Form Methods Syst Des (2011) 38: 1–32
DOI 10.1007/s10703-010-0107-8

Probabilistic contracts: a compositional reasoning methodology for the design of systems with stochastic and/or non-deterministic aspects

Benoît Delahaye · Benoît Caillaud · Axel Legay

Published online: 8 December 2010

Abstract A *contract* allows to distinguish hypotheses made on a system (the guarantees) from those made on its environment (the assumptions). In this paper, we focus on models of Assume/Guarantee contracts for (stochastic) systems. We consider contracts capable of capturing reliability and availability properties of such systems. We also show that classical notions of Satisfaction and Refinement can be checked by effective methods thanks to a reduction to classical verification problems. Finally, theorems supporting compositional reasoning and enabling the scalable analysis of complex systems are also studied.

Keywords Assume/Guarantee reasoning · Contracts · Probabilistic reasoning · Reliability analysis · Availability

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

A preliminary version of this paper appeared in the 10th International Conference on Application of Concurrency to System Design.
This work was supported by the European STREP-COMBEST project no. 215543 and by an "Action de Recherche Collaborative" ARC (TP)I.

B. Delahaye
Université de Rennes 1/IRISA, Rennes, France
e-mail: benoit.delahaye@irisa.fr

B. Caillaud · A. Legay (✉)
INRIA/IRISA, Rennes, France
e-mail: axel.legay@irisa.fr

B. Caillaud
e-mail: benoit.caillaud@irisa.fr

*M*anufacturers (OEMs). OEMs perform system design and integration by importing, combining, or reusing entire subsystems (also called components) provided by ESs.

In this context, techniques that allow to discover errors at the early stage of the design are particularly appealing. 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 analysis. Developing these formal techniques pass by the study of a mathematical formalism characterizing both properties that must be verified and component behaviors/interactions. Results exist (see [18] and [40] 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 combine non-deterministic 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 ompositional reasoning framework for such systems.

The semantics foundations presented in this paper consist of 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 behavior called *contract* [5]. 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) non-stochastic and possibly non-deterministic systems, and (2) stochastic and possibly non-deterministic systems. As in classical non modular verification [18, 51], the satisfaction relation will be Boolean for non-stochastic systems and quantitative otherwise, hence leading to two notions of contracts. In addition, we will consider two measures of satisfaction, namely *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-dependent 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, is needed to propose compositional design operations such as conjunction.

Aside from the satisfaction relation, any good contract-based theory should also support the following requirements.

1. *Refinement and shared refinement. Refinement* of contracts expresses inclusion of sets of models, and therefore allows to compare contracts.
2. *Structural composition.* The contract theory should also provide a combination operator on contracts, reflecting the standard composition of models by, *e.g.* parallel product.
3. *Logical composition/conjunction.* Different aspects of systems are often specified by different teams. The issue of dealing with multiple aspects or multiple viewpoints is thus essential. It should be possible to represent several contracts (viewpoints) for the same system, and to combine them in a logical/conjunctive fashion.

The theory should also support incremental design (contracts can be composed/conjunct in any order) and independent implementability (composable contracts can always be refined separately) [25].

In the paper, we propose mathematical definitions for composition, conjunction and refinement. It is in fact known that most of industrial requirements[1] for component-based design translate to those operations (see [16] 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 produces a contract whose assumptions are the union of the original ones and guarantees are the intersection of the original ones. We say that a contract refines another contract if it guarantees more and assumes less. The definition is Boolean for non deterministic systems and quantitative otherwise.

We also establish a *compositional reasoning verification* theory for those operations and the two notions of satisfiability we consider. This methodology allows to reason on the entire design by only looking at individual components . The theory differs with the type of contracts under consideration. As an example, we will show that if a non stochastic system S_1 reliably satisfies[2] a contract C_1 and a non-stochastic 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 theory is fully general as it assumes that both systems and contracts are 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 [43] or PSL [27]), we observe that checking if a (stochastic) system satisfies a reliability property can be done with classical techniques implemented in tools such as SPIN [49] or LIQUOR [14]. In the paper, we show that satisfaction of availability properties can be checked with an extension of the work presented in [23]. Finally, we also show that operations between and on contracts can easily be performed on the automata-based representations.

Structure of the paper In Sect. 2, we recap all the basic notions that will be used through the rest of the paper. Section 3 introduces the concept of contracts for non-stochastic systems. In Sect. 4, the concept is extended to the stochastic case. Section 5 is dedicated to related work. Finally, Sect. 6 concludes the paper.

2 Preliminaries

In this section, we recap some definitions and concepts related to automata theory. We then introduce some notations and concepts that will be used in the rest of the paper.

Let Σ be an alphabet. A finite word over Σ is a mapping $w : \{0, \ldots, n-1\} \to \Sigma$. An *infinite word* (or *ω-word*) w over Σ is a mapping $w : \mathbb{N} \to \Sigma$. An automaton is a tuple $A = (\Sigma, Q, Q_0, \delta, F)$, where Σ is a finite alphabet, Q is a set of *states*, $Q_0 \in Q$ is the set of *initial states*, $\delta : Q \times \Sigma \to 2^Q$ is a *transition function* ($\delta : Q \times \Sigma \to Q$ if the automaton is deterministic), and $F \subseteq Q$ is a set of *accepting* states. A *finite run* of A on a

[1] Example: those of the European projects COMBEST [20] and SPEEDS [48].

[2] “Reliably satisfy” means that all the runs that satisfy the assumption must satisfy the guarantee.

finite word $w : \{0, \dots, n-1\} \to \Sigma$ is a labeling $\rho : \{0, \dots, n\} \to Q$ such that $\rho(0) \in Q_0$, and $(\forall 0 \le i \le n-1)(\rho(i+1) \in \delta(\rho(i), w(i)))$. A finite run ρ is *accepting* for w if $\rho(n) \in F$. An *infinite run* of A on an infinite word $w : \mathbb{N} \to \Sigma$ is a labeling $\rho : \mathbb{N} \to Q$ such that $\rho(0) \in Q_0$, and $(\forall 0 \le i)(\rho(i+1) \in \delta(\rho(i), w(i))$. An infinite run ρ is *accepting* for w with the Büchi condition if $\mathit{inf}(\rho) \cap F \neq \emptyset$, where $\mathit{inf}(\rho)$ is the set of states that are visited infinitely often by ρ. We distinguish between finite-word automata that are finite automata accepting finite words, and Büchi automata [12] that are finite automata accepting infinite words. A finite-word automaton accepts a finite word w if there exists an accepting finite run for w in this automaton. A Büchi automaton accepts an infinite word w if there exists an accepting infinite run for w in this automaton. The set of words accepted by A is called the *language accepted by* A, and is denoted by $L(A)$. Finite-word and Büchi automata are closed under intersection and union. Inclusion and emptiness are also decidable. Both finite-word and Büchi automata are closed under complementation and, in both cases, the construction is known to be exponential. However, the complementation operation for Büchi automata requires intricate algorithms that not only are worst-case exponential, but are also hard to implement and optimize (see [52] for a survey).

Let $\mathbb{N}_\infty = \mathbb{N} \cup \{\omega\}$ be the closure of the set of natural integers and $\mathbb{N}_n = [0 \dots n-1]$ the interval ranging from 0 to $n-1$. Let V be a finite set of *variables* that takes values in a *domain* D. A *step* $\sigma : V \to D$ is a valuation of variables of V. A *run* on V is a sequence of steps. More precisely, a finite or infinite run is a mapping $w : \mathbb{N}_n \to V \to D$, where $n \in \mathbb{N}_\infty$ is the length of w, also denoted $|w|$. Let ε be the run of length 0. Given $k < n$, the prefix of w of length $k+1$ is denoted $w_{[0,k]}$. Given a variable $v \in V$ and a time $i \ge 0$, the value of v at time i is given by $w(i)(v)$. Given w a finite run on V and σ a step on the same variables, $w.\sigma$ is the run of length $|w|+1$ such that $\forall i < |w|$, $(w.\sigma)(i) = w(i)$ and $(w.\sigma)(|w|) = \sigma$. The set of all finite (respectively infinite) runs on V is denoted by $[V]^*$ (respectively $[V]^\omega$). The set of finite and infinite runs on V is denoted $[V]^\infty = [V]^* \cup [V]^\omega$. Denote $[V]^n$ (respectively $[V]^{\le n}$) the set of all runs on V of length exactly n (respectively not greater than n). The *complement* of $\Omega \subseteq [V]^\infty$ is given by $\neg\Omega = [V]^\infty \setminus \Omega$. The *projection* of w on $V' \subseteq V$ is the run $w \downarrow_{V'}$ such that $|w \downarrow_{V'}| = |w|$ and $\forall v \in V'$, $\forall n \ge 0$, $w \downarrow_{V'} (n)(v) = w(n)(v)$. Given a run w' on V', the *inverse-projection* of w' on V is the set of runs defined by $w' \uparrow^V = \{w \in [V]^\infty \mid w \downarrow_{V'} = w'\}$. The extension of a set of runs Ω on V to V', with $V \subseteq V'$, is the set of runs on V' $\Omega \uparrow^{V'} = \bigcup_{w \in \Omega} w \uparrow^{V'}$.

A *system* over V is a pair (V, Ω), where Ω is a set of (finite and/or infinite) runs on V. Let $S = (V, \Omega)$ and $S' = (V', \Omega')$ be two systems. The extension of S to an alphabet V'' such that $V \subseteq V''$ is the system $S \uparrow^{V''} = (V'', \Omega \uparrow^{V''})$. The *inclusion* of S in S', denoted $S \subseteq S'$, holds whenever $\Omega \uparrow^{V \cup V'} \subseteq \Omega' \uparrow^{V \cup V'}$. The *composition* of S and S', denoted $(V, \Omega) \cap (V', \Omega')$, is given by $(V \cup V', \Omega'')$ with $\Omega'' = \Omega \uparrow^{V \cup V'} \cap \Omega' \uparrow^{V \cup V'}$. The *complement* of S, denoted $\neg S$, is given by $\neg S = (V, \neg\Omega)$. The restriction of system $S = (V, \Omega)$ to runs of length not greater than $n \in \mathbb{N}_\infty$ (respectively exactly n) is the system $S|^{\le n} = (V, \Omega \cap [V]^{\le n})$ (respectively $S|^n = (V, \Omega \cap [V]^n)$). In Sect. 4, it will be assumed that systems can respond to every possible input on a set of probabilistic variables. Such systems are said to be *receptive* to those variables. Given $U \subseteq V$, a set of distinguished variables, system $S = (V, \Omega)$ is *U-receptive* if and only if for all finite run $w \in \Omega \cap [V]^*$ and for all input $\rho : U \to D$, there exists a step $\sigma : V \to D$ such that $\sigma \downarrow_U = \rho$ and $w.\sigma \in \Omega$. Given $U \subseteq V \cap V'$, two U-receptive systems $S = (V, \Omega)$ and $S' = (V', \Omega')$ are U-compatible if and only if $S \cap S'$ is U-receptive.

A *symbolic transition system* over V is a tuple $\mathit{Symb} = (V, Q_s, T, Q_{s0})$, where V is a set of variables defined over a *finite* domain D, Q_s is a set of states (a state is a mapping from V to D), $T \subseteq Q_s \times Q_s$ is the transition relation, and $Q_{s0} \subseteq Q_s$ is the set of initial states. A run of *Symb* is a possibly infinite sequence of states $q_{s0}q_{s1}\ldots$ such that for each $i \geq 0$ $(q_{si}, q_{s(i+1)}) \in T$ and $q_{s0} \in Q_{s0}$. A symbolic transition system for a system (V, Ω) is a symbolic transition system over V whose set of runs is Ω. Operations of (inverse) projection and intersection easily extend from systems to their symbolic representations (such representation may not exist). Let $\mathcal{B}_A = (\Sigma, Q, Q_0, \delta, F \subseteq Q)$ be an automaton such that Σ is a mapping $V \to D$. The *synchronous product* between $\mathcal{B}_A$ and *Symb* is the automaton $\mathcal{B}_{\mathcal{B}_A \times Symb} = (\emptyset, Q', Q'_0, \delta', F')$, where $Q' = Q_s \times Q$, $Q'_0 = Q_{s0} \times Q_0$, $(a', b') \in \delta'((a, b), \emptyset)$ iff $(a, a') \in T$ and $b' \in \delta(b, a)$, $F' = \{(a, b) \in Q' | b \in F\}$. Each state in the product is a pair of states: one for *Symb* and one for $\mathcal{B}_A$. If we do not take the information from $\mathcal{B}_A$ into account, a run of the product corresponds to a run of *Symb*.

3 Non-probabilistic contracts

In this section, we introduce the concept of contract for non stochastic systems. We also study compositional reasoning for such contracts. We will present the theory in the most general case by assuming that contracts and systems are given by (pair of) possibly infinite sets of runs [5]. In practice, a finite representation of such sets is required and there are many ways to instantiate our theory depending on this representation. At the end of the section, we will give an example of such a representation. More precisely, we will follow a successful trend in Model Checking and use automata as a finite representation for systems and contracts. We will also derive effective algorithms based on this symbolic representation.

3.1 Contracts

We first recap the concept of *contract* [4], a mathematical representation that allows to distinguish between assumptions made on the environment and properties of the system.

Definition 1 (Contract) A contract over V is a tuple $C = (V, A, G)$, where V is the set of variables of C, system $A = (V, \Omega_A)$ is the *assumption* and system $G = (V, \Omega_G)$ is the *guarantee*.

The Contract C is said to be in *canonical form* if and only if $\neg A \subseteq G$. As we shall see in Sect. 3.2, the canonical form is needed to have uniform notions of composition and conjunction between contracts. As in [4], this assumption is safe in the sense that we do not lose expressiveness. Indeed, any contract $C = (V, A, G)$ can be turned into an equivalent contract C' in canonical form: $C' = (V, A, G \cup \neg A)$.

We now turn to the problem of deciding whether a system satisfies a contract. A system that satisfies a contract is an *implementation* of the contract. There are two types of implementation relations, depending on the property captured by a contract. A first possible interpretation is when the contract represents properties that are defined on runs of the system. This includes safety properties. In this context, a system satisfies a contract if and only if all system runs that satisfy the assumption are included in the guarantee. This applies to reliability properties, and a system implementing a contract in this way is said to *R-satisfy* the contract. Another possible interpretation is when the contract represents properties that are defined on finite prefixes of the runs of the system and when one wants to evaluate how often the system satisfies the contract. We will say that a system *A-satisfies* a contract with level m $(0 \leq m \leq 1)$ if and only if for each of its runs, the proportion of prefixes of system runs that are either in the guarantee or in the complement of the assumption is greater or equal to m. This concept can be used to check *average safeness* or *reliability*, i.e., to decide for each run whether the average number of positions of the run that do satisfy a local condition is greater or equal to a given threshold.

Definition 2 (R-Satisfaction) System $S = (U, \Omega)$ R-satisfies contract $C = (V, A, G)$ up to time $t \in \mathbb{N}_\infty$, denoted $S \models^{R(t)} C$, if and only if $S|^{\leq t} \cap A \subseteq G$.

