Probabilistic Contracts : A Compositional Reasoning Methodology for the Design of Stochastic Systems

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

Benoît Caillaud
INRIA/IRISA,
Rennes, France
benoit.caillaud@irisa.fr

Axel Legay
INRIA/IRISA,
Rennes, France
axel.legay@irisa.fr

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.*

1 Introduction

Several industrial sectors involving complex embedded systems have recently experienced deep changes in their organization, aerospace and automotive being the most prominent examples. In the past, they were organized around vertically integrated companies, supporting in-house design activities. These sectors have now evolved into more specialized, horizontally structured companies: *E*quipment *S*uppliers (ESs) and *O*riginal *E*quipment *M*anufacturers (OEMs). OEMs perform system design and integration by importing/combining/reusing entire subsystems (also called components) provided by ESs.

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 [8] and [17] 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 compositional reasoning framework for such systems.

The semantics foundations presented in this paper consist in a mathematical formalism designed to support a component based design methodology and to offer modular and scalable verification techniques. At its basis, the mathematical formalism is a language theoretic abstraction of systems behaviour called *contract* [3]. 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 [8, 23], 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-dependant in the sense that runs that do not satisfy the assumptions are considered to be "correct". This interpretation, which has been suggested by many industrial partners, is needed to propose compositional design operations such as conjunction.

We also propose mathematical definitions for crucial component-based design operations including composition, conjunction and refinement. It is known that most of industrial requirements[1] for component-based design translate to those operations (see [5] for an argumentation). Composition between contracts, which mimics classical composition for systems, consists in taking the intersection between the assumptions and the intersection between the guarantees. Conjunction is a more intriguing operation that has no translation at the level of systems; it consists in producing a contract whose assumptions are the union of the original ones and guarantees are the intersection of the original ones. Roughly speaking, the conjunction of two contracts represents their common requirements. We say that a contract refines another contract if it guarantees more and assumes less. The definition is Boolean for non deterministic systems and quantitative otherwise. We also establish a compositional reasoning 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 or PSL), we observe that checking if a (stochastic) system satisfies a reliability property can be done with classical techniques implemented in tools such as SPIN [22] or LIQUOR [6]. In the paper, we show that satisfaction of availability properties can be checked with an extension of the work presented in [12]. Finally, we also show that operations between and on contracts can easily be performed on the automata-based representations.

Related work In [2], 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 [15, 19]. 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 [16, 1]. In both cases, the main difference is that compositional reasoning is possible only in contract theories thanks to the fact that contracts are implications where an assumption implies a guarantee. A second major difference with process algebras, is that contract theories are general and can be instantiated in many different effective automata-based settings. This covers many logical frameworks (CTL, LTL, PCTL, PSL, ...) for specifying properties of components.

Due to space limitation proofs of theorems are presented in appendix.

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 $w : \{0, \ldots, n-1\} \rightarrow \Sigma$. An *infinite word* (or ω-*word*) w over Σ is a mapping $w : \mathbb{N} \rightarrow \Sigma$. 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*, $\delta : Q \times \Sigma \rightarrow 2^Q$ is a *transition function* ($\delta : Q \times \Sigma \rightarrow Q$ if the automaton is deterministic), and $F \subseteq Q$ is a set of *accepting* states. A *finite run* of A on a finite word $w : \{0, \ldots, n-1\} \rightarrow \Sigma$ is a labeling $\rho : \{0, \ldots, n\} \rightarrow Q$ such that $\rho(0) \in Q_0$, and $(\forall 0 \leq i \leq n-1)(\rho(i+1) \in \delta(\rho(i), w(i)))$. A finite run ρ is *accepting* for w if $\rho(n) \in F$. An *infinite run* of A on an infinite word $w : \mathbb{N} \rightarrow \Sigma$ is a labeling $\rho : \mathbb{N} \rightarrow Q$ such that $\rho(0) \in Q_0$, and $(\forall 0 \leq i)(\rho(i+1) \in \delta(\rho(i), w(i))$. 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 [4] 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 [24] for a survey).

Let $\mathbb{N}_\infty = \mathbb{N} \cup \{\omega\}$ be the closure of the set of natural

[1] Example: those of the European projects COMBEST [10] and SPEEDS [21].

