# Compositional Reasoning on (Probabilistic) Contracts

Benoît Delahaye, Benoit Caillaud, Axel Legay

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Compositional Reasoning on (Probabilistic) Contracts*

Benoît Delahaye, Université de Rennes 1 / IRISA
— Benoît Caillaud, INRIA / IRISA
— Axel Legay, INRIA / IRISA

**N° 6970**

Juin 2009

Thèmes COM et SYM


ISSN 0249-6399 ISRN INRIA/RR--6970--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
RENNES - BRETAGNE ATLANTIQUE

# Compositional Reasoning on (Probabilistic) Contracts

Benoît Delahaye, Université de Rennes 1 / IRISA
, Benoît Caillaud, INRIA / IRISA
, Axel Legay, INRIA / IRISA

Thèmes COM et SYM — Systèmes communicants et Systèmes symboliques
Équipe-Projet S4

Rapport de recherche n° 6970 — Juin 2009 — 33 pages

**Abstract:** In this paper, we focus on Assume/Guarantee contracts consisting in (i) a non deterministic model of components behaviour, and (ii) a stochastic and non deterministic model of systems faults. Two types of contracts capable of capturing reliability and availability properties are considered. We show that Satisfaction and Refinement can be checked by effective methods thanks to a reduction to classical verification problems on Markov Decision Processes and transition systems. Theorems supporting compositional reasoning and enabling the scalable analysis of complex systems are also detailed in the paper.

**Key-words:** Assume/Guarantee Reasoning, Contracts, Probabilistic reasoning, Reliability analysis, Availability.

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

# Raisonnement Compositionel pour les Contrats (Probabilistes)

**Résumé :** Ce document présente un modèle de contrats Assume/Guarantee qui consistent en (i) un modèle non-déterministe pour le comportement de composants et (ii) un modèle stochastique et non-déterministe de fautes. Deux types de contrats capables de capturer des propriétés de fiabilité et disponibilité sont présentés. Il est démontré que la satisfaction et le raffinement peuvent être vérifiés par des méthodes effectives en les réduisant à des problèmes de vérification classiques sur les processus de décision Markoviens et les systèmes de transitions. Des théorèmes assurant un raisonnement compositionel et permettant l'analyse modulaire de systèmes complexes sont présentés.

**Mots-clés :** Raisonnement Assume/Guarantee, Contrats, Raisonnement probabiliste, Analyse de fiabilité, Disponibilité.

# 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: equipment suppliers and *O*riginal *E*quipment *M*anufacturers (OEMs). OEMs perform system design and integration by importing/reusing entire subsystems provided by equipment suppliers. As a consequence, part of the design load has been moved from OEMs to suppliers. An inconvenient of this change is the increased occurrence of late error discovery, system level design errors uncovered at integration time. This is particularly true for system reliability, for state of the art reliability analysis techniques are not modular[17, 22].

A corrective action, taken in the last decade is that the OEMs now focus on the part of the system design at the core of their business, and as far as possible, rely on industry-wide standard platforms. This has an impact on design methods and modeling formalisms: Virtual prototyping and design space exploration are required early in the design cycle. Component based design has emerged as the most promising technique to address the challenges resulting from this new organization of the industry.

However, little has been done regarding the capture of reliability requirements, their formalization in behavioural models and the verification techniques capable of analyzing in a modular way the reliability aspects of a system, at an early stage of design. The paper contributes to solve these issues: The semantics foundations presented in the paper consists in a mathematical formalism designed to support a component based design methodology and to offer modular and scalable reliability analysis techniques. At its basis, the mathematical formalism is a language theoretic abstraction of systems behaviour. The central concept of the formalism is the notion of *contract*, built on top of a basic behavioural formalism. Contracts allow to distinguish hypotheses on a component (assumption) from hypotheses made on its environment (guarantee). Contracts are central to component based design methodologies. The contract-based formalism can be instantiated to cover several aspects, including functional [5], timeliness, hybrid and reliability.