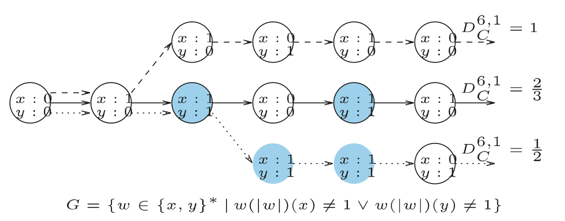
Discussion. In this paper, we assume that runs that do not satisfy the assumptions are "good" runs, i.e., they do not need to satisfy the guarantee. In our theory, assumptions are thus used to distinguish runs that must satisfy the property from those that are not forced to satisfy the property. There are other interpretations of the paradigm of assume/guarantee in which the runs that do not satisfy the assumptions are considered to be bad. We (and our industrial partners) believe that our definition is a more natural interpretation as there is no reason to eliminate runs on which no assumption is made. Another advantage of this approach, which will be made more explicit in Sect. 4, is that this interpretation allows to define a conjunction operation in the stochastic case.

The definition of A-satisfiability is more involved and requires additional notations. The objective is to compute an invariant measure of the amount of time during which the system satisfies a contract. This relation can be combined with *discounting*,[3] which allows to give more weight to faults that arise in the early future. Let $w \in [V]^\infty$ be a (finite or infinite) run and $C = (V, A, G)$ be a contract. We define the function $\varphi_w^C : \mathbb{N}_{|w|} \to \{0, 1\}$ such that $\varphi_w^C(n) = 1 \iff w_{[0,n]} \in G \cup \neg A$. If we fix an horizon in time $t \in \mathbb{N}_\infty$ and a *discount factor* $d \leq 1$, define $D_C^{t,d}(w) = \frac{1}{t} \sum_{i=0}^{t} \varphi_w^C(i)$ if $d = 1$ and $D_C^{t,d}(w) = \frac{1-d}{1-d^{t+1}} \sum_{i=0}^{t} d^i \varphi_w^C(i)$ if $d < 1$. $D_C^{t,d}(w)$ is the mean-availability until position t along the execution corresponding to w with discount factor d. The concept is illustrated in Fig. 1. A-Satisfaction can now be defined.

Definition 3 (A-Satisfaction) A system $S = (U, \Omega)$ A-satisfies at level m contract $C = (V, A, G)$ until position k with discount factor d, denoted $S \models_{d,m}^{A(k)} C$, iff:

$$\min_{w \in (S\uparrow^{U \cup V})|^k} D_{C\uparrow^{U \cup V}}^{k,d}(w) \geq m \quad \text{if } k < \omega$$

$$\inf_{w \in (S\uparrow^{U \cup V})|^k} \liminf_{t \to k} D_{C\uparrow^{U \cup V}}^{t,d}(w) \geq m \quad \text{if } k = \omega.$$



Mean-availability until position 6 is computed for the runs of the system w.r.t a contract with assumption $\{x, y\}^*$ and guarantee the set of finite runs over $\{x, y\}$ such that in the final state $x \neq 1$ or $y \neq 1$. Positions where the contract is satisfied are white.

Fig. 1 Illustration of mean-availability

[3] Discounting is a concept largely used in many areas such as economy.

It is easy to see that the limit in Definition 3 converges, since $D_C^{t,d} \geq 0$. In Sect. 3.4 we will propose techniques to check satisfiability for contracts that are represented with symbolic structures.

In the rest of the section, we propose definitions for composition, conjunction, and refinement. We also study compositional verification with respect to these definitions and the satisfaction relations we considered above.

3.2 Compositional reasoning

In this section, we first define operations between and on contracts and then propose a compositional reasoning framework for contracts. We start with the definition for *composition* and *conjunction*. Composition between contracts mimics classical composition between systems at the abstraction level. It 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.

Definition 4 Let $C_i = (V_i, A_i, G_i)$ with $i = 1, 2$ be two contracts in canonical form. We define

- The *parallel composition* between C_1 and C_2, denoted $C_1 \parallel C_2$, to be the contract $(V_1 \cup V_2, A_1 \cap A_2 \cup \neg(G_1 \cap G_2), G_1 \cap G_2)$.
- The *conjunction* between C_1 and C_2, denoted $C_1 \wedge C_2$, to be the contract $(V_1 \cup V_2, A_1 \cup A_2, G_1 \cap G_2)$.

It is easy to see that both conjunction and composition preserve canonicity.

Discussion. As pointed out in [4], the canonical form is needed to have uniform notions of composition and conjunction between contracts. Indeed, consider two contracts $C_1 = (V, \emptyset, [V]^\infty)$ and $C_2 = (V, \emptyset, \emptyset)$. Observe that C_1 is in canonical form and C_2 is not. Assume also that any system can satisfy both C_1 and C_2. The composition between C_1 and C_2 is the contract $(V, [V]^\infty, \emptyset)$. This contract can only be satisfied by the empty system. Consider now the contract $C_2' = (V, \emptyset, [V]^\infty)$, which is the canonical form for C_2. It is easy to see that the composition between C_1 and C_2' is satisfied by any system. Non-canonical contracts can also be composed. Indeed, the composition of two non-canonical contracts $C_1 = (V_1, A_1, G_1)$ and $C_2 = (V_2, A_2, G_2)$ is given by the following formula $C_1 \parallel_{nc} C_2 = (V_1 \cup V_2, (A_1 \cup \neg G_1) \cap (A_2 \cup \neg G_2), G_1 \cap G_2)$. Observe that this composition requires one more complementation operation, which may be computationally intensive depending on the data-structure used to represent A and G (see Sect. 3.4).

We now turn to the definition of *refinement*, which leads to a preorder relation on contracts.

Definition 5 We say that C_1 *refines* C_2 up to time $t \in \mathbb{N}_\infty$, denoted $C_1 \preceq^{(\leq t)} C_2$, if it guarantees more and assumes less, for all runs of length not greater than t: $A_1 \uparrow^{V_1 \cup V_2} \supseteq (A_2 \uparrow^{V_1 \cup V_2})|^{\leq t}$ and $(G_1 \uparrow^{V_1 \cup V_2})|^{\leq t} \subseteq G_2 \uparrow^{V_1 \cup V_2}$.

Property 1 By a simple inspection of Definitions 4 and 5, one observes that both conjunction and composition are associative, i.e., $C_1 \parallel (C_2 \parallel C_3) = (C_1 \parallel C_2) \parallel C_3$ and $C_1 \wedge (C_2 \wedge C_3) = (C_1 \wedge C_2) \wedge C_3$ (incremental design). Consider $C_2 \parallel C_3$ (respectively, $C_2 \wedge C_3$). We also observe that if $C_1 \preceq^{(\leq t)} C_2$, then $(C_1 \parallel C_3) \preceq^{(\leq t)} (C_2 \parallel C_3)$ (respectively, $(C_1 \wedge C_3) \preceq^{(\leq t)} (C_2 \wedge C_3)$) (independent implementability).

It is interesting to see that the conjunction of two contracts coincide with their *greatest lower bound* with respect to refinement preorder. Thus the following theorem.

Theorem 1 *Consider two contracts C_1 and C_2, we have that*

- $C_1 \wedge C_2 \preceq^{(\leq t)} C_1$ *and* $C_1 \wedge C_2 \preceq^{(\leq t)} C_2$, *and*
- *for each C such that $C \preceq^{\leq t} C_1$ and $C \preceq^{\leq t} C_2$, we have $C \preceq^{\leq t} (C_1 \wedge C_2)$.*

3.3 Compositional verification

In this paper, *compositional verification* refers to a series of results that allow to deduce correctness of a global system by observing its atomic components only. We start with the following theorem for reliability.

Theorem 2 [4] *Consider S_1, S_2 two systems and C_1, C_2 two contracts in canonical form. The following propositions hold for all $t \in \mathbb{N}_\infty$:*

- $S_1 \models^{R(t)} C_1$ *and* $S_2 \models^{R(t)} C_2$ *implies that* $(S_1 \cap S_2) \models^{R(t)} (C_1 \parallel C_2)$;
- $S_1 \models^{R(t)} C_1$ *and* $S_1 \models^{R(t)} C_2$ *iff* $S_1 \models^{R(t)} (C_1 \wedge C_2)$;
- $S_1 \models^{R(t)} C_1$ *and* $C_1 \preceq^{(\leq t)} C_2$ *implies that* $S_1 \models^{R(t)} C_2$.

The above theorem can thus be used to deduce satisfaction w.r.t. to conjunction or disjunction without computing the result of these operations explicitly. The double implication in the second item of the theorem is valid as conjunction is not defined at the level of systems. The theorem can also be used to decide satisfaction on a refined contract without performing any computation. By combining the definitions of composition, conjunction, and refinement, and Theorem 2, we get the following corollary.

Corollary 1 *Let S be a system and C_1, C_2, C_3 be three contracts in canonical form. We have the following results.*

- $S \models^{R(t)} C_1 \parallel (C_2 \parallel C_3)$ *iff* $S \models^{R(t)} (C_1 \parallel C_2) \parallel C_3$;
- $S \models^{R(t)} C_1 \wedge (C_2 \wedge C_3)$ *iff* $S \models^{R(t)} (C_1 \wedge C_2) \wedge C_3$;
- *If* $C_1 \preceq^{(\leq t)} C_2$ *and* $S \models^{R(t)} (C_1 \parallel C_3)$ (*respectively,* $S \models^{R(t)} (C_1 \wedge C_3)$), *then* $S \models^{R(t)} (C_2 \parallel C_3)$ (*respectively,* $S \models^{R(t)} (C_2 \wedge C_3)$).

We now switch to the case of availability. We propose the following theorem that, for example, gives a lower bound on availability for conjunction and disjunction without computing them explicitly.

Theorem 3 *Consider S_1 and S_2 two systems and C_1, C_2 two contracts in canonical form. Let $d \leq 1$ be a discount factor. The following propositions hold for all $t \in \mathbb{N}_\infty$:*

- $S_1 \models^{A(t)}_{d,m_1} C_1$ *and* $S_2 \models^{A(t)}_{d,m_2} C_2$ *implies that* $(S_1 \cap S_2) \models^{A(t)}_{d,m_1+m_2-1} (C_1 \parallel C_2)$;

- $S_1 \models^{A(t)}_{d,m_1} C_1$ *and* $S_1 \models^{A(t)}_{d,m_2} C_2$ *implies that* $S_1 \models^{A(t)}_{d,m_1+m_2-1} (C_1 \wedge C_2)$;
- $S_1 \models^{A(t)}_{d,m} C_1$ *and* $C_1 \preceq^{(\leq t)} C_2$ *implies that* $S_1 \models^{A(t)}_{d,m} C_2$.

The above theorem is an extension of Theorem 2 to the case of availability. It is interesting that the double implication in item two of Theorem 2 does not remain valid in this extension. This is because of the definition of availability. Observe that the last item of Theorems 2 and 3 also holds if C_1 and C_2 are not in canonical form. Observe also that Theorem 3 calls for a direct extension of Corollary 1 to the case of availability. Before we give the proof for Theorem 3 and discuss the extension, we first recap the following classical algebraic properties.

Property 2 Consider $V \subseteq V' \subseteq V''$ three sets of variables and E and E'' two sets of runs over V and V'' respectively. We have:

$$(E \uparrow^{V'}) \uparrow^{V''} = E \uparrow^{V''}; \tag{2.1}$$

$$(E \uparrow^{V''}) \downarrow_{V'} = E \uparrow^{V'}; \tag{2.2}$$

$$(E'' \downarrow_{V'}) \downarrow_{V} = E \downarrow_{V}; \tag{2.3}$$

$$w \in E'' \Rightarrow w \downarrow_{V} \in E'' \downarrow_{V}; \tag{2.4}$$

$$w \in E \Rightarrow w \uparrow^{V'} \subseteq E \uparrow^{V'} . \tag{2.5}$$

We are now ready to give the proof of Theorem 3.

Proof of Theorem 3 For the sake of simplicity, we will consider that $k = \omega$. The proofs for $k < \omega$ are simpler versions of those presented here. We consider the three items of the theorem.

1. Let $S = (U, \Omega) = S_1 \cap S_2$ and $C = (V, A, G) = C_1 \parallel C_2$. Since C_1 and C_2 are contracts in canonical form, we have $G_1 = G_1 \cup \neg A_1$ and $G_2 = G_2 \cup \neg A_2$. Similarly, since composition preserves canonicity, we have $G = G \cup \neg A$.

   Consider $w \in ((S_1 \uparrow^{U_1 \cup U_2} \cap S_2 \uparrow^{U_1 \cup U_2}) \uparrow^{U \cup V})|^k$. Let $w_1 = w \downarrow_{U_1 \cup V_1}$ and $w_2 = w \downarrow_{U_2 \cup V_2}$. By (2.4), we have $w_1 \in (((S_1 \uparrow^{U_1 \cup U_2}) \uparrow^{U \cup V}))|^k \downarrow_{U_1 \cup V_1}$. By (2.1) and (2.2), this implies that $w_1 \in (S_1 \uparrow^{U_1 \cup V_1})|^k$. Similarly, we also have $w_2 \in (S_2 \uparrow^{U_2 \cup V_2})|^k$.

   Consider $t \leq k$ and $i \leq t$. By definition, if $\varphi_w^{C \uparrow^{U \cup V}}(i) = 0$, then $w_{[0,i]} \notin G \uparrow^{U \cup V}$. By (2.5), we deduce $[(w_{1[0,i]} \notin G_1 \uparrow^{U_1 \cup V_1}) \vee (w_{2[0,i]} \notin G_2 \uparrow^{U_2 \cup V_2})]$. As a consequence,

$$\begin{aligned}
&\varphi_w^{C \uparrow^{U \cup V}}(i) \geq \varphi_{w_1}^{C_1 \uparrow^{U_1 \cup V_1}}(i) + \varphi_{w_2}^{C_2 \uparrow^{U_2 \cup V_2}}(i) - 1 \\
&\Rightarrow \quad \forall t \leq k, \quad D^{(t,d)}_{C \uparrow^{U \cup V}}(w) \geq D^{(t,d)}_{C_1 \uparrow^{U_1 \cup V_1}}(w_1) \\
&\qquad\qquad\qquad + D^{(t,d)}_{C_2 \uparrow^{U_2 \cup V_2}}(w_2) - 1 \\
&\Rightarrow \quad \liminf_{t \to k} D^{(t,d)}_{C \uparrow^{U \cup V}}(w) \geq \liminf_{t \to k} D^{(t,d)}_{C_1 \uparrow^{U_1 \cup V_1}}(w_1) \\
&\qquad\qquad\qquad + \liminf_{t \to k} D^{(t,d)}_{C_2 \uparrow^{U_2 \cup V_2}}(w_2) \\
&\qquad\qquad\qquad - 1.
\end{aligned}$$

By hypothesis, we have

$$\begin{cases} \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U_1\cup V_1}}(w_1) \geq m_1 \\ \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U_2\cup V_2}}(w_2) \geq m_2. \end{cases}$$

As a consequence,

$$\liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(w) \geq m_1 + m_2 - 1.$$

Finally,

$$\begin{aligned} &\forall w \in (S \uparrow^{U\cup V})|^k, \quad \liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(w) \geq m_1 + m_2 \\ &\qquad\qquad\qquad\qquad\qquad\qquad - 1 \\ \Rightarrow \quad &\inf_{w\in(S\uparrow^{U\cup V})|^k} \liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(w) \geq m_1 + m_2 \\ &\qquad\qquad\qquad\qquad\qquad\qquad - 1. \end{aligned}$$

2. Let $C = (V, A, G) = C_1 \wedge C_2$. Since C_1 and C_2 are contracts in canonical form, we have $G_1 = G_1 \cup \neg A_1$ and $G_2 = G_2 \cup \neg A_2$. Similarly, since conjunction preserves canonicity, we have $G = G \cup \neg A$.