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

integers and $\mathbb{N}_n = [0 \ldots n-1]$ 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 valuations of variables of V. 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 a variable $v \in V$ and a time $i \geq 0$, 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 $[V]^{\leq n}$) 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 $\forall v \in V'$, $\forall n \geq 0$, $w \downarrow_{V'} (n)(v) = w(n)(v)$. Given a run w' on V', the *inverse-projection* of w' on V is the set of runs defined by $w' \uparrow^V = \{w \in [V]^\infty \mid w \downarrow_{V'} = w'\}$. 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 *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 $S|^{\leq n} = (V, \Omega \cap [V]^{\leq n})$ (respectively $S|^n = (V, \Omega \cap [V]^n)$).In Section 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 $Symb = (V, Q_s, T, Q_{s0})$, 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 [3]. 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* [2], 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 Section 3.2, the canonical form is needed to have uniform notions of composition and conjunction between contracts.

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 Section 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 $\varphi_w^C : \mathbb{N}_{|w|} \rightarrow \{0, 1\}$ 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 $D_C^{t,d}(w) = \frac{1}{t} \sum_{i=O}^{t} \varphi_w^C(i)$ 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 Figure 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 τ with discount factor d, denoted $S \models_{d,m}^{A(\tau)} C$, iff:*

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

$$\inf_{w \in (S \uparrow^{U \cup V})|^\tau} \liminf_{t \rightarrow \tau} D_{C \uparrow^{U \cup V}}^{t,d}(w) \geq m \qquad \textit{if } \tau = \omega.$$

It is easy to see that the limit in Definition 3 converges, since $D_C^{t,d} \geq 0$. In Section 3.3 we will propose techniques to check satisfiability for contracts that are represented with symbolic structures.

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.

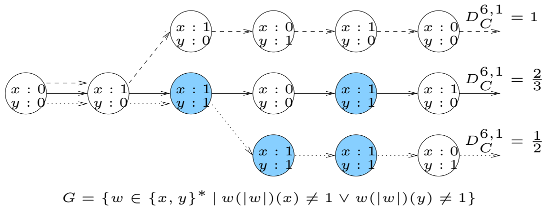
Definition 4 *Let $C_i = (V_i, A_i, G_i)$ with $i = 1, 2$ be two contracts in canonical form. We define*

- *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 [2], 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 $C_2 = (V, \emptyset, \emptyset)$. Suppose 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. Assume 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 contract 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 $C_1 \parallel_{nc} C_2 = (V_1 \cup V_2, (A_1 \cup \neg G_1) \cap (A_2 \cup \neg G_2), G_1 \cap G_2)$.

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



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.

Figure 1. Illustration of mean-availability.

Observe that this composition requires one more complementation operation, which may be computationally intensive depending of the data-structure used to represented A and G (see Section 3.3).

We now turn to the definition of *refinement*, which leads to an order 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* $(G_1 \uparrow^{V_1 \cup V_2})|^{\leq t} \subseteq G_2 \uparrow^{V_1 \cup V_2}$.

Compositional Reasoning We now propose the following results for compositional reasoning in a contract-based setting.

Theorem 1 ([2]) *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* $S_1 \models^{R(t)} C_2$.

Theorem 2 *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 last item of each of the theorems also stands if C_1 and C_2 are not in canonical form.

3.3 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 Section 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 [25, 26], implemented in the *SPIN* toolset [22], 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* [18] and *ETL* [27]. The theory generalizes to other classes of infinite word automata closed under negation and union and other logical formalisms such as *CTL* [9] or *PSL* [13].

Availability with level m and discount factor d. In [12], de Alfaro et al. proposed *DCTL*, a quantitative version of the CTL logic [9]. 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 [12]) 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 labelled 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. 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.

In the spirit of [16], we now consider that the valuations of some variables depend on a probability distribution. This allows to model systems failures. The easiest way to describe probabilistic variables that will be shared between contracts and implementations is to fix a set of global probabilistic variables P. We consider a probability distribution $\mathbb{P}$ over $[P]^\omega$ and extend 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.

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 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.

Our first step is to introduce the definition of scheduler that will be used to resolve non determinism in assumption and guarantee of contracts. Given a system $S = (U, \Omega)$, a scheduler f maps every finite run w on probabilistic variables P to a run $f(w)$ of S which coincides with w for every probabilistic variable. In addition, it is assumed that schedulers are causal, meaning that they resolve non-determinism on a step by step basis. This is ensured by a monotonicity assumption of the schedulers: $\forall w, w' \in [P]^*, w < w' \Rightarrow f(w) < f(w')$.

