a bounded measurable function* $\phi : S^\omega \to \mathbb{R}$*.*

Considering a state s *and a strategy* σ*, the expected payoff under probability* $\mathbb{P}^\sigma_s$ *is* $\mathbb{E}^\sigma_s[\phi(S_0S_1 \dots)]$ *that we will denote* $\mathbb{E}^\sigma_s[\phi]$*.*

A strategy is said to be *optimal* if for all $s \in S$, $\mathbb{E}^\sigma_s$ is optimal over all strategies (the optimality condition could be either sup or inf for example).

In the following section, we will consider mean-payoff MDPs, where a reward $r(s\ a\ t)$ is given on each transition. The payoff associated to an infinite history $s_0a_0s_1 \dots$ is

$$\limsup_{n\in\mathbb{N}} \frac{1}{n+1} \sum_{i=0}^{n} r(s_ia_is_{i+1})$$

It is proved in [Gim07] that mean-payoff MDPs have a pure stationary optimal strategy.

3 Probabilistic extension and application to MDPs

In this section we will show how we can use the existence of pure stationary optimal strategies in mean-payoff MDPs in order to compute probabilistic satisfaction. In order to use this result, we must consider that the probabilities in the contracts are contained in some of the inputs. In the first subsection we will define probabilistic contracts and satisfaction. In the second and third subsections we will give a probabilistic adaptation of the classical dominance and composition operations.

3.1 Probabilistic contracts and satisfaction

A probabilistic contract is a deterministic Input/Output machine with assumptions on its inputs. We have the following definition:

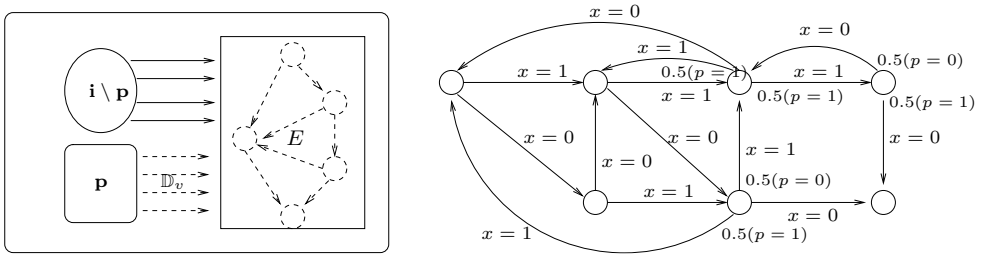
Definition 3. *A probabilistic contract is a tuple* $\mathcal{C} = (\mathbf{u}, \mathbf{c}, \mathbf{p}, (\mathbb{D}_v)_{v\in\mathbf{p}}, E)$ *where*

- $\mathbf{u} \cup \mathbf{c}$ *is the signature of the contract (i.e. all the ports and variables used) where* $\mathbf{u}$ *are the* uncontrolled *variables (inputs) and* $\mathbf{c}$ *the* controlled *variables (internal variables and outputs).*
- $\mathbf{p} \subseteq \mathbf{u}$ *is the set of probabilistic inputs and* $(\mathbb{D}_v)_{v\in\mathbf{p}}$ *the distributions on their values.*
- E *is a deterministic transition system where transitions are labelled with the values of the inputs.*

This definition is quite similar to the Assume/Guarantee formalism defined in [SPE]. In this formalism, a contract is a pair (A, G) of sets of runs defining respectively the assumptions on the environment and the guaranteed behavior. These non-probabilistic contracts have a canonical form given by $(A, G \cup \neg A)$, and knowing the maximal implementation $G \cup \neg A$ is enough to capture all the behaviors. The machine E in the definition above can be viewed as the maximal implementation of a non-probabilistic contract (A, G), where A would contain the assumptions on the non-probabilistic inputs. There will then remain inputs for which no assumptions are made $(\mathbf{u} \setminus \mathbf{p})$ - we will call them non-deterministic inputs; and inputs for which probabilistic assumptions are made $(\mathbf{p})$ but E does not distinguish them.

Probabilistic contracts can be seen as controllable Markov chains. Indeed, if we map the distributions of the inputs on the transition system E, we obtain another transition system $E_{\mathbf{p}}$ where, in each state s, a controller could choose an action (the value of an input for which no assumption is made), and the state changes to t with a probability that depends on the values of the probabilistic ports (see Example in Figure 1(b) where x is non-deterministic and p probabilistic).