In this paper, we focus on two models of contracts : (i) a non-deterministic model of components behaviour, and (ii) a stochastic and nondeterministic model of systems faults. These contracts are capable of capturing reliability aspects of components and systems. We consider two types of systems properties : 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. While reliability is the notion that is generally considered in formal verification, we observe that availability is crucial when designing, for instance, fault-tolerant systems.

Our second contribution is to propose definitions of (probabilistic) composition, conjunction, refinement, and quotient relations for (probabilistic) contracts. Conjonction and compositon are the classical notions considered in [5]. We say that a contract refines another contract if it guarantees more and assumes less. The definition is boolean for nondeterministic systems and stochastic otherwise. The quotient operation corresponds to the so called “component reuse”, which consists in syntethizing a contract from a global specification and one of its components which is assumed to

be reusable in several designs. We also establish a compositional reasoning theory for those operations and the two notions of satisfiability we consider. The theory differs with the type of contracts under consideration. As an example, we will show that if a non stochastic system $S_1$ reliably satisfies[1] 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 $\alpha$ and $S_2$ satisfies $C_2$ with probability $\beta$, then their composition satisfies the composition of $C_1$ and $C_2$ with probability at least $\alpha + \beta - 1$. The advantage being that the composition, which may be large, does not need to be computed. Our theory is fully general as it assumes that both systems and contracts are possibly infinite sets of runs.

Our last contribution is to propose effective and symbolic representations for contracts and systems. Those representations, which are nothing more than an instance of what we can be handled by automated methods, rely on an automata-based representation of possibly infinite sets of runs. Assuming that assumptions and guarantees are represented with Büchi automata, we observe that checking if a (stochastic) system satisfies a reliability property can be done with classical technics implemented in tools such as SPIN [24] or LIQUOR [8]. In the paper, we show that satisfaction of availability properties can be checked with an extension of the work presented in [12]. Another contribution is to show that operations between and on contracts can easily be performed on the automata-based representations.

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

**Related work** This work is based on previous work on non-probabilistic contracts presented in [5] and also in [16], where the same mathematical theory is recast in a reactive synchronous language setting. Remark that none of the two papers consider system availability, a key contribution of the present paper.

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 [18, 3]. In both cases, the main difference is that compositional reasoning is possible only in contract theories thanks to the fact that contracts are implications where an assumption implies a guarantee. A second major difference with process agebras, is that contract theories are general and can be instantiated in many different effective automata-based settings. This covers many logical frameworks (CTL, LTL, PCTL, PSL, ...) for specifying properties of components. In [7], Chatterjee et al. proposes compositionality results in a quantitative setting. Their approach differs from our approach as they do not consider stochastic aspects and satisfiability.

**Organization of the paper** Section 2 recalls basic language-theoretic concepts of runs and systems. Section 3 recalls non-probabilistic contracts, their compositions, introduces their quotients and two types of satisfaction/refinement relations: One for reliability and one for availability (contribution of the paper). Both types of relations will play an important role in Section 4, where the main contribution of the paper will be presented: A probabilistic contract theory with both reliability and availability satisfaction/refinement/ quotient relations. Compositional theorems of Section 3 are

[1] "Reliably satisfy" means that all the runs that satisfy the assumtion must satisfy the guarantee

generalized to probabilistic systems/contracts, where systems faults are captured in a probability distribution over a set of global stochastic variables. Section 5 deals with effective, automata and logic based instantiations of the probabilistic contract theory, allowing scalable compositional reasoning on possibly large systems.
Some proofs had to be omitted due to space constraints. A self-contained long version of this paper is available at [13].

# 2 Preliminaries

Denote $\mathbb{N}_\infty = \mathbb{N} \cup \{\omega\}$ the closure of the set of natural integers and $\mathbb{N}_n = [0 \dots n-1]$ the interval ranging from $0$ to $n-1$. For the sake of generality, denote $\mathbb{N}_\omega = \mathbb{N}$.

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 $|w|$. Denote $\varepsilon$ 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 $\sigma$ 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 $w' \uparrow^V = \{w \in [V]^\infty \mid w \downarrow_{V'} = w'\}$.