Definition 6 (Scheduler) *A scheduler f of system $S = (U, \Omega)$ 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 $\mathsf{Sched}(S)$.*

In Section 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 *qualitative* 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 $S \models^{R(k)}_{\alpha} \mathcal{C}$, 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 Section 3.1, we will see that using such an interpretation is needed when considering the conjunction operation (see the observation after Theorem 3).

Though A-Satisfaction was already qualitative, 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

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

4.2 Operations on probabilistic contracts and Compositional reasoning

We now leverage the compositional reasoning results of Section 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 1 and 2 which takes the probabilistic aspects into account.

Theorem 3 (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 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 0 otherwise.*

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.

We now switch to the case of P-A-Satisfaction.

Theorem 4 (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 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 0 otherwise.*

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 Section 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*

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

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

Theorem 5 *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*

$$S \models^{R(k)}_{\alpha} \mathcal{C}_1 \wedge \mathcal{C}_1 \preceq^{R(k)}_{\beta} \mathcal{C}_2 \Rightarrow S \models^{R(k)}_{\alpha+\beta-1} \mathcal{C}_2.$$

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 6 *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 $S \Vdash^{A(k)}_{d,\alpha} \mathcal{C}_1$ and $\mathcal{C}_1 \preceq^{(\leq k)} \mathcal{C}_2$, then $S \models^{A(k)}_{d,\alpha} \mathcal{C}_2$.*

4.2.3 An illustration

Consider the systems and contracts given in Figure 2. 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 3, we know that this probability is at least $(0.999)^{50} + (0.998)^{50} - 1 = 0.86$. Considering $\mathcal{C}_3 = (\{f_1, f_2, a, c, d\}, \text{"}\mathsf{true}\text{"}, \text{"}\square(d = ((a \wedge \neg f_1) \vee c) \wedge \neg f_2)\text{"})$, 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 Section 3.3. We assume the reader to be familiar with the concepts of (discrete) Markov Chain and turn-based Markov Decision Processes. Roughly speaking, a Markov Chain is a symbolic transition system whose states are labeled with valuations for variables in P and transitions by probabilities. The labelling 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 Figure 4(a), 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 $(1/2) \times (1/4) \times (1/3) = 1/24$.

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 evaluate $V \setminus P$ and then P.

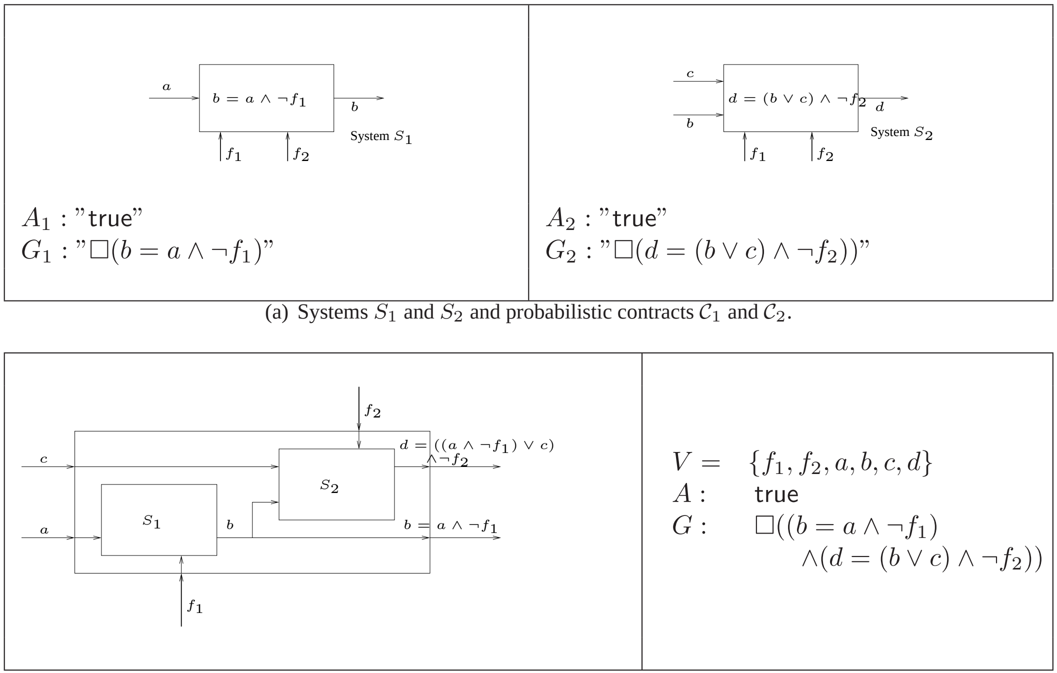
Example 1 *The split is illustrated in Figure 3. Consider the state $X = \{a = 1, b = 0, c = 1\}$ in the system given in Figure (a). This state can be split into two states, $A = \{a = 1, c = 1\}$ and $E = \{b = 0\}$. The state $Y = \{a = 1, b = 1, c = 1\}$ can be split into $B = \{a = 1, c = 1\}$ and $F = \{b = 1\}$. 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 Figure (b).*