   Consider $w \in (S_1 \uparrow^{U_1\cup V})|^k$. Let $w_1 = w \downarrow_{U_1\cup V_1}$ and $w_2 = w \downarrow_{U_1\cup V_2}$. By (2.4), we have $w_1 \in ((S_1 \uparrow^{U_1\cup V}))|^k \downarrow_{U_1\cup V_1}$. By (2.2), this implies that $w_1 \in (S_1 \uparrow^{U_1\cup V_1})|^k$. Similarly, we also have $w_2 \in (S_1 \uparrow^{U_1\cup V_2})|^k$.

   Consider $t \leq k$ and $i \leq t$. By definition, if $\varphi_w^{C\uparrow^{U_1\cup V}}(i) = 0$, then $w_{[0,i]} \notin G \uparrow^{U_1\cup V}$. By (2.5), we deduce $[(w_{1[0,i]} \notin G_1 \uparrow^{U_1\cup V_1}) \vee (w_{2[0,i]} \notin G_2 \uparrow^{U_1\cup V_2})]$. As a consequence,

$$\begin{aligned} &\varphi_w^{C\uparrow^{U_1\cup V}}(i) \geq \varphi_{w_1}^{C_1\uparrow^{U_1\cup V_1}}(i) + \varphi_{w_2}^{C_2\uparrow^{U_1\cup V_2}}(i) - 1 \\ \Rightarrow \quad &\forall t \leq k, \quad D^{(t,d)}_{C\uparrow^{U_1\cup V}}(w) \geq D^{(t,d)}_{C_1\uparrow^{U_1\cup V_1}}(w_1) \\ &\qquad\qquad\qquad + D^{(t,d)}_{C_2\uparrow^{U_1\cup V_2}}(w_2) - 1 \\ \Rightarrow \quad &\liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U_1\cup V}}(w) \geq \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U_1\cup V_1}}(w_1) \\ &\qquad\qquad + \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U_1\cup V_2}}(w_2) \\ &\qquad\qquad - 1. \end{aligned}$$

   By hypothesis, we have

$$\begin{cases} \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U_1\cup V_1}}(w_1) \geq m_1 \\ \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U_1\cup V_2}}(w_2) \geq m_2. \end{cases}$$

   As a consequence,

$$\liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U_1\cup V}}(w) \geq m_1 + m_2 - 1.$$

   Finally,

$$\forall w \in (S_1 \uparrow^{U_1 \cup V})|^k, \quad \liminf_{t \to k} D^{(t,d)}_{C \uparrow^{U_1 \cup V}}(w) \geq m_1 + m_2 - 1$$

$$\Rightarrow \quad \inf_{w \in (S_1 \uparrow^{U_1 \cup V})|^k} \liminf_{t \to k} D^{(t,d)}_{C \uparrow^{U_1 \cup V}}(w) \geq m_1 + m_2 - 1.$$

3. Consider $w \in (S_1 \uparrow^{U_1 \cup V_2})|^k$. Let $w' \in w \uparrow^{U_1 \cup V_1 \cup V_2}$ and $w_1 = w' \downarrow_{U_1 \cup V_1}$. By (2.1) and (2.2), we have $w_1 \in (S_1 \uparrow^{U_1 \cup V_1})|^k$.

   Consider now $t \leq k$ and $i \leq t$. By definition, $\varphi^{C_1 \uparrow^{U_1 \cup V_1}}_{w_1}(i) = 1 \iff w_{1[0,i]} \in (G_1 \cup \neg A_1) \uparrow^{U_1 \cup V_1}$. By hypothesis, $((G_1 \cup \neg A_1) \uparrow^{V_1 \cup V_2})|^{\leq k} \subseteq ((G_2 \cup \neg A_2) \uparrow^{V_1 \cup V_2})|^{\leq k}$. Thus, by (2.5), $((G_1 \cup \neg A_1) \uparrow^{U_1 \cup V_1 \cup V_2})|^{\leq k} \subseteq ((G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_1 \cup V_2})|^{\leq k}$.
   If $\varphi^{C_1 \uparrow^{U_1 \cup V_1}}_{w_1}(i) = 1$, then

$$\begin{aligned}
& w_{1[0,i]} \in ((G_1 \cup \neg A_1) \uparrow^{U_1 \cup V_1})|^{\leq k} \\
\Rightarrow \quad & w_1[0,i] \uparrow^{U_1 \cup V_1 \cup V_2} \subseteq ((G_1 \cup \neg A_1) \uparrow^{U_1 \cup V_1 \cup V_2})|^{\leq k} \\
\Rightarrow \quad & w_1[0,i] \uparrow^{U_1 \cup V_1 \cup V_2} \subseteq ((G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_1 \cup V_2})|^{\leq k} \\
\Rightarrow \quad & w'_{[0,i]} \in (G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_1 \cup V_2} \\
\Rightarrow \quad & w'_{[0,i]} \downarrow_{U_1 \cup V_2} \in (G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_1 \cup V_2} \downarrow_{U_1 \cup V_2} \quad \text{by (2.4)} \\
\Rightarrow \quad & w_{[0,i]} \in (G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_2} \quad \text{by (2.2)} \\
\Rightarrow \quad & \varphi^{C_2 \uparrow^{U_1 \cup V_2}}_{w}(i) = 1.
\end{aligned}$$

   Thus,

$$\begin{aligned}
& \forall t \leq k,\ \forall i \leq t, \quad \varphi^{C_2 \uparrow^{U_1 \cup V_2}}_{w}(i) \geq \varphi^{C_1 \uparrow^{U_1 \cup V_1}}_{w_1}(i) \\
\Rightarrow \quad & \forall t \leq k, \quad D^{t,d}_{C_2 \uparrow^{U_1 \cup V_2}}(w) \geq D^{t,d}_{C_1 \uparrow^{U_1 \cup V_1}}(w_1) \\
\Rightarrow \quad & \liminf_{t \to k} D^{t,d}_{C_2 \uparrow^{U_1 \cup V_2}}(w) \geq \liminf_{t \to k} D^{t,d}_{C_1 \uparrow^{U_1 \cup V_1}}(w_1).
\end{aligned}$$

   By hypothesis,

$$\liminf_{t \to k} D^{t,d}_{C_1 \uparrow^{U_1 \cup V_1}}(w_1) \geq m.$$

   As a consequence,

$$\forall w \in (S_1 \uparrow^{U_1 \cup V_2})|^k, \quad \liminf_{t \to k} D^{t,d}_{C_2 \uparrow^{U_1 \cup V_2}}(w) \geq m$$

$$\Rightarrow \quad \inf_{w \in (S_1 \uparrow^{U_1 \cup V_2})|^k} \liminf_{t \to k} D^{t,d}_{C_2 \uparrow^{U_1 \cup V_2}}(w) \geq m.$$

□

Theorem 1 also extends to the case of availability. Hence, we have the following corollary

Corollary 2 *Let S be a system and C_1, C_2, C_3 be three contracts in canonical form. We have the following results.*

- $S \models^{A(k)}_{d,m} C_1 \parallel (C_2 \parallel C_3)$ *iff* $S \models^{A(k)}_{d,m} (C_1 \parallel C_2) \parallel C_3$;

- *$S \models^{A(k)}_{d,m} C_1 \wedge (C_2 \wedge C_3)$ iff $S \models^{A(k)}_{d,m} (C_1 \wedge C_2) \wedge C_3$;*
- *If $C_1 \preceq^{(\leq t)} C_2$ and $S \models^{A(k)}_{d,m} C_1 \parallel C_3$ (respectively, $S \models^{A(k)}_{d,m} C_1 \wedge C_3$), then $S \models^{A(k)}_{d,m} (C_2 \parallel C_3)$ (respectively, $S \models^{A(k)}_{d,m} (C_2 \wedge C_3)$).*

3.4 Effective algorithms/representations

We propose *symbolic* and *effective* automata-based representations for contracts and systems. Those representations are needed to handle possibly infinite sets of runs with a finite memory. We will be working with variables defined over a *finite* domain D. According to our theory, a symbolic representation is effective for an assumption (resp. a guarantee) if inclusion is decidable and the representation is closed under complementation (needed for refinement), union, and intersection. A representation is effective for a system (that is not an assumption or a guarantee) if it is closed under intersection and (inverse) projection, and reliability/availability are decidable.

We assume that systems that are not assumptions or guarantees are represented with *symbolic transition systems* (see Sect. 2 for properties) and that assumptions and guarantees are represented with either finite-word or Büchi automata. Let $C = (V, A, G)$ be a contract, a *symbolic contract* for C is thus a tuple $(V, \mathcal{B}_A, \mathcal{B}_G)$, where $\mathcal{B}_A$ and $\mathcal{B}_G$ are automata with $L(\mathcal{B}_A) = A$ and $L(\mathcal{B}_G) = G$. Observe that there are systems and contracts for which there exists no symbolic representation.

Since both finite-word and Büchi automata are closed under complementation, union and intersection, it is easy to see that the composition and the conjunction of two symbolic contracts is still a symbolic contract. Moreover, since inclusion is decidable for those automata, we can always check whether refinement holds. We now focus on the satisfaction relations. We distinguish between R-Satisfiability and A-Satisfiability. We consider a symbolic contract $C = (V, \mathcal{B}_A, \mathcal{B}_G)$ and a symbolic transition system $Symb = (V, Q_s, T, Q_{s0})$.

- **Reliability**. When considering R-satisfaction, we will assume that $\mathcal{B}_A$ and $\mathcal{B}_G$ are Büchi automata. It is conceptually easy to decide whether *Symb* R-satisfies C. Indeed, following results obtained for temporal logics [53, 54], implemented in the *SPIN* toolset [49], this amounts to check whether the Büchi automaton obtained by taking the synchronous product between *Symb* and $\neg(\mathcal{B}_G \cup \neg\mathcal{B}_A)$ is empty. Observe that assumptions and guarantees can also be represented by logical formalisms that have a translation to Büchi automata—this includes *LTL* [43] and *ETL* [55]. The theory generalizes to other classes of infinite word automata closed under negation and union and other logical formalisms such as *CTL* [17] or *PSL* [27].
- **Availability with level *m* and discount factor *d***. In [23], de Alfaro et al. proposed *DCTL*, a quantitative version of the CTL logic [17]. DCTL has the same syntax as CTL, but its semantics differs: in DCTL, formulas and atomic propositions take values between 0 and 1 rather than in $\{0, 1\}$. Let φ_1 and φ_2 be two DCTL formulas, the value of $\varphi_1 \wedge \varphi_2$ (resp. $\varphi_1 \vee \varphi_2$) is the minimum (resp. maximum) between the values of φ_1 and φ_2. The value of $\forall\varphi_1$ (resp. $\exists\varphi_1$) is the minimum (resp. maximum) valuation of φ_1 over all the runs. In addition to its quantitative aspect, DCTL also allows to discount on the value of the formula as well as to compute its average ($\triangle_d$ operator, where d is the discount: see the semantics with $d = 1$ and $d < 1$ page 6 of [23]) on a possibly infinite run. We assume that $\mathcal{B}_A$ and $\mathcal{B}_G$ are *complete* finite-word automata and show how to reduce A-satisfaction to the evaluation of a DCTL property. Our first step is to compute *Symb*$'$, the synchronous product between *Symb* and $\mathcal{B}_G \cup \neg\mathcal{B}_A$. The resulting automaton can also be viewed as a symbolic transition system whose states are labeled with a proposition p which is true

if the state is accepting and false otherwise. In fact, finite sequences of states of $Symb'$ whose last state is accepting are prefixes of runs of $Symb$ that satisfy $\mathcal{B}_G \cup \neg\mathcal{B}_A$. Hence, checking whether $Symb$ A-satisfies C boils down to compute the minimal average to see $p = 1$ in $Symb'$. Our problem thus reduces to the one of checking for each initial state of $Symb'$ whether the value of the DCTL property $\forall\triangle_d\ p$ is greater or equal to m.

4 Probabilistic contracts

We now extend the results of the previous section to systems that mix stochastic and non-deterministic aspects. Stochastic information is needed to model systems with failures. As for the previous section, all our results will be developed assuming that contracts and systems are represented by sets of runs and then an automata-based representation will be proposed.

Consider a system whose set of variables is U. Our way to mix stochastic and non-deterministic information consists in assuming that, at any moment of time, the value of a set of variables P are chosen with respect to a given probability distribution. The value of the variables in $U \setminus P$ are chosen in a non-deterministic manner. From the point of view of compositional reasoning, it matters whether variables in P are local to a given system or global and shared by all the systems. Indeed, without going to the details, dealing with local probabilistic variables would require to handle conditional probabilities in composition and conjunction operations. To simplify the problem, we assume that variables in P are global and shared by all the systems involved in the design. Remark that one can already model a lot with global variables. Classically, the idea is to view some of the variables as "don't care" in the systems in where they do not matter. Without loss of generality, we also assume that for a given system, the value of the non-deterministic variables remain the same for the initial position of all the runs. This allows to select the initial value of the variables of the run by using the probability distribution only.

We will assume that systems are receptive on P. Due to this property, one can see that runs of a system on a set of variables U with $P \subseteq U$ are runs on P in where each position is augmented with an assignment for the variables in $U \setminus P$. In addition, we suppose that, in a given position, the probability to select the next values of the variables in P is independent from the non-deterministic choice. This is done by assuming the existence of a unique probability distribution $\mathbb{P}$ over $[P]^\omega$ and extending it to $[P]^*$ as follows: $\forall w \in [P]^*,\ \mathbb{P}(w) = \int_{\{w' \in P^\omega \mid w < w'\}} \mathbb{P}(w')dw'$, where $<$ is the prefix order on runs.

Remark 1 Our model of computation is clearly not as powerful as Markov Decision Processes (MDPs). Indeed, in an MDP, at any given moment of time, the choice of the values of variables in $U \setminus P$ may influence the distribution on the next values of variables in P. As we assume a unique global distribution on the set of runs, the choice of the values of the variables in $U \setminus P$ does not influence the probability distribution that is fixed in advance and only depend on the probabilistic choices.

Before defining relations between systems and contracts, it is first necessary to define a probability measure on the set of runs of the system. By hypothesis, this measure has been defined on the set of runs over P and we have to lift it to runs on U. As the system is receptive on P, one could think that the measure directly extends to the runs of the system. This is actually not true. Indeed, one can associate several different values of the non-deterministic variables to a given run of the stochastic variables. This problem can be solved with the help of a scheduler that, in a given moment of time, associates a unique value to each non-deterministic variable with a given value of the probabilistic variables. In practice, systems are not defined as sets of runs but rather as symbolic objects, e.g., Markov Decision processes, that generate runs from a set of initial states. In such context, the resolution of the non-determinism is incremental. The process starts from an initial value of the probabilistic variables to which is associated a unique value of the non-deterministic variables. Then, at any moment of time and for any run, the scheduler associates a unique non-deterministic choice to a given value of the probabilistic variables. As the system is receptive on P, a scheduler basically associates to any position of any run on P a value for the non-deterministic variables in order to retrieve a run of the system. This is sufficient to define a probability measure on subsets of runs of the system. The assignments can either depend (1) on the last position of the run, in which case the scheduler is said to be memoryless, or (2) on a prefix of the run, in which case the scheduler is said to be history-dependent.