We now define *systems*: Let $V$ be a set of variables. A system over $V$ is a pair $(V, \Omega)$, where $\Omega$ 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.

# 3 Non-Probabilistic Contracts

We introduce the concept of contract and its composition / conjunction / quotient operators and implementation/refinement relations. Finally we conclude with results related to compositional reasoning on contracts.

## 3.1 Contracts and Satisfiability

We recap the concept of *contract* [5], supporting assume-guarantee style of reasoning on systems of components.

**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* asumption *and system $G = (V, \Omega_G)$ is the* guarantee.

Contract $C$ is in *canonical form* if and only if $\neg A \subseteq G$. The canonical form is needed to have uniform notions of composition and conjunction between contracts (see Section 3.2).

We 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 $m$ 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 safetiness* 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 $S|^{\leq t} \cap A \subseteq G$.*

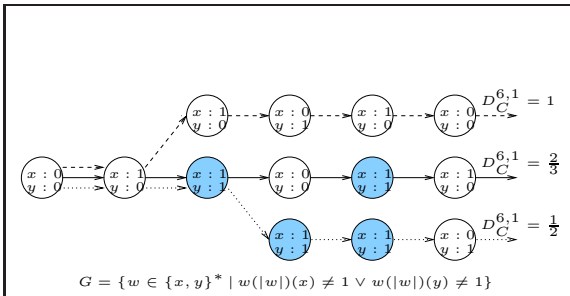
Definition of A-satisfiability is more involved and requires additional notations. As already explained above, the idea is to compute an invariant measure of the amount of time during which the system satisfies a contract. Let $w \in [V]^\infty$ be a (finite or infinite) run and $C = (V, A, G)$ be a contract. Define 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:

$$D_C^{t,d}(\omega) = \frac{1}{t} \sum_{i=O}^{t} \varphi_\omega^C(i) \qquad \text{if } d = 1$$

$$D_C^{t,d}(\omega) = \frac{1-d}{1-d^{t+1}} \sum_{i=0}^{t} d^i \varphi_\omega^C(i) \qquad \text{if } d < 1.$$

$D_C^{t,d}(\omega)$ is the mean-availability until position t along the execution corresponding to $w$ with discount factor $d$. The concept is illustrated in Appendix 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 $\tau$ with discount factor $d$, denoted $S \models_{d,m}^{A(\tau)} C$, iff:*




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.

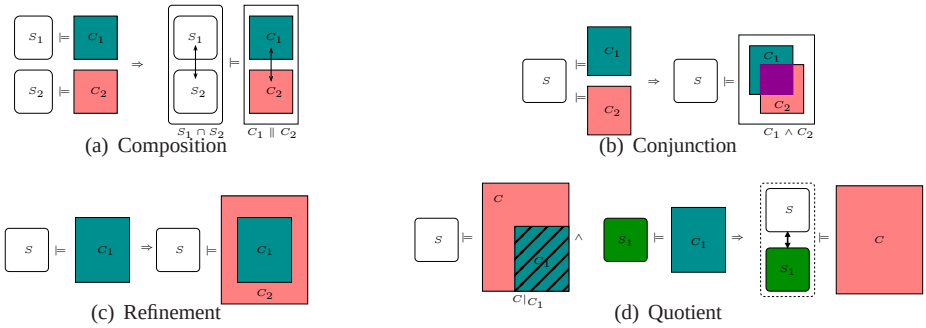



Figure 2: Illustration of operations between / on contracts.

$$\min_{\omega \in (S\uparrow^{U\cup V})|^\tau} D^{\tau,d}_{C\uparrow^{U\cup V}}(\omega) \geq m \qquad \text{if } \tau < \omega$$
$$\min_{\omega \in (S\uparrow^{U\cup V})|^\tau} \liminf_{t\to\tau} D^{t,d}_{C\uparrow^{U\cup V}}(\omega) \geq m \qquad \text{if } \tau = \omega.$$

It is easy to see that the limit in Definition 3 converges, since $D^{t,d}_C \geq 0$. In Section 5 we will propose techniques to check satisfiability for contracts that are represented with symbolic structures.