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 MDPs mixes both non-determinism and probability. In our setting, non-determinism thus comes from the choice of the values for the non-probabilistic variables, while probability arises when evaluating variables in P. The transitions from states that are labeled with probability 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 alternance 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 Figure 4. Observe that the state $\{a = 1, c = 1\}$ has been duplicated. Indeed, according to the Markov Chain in Figure 5.(a), the probability to select $\{b = 0\}$ in the first step is not the same as the one to select it after the first step.*

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. A *scheduler* for a Markov Decision Process [7] is a mechanism that, in a non deterministic state, selects the successor state without taking predecessors into account. This definition matches the one we proposed in Definition 6. In this context, we have the following methodology.

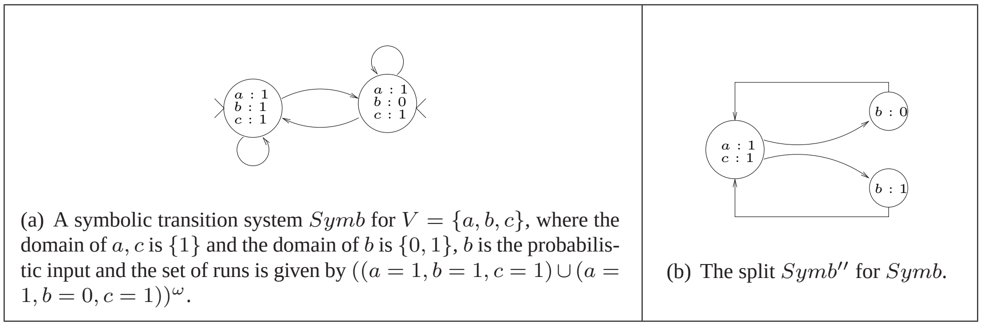
P-R-Satisfaction. Assuming that $\mathcal{B}_A$ and $\mathcal{B}_G$ are Büchi automata, P-R-Satisfaction can be checked with the technique introduced in [23, 11] (which requires a determinization step from Büchi to deterministic Rabin [20]) and implemented in the *LIQUOR* toolset [6]. Indeed, this technique allows to compute the minimal probability for a Markov decision process to satisfy a property which is representable



(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 $\mathcal{C}_1 \parallel \mathcal{C}_2$.

Figure 2. Reliability : Example



(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 $Symb$.

Figure 3. A symbolic transition system and its split.

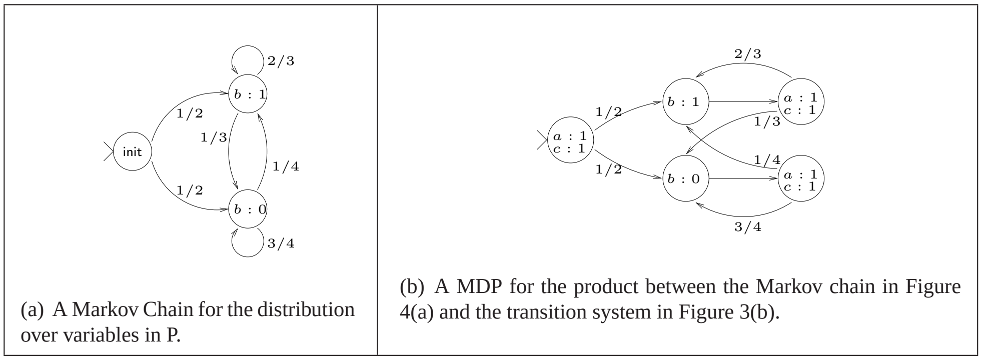
with a Büchi automaton. We can thus consider assumptions and guarantees represented with logical formalism that have a translation to Büchi automata, e.g., ETL [27].