In order to define the probabilistic satisfaction, the idea is to build a Markov decision process with the machine and the controllable Markov chain that represents the contract, and to use the results of [Gim07] to "measure" satisfaction. We propose the following algorithm:



(a) A probabilistic contract (b) A controllable Markov chain

Fig. 1. A probabilistic contract and a controllable Markov chain

1. Consider the controllable Markov chain associated to the contract $E_{\mathbf{p}} = (S_1, A, (A(s)_{s \in S_1}), \mathbb{P})$.
2. Compute the synchronous product between $E_{\mathbf{p}}$ and the implementation $M = (S_2, A, \delta)$, which gives a Markov decision process $\mathcal{A} = (S', A, (A'(s))_{s \in S'}, \mathbb{P}', \phi)$ where $S' = S_1 \times S_2$. The accepted actions in (s_1, s_2) are the accepted actions of s_1 *and* those of s_2, and the transitions and rewards are as follow:
   - If $a \in A(s_1)$ and $\delta(s_2, a) = (s_2')$ then $\mathbb{P}'((t_1, s_2')|(s_1, s_2), a) = \mathbb{P}(t_1|s_1, a)$ and $r((s_1, s_2)a(t_1, s_2')) = 0$.
   - If $a \notin A(s_1)$ and $\delta(s_2, a) = (s_2')$ then $\mathbb{P}'((s_1, s_2')|(s_1, s_2), a) = 1$ and $r((s_1, s_2)a(s_1, s_2')) = 1$.
   - If $a \in A(s_1)$ and $\delta(s_2, a)$ is not defined then $\mathbb{P}'((t_1, s_2)|(s_1, s_2), a) = \mathbb{P}(t_1|s_1, a)$ and $r((s_1, s_2)a(t_1, s_2)) = 0$.
3. Consider the mean-payoff of the infinite runs and compute the maximal expected value β that you can obtain. This value can be seen as the maximal expected probability for the implementation to step outside of the behaviour guaranteed by the contract. Thus we will say that M satisfies the contract $\mathcal{C}$ with a probability at least $1 - \beta$.

Definition 4. *A machine M satisfies the probabilistic contract $\mathcal{C}$ with level α (written $M \models_\alpha \mathcal{C}$) if the maximal expected value of the MDP defined in the above algorithm is $1 - \alpha$.*

The soundness of this definition is assured by the results obtained in [Gim07] that prove the existence of pure stationary optimal strategies in the case of mean-payoff MDPs.

We now need to adapt the classical operations on non-probabilistic contracts in order to keep a compositional formalism. In the following subsections, we will present a possible adaptation of the classical definitions of composition and dominance to probabilistic contracts. We expect that these definitions should enable us to prove that satisfaction and dominance are compositional, but we are still working on them.

3.2 Composition

The main idea behind composition is that we want to prevent the contracts to be incompatible, so we strictly forbid the bad cases (for example two contracts that have the same probabilistic ports). We then use synchronous composition between the transition systems.

Definition 5. *If* $\mathcal{C}_1 = (\mathbf{u}_1, \mathbf{c}_1, \mathbf{p}_1, (\mathbb{D}_{1,v})_{v\in\mathbf{p}_1}, E_1)$ *and* $\mathcal{C}_2 = (\mathbf{u}_2, \mathbf{c}_2, \mathbf{p}_2, (\mathbb{D}_{2,v})_{v\in\mathbf{p}_2}, E_2)$ *are two probabilistic contracts, then their composition* $\mathcal{C}_1 \parallel \mathcal{C}_2$ *is defined if and only if*

- $\mathbf{c}_1 \cap \mathbf{c}_2 = \emptyset$
- $\mathbf{p}_1$ *and* $\mathbf{p}_2$ *are* disjointed *sets of* uncontrolled *ports in both contracts.*

In this case $\mathcal{C}_1 \parallel \mathcal{C}_2 = (\mathbf{u}, \mathbf{c}, \mathbf{p}, (\mathbb{D}_v)_{v\in\mathbf{p}}, E)$*, with*

- $\mathbf{u} = \mathbf{u}_1 \cup \mathbf{u}_2 \setminus (\mathbf{c}_1 \cup \mathbf{c}_2)$
- $\mathbf{c} = \mathbf{c}_1 \cup \mathbf{c}_2$
- $\mathbf{p} = \mathbf{p}_1 \cup \mathbf{p}_2$
- $(\mathbb{D}_v)_{v\in\mathbf{p}} = (\mathbb{D}_{1,v})_{v\in\mathbf{p}_1} \cup (\mathbb{D}_{2,v})_{v\in\mathbf{p}_2}$
- $E = E_1 \times E_2$ *(the synchronous product).*

As the composition between implementations is also synchronous, we expect that we will be able to compare $\alpha \cdot \beta$ and γ with $M_1 \models_\alpha \mathcal{C}_1$, $M_2 \models_\beta \mathcal{C}_2$, and $M_1 \times M_2 \models_\gamma \mathcal{C}_1 \parallel \mathcal{C}_2$.