**Example 1** *The concept of A-Satisfaction is illustrated in Figure 1.*

## 3.2 Compositional reasoning

We first define operations between and on contracts (see Figure 3.2 for a summary) and then propose a compositional reasoning framework for contracts. We start with the definition for composition and conjunction.

**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 conjuction and parallel composition preserve canonicity.

**Remark 1** *The following observation (which is missing in [5]) clarifies the choice of working with contracts that are in canonical form. Assume two contracts $C_1 = (V, \emptyset, [V]^\infty)$ and $C_2 = (V, \emptyset, \emptyset)$. Suppose that $C_1$ is in canonical form, while $C_2$ is not. Assume also that every system satisfies both $C_1$ and $C_2$. The composition between $C_1$ and $C_2$ as defined in the paper is the following contract $(V, [V]^\infty, \emptyset)$. This contract is only 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 $C_2'$ as defined in the paper is satisfied by any system. We did not state that non-canonical contract cannot be composed. Indeed, two non-canonical contracts $C_1 = (V_1, A_1, G_1)$ and $C_2 = (V_2, A_2, G_2)$ can be composed as follows $C_1 \parallel_{nc} C_2 = (V_1 \cup V_2, (A_1 \cup \neg G_1) \cap (A_2 \cup \neg G_2), G_1 \cap G_2)$. Observe that this new combination requires one more complementation operation, which may be computationnaly intensive depending of the data-structure used to represented $A$ and $G$ (see Section 5).*

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

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

**Theorem 1 ([5])** *Consider $S_1$, $S_2$ two systems and $C_1$, $C_2$ two contracts in canonical form. The following propositions hold for all $t \in \mathbb{N}_\infty$:*

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

**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 $t \in \mathbb{N}_\infty$:*

- $S_1 \models^{A(t)}_{d,m_1} C_1 \wedge S_2 \models^{A(t)}_{d,m_2} C_2 \Rightarrow (S_1 \cap S_2) \models^{A(t)}_{d,m_1+m_2-1} (C_1 \parallel C_2)$;
- $S_1 \models^{A(t)}_{d,m_1} C_1 \wedge S_1 \models^{A(t)}_{d,m_2} C_2 \Rightarrow S_1 \models^{A(t)}_{d,m_1+m_2-1} (C_1 \wedge C_2)$;
- $S_1 \models^{A(t)}_{d,m} C_1 \wedge C_1 \preceq^{(\leq t)} C_2 \Rightarrow S_1 \models^{A(t)}_{d,m} C_2$.

The last item of each of the theorems also stands if $C_1$ and $C_2$ are not in canonical form. Theorem 1 was already proposed in [5]. Theorem 2 is our contribution.

Reusing a system $S_1$ that satisfies a contract $C_1$ to realize a global system $S$ that satisfies a contract $C$ amounts to exhibit a residual contract $C|_{C_1}$ such that any system $S_2$ that satisfies $C|_{C_1}$ is such that the composition of $S_1$ and $S_2$ satisfies the contract $C$. This correspond to the notion of quotient which is considered hereafter. We again make the distinction between A-Satisfaction and R-Satisfaction.

**Definition 6 (R-Quotient)** *Consider $C = (V, A, G)$ and $C_1 = (V_1, A_1, G_1)$ two contracts in canonical form and let $\tau \in \mathbb{N}_\infty$. Assume $V_1 \subseteq V$ and $G \subseteq G_1 \uparrow^V$. The set of residuations of $C$ by $C_1$, denoted $C|_{C_1}^{R(\tau)}$, is the set of contracts $C'$ that satisfy the following relation*

$$C' \in C|_{C_1}^{R(\tau)} \iff$$
$$S \models^{R(\tau)} C' \Rightarrow \forall S_1 \models^{R(\tau)} C_1,\ S \cap S_1 \models^{R(\tau)} C.$$

The following theorem states that $C|_{C_1}^{R(\tau)}$ has a largest element w.r.t refinement, and allows to compute it.