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 labelling 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 Section 2.2 of [12]. The overall formula we model check is $\forall E[\triangle_d\, p]$, where E states for "expected value".

5 Conclusion

We have proposed a new theory for (probabilistic) contracts. 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 [22] and we plan to implement the probabilistic approach in the LIQUOR toolset [6].

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 A-



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

(b) A MDP for the product between the Markov chain in Figure 4(a) and the transition system in Figure 3(b).

Figure 4. The product of a split symbolic transition system with a Markov Chain.

satisfaction. We also plan to consider other symbolic representations such as visibly pushdown systems [14]. Considering such representations will require new DCTL model checking algorithms. We also plan to extend our results to the timed setting. Finally, it would be worth considering the case of dependent probability distributions.

References

[1] S. Andova. Process algebra with probabilistic choice. In *ARTS*, volume 1601 of *LNCS*, pages 111–129. Springer, 1999.

[2] A. Benveniste, B. Caillaud, A. Ferrari, L. Mangeruca, R. Passerone, and C. Sofronis. Multiple viewpoint contract-based specification and design. In *FMCO'07*, volume 5382 of *LNCS*, pages 200–225. Springer, October 2008.

[3] A. Benveniste, B. Caillaud, and R. Passerone. A generic model of contracts for embedded systems. *CoRR*, abs/0706.1456, 2007.

[4] J. R. Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift Math. Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[5] B. Caillaud, B. Delahaye, K. Larsen, A. Legay, M. Pedersen, and A. Wasowski. Compositional design methodology with constraint markov chains. Technical report, INRIA/IRISA Rennes, 2009.

[6] F. Ciesinski and C. Baier. Liquor: A tool for qualitative and quantitative linear time analysis of reactive systems. In *QEST*, pages 131–132. IEEE Computer Society, 2006.

[7] F. Ciesinski and M. Größer. On probabilistic computation tree logic. In *Validation of Stochastic Systems*, volume 2925 of *LNCS*, pages 147–188. Springer, 2004.

[8] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[9] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs*, volume 131 of *LNCS*, pages 52–71. Springer, 1981.

[10] Combest. http://www.combest.eu.com.

[11] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1997.

[12] L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. Model checking discounted temporal properties. In *TACAS*, volume 2988 of *LNCS*, pages 77–92. Springer, 2004.

[13] C. Eisner and D. Fisman. *A Practical Introduction to PSL*. Springer, 2006.

[14] A. Finkel, B. Willems, and P. Wolper. A direct symbolic approach to model checking pushdown systems. *Electr. Notes Theor. Comput. Sci.*, 9, 1997.

[15] Y. Glouche, P. L. Guernic, J.-P. Talpin, and T. Gautier. A boolean algebra of contracts for logical assume-guarantee reasoning. *CoRR*, inria-00292870, 2009.

[16] N. López and M. Núñez. An overview of probabilistic process algebras and their equivalences. In *Validation of Stochastic Systems*, volume 2925 of *LNCS*, pages 89–123. Springer, 2004.

[17] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[18] A. Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE, 1977.

[19] S. Quinton and S. Graf. Contract-based verification of hierarchical systems of components. In *Proc. of the 6th IEEE International Conference on Software Engineering and Formal Methods (SEFM'08)*, pages 377–381. IEEE Computer Society, 2008.

[20] M. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, pages 115–125, 1959.

[21] Speeds. http://www.speeds.eu.com.

[22] The spin tool (spin). Available at http://spinroot.com/spin/whatispin.html.

[23] M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS*, pages 327–338. IEEE, 1985.

[24] M. Y. Vardi. From church and prior to psl, 2007. Available at http://www.cs.rice.edu/˜vardi/papers/index.html.

[25] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *LICS*, pages 332–344. IEEE Computer Society, 1986.

[26] M. Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.

[27] P. Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1/2):72–99, 1983.

A Proofs of the Theorems

A.1 Properties common to all proofs

In this section, we present properties and Lemmas that will be used in all proofs.