We now propose a general definition of the "effect of a scheduler", i.e., computing a subsets of runs of S receptive on P and on which a probability measure can be defined. Characterizing the effect of the scheduler is enough to reason on compositional design. This is different from the application of the scheduler itself, i.e., the choice made at a given position. Consider a system $S = (U, \Omega)$. From a definition point of view, since the system is receptive on P, the effect of a scheduler f can be characterized by a mapping from every finite (or infinite) run w on probabilistic variables P to a run $f(w)$ of S which coincides with w for every probabilistic variable. This can be formalized with the following definition.

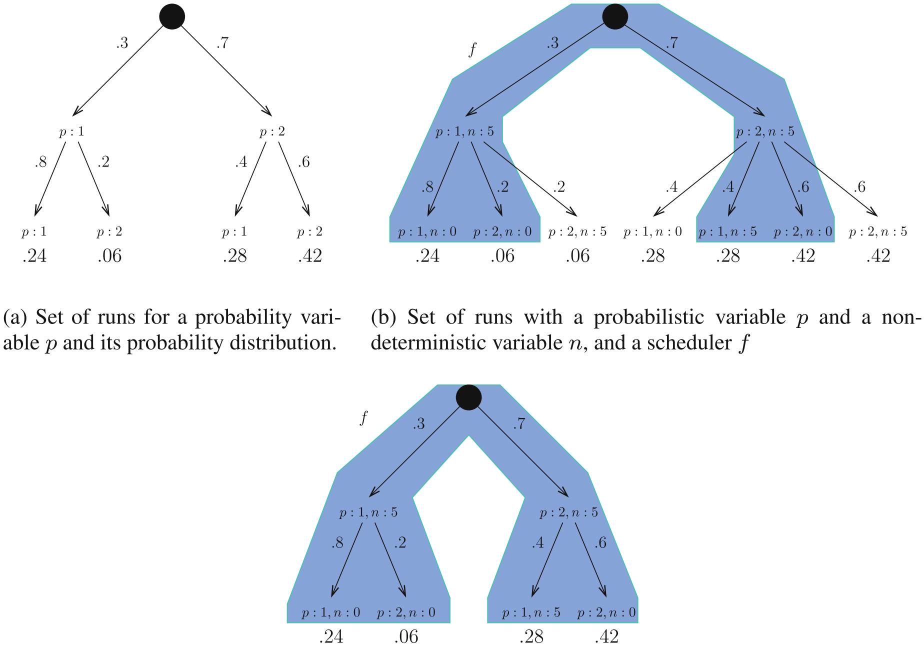
Definition 6 (Scheduler) A scheduler f of system $S = (U, \Omega)$, with $P \subseteq U$, is a monotonous mapping $[P]^* \to \Omega$ such that for all $w \in [P]^*$, $f(w) \downarrow_P = w$. The set of schedulers corresponding to a system S is denoted by $\mathsf{Sched}(S)$.

For simplicity of the presentation, we use the term scheduler to refer either to the resolution of the non-determinism in a given position (which will be needed in Sect. 4.3) of the run or to the effect of applying the scheduler to generate a subset of runs of the system whose probability measure is defined. Let f be a scheduler defined on a finite set of runs of length k. To be coherent with classical definitions of schedulers that resolve non-determinism starting from the initial set of states, we have to suppose that f is causal. More precisely, given a run of length $k + 1$, this means that f cannot change the non-deterministic assignments to the prefix of length k of the run. Formally, $\forall w, w' \in [P]^*, w < w' \Rightarrow f(w) < f(w')$. In practice, this is a natural assumption that is only emphasized as it will be used in the proofs.

The above theory is illustrated in Fig. 2. Figure 2a presents the set of runs of a probabilistic variable p that can take two values: 1 and 2. Figure 2b presents the set of runs of a system whose unique probabilistic variable is p. The runs colored in dark are those selected by the schedulers. One can see that the probability measure of these runs is $1 = 0.24 + 0.06 + 0.28 + 0.42$, while the measure on all runs is 1.76. The reason is that probability values are duplicated due to non-determinism. As an example, from the state $p : 1, n : 5$, the probability that $p = 2$ in the next step is 0.2. However, this probability is duplicated because $p : 2$ can either be associated to $n : 0$ or to $n : 5$. The scheduler will choose between those two values. For doing so, it may use the history of the run.

4.1 Probabilistic contracts

We will say that a contract $C = (V, A, G)$ is a *probabilistic contract* iff $P \subseteq V$, i.e. iff its set of variables contains all the probabilistic variables. We now turn to the problem of



(a) Set of runs for a probability variable p and its probability distribution.

(b) Set of runs with a probabilistic variable p and a non-deterministic variable n, and a scheduler f

(c) Measure on the sets of runs after applying the scheduler f

Fig. 2 Illustration of a scheduler defining a probability measure on a set of executions

deciding whether a system $S = (U, \Omega)$ satisfies a probabilistic contract $\mathcal{C} = (V, A, G)$. As it was already the case for non-probabilistic contracts, we will distinguish R-Satisfaction and A-Satisfaction.

In Sect. 3, R-Satisfaction was defined with respect to a Boolean interpretation: either the system R-satisfies a contract or it does not. When moving to the probabilistic setting, we can give a *quantitative* definition for R-Satisfaction that is: *for any scheduler, is the probability to satisfy the contract greater or equal to a certain threshold?*

Definition 7 (P-R-Satisfaction) A system $S = (U, \Omega)$ R-satisfies a probabilistic contract $\mathcal{C} = (V, A, G)$ for runs of length k ($k \in \mathbb{N}^\infty$) with level α, denoted $S \VDash_\alpha^{R(k)} \mathcal{C}$, iff

$$\inf_{f \in \mathsf{Sched}(S\uparrow^{U\cup V})} \mathbb{P}([f([P]^k) \cap (G \cup \neg A)\uparrow^{U\cup V}]\downarrow_P) \geq \alpha.$$

Observe that, as for the non-probabilistic case, we consider that runs that do not satisfy the assumption are good runs. In addition to the motivation given in Sect. 3.1, we will see that using such an interpretation is needed when considering the conjunction operation (see the observation after Theorem 4).

Though A-Satisfaction was already quantitative, we now have to take into account the probabilistic point of view: instead of considering the minimal value of the mean-availability for all runs of the system, we now consider the *minimal expected value* of the mean-availability for all schedulers.

Definition 8 (P-A-Satisfaction) A system $S = (U, \Omega)$ A-satisfies a probabilistic contract $\mathcal{C} = (V, A, G)$ for runs of length k ($k \in \mathbb{N}^\infty$) with level α and discount factor d, denoted $S \models^{A(k)}_{d,\alpha} \mathcal{C}$, iff

$$\inf_{f \in \mathsf{Sched}(S\uparrow^{U\cup V})} \int_{w\in[P]^k} \mathbb{P}(w) \cdot F(w) dw \geq \alpha$$

with

$$F(w) = \begin{cases} D^{k,d}_{\mathcal{C}\uparrow^{U\cup V}}(f(w)) & \text{if } k < \omega \\ \liminf\limits_{t\to k} D^{t,d}_{\mathcal{C}\uparrow^{U\cup V}}(f(w)) & \text{if } k = \omega. \end{cases}$$

4.2 Operations on probabilistic contracts and compositional reasoning

We now leverage the compositional reasoning results of Sect. 3.2 to probabilistic contracts. We consider composition/conjunction and refinement separately.

4.2.1 Composition and conjunction

Composition and conjunction of probabilistic contracts is defined as for non-probabilistic contracts (see Definition 4). We thus propose an extension of Theorems 2 and 3 which takes the probabilistic aspects into account.

Theorem 4 (P-R-Satisfaction) *Consider three systems $S = (U, \Omega)$, $S_1 = (U_1, \Omega_1)$ and $S_2 = (U_2, \Omega_2)$ and two probabilistic contracts $\mathcal{C}_1 = (V_1, A_1, G_1)$ and $\mathcal{C}_2 = (V_2, A_2, G_2)$ that are in canonical form. We have the following results*:

1. *Composition. Assume that S_1 and S_2 are P-compatible. If $S_1 \models^{R(k)}_{\alpha} \mathcal{C}_1$ and $S_2 \models^{R(k)}_{\beta} \mathcal{C}_2$, then $S_1 \cap S_2 \models^{R(k)}_{\gamma} \mathcal{C}_1 \parallel \mathcal{C}_2$ with $\gamma \geq \alpha + \beta - 1$ if $\alpha + \beta \geq 1$ and $\gamma \geq 0$ otherwise.*
2. *Conjunction. Assume that S is P-receptive. If $S \models^{R(k)}_{\alpha} \mathcal{C}_1$ and $S \models^{R(k)}_{\beta} \mathcal{C}_2$, then $S \models^{R(k)}_{\gamma} \mathcal{C}_1 \wedge \mathcal{C}_2$ with $\gamma \geq \alpha + \beta - 1$ if $\alpha + \beta \geq 1$ and $\gamma \geq 0$ otherwise.*

We first state a classical algebraic property, which in fact justify the choice for γ in the theorem, and two lemmas that will be needed in the proof of Theorem 4. We then present the proof. We start with the following property.

Property 3 Let E_1 and E_2 be two sets of runs over P. We have:

$$\begin{aligned} &\mathbb{P}(\neg(E_1 \cap E_2)) \leq \mathbb{P}(\neg E_1) + \mathbb{P}(\neg E_2) \\ &\quad\Rightarrow \quad 1 - \mathbb{P}(E_1 \cap E_2) \leq (1 - \mathbb{P}(E_1)) + (1 - \mathbb{P}(E_2)) \\ &\quad\Rightarrow \quad \mathbb{P}(E_1 \cap E_2) \geq \mathbb{P}(E_1) + \mathbb{P}(E_2) - 1. \end{aligned} \tag{3.1}$$

We now propose the two lemmas.

Lemma 1 *Consider $S = (U, \Omega)$ a P-receptive system, $f \in \mathsf{Sched}(S)$ a scheduler of S and U' a set of variables. If $P \subseteq U' \subseteq U$, then we have*:

$$f\downarrow_{U'} : \left\{ \begin{array}{c} [P]^\infty \to S\downarrow_{U'} \\ w \mapsto f(w)\downarrow_{U'} \end{array} \right\} \in \mathsf{Sched}(S\downarrow_{U'}).$$

Proof Let $f' = f \downarrow_{U'}$. By definition, $f' : [P]^* \to S \downarrow_{U'}$. Consider now $w \in [P]^*$ and $w' < w$. Since $w' < w$, we have $f(w') < f(w)$. As a consequence, $f'(w') < f'(w)$. Moreover, $f(w) \downarrow_P = w$ and $P \subseteq U'$, thus by (2.3), $(f(w) \downarrow_{U'}) \downarrow_P = w$. □

Lemma 2 *Consider $S = (U, \Omega)$ a P-receptive system, $f \in \mathsf{Sched}(S)$ a scheduler of S and U' and U'' two sets of variables. If $P \subseteq U' \subseteq U$, $P \subseteq U'' \subseteq U$ and $U' \cup U'' = U$, then*

$$\forall w \in (P)^\infty, \quad f \downarrow_{U'} (w) \cap f \downarrow_{U''} (w) = \{f(w)\}.$$

Proof Let $w' = f \downarrow_{V'} (w)$ and $w'' = f \downarrow_{V''} (w)$. w, w' and w'' are such that $\forall i \in \mathbb{N}$, $\forall v \in V'$, $f(w)(i)(v) = w'(i)(v)$ and $\forall i \in \mathbb{N}, \forall v \in V''$, $f(w)(i)(v) = w''(i)(v)$. Moreover, because w' and w'' are both projections of $f(w)$, $\forall i \in \mathbb{N}, \forall v \in V' \cap V''$, $f(w)(i)(v) = w'(i)(v) = w''(i)(v)$.

Now, consider $w_0 \in f \downarrow_{V'} (w) \cap f \downarrow_{V''} (w)$. Since $w_0 \in (f \downarrow_{V'} (w)) \uparrow^V$, we have $w_0 \downarrow_{V'} = w'$. Thus $\forall i \in \mathbb{N}, \forall v \in V'$, $w_0(i)(v) = w'(i)(v) = f(w)(i)(v)$.

Similarly, since $w_0 \in (f \downarrow_{V''} (w)) \uparrow^V$, we have $\forall i \in \mathbb{N}, \forall v \in V'$, $w_0(i)(v) = w''(i)(v) = f(w)(i)(v)$.

Finally, $\forall i \in \mathbb{N}, \forall v \in V = V' \cup V''$, $w''(i)(v) = f(w)(i)(v)$, thus $w'' = f(w)$. □

We are now ready to give the proof of Theorem 4

Proof of Theorem 4 We consider the two items of the theorem.

1. Let $S = (U, \Omega) = S_1 \cap S_2$ and $\mathcal{C} = (V, A, G) = \mathcal{C}_1 \parallel \mathcal{C}_2$. Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are in canonical form and since composition preserves canonicity, we will consider that $G_1 = G_1 \cup \neg A_1$, $G_2 = G_2 \cup \neg A_2$ and $G = G \cup \neg A$.

   Consider $f \in \mathsf{Sched}(S \uparrow^{U \cup V})$. Since S_1 and S_2 are P-compatible, f is defined over all runs in $[P]^k$. Moreover, since $S = (S_1 \uparrow^{U_1 \cup U_2}) \cap (S_2 \uparrow^{U_1 \cup U_2})$, we have $(f \in \mathsf{Sched}((S_1 \uparrow^{U_1 \cup U_2}) \uparrow^{U \cup V})) \wedge (f \in \mathsf{Sched}((S_2 \uparrow^{U_1 \cup U_2}) \uparrow^{U \cup V}))$. By (2.1), we obtain

$$(f \in \mathsf{Sched}(S_1 \uparrow^{U \cup V})) \wedge (f \in \mathsf{Sched}(S_2 \uparrow^{U \cup V})).$$

   Let $f_1 = f \downarrow_{U_1 \cup V_1}$ and $f_2 = f \downarrow_{U_2 \cup V_2}$. By Lemma 1, we have

$$\begin{cases} (f_1 \in \mathsf{Sched}((S_1 \uparrow^{U \cup V}) \downarrow_{U_1 \cup V_1})) \\ \wedge \\ (f_2 \in \mathsf{Sched}((S_2 \uparrow^{U \cup V}) \downarrow_{U_2 \cup V_2})). \end{cases}$$

   Thus, by (2.2),

$$(f_1 \in \mathsf{Sched}(S_1 \uparrow^{U_1 \cup V_1}) \wedge (f_2 \in \mathsf{Sched}(S_2 \uparrow^{U_2 \cup V_2})).$$

   Consider now $w \in [P]^k$. If $f_1(w) \in G_1 \uparrow^{U_1 \cup V_1}$, then by (2.5) and (2.1), $f_1(w) \uparrow^{U \cup V} \subseteq G_1 \uparrow^{U \cup V}$. Similarly, if $f_2(w) \in G_2 \uparrow^{U_2 \cup V_2}$, then $f_2(w) \uparrow^{U \cup V} \subseteq G_2 \uparrow^{U \cup V}$. As a consequence, $f_1(w) \uparrow^{U \cup V} \cap f_2(w) \uparrow^{U \cup V} \subseteq (G_1 \cap G_2) \uparrow^{U \cup V}$, and, by Lemma 2, $f(w) \in (G_1 \cap G_2) \uparrow^{U \cup V}$. As a consequence,

$$\overbrace{[f_1([P]^k) \cap G_1 \uparrow^{U_1 \cup V_1}] \downarrow_P}^{E_1} \cap \overbrace{[f_2([P]^k) \cap G_2 \uparrow^{U_2 \cup V_2}] \downarrow_P}^{E_2}$$
$$\subseteq \underbrace{[f([P]^k) \cap G \uparrow^{U \cup V}] \downarrow_P}_{E}.$$

This implies, by (3.1), that $\mathbb{P}(E) \geq \mathbb{P}(E_1) + \mathbb{P}(E_2) - 1$. Moreover, by hypothesis,

$$\begin{cases} \mathbb{P}(E_1) \geq \alpha \\ \mathbb{P}(E_2) \geq \beta. \end{cases}$$

Thus, $\mathbb{P}(E) \geq \alpha + \beta - 1$ and

$$\begin{aligned} &\forall f \in \mathsf{Sched}(S \uparrow^{U \cup V}), \\ &\mathbb{P}([f([P]^k) \cap G \uparrow^{U \cup V}] \downarrow_P) \geq \alpha + \beta - 1 \\ &\quad \Rightarrow \inf_{f \in \mathsf{Sched}(S \uparrow^{U \cup V})} \mathbb{P}([f([P]^k) \cap G \uparrow^{U \cup V}] \downarrow_P) \\ &\qquad \geq \alpha + \beta - 1. \end{aligned}$$

2. We will use $\mathcal{C} = (V, A, G) = \mathcal{C}_1 \wedge \mathcal{C}_2$. Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are in canonical form and since conjunction preserves canonicity, we will consider that $G_1 = G_1 \cup \neg A_1$, $G_2 = G_2 \cup \neg A_2$ and $G = G \cup \neg A$.