**Theorem 3** *Consider $C = (V, A, G)$ and $C_1 = (V_1, A_1, G_1)$ two contracts in canonical form and let $\tau \in \mathbb{N}_\infty$. Assume $V_1 \subseteq V$ and $G \subseteq G_1 \uparrow^V$. Define $C_2$ to be the contract $(V, \neg G \cap G_1, G \cup \neg G_1)$, we have*

- $C_2 \in C|_{C_1}^{R(\tau)}$,
- $\forall C' \in C|_{C_1}^{R(\tau)},\ C' \preceq^{(\leq\tau)} C_2$.

We now switch to the case of A-Satisfaction. Given two contracts $C$ and $C_1$ and two levels of A-Satisfaction $\alpha$ and $x$, we aim at finding a contract $C'$ and a level of satisfaction $\beta$ such that if $S'$ A-Satisfies $C'$ with level at least $\beta$, then for all the systems $S_1$ that A-Satisfy $C_1$ with level alpha, we will have $S' \cap S_1 \models_\alpha^A C$. This is formalized with the following definition.

**Definition 7 (A-Quotient)** *Consider $C = (V, A, G)$ and $C_1 = (V_1, A_1, G_1)$, two contracts in canonical form. Let $\tau \in \mathbb{N}_\infty$ and $d \in [0, 1]$ and assume $V_1 \subseteq V$ and $G \subseteq G_1 \uparrow^V$. Given $\alpha$ and $x \in [0, 1]$, the set of A-residuations of $C$ by $C_1$ with parameters $\alpha$ and $x$, denoted $C|_{C_1}^{A(\tau,d),\alpha,x}$ is the set of pairs $(C', \beta)$ that satisfy the following relation.*

$$(C', \beta) \in C|_{C_1}^{A(\tau,d),\alpha,x} \iff$$
$$\forall S, S_1,\ (S \models_{d,\beta}^{A(\tau)} C') \wedge\ (S_1 \models_{d,x}^{A(\tau)} C_1) \Rightarrow S \cap S_1 \models_{d,\alpha}^{A(\tau)} C.$$

Observe that, as A-Satisfaction is a mean-value, a system will A-Satisfy with the same level several contracts that only differ for a small amount of time / states / runs. There is thus no notion of largest quotient linked to A-Satisfaisability. Nevertheless, the following theorem suggests a methodology to compute an element in $C|_{C_1}^{A(\tau,d),\alpha,x}$.

**Theorem 4** *Consider $C = (V, A, G)$, $C_1 = (V_1, A_1, G_1)$ two contracts in canonical form. Let $\tau \in \mathbb{N}_\infty$, $d, \alpha$ and $x \in [0, 1]$. Let $C_2 = (V, \neg G \cap G_1, G \cup \neg G_1)$. We have*

$$(C_2, \alpha + 1 - x) \in C|_{C_1}^{A(\tau,d),\alpha,x}.$$

# 4 Probabilistic Contracts

In the spirit of [18], we now consider that the valuation 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: $\forall w \in [P]^*$, $\mathbb{P}(w) = \int_{\{w' \in P^\omega \mid w < w'\}} \mathbb{P}(w')dw'$.

## 4.1 Probabilistic contracts and satisfiability

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

**Definition 8 (Scheduler)** *A scheduler $f$ of system $S = (U, \Omega)$ is a monotonous mapping $[P]^* \to \Omega$ such that that for all $w \in [P]^*$, $f(w) \downarrow_P = w$.*

The set of schedulers corresponding to a system $S$ is denoted by $\mathsf{Sched}(S)$. Our notion of schedulers is a generalization of the one proposed for Markov Decision Processes (see also Section 5.3).

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 : *for any scheduler, is the probability to satisfy the contract greater or equal to a certain threshold?* We propose the following definition.

**Definition 9 (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 $\alpha$, denoted $S \models_\alpha^{R(k)} \mathcal{C}$, iff*

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

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-disponibility for all runs of the system, we now consider the *minimal expected value* of the mean-disponibility for all schedulers.