Property 1 *Let E_1 and E_2 be two sets of runs over P. We have:*

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

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:*

$$\begin{aligned}
(E \uparrow^{V'}) \uparrow^{V''} &= E \uparrow^{V''}; &(2)\\
(E \uparrow^{V''}) \downarrow_{V'} &= E \uparrow^{V'}; &(3)\\
(E'' \downarrow_{V'}) \downarrow_{V} &= E \downarrow_{V}; &(4)\\
w \in E'' &\Rightarrow w \downarrow_{V} \in E'' \downarrow_{V}; &(5)\\
w \in E &\Rightarrow w \uparrow^{V'} \subseteq E \uparrow^{V'}\ . &(6)
\end{aligned}$$

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:*

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

Proof :
Let $f' = f \downarrow_{U'}$. By definition, $f' : [P]^* \rightarrow S \downarrow_{U'}$. Nonsider 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 (4), $(f(w) \downarrow_{U'}) \downarrow_P = w$.

□

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*

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

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 $w'' = f(w)$.

□

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*

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

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

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

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 $\sigma \in P \rightarrow D$, there exists $\sigma' \in U \rightarrow D$ 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 $\sigma \in P \rightarrow D$.*
  - *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, $f'(w') < f'(w) < f'(w).\sigma'$.*

□

A.2 Proof of Theorem 2

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

1. Proof :

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 (5), 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) and (3), 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 (6), 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,

$$\varphi_w^{C \uparrow^{U \cup V}}(i) \geq \varphi_{w_1}^{C_1 \uparrow^{U_1 \cup V_1}}(i) + \varphi_{w_2}^{C_2 \uparrow^{U_2 \cup V_2}}(i) - 1$$

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

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

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,

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

□

2. Proof :

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 (5), we have $w_1 \in ((S_1 \uparrow^{U_1 \cup V}))|^k \downarrow_{U_1 \cup V_1}$. By (3), 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 (6), 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,

$$\varphi_w^{C \uparrow^{U_1 \cup V}}(i) \geq \varphi_{w_1}^{C_1 \uparrow^{U_1 \cup V_1}}(i) + \varphi_{w_2}^{C_2 \uparrow^{U_1 \cup V_2}}(i) - 1$$

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

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

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,

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

□

3. Proof :
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) and (3), 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, $\varphi_{w_1}^{C_1 \uparrow^{U_1 \cup V_1}}(i) = 1 \iff w_{1[0,i]} \in (G_1 \cup \neg A_1) \uparrow^{U_1 \cup V_1}$. 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 (6), $((G_1 \cup \neg A_1) \uparrow^{U_1 \cup V_1 \cup V_2})|^{\leq k} \subseteq ((G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_1 \cup V_2})|^{\leq k}$.
If $\varphi_{w_1}^{C_1 \uparrow^{U_1 \cup V_1}}(i) = 1$, then

$$\begin{aligned}
& w_{1[0,i]} \in ((G_1 \cup \neg A_1) \uparrow^{U_1 \cup V_1})|^{\leq k} \\
\Rightarrow\ & w_1[0,i] \uparrow^{U_1 \cup V_1 \cup V_2} \subseteq ((G_1 \cup \neg A_1) \uparrow^{U_1 \cup V_1 \cup V_2})|^{\leq k} \\
\Rightarrow\ & w_1[0,i] \uparrow^{U_1 \cup V_1 \cup V_2} \subseteq ((G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_1 \cup V_2})|^{\leq k} \\
\Rightarrow\ & w'_{[0,i]} \in (G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_1 \cup V_2} \\
\Rightarrow\ & w'_{[0,i]} \downarrow_{U_1 \cup V_2} \in (G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_1 \cup V_2} \downarrow_{U_1 \cup V_2} && \textit{by (5)} \\
\Rightarrow\ & w_{[0,i]} \in (G_2 \cup \neg A_2) \uparrow^{U_1 \cup V_2} && \textit{by (3)} \\
\Rightarrow\ & \varphi_w^{C_2 \uparrow^{U_1 \cup V_2}}(i) = 1.
\end{aligned}$$

Thus,

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

By hypothesis,

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

As a consequence,

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

□

A.3 Proof of Theorem 3

1. Proof :
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), 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 (3),

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

Consider now $w \in [P]^k$. If $f_1(w) \in G_1 \uparrow^{U_1 \cup V_1}$, then by (6) and (2), $f_1(w) \uparrow^{U \cup V} \subseteq G_1 \uparrow^{U \cup V}$. 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,