   Consider $f \in \mathsf{Sched}(S \uparrow^{U \cup V})$. Since S is P-receptive, f is defined over all runs in $[P]^k$.

   Let $f_1 = f \downarrow_{U \cup V_1}$ and $f_2 = f \downarrow_{U \cup V_2}$. By Lemma 1, we have

$$\begin{cases} \quad (f_1 \in \mathsf{Sched}((S \uparrow^{U \cup V}) \downarrow_{U \cup V_1})) \\ \wedge \\ \quad (f_2 \in \mathsf{Sched}((S \uparrow^{U \cup V}) \downarrow_{U \cup V_2})). \end{cases}$$

   Thus, by (2.2),

$$(f_1 \in \mathsf{Sched}(S \uparrow^{U \cup V_1}) \wedge (f_2 \in \mathsf{Sched}(S \uparrow^{U_2 \cup V_2})).$$

   Consider now $w \in [P]^k$. If $f_1(w) \in G_1 \uparrow^{U \cup V_1}$, then by (2.5) and (2.1), $f_1(w) \uparrow^{U \cup V} \subseteq G_1 \uparrow^{U \cup V}$. Similarly, if $f_2(w) \in G_2 \uparrow^{U \cup V_2}$, then $f_2(w) \uparrow^{U \cup V} \subseteq G_2 \uparrow^{U \cup V}$. As a consequence, $f_1(w) \uparrow^{U \cup V} \cap f_2(w) \uparrow^{U \cup V} \subseteq (G_1 \cap G_2) \uparrow^{U \cup V}$, and, by Lemma 2, $f(w) \in (G_1 \cap G_2) \uparrow^{U \cup V}$. As a consequence,

$$\overbrace{[f_1([P]^k) \cap G_1 \uparrow^{U \cup V_1}] \downarrow_P}^{E_1} \cap \overbrace{[f_2([P]^k) \cap G_2 \uparrow^{U \cup V_2}] \downarrow_P}^{E_2}$$
$$\subseteq \underbrace{[f([P]^k) \cap G \uparrow^{U \cup V}] \downarrow_P}_{E}.$$

   This implies, by (3.1), that $\mathbb{P}(E) \geq \mathbb{P}(E_1) + \mathbb{P}(E_2) - 1$. Moreover, by hypothesis,

$$\begin{cases} \mathbb{P}(E_1) \geq \alpha \\ \mathbb{P}(E_2) \geq \beta. \end{cases}$$

Thus, $\mathbb{P}(E) \geq \alpha + \beta - 1$ and

$$\begin{aligned} &\forall f \in \mathsf{Sched}(S \uparrow^{U \cup V}),\\ &\mathbb{P}([f([P]^k) \cap G \uparrow^{U \cup V}] \downarrow_P) \geq \alpha + \beta - 1\\ &\Rightarrow \inf_{f \in \mathsf{Sched}(S \uparrow^{U \cup V})} \mathbb{P}([f([P]^k) \cap G \uparrow^{U \cup V}] \downarrow_P)\\ &\qquad \geq \alpha + \beta - 1. \end{aligned}$$

□

Remark 2 Consider two contracts (A_1, G_1) and (A_2, G_2) such that $A_1 \subset G_1$, $A_2 \subset G_2$ and $(A_1 \cup A_2) \cap (G_1 \cap G_2) = \emptyset$. It is easy to see that any system will reliably satisfy both contracts with probability 1. According to an interpretation where one considers that runs that do not satisfy assumptions are bad runs, the probability that a system satisfies the conjunction is always 0. With our interpretation, there are situations where this probability is strictly higher than 0: those where there are runs that do not belong to A_1 or A_2.

Let us now consider to the case of P-A-Satisfaction. We propose the following theorem.

Theorem 5 (P-A-Satisfaction) *Consider three systems $S = (U, \Omega)$, $S_1 = (U_1, \Omega_1)$ and $S_2 = (U_2, \Omega_2)$ and two probabilistic contracts $\mathcal{C}_1 = (V_1, A_1, G_1)$ and $\mathcal{C}_2 = (V_2, A_2, G_2)$ that are in canonical form. We have the following results*:

1. *Composition. Assume that S_1 and S_2 are P-compatible. If $S_1 \models^{A(k)}_{d,\alpha} \mathcal{C}_1$ and $S_2 \models^{A(k)}_{d,\beta} \mathcal{C}_2$, then $S_1 \cap S_2 \models^{A(k)}_{d,\gamma} \mathcal{C}_1 \parallel \mathcal{C}_2$ with $\gamma \geq \alpha + \beta - 1$ if $\alpha + \beta \geq 1$ and $\gamma \geq 0$ otherwise.*
2. *Conjunction. Assume that S is P-receptive. If $S \models^{A(k)}_{d,\alpha} \mathcal{C}_1$ and $S \models^{A(k)}_{d,\beta} \mathcal{C}_2$, then $S \models^{A(k)}_{d,\gamma} \mathcal{C}_1 \wedge \mathcal{C}_2$ with $\gamma \geq \alpha + \beta - 1$ if $\alpha + \beta \geq 1$ and $\gamma \geq 0$ otherwise.*

For the sake of simplicity, we will consider that $k = \omega$. The proofs for $k < \omega$ are simpler versions of the ones presented here.

Proof For the sake of simplicity, we will consider that $k = \omega$. The proofs for $k < \omega$ are simpler versions of the ones presented here. We consider the two items of the theorem.

1. Let $S = (U, \Omega) = S_1 \cap S_2$ and $\mathcal{C} = (V, A, G) = \mathcal{C}_1 \parallel \mathcal{C}_2$. Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are in canonical form and since composition preserves canonicity, we will consider that $G_1 = G_1 \cup \neg A_1$, $G_2 = G_2 \cup \neg A_2$ and $G = G \cup \neg A$.

   Consider $f \in \mathsf{Sched}(S \uparrow^{U \cup V})$. Since S_1 and S_2 are P-compatible, f is defined over all runs in $[P]^k$. Moreover, since $S = (S_1 \uparrow^{U_1 \cup U_2}) \cap (S_2 \uparrow^{U_1 \cup U_2})$, it is clear that $(f \in \mathsf{Sched}((S_1 \uparrow^{U_1 \cup U_2}) \uparrow^{U \cup V})) \wedge (f \in \mathsf{Sched}((S_2 \uparrow^{U_1 \cup U_2}) \uparrow^{U \cup V}))$. Thus, by (2.1),

   $$\Rightarrow \quad (f \in \mathsf{Sched}(S_1 \uparrow^{U \cup V})) \wedge (f \in \mathsf{Sched}(S_2 \uparrow^{U \cup V})).$$

   Let $f_1 = f \downarrow_{U_1 \cup V_1}$ and $f_2 = f \downarrow_{U_2 \cup V_2}$. By Lemma 1, we have

   $$\Rightarrow \quad \begin{cases} (f_1 \in \mathsf{Sched}((S_1 \uparrow^{U \cup V}) \downarrow_{U_1 \cup V_1}))\\ \wedge\\ (f_2 \in \mathsf{Sched}((S_2 \uparrow^{U \cup V}) \downarrow_{U_2 \cup V_2})). \end{cases}$$

Thus, by (2.2),

$$(f_1 \in \mathsf{Sched}(S_1 \uparrow^{U_1\cup V_1}) \wedge (f_2 \in \mathsf{Sched}(S_2 \uparrow^{U_2\cup V_2})).$$

Consider $w \in [P]^k$, $t \leq k$ and $i \leq t$. If $\varphi_{f(w)}^{C\uparrow^{U\cup V}}(i) = 0$, then $f(w)_{[0,i]} \notin G\uparrow^{U\cup V}$. By (2.5) and (2.2), we deduce that $[(f_1(w)_{[0,i]} \notin G_1 \uparrow^{U_1\cup V_1}) \vee (f_2(w)_{[0,i]} \notin G_2 \uparrow^{U_2\cup V_2})]$. As a consequence,

$$\begin{aligned}
&\varphi_{f(w)}^{C\uparrow^{U\cup V}}(i) \geq \varphi_{f_1(w)}^{\mathcal{C}_1\uparrow^{U_1\cup V_1}}(i) + \varphi_{f_2(w)}^{\mathcal{C}_2\uparrow^{U_2\cup V_2}}(i) - 1\\
&\quad\Rightarrow\quad \forall t \leq k,\quad D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w)) \geq D^{(t,d)}_{\mathcal{C}_1\uparrow^{U_1\cup V_1}}(f_1(w))\\
&\qquad\qquad\qquad + D^{(t,d)}_{\mathcal{C}_2\uparrow^{U_2\cup V_2}}(f_2(w))\\
&\qquad\qquad\qquad - 1\\
&\quad\Rightarrow\quad \liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w)) \geq \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_1\uparrow^{U_1\cup V_1}}(f_1(w))\\
&\qquad\qquad\qquad + \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_2\uparrow^{U_2\cup V_2}}(f_2(w))\\
&\qquad\qquad\qquad - 1.
\end{aligned}$$

As a consequence, $\forall w \in [P]^k$,

$$\begin{aligned}
&\liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w)) \geq \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_1\uparrow^{U_1\cup V_1}}(f_1(w))\\
&\qquad + \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_2\uparrow^{U_2\cup V_2}}(f_2(w))\\
&\qquad - 1\\
&\quad\Rightarrow\quad \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w))dw\\
&\qquad \geq \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_1\uparrow^{U_1\cup V_1}}(f_1(w))dw\\
&\qquad + \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_2\uparrow^{U_2\cup V_2}}(f_2(w))dw\\
&\qquad - 1.
\end{aligned}$$

By hypothesis, we have

$$\begin{cases}
\int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_1\uparrow^{U_1\cup V_1}}(f_1(w))dw \geq \alpha\\
\int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_2\uparrow^{U_2\cup V_2}}(f_2(w))dw \geq \beta.
\end{cases}$$

Thus, $\forall f \in \mathsf{Sched}(S\uparrow^{U\cup V})$,

$$\int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w))dw \geq \alpha + \beta - 1.$$

2. Let $\mathcal{C} = (V, A, G) = \mathcal{C}_1 \wedge \mathcal{C}_2$. Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are in canonical form and since conjunction preserves canonicity, we will consider that $G_1 = G_1 \cup \neg A_1$, $G_2 = G_2 \cup \neg A_2$ and $G = G \cup \neg A$.

   Consider $f \in \mathsf{Sched}(S \uparrow^{U\cup V})$. Since S is P-receptive, f is defined over all runs in $[P]^k$. Let $f_1 = f \downarrow_{U\cup V_1}$ and $f_2 = f \downarrow_{U\cup V_2}$. By Lemma 1, we have

$$\Rightarrow \quad \begin{cases} (f_1 \in \mathsf{Sched}((S \uparrow^{U\cup V}) \downarrow_{U\cup V_1})) \\ \wedge \\ (f_2 \in \mathsf{Sched}((S \uparrow^{U\cup V}) \downarrow_{U\cup V_2})). \end{cases}$$

   Thus, by (2.2)

$$(f_1 \in \mathsf{Sched}(S \uparrow^{U\cup V_1}) \wedge (f_2 \in \mathsf{Sched}(S \uparrow^{U\cup V_2})).$$

   Consider $w \in [P]^k$, $t \le k$ and $i \le t$. If $\varphi_{f(w)}^{C\uparrow^{U\cup V}}(i) = 0$, then $f(w)_{[0,i]} \notin G \uparrow^{U\cup V}$. By (2.5) and (2.2), we deduce that $[(f_1(w)_{[0,i]} \notin G_1 \uparrow^{U\cup V_1}) \vee (f_2(w)_{[0,i]} \notin G_2 \uparrow^{U\cup V_2})]$. As a consequence,

$$\begin{aligned} &\varphi_{f(w)}^{C\uparrow^{U\cup V}}(i) \ge \varphi_{f_1(w)}^{\mathcal{C}_1\uparrow^{U\cup V_1}}(i) + \varphi_{f_2(w)}^{\mathcal{C}_2\uparrow^{U\cup V_2}}(i) - 1 \\ &\quad\Rightarrow \quad \forall t \le k, \quad D_{C\uparrow^{U\cup V}}^{(t,d)}(f(w)) \ge D_{\mathcal{C}_1\uparrow^{U\cup V_1}}^{(t,d)}(f_1(w)) \\ &\qquad\qquad\qquad\qquad + D_{\mathcal{C}_2\uparrow^{U\cup V_2}}^{(t,d)}(f_2(w)) \\ &\qquad\qquad\qquad\qquad - 1 \\ &\quad\Rightarrow \quad \liminf_{t\to k} D_{C\uparrow^{U\cup V}}^{(t,d)}(f(w)) \ge \liminf_{t\to k} D_{\mathcal{C}_1\uparrow^{U\cup V_1}}^{(t,d)}(f_1(w)) \\ &\qquad\qquad\qquad\qquad + \liminf_{t\to k} D_{\mathcal{C}_2\uparrow^{U\cup V_2}}^{(t,d)}(f_2(w)) \\ &\qquad\qquad\qquad\qquad - 1. \end{aligned}$$

   As a consequence, $\forall w \in [P]^k$,

$$\begin{aligned} &\liminf_{t\to k} D_{C\uparrow^{U\cup V}}^{(t,d)}(f(w)) \ge \liminf_{t\to k} D_{\mathcal{C}_1\uparrow^{U\cup V_1}}^{(t,d)}(f_1(w)) \\ &\qquad + \liminf_{t\to k} D_{\mathcal{C}_2\uparrow^{U\cup V_2}}^{(t,d)}(f_2(w)) \\ &\qquad - 1 \\ &\quad\Rightarrow \int_{w\in[P]^k} \mathbb{P}(w) \cdot \liminf_{t\to k} D_{C\uparrow^{U\cup V}}^{(t,d)}(f(w))dw \\ &\qquad \ge \int_{w\in[P]^k} \mathbb{P}(w) \cdot \liminf_{t\to k} D_{\mathcal{C}_1\uparrow^{U\cup V_1}}^{(t,d)}(f_1(w))dw \\ &\qquad + \int_{w\in[P]^k} \mathbb{P}(w) \cdot \liminf_{t\to k} D_{\mathcal{C}_2\uparrow^{U\cup V_2}}^{(t,d)}(f_2(w))dw \\ &\qquad - 1. \end{aligned}$$

By hypothesis, we have

$$\begin{cases} \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_1\uparrow^{U\cup V_1}}(f_1(w))dw \geq \alpha \\ \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}_2\uparrow^{U\cup V_2}}(f_2(w))dw \geq \beta. \end{cases}$$

Thus, $\forall f \in \mathsf{Sched}(S\uparrow^{U\cup V})$,

$$\int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{\mathcal{C}\uparrow^{U\cup V}}(f(w))dw \geq \alpha+\beta-1. \qquad \square$$

We now discuss the incremental design property. In fact, as Property 1 is independent from the systems and because of Theorems 4 and 5, we directly obtain extensions to the availability case for the two first items of Theorems 1 and 2. More precisely, we have the following results.