3.3 Dominance

Dominance must be seen as a way to simplify a contract. It must enable to be less precise, but keep some information. This could be done by changing some ports that were considered probabilistic into ports that are non-deterministic. This can be seen as taking the worst possible behavior for the ports we change into non-deterministic, and the level of satisfaction for a machine would be less than for the original contract.

In fact we also want to give the possibility to slightly change the behavior guaranteed in the contracts. Thus the following definition:

Definition 6. *A contract* $\mathcal{C}_1 = (\mathbf{u}_1, \mathbf{c}_1, \mathbf{p}_1, (\mathbb{D}_{1,v})_{v\in\mathbf{p}_1}, E_1)$ *dominates a contract* $\mathcal{C}_2 = (\mathbf{u}_2, \mathbf{c}_2, \mathbf{p}_2, (\mathbb{D}_{2,v})_{v\in\mathbf{p}_2}, E_2)$ *with level* α *(written* $\mathcal{C}_1 \preceq_\alpha \mathcal{C}_2$*) if and only if*

- $\mathbf{u}_1 = \mathbf{u}_2$ *and* $\mathbf{c}_1 = \mathbf{c}_2$*,*
- $\mathbf{p}_1 \subseteq \mathbf{p}_2$ *and* $\forall v \in \mathbf{p}_1,\ \mathbb{D}_{1,v} = \mathbb{D}_{2,v}$
- $E_1 \models_\alpha \mathcal{C}_2$*.*

This approach makes it quite easy to check the level of refinement between two contracts once we know how to check satisfaction. Once again we expect that we will be able to say something like "If $M \models_\alpha \mathcal{C}_1$ and $\mathcal{C}_1 \preceq_\beta \mathcal{C}_2$, then $M \models_{\alpha\cdot\beta} \mathcal{C}_2$", but the proof is still to be formally written.

4 Conclusion and future work

In this paper, we have presented a formalism that uses classical results about the existence of pure stationary optimal strategies in mean-payoff MDPs [Gim07] in order to define probabilistic contracts and their satisfaction. We have also given definitions of composition and dominance for probabilistic contracts. We now need to prove that, as we expect, these operations are compatible with the definition of probabilistic satisfaction that we gave above.

This formalism can be used in order to compute modular reliability analysis, but we still have to write it formally and to compare it to other notions like fault trees and binary decision diagrams ([DR05], [BDRS04]), or even dynamic fault trees ([BCS07]).

Another extension of this work would be to use modal or acceptance automata instead of deterministic input/output machines, which would be a quite different approach to the assume/guarantee formalism.

References

[BCS07] Hichem Boudali, Pepijn Crouzen, and Mariëlle Stoelinga, *A compositional semantics for dynamic fault trees in terms of interactive markov chains*, ATVA (Kedar S. Namjoshi, Tomohiro Yoneda, Teruo Higashino, and Yoshio Okamura, eds.), Lecture Notes in Computer Science, vol. 4762, Springer, 2007, pp. 441–456.

[BDRS04] M. Boiteau, Y. Dutuit, A. Rauzy, and J.-P. Signoret, *The altarica data-flow language in use : Assessment of production availability of a multistates system*, Reliability Engineering and System Safety, 2004.

[dAH05] L. de Alfaro and T. Henzinger, *Interface-based design*, Engineering Theories of Software-intensitive Systems (M. Broy, J. Gruenbauer, D. Harel and C.A.R Hoare, eds.), vol. 195, NATO Science Series : Mathematics, Physics, and Chemistry, Springer, 2005, pp. 83–104.

[Dam05] W. Damm, *Controlling speculative design processes using rich component models*, ACSD, IEEE Computer Society, 2005, pp. 118–119.

[DR05] Y. Dutuit and A. Rauzy, *Approximate estimation of system reliability via fault trees*, Reliability Engineering and System Safety, vol. 87(2), 2005, pp. 163–172.

[Gim07] Hugo Gimbert, *Pure stationary optimal strategies in markov decision processes*, STACS (Wolfgang Thomas and Pascal Weil, eds.), Lecture Notes in Computer Science, vol. 4393, Springer, 2007, pp. 200–211.

[SPE] SPEEDS : Speculative and Exploratory Design in System Engineering. http://www.embedded-computing.com/news/db/?3398.