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

This implies, by (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

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

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

□

2. Proof :

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

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

Thus, by (3),

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

Consider now $w \in [P]^k$. If $f_1(w) \in G_1 \uparrow^{U \cup V_1}$, then by (6) and (2), $f_1(w) \uparrow^{U \cup V} \subseteq G_1 \uparrow^{U \cup V}$. 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 (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

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

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

□

A.4 Proof of Theorem 4

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

1. Proof :

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),

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

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

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

Thus, by (3),

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

Consider $w \in [P]^k$, $t \leq k$ and $i \leq t$. If $\varphi_{f(w)}^{\mathcal{C} \uparrow^{U \cup V}}(i) = 0$, then $f(w)_{[0,i]} \notin G \uparrow^{U \cup V}$. By (6) and (3), 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,

$$\varphi_{f(w)}^{C \uparrow^{U \cup V}}(i) \geq \varphi_{f_1(w)}^{C_1 \uparrow^{U_1 \cup V_1}}(i) + \varphi_{f_2(w)}^{C_2 \uparrow^{U_2 \cup V_2}}(i) - 1$$

$$\Rightarrow \forall t \leq k,\ D_{C \uparrow^{U \cup V}}^{(t,d)}(f(w)) \geq D_{C_1 \uparrow^{U_1 \cup V_1}}^{(t,d)}(f_1(w)) \\ + D_{C_2 \uparrow^{U_2 \cup V_2}}^{(t,d)}(f_2(w)) \\ - 1$$

$$\Rightarrow \liminf_{t \to k} D_{C \uparrow^{U \cup V}}^{(t,d)}(f(w)) \geq \liminf_{t \to k} D_{C_1 \uparrow^{U_1 \cup V_1}}^{(t,d)}(f_1(w)) \\ + \liminf_{t \to k} D_{C_2 \uparrow^{U_2 \cup V_2}}^{(t,d)}(f_2(w)) \\ - 1.$$

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

$$\begin{aligned}\liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w)) \geq{}& \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U_1\cup V_1}}(f_1(w))\\ &+ \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U_2\cup V_2}}(f_2(w))\\ &- 1\end{aligned}$$

$$\begin{aligned}\Rightarrow \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w))dw \geq&\\ \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U_1\cup V_1}}(f_1(w))dw&\\ + \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U_2\cup V_2}}(f_2(w))dw&\\ - 1.&\end{aligned}$$

By hypothesis, we have

$$\begin{cases}\displaystyle\int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U_1\cup V_1}}(f_1(w))dw \geq \alpha\\ \displaystyle\int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U_2\cup V_2}}(f_2(w))dw \geq \beta.\end{cases}$$

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. Proof :
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*

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

Thus, by (3)

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

Consider $w \in [P]^k$, $t \leq k$ *and* $i \leq t$. *If* $\varphi^{\mathcal{C}\uparrow^{U\cup V}}_{f(w)}(i) = 0$, *then* $f(w)_{[0,i]} \notin G\uparrow^{U\cup V}$. *By (6) and (3), 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,*

$$\varphi^{C\uparrow^{U\cup V}}_{f(w)}(i) \geq \varphi^{C_1\uparrow^{U\cup V_1}}_{f_1(w)}(i) + \varphi^{C_2\uparrow^{U\cup V_2}}_{f_2(w)}(i) - 1$$

$$\begin{aligned}\Rightarrow \forall t \leq k,\ D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w)) \geq{}& D^{(t,d)}_{C_1\uparrow^{U\cup V_1}}(f_1(w))\\ &+ D^{(t,d)}_{C_2\uparrow^{U\cup V_2}}(f_2(w))\\ &- 1\end{aligned}$$

$$\begin{aligned}\Rightarrow \liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w)) \geq{}& \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U\cup V_1}}(f_1(w))\\ &+ \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U\cup V_2}}(f_2(w))\\ &- 1.\end{aligned}$$

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

$$\begin{aligned}\liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w)) \geq{}& \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U\cup V_1}}(f_1(w))\\ &+ \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U\cup V_2}}(f_2(w))\\ &- 1\end{aligned}$$

$$\begin{aligned}\Rightarrow \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C\uparrow^{U\cup V}}(f(w))dw \geq&\\ \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U\cup V_1}}(f_1(w))dw&\\ + \int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U\cup V_2}}(f_2(w))dw&\\ - 1.&\end{aligned}$$