Theorem 6 *Consider three probabilistic contracts $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ and a system S. Assume that $S \models^{R(k)}_{\alpha_1} \mathcal{C}_1$, $S \models^{R(k)}_{\alpha_2} \mathcal{C}_2$, $S \models^{R(k)}_{\alpha_3} \mathcal{C}_3$. Let $\gamma = \alpha_1+\alpha_2+\alpha_3-2$ if $\alpha_1+\alpha_2+\alpha_3 > 2$ and 0 otherwise. We have*

- $S \models^{R(k)}_{\gamma} \mathcal{C}_1 \parallel (\mathcal{C}_2 \parallel \mathcal{C}_3)$ *iff* $S \models^{R(k)}_{\gamma} (\mathcal{C}_1 \parallel \mathcal{C}_2) \parallel \mathcal{C}_3$.
- $S \models^{R(k)}_{\gamma} \mathcal{C}_1 \wedge (\mathcal{C}_2 \wedge \mathcal{C}_3)$ *iff* $S \models^{R(k)}_{\gamma} (\mathcal{C}_1 \wedge \mathcal{C}_2) \wedge \mathcal{C}_3$.

Theorem 7 *Consider three probabilistic contracts $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ and a system S. Assume that $S \models^{A(k)}_{d,\alpha_1} \mathcal{C}_1$, $S \models^{A(k)}_{d,\alpha_2} \mathcal{C}_2$, $S \models^{A(k)}_{d,\alpha_3} \mathcal{C}_3$. Let $\gamma = \alpha_1+\alpha_2+\alpha_3-2$ if $\alpha_1+\alpha_2+\alpha_3 > 2$ and 0 otherwise. We have*

- $S \models^{A(k)}_{d,\gamma} \mathcal{C}_1 \parallel (\mathcal{C}_2 \parallel \mathcal{C}_3)$ *iff* $S \models^{A(k)}_{d,\gamma} (\mathcal{C}_1 \parallel \mathcal{C}_2) \parallel \mathcal{C}_3$.
- $S \models^{A(k)}_{d,\gamma} \mathcal{C}_1 \wedge (\mathcal{C}_2 \wedge \mathcal{C}_3)$ *iff* $S \models^{A(k)}_{d,\gamma} (\mathcal{C}_1 \wedge \mathcal{C}_2) \wedge \mathcal{C}_3$.

4.2.2 Refinement

We consider refinement for probabilistic contracts. Contrarily to the case of non-probabilistic contracts, we will distinguish between R-Satisfaction and A-Satisfaction.

Following our move from R-Satisfaction to P-R-Satisfaction, we propose the notion of *P-Refinement* that is the quantitative version of the refinement we proposed in Sect. 3. We have the following definition.

Definition 9 (P-Refinement) A probabilistic contract $\mathcal{C}_1 = (V_1, A_1, G_1)$ P-Refines a probabilistic contract $\mathcal{C}_2 = (V_2, A_2, G_2)$ for runs of length k ($k \in \mathbb{N}^\infty$) with level α, denoted $\mathcal{C}_1 \preceq^{R(k)}_{\alpha} \mathcal{C}_2$, iff

$$\forall f \in \mathsf{Sched}((G_1 \cup \neg A_1)\uparrow^{V_1\cup V_2}),$$
$$\mathbb{P}([f([P]^k) \cap (G_2 \cup \neg A_2)\uparrow^{V_1\cup V_2}]\downarrow_P) \geq \alpha.$$

Consider $\mathcal{C}_2 \parallel \mathcal{C}_3$ (respectively, $\mathcal{C}_2 \wedge \mathcal{C}_3$). If $\mathcal{C}_1 \preceq^{R(k)}_{\alpha} \mathcal{C}_2$, then $(\mathcal{C}_1 \parallel \mathcal{C}_3) \preceq^{R(k)}_{\alpha} (\mathcal{C}_2 \parallel \mathcal{C}_3)$ (respectively, $(\mathcal{C}_1 \wedge \mathcal{C}_3) \preceq^{R(k)}_{\alpha} (\mathcal{C}_2 \wedge \mathcal{C}_3)$). Observe that P-Refinement is not a preorder relation. As a consequence, conjunction is not a greatest lower bound with respect to

P-Refinement. Quantitative refinement is compatible with the definition of P-R-Satisfaction, which brings the following result.

Theorem 8 *Consider a P-receptive system $S = (U, \Omega)$ and two probabilistic contracts $\mathcal{C}_i = (V_i, A_i, G_i)$ for $i = 1, 2$. If $(G_1 \cup \neg A_1)$ is P-receptive and prefix-closed, then*

$$S \models_{\alpha}^{R(k)} \mathcal{C}_1 \wedge \mathcal{C}_1 \preceq_{\beta}^{R(k)} \mathcal{C}_2 \quad \Rightarrow \quad S \models_{\alpha+\beta-1}^{R(k)} \mathcal{C}_2.$$

Before giving the proof of the theorem, we propose the following lemma, which proves the existence of corresponding schedulers in two P-receptive systems.

Lemma 3 *Consider $S = (U, \Omega)$ and $S' = (U, \Omega')$ two systems over the same set of variables U. If S and S' are P-receptive and if S' is prefix-closed, then for all $f \in \mathsf{Sched}(S)$, there exists $f' \in \mathsf{Sched}(S')$ such that*

$$\forall w \in [P]^*, \quad f(w) \in S' \quad \Rightarrow \quad f'(w) = f(w).$$

Proof Consider $f \in \mathsf{Sched}(S)$ and let $f' : [P]^* \to S'$ such that:

$$\begin{cases} f'(\varepsilon) = \varepsilon \\ f'(w.\sigma) = f(w.\sigma) \quad \text{if } f(w.\sigma) \in S' \\ f'(w.\sigma) = f'(w).\sigma' \quad \text{s.t.} \quad f'(w).\sigma' \in S' \text{ and } \sigma' \downarrow_P = \sigma. \end{cases}$$

First of all, since S' is prefix-closed, if $f(w) \in S'$, then for all $w' < w$, $f(w') \in S'$, and as a consequence $f'(w') = f(w')$. Moreover, since S' is P-receptive, if $f'(w) \in S'$, then for all $\sigma \in P \to D$, there exists $\sigma' \in U \to D$ such that $\sigma' \downarrow_P = \sigma$ and $f'(w).\sigma' \in S'$. This ensures that the definition of f' is coherent.

We will now prove by induction that $f' \in \mathsf{Sched}(S')$.

- $f'(\varepsilon) = \varepsilon$ satisfies the prefix property.
- Let $w \in [P]^k$ and $w' < w$. Suppose that $f'(w') < f'(w)$. Let $\sigma \in P \to D$.
  - If $f(w.\sigma) \in S'$, then $f'(w.\sigma) = f(w.\sigma)$ and $\forall w'' < w,\ f'(w'') = f(w'')$. Since f is a scheduler, we have $f(w') < f(w.\sigma)$.
  - Else, $f'(w.\sigma) = f'(w).\sigma'$ and as a consequence, $f'(w') < f'(w) < f'(w).\sigma'$. □

We now give the proof for Theorem 8.

Proof of Theorem 8 Consider $f \in \mathsf{Sched}(S \uparrow^{U \cup V_2})$. By Lemma 1, there exists $f' \in \mathsf{Sched}(S \uparrow^{U \cup V_1 \cup V_2})$ such that $f' \downarrow_{U \cup V_2} = f$. Let $f_1 = f' \downarrow_{U \cup V_1}$. By Lemma 1, we have $f_1 \in \mathsf{Sched}(S \uparrow^{U \cup V_1})$. Lemma 3 states that there exists $f_2' \in \mathsf{Sched}((G_1 \cup \neg A_1) \uparrow^{U \cup V_1 \cup V_2})$ such that $\forall w \in [P]^*,\ f'(w) \in (G_1 \cup \neg A_1) \uparrow^{U \cup V_1 \cup V_2} \Rightarrow f_2'(w) = f'(w)$. Let $f_2 = f_2' \downarrow_{V_1 \cup V_2}$. By Lemma 1, we have $f_2 \in \mathsf{Sched}((G_1 \cup \neg A_1) \uparrow^{V_1 \cup V_2}$.

Consider $w \in [P]^k$. If $f_1(w) \in (G_1 \cup \neg A_1) \uparrow^{U \cup V_1}$, then by (2.5), $f'(w) \in (G_1 \cup \neg A_1) \uparrow^{U \cup V_1 \cup V_2} \Rightarrow f_2'(w) = f'(w)$. Moreover, if $f_2(w) \in (G_2 \cup \neg A_2) \uparrow V_1 \cup V_2$, then by (2.5), $f_2'(w) \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2}$. Thus,

$$\begin{aligned}
&f'(w) \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2} \\
&\quad \Rightarrow \quad f(w) \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_2} \quad \text{by (2.4).}
\end{aligned}$$

As a consequence, let

$$\begin{aligned}
E_1 &= [f_1([P]^k) \cap (G_1 \cup \neg A_1) \uparrow^{U \cup V_1}] \downarrow_P \\
E_2 &= [f_2([P]^k) \cap (G_2 \cup \neg A_2) \uparrow^{V_1 \cup V_2}] \downarrow_P \\
E &= [f([P]^k) \cap (G_2 \cup \neg A_2) \uparrow^{U \cup V_2}] \downarrow_P .
\end{aligned}$$

We have $E_1 \cap E_2 \subseteq E$.

This implies, by (3.1), that $\mathbb{P}(E) \geq \mathbb{P}(E_1) + \mathbb{P}(E_2) - 1$. Moreover, by hypothesis,

$$\begin{cases} \mathbb{P}(E_1) \geq \alpha \\ \mathbb{P}(E_2) \geq \beta. \end{cases}$$

Thus, $\mathbb{P}(E) \geq \alpha + \beta - 1$ and $\forall f \in \mathsf{Sched}(S \uparrow^{U \cup V_2})$,

$$\mathbb{P}([f([P]^k) \cap (G_2 \cup \neg A_2) \uparrow^{U \cup V_2}] \downarrow_P) \geq \alpha + \beta - 1. \qquad \square$$

P-A-satisfaction and quantitative refinement are orthogonal measures. Indeed, P-A-satisfaction measures the infimal expected availability of a system for all schedulers, while quantitative refinement measures the infimal set of traces of a probabilistic contract that corresponds to another probabilistic contract. In such context, the minimal schedulers for the two notions may differ. We propose the following result, which links P-A-Satisfaction with the definition of refinement proposed for non-probabilistic contracts.

Theorem 9 *Consider a P-receptive system $S = (U, \Omega)$ and two probabilistic contracts $\mathcal{C}_i = (V_i, A_i, G_i)$ for $i = 1, 2$. If $S \models^{A(k)}_{d,\alpha} \mathcal{C}_1$ and $\mathcal{C}_1 \preceq^{(\leq k)} \mathcal{C}_2$, then $S \models^{A(k)}_{d,\alpha} \mathcal{C}_2$.*

Proof For the sake of simplicity, we will consider that $k = \omega$. The proof for $k < \omega$ is a simpler version of the one presented here.

Consider $f \in \mathsf{Sched}(S \uparrow^{U \cup V_2})$. By Lemma 1, there exists $f' \in \mathsf{Sched}(S \uparrow^{U \cup V_1 \cup V_2})$ such that $f' \downarrow_{U \cup V_2} = f$. Let $f_1 = f' \downarrow_{U \cup V_1}$. By Lemma 1, we also have $f_1 \in \mathsf{Sched}(S \uparrow^{U \cup V_1})$. Consider now $w \in [P]^k$, $t \leq k$ and $i \leq t$. By definition, $\varphi^{\mathcal{C}_1 \uparrow^{U \cup V_1}}_{f_1(w)}(i) = 1 \iff f_1(w)_{[0,i]} \in (G_1 \cup \neg A_1) \uparrow^{U \cup V_1}$. By hypothesis,

$$((G_1 \cup \neg A_1) \uparrow^{V_1 \cup V_2})|^{\leq k} \subseteq ((G_2 \cup \neg A_2) \uparrow^{V_1 \cup V_2})|^{\leq k}.$$

Thus, by (2.5),

$$((G_1 \cup \neg A_1) \uparrow^{U \cup V_1 \cup V_2})|^{\leq k} \subseteq ((G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2})|^{\leq k}.$$

If $\varphi^{\mathcal{C}_1 \uparrow^{U \cup V_1}}_{f_1(w)}(i) = 1$, then

$$\begin{aligned}
&f_1(w)_{[0,i]} \in ((G_1 \cup \neg A_1) \uparrow^{U \cup V_1})|^{\leq k} \\
&\quad \Rightarrow \quad f_1(w) w[0,i] \uparrow^{U \cup V_1 \cup V_2} \subseteq ((G_1 \cup \neg A_1) \uparrow^{U \cup V_1 \cup V_2})|^{\leq k} \\
&\quad \Rightarrow \quad f_1(w) w[0,i] \uparrow^{U \cup V_1 \cup V_2} \subseteq ((G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2})|^{\leq k}
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow \quad f'(w)_{[0,i]} \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2} \\
&\Rightarrow \quad f'(w)_{[0,i]} \downarrow_{U \cup V_2} \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2} \downarrow_{U \cup V_2} \quad \text{by (2.4)} \\
&\Rightarrow \quad f(w)_{[0,i]} \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_2} \quad \text{by (2.2)} \\
&\Rightarrow \quad \varphi_{f(w)}^{\mathcal{C}_2 \uparrow^{U \cup V_2}}(i) = 1.
\end{aligned}$$

Thus,

$$\begin{aligned}
&\forall t \leq k,\ \forall i \leq t, \quad \varphi_{f(w)}^{\mathcal{C}_2 \uparrow^{U \cup V_2}}(i) \geq \varphi_{f_1(w)}^{\mathcal{C}_1 \uparrow^{U \cup V_1}}(i) \\
&\quad \Rightarrow \quad \forall t \leq k, \quad D^{t,d}_{\mathcal{C}_2 \uparrow^{U \cup V_2}}(f(w)) \geq D^{t,d}_{\mathcal{C}_1 \uparrow^{U \cup V_1}}(f_1(w)) \\
&\quad \Rightarrow \quad \liminf_{t \to k} D^{t,d}_{\mathcal{C}_2 \uparrow^{U \cup V_2}}(f(w)) \geq \liminf_{t \to k} D^{t,d}_{\mathcal{C}_1 \uparrow^{U \cup V_1}}(f_1(w)).
\end{aligned}$$

