Compositinal Reasoning for Assume/Guarantee Contracts Combining Stochastic and Nondeterministic Aspects

Benoît Delahaye[1], Benoît Caillaud[2], and Axel Legay[2]

[1] Université de Rennes 1 / IRISA, Rennes, France
[2] INRIA/IRISA, Rennes, France
{benoit.delahaye, benoit.caillaud, axel.legay}@irisa.fr

Abstract. In this paper, we present an Assume/Guarantee contract formalism for systems combining nondeterministic and stochastic aspects. Contracts dictate how component variables behave, being either non-deterministic, or probabilistic. As shown in the paper, our theory preserves compositionality and therefore enables a modular but behavior dependent analysis of systems behaviors.

1 Introduction

Several industrial sectors involving complex embedded systems have recently experienced deep changes in their organization, aerospace and automotive being the most prominent examples. In the past, they were organized around vertically integrated companies, supporting in-house design activities. These sectors have now evolved into more specialized, horizontally structured companies: *E*quipment *S*uppliers (ESs) and *O*riginal *E*quipment *M*anufacturers (OEMs). OEMs perform system design and integration by importing/combining/reusing entire subsystems (also called components) provided by ESs. As a consequence, part of the design load has been moved from OEMs to ESs. An inconvenient of this change is the increased occurrence of late error discovery, i.e., system level design errors uncovered at integration time. Such errors could call for a completely new design of the system; moreover verifying complex systems is generally a tedious task.

There is thus the need for modular verification techniques that allows to discover errors at the early stage of the design. Such techniques should be independent from the way components are combined and must give strong confidence regarding the correctness of the entire system without proceeding to a complete analyse. Developing these formal techniques pass by the study of a mathematical formalism characterizing both properties that must be verified and component behaviors/interactions. Results exists (see [6] and [11] for illustrations), but only for limited classes of components, properties, and interactions. The objective of this paper is to go one step further by studying systems that combines nondeterministic and stochastic aspects. More precisely, we will propose : (1) a more complete set of component-based design operations, (2) more complex properties than the classical safety/liveness properties that are usually considered in the literature, and (3) a compositional reasoning framework for such systems.

The semantics foundations presented in this paper consists in a mathematical formalism designed to support a component based design methodology and to offer modular and scalable verification techniques. At its basis, the mathematical formalism is a language theoretic abstraction of systems behaviour called *contract* [4]. Contracts allow to distinguish hypotheses on a component (*guarantees*), from hypotheses made on its environment (*assumptions*). In the paper we will focus on developing a contract-based compositional theory for two classes of systems, that are (1) nonstochastic and possibly nondeterministic systems, and (2) stochastic and possibly nondeterministic systems. As in classical nonmodular verification [6,14], the satisfaction relation will be Boolean for nonstochastic systems and quantitative otherwise, hence leading to two notions of contracts. In addition, we will consider two measures of satisfaction : *reliability* and *availability*. Availability is a measure of the time during which a system satisfies a given property, for all possible runs of the system. In contrast, reliability is a measure of the set of runs of a system that satisfy a given property. Both quantities play an important role when designing, for instance, mission-critical systems. Our notion of satisfaction is assumption-dependant in the sense that runs that do not satisfy the assumptions are considered to be "correct". This interpretation, which has been suggested by many industrial partners in European projects such as COMBEST [8] or SPEEDS [12], should not be confused with the notion of assume/guarantee reasoning in model checking [7], where assumptions are used as constraints on the environment's behavior. We will show that the model checking interpretation is in fact incompatible with compositional design operations such as conjunction.

We also propose mathematical definitions for crucial component-based design operations including composition, conjunction and refinement. It is known that most of industrial requirements for component-based design translates to those operations (see [2] for an argumentation). Composition between contracts, which mimics classical composition for systems, consists in taking the intersection between the assumptions and the intersection between the guarantees. Conjunction is a more intriguing operation that has no translation at the level of systems; its consists in producing a contract whose assumptions are the union of the original ones and guarantees are the intersection of the original ones. Roughly speaking, the conjunction of two contracts represents their common requirements. We say that a contract refines another contract if it guarantees more and assumes less. The definition is boolean for nondeterministic systems and quantitative otherwise. We also establish a compositional reasoning theory for those operations and the two notions of satisfiability we consider. The theory differs with the type of contracts under consideration. As an example, we will show that if a nonstochastic system S_1 reliably satisfies[3] a contract C_1 and a nonstochastic system S_2 reliably satisfies a contract C_2, then the composition of the two systems reliably satisfies the composition of the two contracts. When moving to stochastic systems, we will show that if S_1 satisfies C_1 with probability α and S_2 satisfies C_2 with probability β, then their composition satisfies the composition of C_1 and C_2 with probability at least $\alpha + \beta - 1$. The advantage being that the composition, which may be large, does not need to be