By hypothesis, we have

$$\begin{cases}\displaystyle\int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C_1\uparrow^{U\cup V_1}}(f_1(w))dw \geq \alpha\\ \displaystyle\int_{w\in[P]^k} \mathbb{P}(w)\cdot \liminf_{t\to k} D^{(t,d)}_{C_2\uparrow^{U\cup V_2}}(f_2(w))dw \geq \beta.\end{cases}$$

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

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

□

A.5 Proof of Theorem 5

Proof :

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 (6),* $f'(w) \in (G_1 \cup \neg A_1) \uparrow^{U \cup V_1 \cup V_2} \Rightarrow f_2'(w) = f'(w)$. *Moreover, if* $f_2(w) \in (G_2 \cup \neg A_2) \uparrow V_1 \cup V_2$, *then by (6),* $f_2'(w) \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2}$. *Thus,*

$$\begin{aligned} & f'(w) \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2} \\ & \Rightarrow f(w) \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_2} \quad \textit{by (5)}. \end{aligned}$$

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

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

□

A.6 Proof of Theorem 6

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.

Proof :

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 again, we 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,* $\varphi_{f_1(w)}^{C_1 \uparrow^{U \cup V_1}}(i) = 1 \iff f_1(w)_{[0,i]} \in (G_1 \cup \neg A_1) \uparrow^{U \cup V_1}$. *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 (6),

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

If $\varphi_{f_1(w)}^{C_1 \uparrow^{U \cup V_1}}(i) = 1$, *then*

$$\begin{aligned} & f_1(w)_{[0,i]} \in ((G_1 \cup \neg A_1) \uparrow^{U \cup V_1})|^{\leq k} \\ & \Rightarrow f_1(w)w[0,i] \uparrow^{U \cup V_1 \cup V_2} \subseteq ((G_1 \cup \neg A_1) \uparrow^{U \cup V_1 \cup V_2})|^{\leq k} \\ & \Rightarrow f_1(w)w[0,i] \uparrow^{U \cup V_1 \cup V_2} \subseteq ((G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2})|^{\leq k} \\ & \Rightarrow f'(w)_{[0,i]} \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2} \\ & \Rightarrow f'(w)_{[0,i]} \downarrow_{U \cup V_2} \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_1 \cup V_2} \downarrow_{U \cup V_2} \quad \textit{by (5)} \\ & \Rightarrow f(w)_{[0,i]} \in (G_2 \cup \neg A_2) \uparrow^{U \cup V_2} \qquad \textit{by (3)} \\ & \Rightarrow \varphi_{f(w)}^{C_2 \uparrow^{U \cup V_2}}(i) = 1. \end{aligned}$$

Thus,

$$\begin{aligned} & \forall t \leq k,\ \forall i \leq t,\ \varphi_{f(w)}^{C_2 \uparrow^{U \cup V_2}}(i) \geq \varphi_{f_1(w)}^{C_1 \uparrow^{U \cup V_1}}(i) \\ & \Rightarrow \forall t \leq k,\ D^{t,d}_{C_2 \uparrow^{U \cup V_2}}(f(w)) \geq D^{t,d}_{C_1 \uparrow^{U \cup V_1}}(f_1(w)) \\ & \Rightarrow \liminf_{t \to k} D^{t,d}_{C_2 \uparrow^{U \cup V_2}}(f(w)) \geq \liminf_{t \to k} D^{t,d}_{C_1 \uparrow^{U \cup V_1}}(f_1(w)). \end{aligned}$$

By hypothesis,

$$\liminf_{t \to k} D^{t,d}_{C_1 \uparrow^{U \cup V_1}}(f_1(w)) \geq \alpha.$$

As a consequence,

$$\begin{aligned} & \forall w \in [P]^k,\ \liminf_{t \to k} D^{t,d}_{C_2 \uparrow^{U \cup V_2}}(f(w)) \geq m \\ & \Rightarrow \int_{w \in [P]^k} \mathbb{P}(w) \cdot \liminf_{t \to k} D^{t,d}_{C_2 \uparrow^{U \cup V_2}}(f(w)) dw \geq m. \end{aligned}$$

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

$$\int_{w \in [P]^k} \mathbb{P}(w) \cdot \liminf_{t \to k} D^{t,d}_{C_2 \uparrow^{U \cup V_2}}(f(w)) dw \geq m$$

□