By hypothesis,

$$\liminf_{t \to k} D^{t,d}_{\mathcal{C}_1 \uparrow^{U \cup V_1}}(f_1(w)) \geq \alpha.$$

As a consequence,

$$\begin{aligned}
&\forall w \in [P]^k, \quad \liminf_{t \to k} D^{t,d}_{\mathcal{C}_2 \uparrow^{U \cup V_2}}(f(w)) \geq m \\
&\quad \Rightarrow \quad \int_{w \in [P]^k} \mathbb{P}(w) \cdot \liminf_{t \to k} D^{t,d}_{\mathcal{C}_2 \uparrow^{U \cup V_2}}(f(w)) dw \geq m.
\end{aligned}$$

Finally, $\forall f \in \mathsf{Sched}(S \uparrow^{U \cup V_2})$,

$$\int_{w \in [P]^k} \mathbb{P}(w) \cdot \liminf_{t \to k} D^{t,d}_{\mathcal{C}_2 \uparrow^{U \cup V_2}}(f(w)) dw \geq m. \qquad \square$$

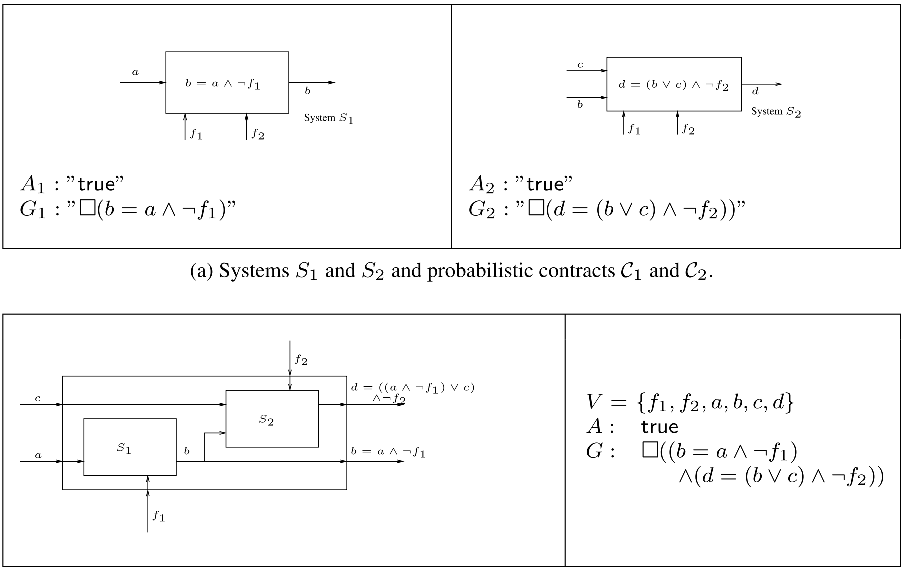
We now briefly discuss independent implementability in the probabilistic case. For P-R-Satisfaction, the property is defined with respect to P-Refinement. For P-A-satisfaction we use the notion of refinement introduced for non-probabilistic contracts. We have the following theorem, whose proof is a direct consequence of Theorems 4, 5, 8 and 9.

Theorem 10 *Let S be a P-receptive system and $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ be three probabilistic contracts such that $\mathcal{C}_1$ and $\mathcal{C}_3$ are P-compatible, and $\mathcal{C}_2$ and $\mathcal{C}_3$ are also P-compatible. We have the following results.*

- *Assume that $(G_1 \cup \neg A_1)$ is prefix-closed and P-receptive. If $\mathcal{C}_1 \preceq_\alpha^{R(k)} \mathcal{C}_2$ and $S \models_\beta^{R(k)} (\mathcal{C}_1 \parallel \mathcal{C}_3)$ (respectively, $S \models_\beta^{R(k)} (\mathcal{C}_1 \wedge \mathcal{C}_3)$), then $S \models_\gamma^{R(k)} (\mathcal{C}_2 \parallel \mathcal{C}_3)$ (respectively, $S \models_\gamma^{R(k)} (\mathcal{C}_2 \wedge \mathcal{C}_3)$), with $\gamma \geq \alpha + \beta - 1$ if $\alpha + \beta \geq 1$ and 0 else.*
- *If $\mathcal{C}_1 \preceq^{(\leq k)} \mathcal{C}_2$ and $S \models_{d,\alpha}^{A(k)} (\mathcal{C}_1 \parallel \mathcal{C}_3)$ (respectively, $S \models_{d,\alpha}^{A(k)} (\mathcal{C}_1 \wedge \mathcal{C}_3)$), then $S \models_{d,\alpha}^{A(k)} (\mathcal{C}_2 \parallel \mathcal{C}_3)$ (respectively, $S \models_{d,\alpha}^{A(k)} (\mathcal{C}_2 \wedge \mathcal{C}_3)$).*

4.2.3 An illustration

The objective of this paper is to introduce the theoretical foundations for contracts and their stochastic extensions. In the rest of this section, we give a simple example that illustrates the approach. Deliverable 5.1.1 of the SPEEDS project (available at [48]) shows the interest of


a
$b = a \wedge \neg f_1$
b
System S_1
f_1
f_2
A_1 : "true"
G_1 : "$\Box(b = a \wedge \neg f_1)$"
c
b
$d = (b \vee c) \wedge \neg f_2$
d
System S_2
f_1
f_2
A_2 : "true"
G_2 : "$\Box(d = (b \vee c) \wedge \neg f_2))$"
f_2
c
S_2
$d = ((a \wedge \neg f_1) \vee c)$
$\wedge \neg f_2$
a
S_1
b
$b = a \wedge \neg f_1$
f_1
$V = \{f_1, f_2, a, b, c, d\}$
A : true
G : $\Box((b = a \wedge \neg f_1)$
$\wedge(d = (b \vee c) \wedge \neg f_2))$

(a) Systems S_1 and S_2 and probabilistic contracts $\mathcal{C}_1$ and $\mathcal{C}_2$.

(b) Systems $S_1 \cap S_2$ and probabilistic contract $\mathcal{C}_1 \parallel \mathcal{C}_2$.

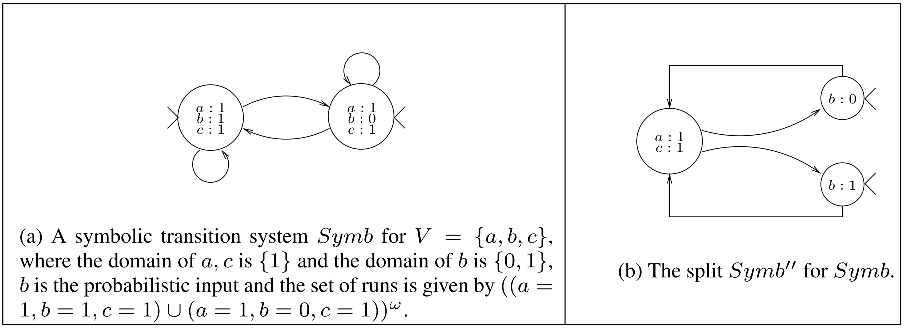
Fig. 3 Reliability: example

industrials for our methodology and discuss other examples for the case of non-stochastic contracts. Also, the work in [32], which can be subsumed by our contribution, has been applied to an interesting case study. We now present the example.

Consider the systems and contracts given in Fig. 3. Assume that $\forall i \in \mathbb{N}$, $\mathbb{P}(f_1(i) = 1) = 10^{-3}$ and $\mathbb{P}(f_2(i) = 1) = 2.10^{-3}$. It is easy to show that $S_1 \VDash^{R(50)}_{(1-10^{-3})^{50}} \mathcal{C}_1$ and $S_2 \VDash^{R(50)}_{(1-2.10^{-3})^{50}} \mathcal{C}_2$. It is however more difficult to deduce the probability for which $S_1 \cap S_2$ satisfies the contract $\mathcal{C}_1 \parallel \mathcal{C}_2$. Thanks to Theorem 4, we know that this probability is at least $(0.999)^{50} + (0.998)^{50} - 1 = 0.86$. Considering $\mathcal{C}_3 = (\{f_1, f_2, a, c, d\}$, "true", "$\Box(d = ((a \wedge \neg f_1) \vee c) \wedge \neg f_2)$"), it is clear that $\mathcal{C}_1 \parallel \mathcal{C}_2 \preceq^{R(50)}_{1} \mathcal{C}_3$, which implies that $S_1 \cap S_2 \models^{R(50)}_{0.86} \mathcal{C}_3$.

4.3 Effective algorithms/representations

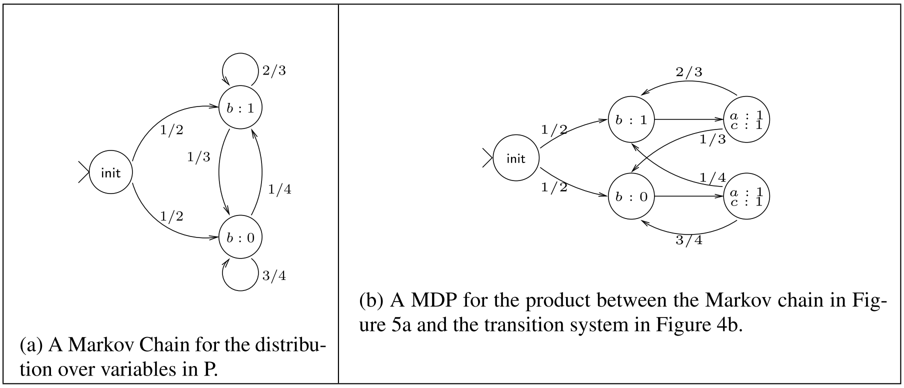
The constructions are similar to those given in Sect. 3.4. We assume the reader to be familiar with the concepts of (discrete) Markov Chains and turn-based Markov Decision Processes (else, see [10, 11, 19, 46] for an introduction and references). Roughly speaking, a Markov Chain is a symbolic transition system whose states are labeled with valuations for variables in P and whose transitions are labeled by probabilities. The labeling by probabilities follows a probability distribution, i.e., for a given state, the sum of the probability values for all outgoing transitions must be less or equal to one. In a given state, one picks up the next valuation for the probability variables, i.e., the next state. The probability to pick up a valuation is the value given on the transition that links the current state to the next chosen one. There is a special state called "*init*" from where one has to chose the first value. The concept of representing P with a Markov Chain is illustrated in Fig. 5a, where $P = \{b\}$ and $D = \{0, 1\}$. In this example, the probability that a run starts with $b = 0$ is $1/2$. The probability that a run starts with the prefix $(b = 0)(b = 1)(b = 0)$ is given by $(1/2) \times (1/4) \times (1/3) = 1/24$.



(a) A symbolic transition system $Symb$ for $V = \{a, b, c\}$, where the domain of a, c is $\{1\}$ and the domain of b is $\{0, 1\}$, b is the probabilistic input and the set of runs is given by $((a = 1, b = 1, c = 1) \cup (a = 1, b = 0, c = 1))^\omega$.

(b) The split $Symb''$ for $Symb$.

Fig. 4 A symbolic transition system and its split



(a) A Markov Chain for the distribution over variables in P.

(b) A MDP for the product between the Markov chain in Figure 5a and the transition system in Figure 4b.

Fig. 5 The product of a split symbolic transition system with a Markov chain

Let $C = (V, \mathcal{B}_A, \mathcal{B}_G)$ be a symbolic contract and $Symb = (V, Q_s, T, Q_{s0})$ be a symbolic transition system. We consider a set $P \subseteq V$ of probabilistic variables. We assume that the distribution over P is symbolically represented with a Markov Chain. At each state, we have a probability distribution over the possible set of valuations for the variables. The Markov chain is finitely-branching as D is finite. Observe that each state of $Symb$ can be split into two states, one for the valuations of the non-probabilistic variables followed by one for the valuations of the probabilistic variables. The result is a new symbolic system $Symb''$ where one first evaluates $V \setminus P$ and then P.

Example 1 The split is illustrated in Fig. 4. Consider the state $X = \{a = 1, b = 0, c = 1\}$ in the system given in Fig. 4a. This state can be split into two states, $A = \{a = 1, c = 1\}$ and $E = \{b = 0\}$. The state $Y = \{a = 1, b = 1, c = 1\}$ can be split into $B = \{a = 1, c = 1\}$ and $F = \{b = 1\}$. In the split, there will be transitions from A to E and from B to F. Any transition from X (resp. Y) to Y (resp. X) will now be from E (resp. F) to B (resp. A). Since A and B have the same label and successors, they can be merged, which gives the split in Fig. 4b.

It is easy to see that we can use the Markov Chain that represents the probability distribution in order to "transform" the transitions from a non-deterministic variable state of $Symb''$

into a probability distribution over the probabilistic variable states simply by synchronizing the two systems. By doing so, *Symb''* becomes a *turn-based Markov Decision Process* (MDP). Recall that a turn-based MDP mixes both non-determinism and probabilities. In our setting, non-determinism thus comes from the choice of the values for the non-probabilistic variables, while probabilities arise when evaluating variables in P. The transitions from states that are labeled with probabilistic variables are thus non-deterministic (since one has to pick up the next values for the non-probabilistic variables). Transitions from states that are labeled with non-probabilistic variables form a probability distribution on the possible values of the probabilistic variables. In this context, a run for the MDP is simply an alternation of valuations of the non-probabilistic and the probabilistic variables.

Example 2 The concept of turn-based Markov Decision Process resulting from the product of a split and a Markov Chain for P is illustrated in Fig. 5(b). Observe that the state $\{a = 1, c = 1\}$ has been duplicated. Indeed, according to the Markov Chain in Fig. 5a, the probability to select $\{b = 0\}$ in the first step is not the same as the one to select it after the first step. The role of the "init" state is to decide (with some probability) of the initial value of the probabilistic variable b. This is a very simple example as in each step, there is only one non-deterministic choice. In more complex designs several choices will be available and a scheduler will have to select one of them, depending or not of the history of the execution.

Remark 3 The above example is somehow naive as it only considers one value for the non-deterministic variables. However, the construction easily generalizes to several values. In the example, there is a clear alternance between states for stochastic variables and states for non-deterministic variables. The latter can be eliminated with a merging of the probabilistic state with its non-deterministic successors (this is the classical transformation of a turn-based Markov Decision Process into a concurrent one). The situation is illustrated in Fig. 6. If we apply this principle to the Markov Decision Process of Fig. 5 and then remove the initial state and the stochastic information from the resulting system, then one obtains the system given in Fig. 4a.

Assuming that the combination of the system with the distribution can be represented with a MDP, we now briefly discuss P-R-Satisfaction and P-A-Satisfaction. In this context, we have the following methodology.