[3] "Reliably satisfy" means that all the runs that satisfy the assumption must satisfy the guarantee

computed. Our theory is fully general as it assumes that both systems and contracts are possibly infinite sets of runs.

Most of our theory is developed assuming that both assumptions and guarantees are represented by assertions on systems which are themselves represented by sets of runs. Our last contribution is to propose effective and symbolic representations for contracts and systems. Those representations rely on an automata-based representation of possibly infinite sets of runs. Assuming that assumptions and guarantees are represented with Büchi automata (which allows to specify assumptions and guarantees with logics such as LTL or PSL), we observe that checking if a (stochastic) system satisfies a reliability property can be done with classical techniques implemented in tools such as SPIN [13] or LIQUOR [5]. In the paper, we show that satisfaction of availability properties can be checked with an extension of the work presented in [9]. Another contribution is to show that operations between and on contracts can easily be performed on the automata-based representations.

From the theoretical point of view, our work is the first contribution on (probabilistic) contracts that consider both reliability and availability apsects with compositional reasoning theorems. From the practical point of view, our work is an inspiration for extending tools such as SPIN and LIQUOR from nonmodular to modular verification.

Related work This work is based on previous work on contracts presented in [3], where availability and stochastic aspects are not considered. Works on behavioral types in process algebras bear commonalities with contract theories. In a similar way, the probabilistic contract theory must be compared with stochastic process algebras [10,1]. In both cases, the main difference is that compositional reasoning is possible only in contract theories thanks to the fact that contracts are implications where an assumption implies a guarantee. A second major difference with process agebras, is that contract theories are general and can be instantiated in many different effective automata-based settings. This covers many logical frameworks (CTL, LTL, PCTL, PSL, . . .) for specifying properties of components.

References

1. S. Andova. Process algebra with probabilistic choice. In *ARTS*, volume 1601 of *Lecture Notes in Computer Science*, pages 111–129. Springer, 1999.
2. E. Badouel, A. Benveniste, B. Caillaud, T. Henzinger, A. Legay, and R. Passerone. Contract theories for embedded systems, a white paper. Technical report, INRIA/IRISA Rennes, 2009.
3. A. Benveniste, B. Caillaud, A. Ferrari, L. Mangeruca, R. Passerone, and C. Sofronis. Multiple viewpoint contract-based specification and design. In *FMCO'07*, volume 5382 of *Lecture Notes in Computer Science*, pages 200–225. Springer, October 2008.
4. A. Benveniste, B. Caillaud, and R. Passerone. A generic model of contracts for embedded systems. *CoRR*, abs/0706.1456, 2007.
5. F. Ciesinski and C. Baier. Liquor: A tool for qualitative and quantitative linear time analysis of reactive systems. In *QEST*, pages 131–132. IEEE Computer Society, 2006.
6. E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

7. J. M. Cobleigh, D. Giannakopoulou, and C. S. Pasareanu. Learning assumptions for compositional verification. In H. Garavel and J. Hatcliff, editors, *TACAS*, volume 2619 of *Lecture Notes in Computer Science*, pages 331–346. Springer, 2003.
8. Combest. http://www.combest.eu.com.
9. L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. Model checking discounted temporal properties. In *TACAS*, volume 2988 of *Lecture Notes in Computer Science*, pages 77–92. Springer, 2004.
10. N. López and M. Núñez. An overview of probabilistic process algebras and their equivalences. In *Validation of Stochastic Systems*, volume 2925 of *Lecture Notes in Computer Science*, pages 89–123. Springer, 2004.
11. R. Milner. *Communication and Concurrency*. prentice hall, 1989.
12. Speeds. http://www.speeds.eu.com.
13. The spin tool (spin). Available at `http://spinroot.com/spin/whatispin.html`.
14. M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS*, pages 327–338. IEEE, 1985.