- **P-R-Satisfaction**. Assuming that $\mathcal{B}_A$ and $\mathcal{B}_G$ are Büchi automata, P-R-Satisfaction can be checked with the technique introduced in [9, 21, 51] (which requires a determinization step from Büchi to deterministic Rabin [47]) and implemented in the *LIQUOR* toolset [14]. Indeed, this technique allows to compute the minimal probability for a

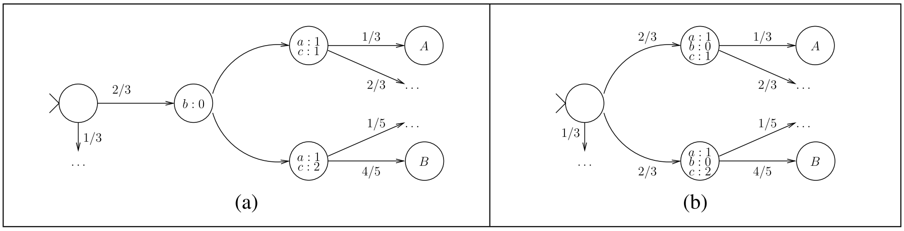


Fig. 6 From turn-based to concurrent Markov decision process

Markov decision process to satisfy a property which is representable with a Büchi automaton. We can thus consider assumptions and guarantees represented with logical formalisms that have a translation to Büchi automata, e.g., ETL [55].

- **P-A-Satisfaction with level m and discount factor d.** The DCTL logic can also be interpreted over MDPs. The definition of synchronous product easily extends to MDPs. The product between a MDP and an automaton can be interpreted as a MDP. We can thus use the labeling technique with propositions that was proposed for the non-probabilistic case (assuming that the states of the automaton have also been split (see the split for transition system)). For a given scheduler (which transforms the MDP into a Markov chain), we can compute the *expected value* for the formula $\triangle_d\ p$. We then compute the minimum between the expected values for all schedulers and check whether it is greater than m. More details about model checking DCTL over MDPs can be found in Sect. 2.2 of [23]. The overall formula we model check is $\forall E[\triangle_d\ p]$, where E states for "expected value".

Remark 4 Memoryless schedulers are sufficient for P-R-Satisfaction, but history-dependent schedulers are needed for the case of P-A-Satisfaction (see page 13 of [23]).

5 Some related work

In this section, we compare our work with related work on contracts, process algebra, modal automata, and interface automata.

Contracts In [4], Benveniste et al. have presented a contract theory where availability, effective representations, and stochastic aspects are not considered. Other definitions of contracts have been proposed in [33, 44] and in [32], where the mathematical theory of [4] is recast in a reactive synchronous language setting. In [42], Pace and Schneider study the satisfaction of contracts that combines deontic and temporal concepts. Composition for such contracts is studied in [29, 30].

Probabilistic contracts In the probabilistic setting, Xu et al. [56] have recently proposed another formalism for probabilistic contracts. Their proposal differs from ours in the sense that they directly focus on Interactive Markov Chains [34], that is a well-known abstract model for stochastic systems with non-determinism on actions. The drawbacks of the approach are that (1) the model they propose does not embed any notion of (global) variables, while our framework can be instantiated with variables, and (2) availability is not considered.

Process algebra 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 [2, 38]. 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 algebras, is that contract theories are general and can be instantiated in many different effective automata-based settings. This covers many logical frameworks (CTL [17], LTL [43], PCTL [35], PSL [27], . . .) for specifying properties of components.

Modal specifications In [37], Larsen proposed *modal specifications* that correspond to *deterministic modal automata*, i.e., automata whose transitions are typed with *may* and *must* modalities. A modal specification thus represents a set of models; informally, a must transition is available in every component that implements the modal specification, while a may transition needs not be. The components that implement modal specifications are prefix-closed languages, or equivalently deterministic automata. As contracts, modal specifications support both refinement, conjunction, and composition operations. Moreover, modal specifications support a quotient operation which is the adjunct of parallel composition [45]. The theory has recently been extended to the timed setting [7, 8]. However, contrary to contracts, modal specifications do not allow an explicit treatment of assumptions and guarantees. It is also known that modal specifications are not more expressive than nu-calculus [28], while the theory of contracts is general and could potentially embed any type of property. Finally, aside from some attempts [16] there is no stochastic extension for modal specifications.

Interface automata In interface automata [22, 24], an interface is represented by an input/output automaton [39], i.e., an automaton whose transitions are labeled with *input* or *output* actions. The semantics of such an automaton is given by a two-player game: an *Input* player represents the environment, and an *Output* player represents the component itself. Interface automata do not encompass any notion of model, because one cannot distinguish between interfaces and implementations. Alternatively, properties of interfaces are described in game-based logics, e.g., ATL [1], with a high-cost complexity. The game-based interpretation offers a more elaborated version of the composition operation than our contract approach. More precisely, the game-based interpretation offers an *optimistic* treatment of composition: two interfaces can be composed if there exists at least one environment (i.e., one strategy for the Input player) in which they can interact together in a safe way (i.e., whatever the strategy of the Output player is). This is referred as compatibility of interfaces. However, contrary to contracts, interface automata do not allow an explicit treatment of assumptions and guarantees. In [41], Pavese et al. propose a quantitative analysis of non-probabilistic models using probabilistic environments. In their setting, a given non-probabilistic system is composed with a probabilistic environment, which allows to perform quantitative analysis. However, the framework does not consider any notion of composition, conjunction, or refinement.

Compositional reasoning Another assume-guarantee approach for the verification of systems consists in decomposing the system into sub-systems and choosing an adequate assumption for a particular decomposition (see [13] for a survey). As we already said in the paper, those works clearly differ from ours. First, they have to find a decomposition of the system in sub-systems, and second, they do not support compositional design operators (conjunction, refinement). In [36], Kwiatkowska et al. propose a compositional verification technique based on assume-guarantee reasoning. In this approach, both assumption and guarantees are regular safety properties represented by finite automata. This work differs from ours in several ways: their satisfaction relation is restricted to safety properties, their compositional rules are qualitative and they consider neither refinement nor conjunction. In [26], another framework is proposed in order to handle dependent probability distributions. However, conjunction is not considered and the stochastic model remains rather simple. Our work is much related to the work by Basu et al. [3] on the BIP toolset [6]. In their work, they do consider a much more elaborated composition operation. However, they do not consider conjunction, availability (they mostly restrict themselves to safety properties), and stochastic aspects.

6 Conclusion

We have proposed a new theory for (probabilistic) contracts, which extends the one we developed for the European project *SPEEDS* [48]. Our contributions are: (1) a theory for reliability and availability, (2) a treatment of the stochastic aspects and (3) a discussion on effective symbolic representations. We are currently implementing the non-probabilistic approach in the SPIN toolset [49] and we plan to implement the probabilistic approach in the LIQUOR toolset [14].

In addition to implementation, there are various other directions for future research. A first direction is to develop a notion of quantitative refinement that is compatible with P-A-satisfaction. We also plan to consider other symbolic representations such as visibly pushdown systems [31]. Considering such representations will require new DCTL model checking algorithms. We also plan to extend our results to the timed setting and consider a more elaborated version of composition. Considering the case of dependent probability distributions like in [26] is also a challenging issue. Finally, it would be interesting to define another satisfaction for contracts based on statistical techniques in the spirit of [15, 50, 57, 58].

References

1. Alur R, Henzinger TA, Kupferman O (2002) Alternating-time temporal logic. J ACM 49(5):672–713
2. Andova S (1999) Process algebra with probabilistic choice. In: ARTS. LNCS, vol 1601. Springer, Berlin, pp 111–129
3. Bensalem S, Bozga M, Nguyen T, Sifakis J (2009) D-finder: A tool for compositional deadlock detection and verification. In: CAV. Lecture notes in computer science, vol 5643. Springer, Berlin, pp 614–619
4. Benveniste A, Caillaud B, Ferrari A, Mangeruca L, Passerone R, Sofronis C (2008) Multiple viewpoint contract-based specification and design. In: FMCO'07. LNCS, vol 5382. Springer, Berlin, pp 200–225
5. Benveniste A, Caillaud B, Passerone R (2007) A generic model of contracts for embedded systems. CoRR, abs/0706.1456
6. Bip—incremental component-based construction of real-time systems. http://www-verimag.imag.fr/async/bip.php
7. Bertrand N, Legay A, Pinchinat S, Raclet J-B (2009) A compositional approach on modal specifications for timed systems. In: ICFEM. LNCS, vol 679–697. Springer, Berlin, p 5885
8. Bertrand N, Pinchinat S, Raclet J-B (2009) Refinement and consistency of timed modal specifications. In: Proc of the 3rd international conference on language and automata theory and applications (LATA'09), Tarragona, Spain, 2009. LNCS, vol 5457. Springer, Berlin, pp 152–163
9. Bustan D, Rubin S, Vardi MY (2004) Verifying omega-regular properties of Markov chains. In: CAV. LNCS, vol 3114. Springer, Berlin, pp 189–201
10. Bertsekas DP, Tsitsiklis JN (2002) Introduction to probability. Scientific, Athena
11. Bertsekas DP, Tsitsiklis JN (2008) Introduction to probability. MIT Press, New York
12. Büchi JR (1960) Weak second-order arithmetic and finite automata. Z Math Log Grundl Math 6:66–92
13. Cobleigh JM, Avrunin GS, Clarke LA (2008) Breaking up is hard to do: An evaluation of automated assume-guarantee reasoning. ACM Trans Softw Eng Methodol 17(2):1–52
14. Ciesinski F, Baier C (2006) Liquor: A tool for qualitative and quantitative linear time analysis of reactive systems. In: QEST. IEEE Computer Society, New York, pp 131–132
15. Clarke EM, Donzé A, Legay A (2010) On simulation-based probabilistic model checking of mixed-analog circuits. Formal Methods Syst Des 36(2):97–113
16. Caillaud B, Delahaye B, Larsen KG, Legay A, Pedersen ML, Wasowski A (2010) Compositional design methodology with constraint Markov chains. In: QEST. IEEE, New York
17. Clarke EM, Emerson EA (1981) Design and synthesis of synchronization skeletons using branching-time temporal logic. In: Logic of programs. LNCS, vol 131. Springer, Berlin, pp 52–71
18. Clarke E, Grumberg O, Peled D (1999) Model checking. MIT Press, New York
19. Cox DR, Miller HD (1965) The theory of stochastic processes / d r cox, h d miller
20. Combest http://www.combest.eu.com
21. de Alfaro L (1997) Formal verification of probabilistic systems. PhD thesis, Stanford University
22. de Alfaro L, da Silva LD, Faella M, Legay A, Roy P, Sorea M (2005) Sociable interfaces. In: FroCos. LNCS, vol 3717. Springer, Berlin, pp 81–105

23. de Alfaro L, Faella M, Henzinger TA, Majumdar R, Stoelinga M (2004) Model checking discounted temporal properties. In: TACAS. LNCS, vol 2988. Springer, Berlin, pp 77–92
24. de Alfaro L, Henzinger TA (2001) Interface automata. In: FSE. ACM Press, New York, pp 109–120
25. de Alfaro L, Henzinger TA (2005) Interface-based design. In: Engineering theories of software-intensive systems. NATO science series: mathematics, physics, and chemistry, vol 195. Springer, Berlin, pp 83–104
26. de Alfaro L, Henzinger TA, Jhala R (2001) Compositional methods for probabilistic systems. In: CONCUR. LNCS, vol 2154. Springer, Berlin, pp 351–365
27. Eisner C, Fisman D (2006) A practical introduction to PSL. Springer, Berlin
28. Feuillade G, Pinchinat S (2007) Modal specifications for the control theory of discrete-event systems. Discrete Event Dyn Syst 17(2):181–205
29. Fenech S, Pace GJ, Schneider G (2009) Automatic conflict detection on contracts. In: ICTAC. Lecture notes in computer science, vol 5684. Springer, Berlin, pp 200–214
30. Fenech S, Pace GJ, Schneider G (2009) Clan: A tool for contract analysis and conflict discovery. In: ATVA. LNCS, vol 5799. Springer, Berlin, pp 90–96
31. Finkel A, Willems B, Wolper P (1997) A direct symbolic approach to model checking pushdown systems. In: ENTCS, vol 9
32. Glouche Y, Le Guernic P, Talpin J-P, Gautier T (2009) A boolean algebra of contracts for logical assume-guarantee reasoning. CoRR, inria-00292870
33. Goessler G, Raclet J-B (2009) Modal contracts for component-based design. In: SEFM. IEEE Computer Society, New York, pp 295–303
34. Hermanns H (2002) Interactive Markov chains: the quest for quantified quality. LNCS, vol 2428. Springer, Berlin
35. Hansson H, Jonsson B (1994) A logic for reasoning about time and reliability. Formal Asp Comput 6(5):512–535
36. Kwiatkowska MZ, Norman G, Parker D, Qu H (2010) Assume-guarantee verification for probabilistic systems. In: TACAS. LNCS, vol 6015. Springer, Berlin, pp 23–37
37. Larsen KG (1989) Modal specifications. In: Automatic verification methods for finite state systems. Lecture notes in computer science, vol 407. Springer, Berlin, pp 232–246
38. López N, Núñez M (2004) An overview of probabilistic process algebras and their equivalences. In: Validation of stochastic systems. LNCS, vol 2925. Springer, Berlin, pp 89–123
39. Lynch N, Tuttle MR (1989) An introduction to Input/Output automata. CWI Q 2(3):219–246
40. Milner R (1989) Communication and concurrency. Prentice Hall, New York
41. Pavese E, Braberman VA, Uchitel S (2009) Probabilistic environments in the quantitative analysis of (non-probabilistic) behaviour models. In: Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT international symposium on foundations of software engineering, Amsterdam, The Netherlands, August 24–28, 2009. ACM Press, New York, pp 335–344
42. Pace GJ, Schneider G (2009) Challenges in the specification of full contracts. In: IFM. Lecture notes in computer science, vol 5423. Springer, Berlin, pp 292–306
43. Pnueli A (1977) The temporal logic of programs. In: FOCS. IEEE, New York, pp 46–57
44. Quinton S, Graf S (2008) Contract-based verification of hierarchical systems of components. In: SEFM. IEEE Computer Society, New York, pp 377–381
45. Raclet J-B (2007) Residual for component specifications. In: FACS
46. Rutten JJMM, Kwiatkowska M, Norman G, Parker D (2004) Mathematical techniques for analyzing concurrent and probabilistic systems, vol 23. American Mathematical Society, Providence
47. Rabin MO, Scott D (1959) Finite automata and their decision problems. IBM J Res Dev 115–125
48. Speeds. http://www.speeds.eu.com
49. The spin tool (spin). Available at http://spinroot.com/spin/whatispin.html
50. Sen K, Viswanathan M, Agha G (2005) On statistical model checking of stochastic systems. In: CAV. LNCS, vol 3576. Springer, Berlin, pp 266–280
51. Vardi MY (1985) Automatic verification of probabilistic concurrent finite-state programs. In: FOCS. IEEE, New York, pp 327–338
52. Vardi MY (2007) From church and prior to psl. Available at http://www.cs.rice.edu/~vardi/papers/index.html
53. Vardi MY, Wolper P (1986) An automata-theoretic approach to automatic program verification (preliminary report). In: LICS. IEEE Computer Society, New York, pp 332–344
54. Vardi MY, Wolper P (1994) Reasoning about infinite computations. Inf Comput 115(1):1–37
55. Wolper P (1983) Temporal logic can be more expressive. Inf Control 56(1/2):72–99
56. Xu DN, Gößler G, Girault A (2010) In: ATVA. LNCS, vol 6252. Springer, Berlin, pp 325–340
57. Younes HLS (2005) Verification and planning for stochastic processes with asynchronous events. PhD thesis, Carnegie Mellon
58. Younes HLS (2006) Error control for probabilistic model checking. In: VMCAI. LNCS, vol 3855. Springer, Berlin, pp 142–156