**Definition 10 (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 $\alpha$ and discount factor $d$, denoted $S \|\!\models^{A(k)}_{d,\alpha} \mathcal{C}$, iff*
*If $k < \omega$ :*

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

*If $k = \omega$ :*

$$\inf_{f \in \mathsf{Sched}(S\uparrow^{U\cup V})} \int_{w\in[P]^k} \mathbb{P}(w) \cdot [\liminf_{t\to k} D^{t,d}_{\mathcal{C}\uparrow^{U\cup V}}(f(w))]dw \geq \alpha.$$

## 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/quotient separately. The theory is then illustrated with a toy example.

## 4.3 Composition and Conjunction

Composition and conjunction of probabilistic contracts is defined as for nonprobabilistic contracts (see Definition 4). We thus propose an extension of Theorems 1 and 2 which takes the probabilistic aspects into account.

**Theorem 5 (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 $\gamma \geq \alpha + \beta - 1$.*
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 $\gamma \geq \alpha + \beta - 1$.*

**Theorem 6 (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 $\gamma \geq \alpha + \beta - 1$.*
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 $\gamma \geq \alpha + \beta - 1$.*

## 4.4 Refinement and Quotient

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

Following our move from R-Satisfaction to P-R-Satifaction, 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 11 (P-Refinement)** *A probabilistic contract $\mathcal{C}_1 = (V_1, A_1, G_1)$ P-Refines a second probabilistic contract $\mathcal{C}_2 = (V_2, A_2, G_2)$ for runs of length $k$ ($k \in \mathbb{N}^\infty$) with level $\alpha$, denoted $\mathcal{C}_1 \preceq_\alpha^{R(k)} \mathcal{C}_2$, iff*

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

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

**Theorem 7** *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*

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

P-A-satisfaction and qualitative refinement are orthogonal qualitative measures. Indeed, P-A-satisfaction measures the infimal expected availability of a system for all schedulers, while qualitative 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. Consequently, we are only able to propose the following result, which links P-A-Satisfaction with the definition of refinement proposed for non-probabilistic contracts.

**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 $S \parallel\!\models_{d,\alpha}^{A(k)} \mathcal{C}_1$ and $\mathcal{C}_1 \preceq^{(\leq k)} \mathcal{C}_2$, then $S \models_{d,\alpha}^{A(k)} \mathcal{C}_2$.*

We now leverage the notion of quotient to the probabilistic setting. We again make the distinction between R-satisfaction and A-satisfaction.

**Definition 12 (P-R-Quotient)** *Consider $\mathcal{C} = (V, A, G)$ and $\mathcal{C}_1 = (V_1, A_1, G_1)$, two probabilistic contracts in canonical form. Let $\alpha$ and $x \in [0, 1]$, and $\tau \in \mathbb{N}_\infty$. Assume $V_1 \subseteq V$ and $G \subseteq G_1 \uparrow^V$. The set of P-R-Residuations of $\mathcal{C}$ by $\mathcal{C}_1$ with parameters $\alpha$ and $x$, denoted $\mathcal{C}|_{\mathcal{C}_1}^{R(\tau),\alpha,x}$, is the set of pairs $(\mathcal{C}', \beta)$ that satisfy the following relation*

$$(\mathcal{C}', \beta) \in \mathcal{C}|_{\mathcal{C}_1}^{R(\tau),\alpha,x} \iff$$
$$\forall S, S_1,\ (S \parallel\!\models_\beta^{R(\tau)} \mathcal{C}') \wedge\ (S_1 \parallel\!\models_x^{R(\tau)} \mathcal{C}_1) \Rightarrow S \cap S_1 \parallel\!\models_\alpha^{R(\tau)} \mathcal{C}.$$

Observe that, as P-R-Satisfaction is a probability measure, a system will P-R-Satisfy with the same level several contracts that only differ for a small amount of time / states / runs. Thus, as for A-Satisfactiability, there is no notion of largest quotient linked to P-R-Satisfaisability. Nevertheless, the following theorem suggests a methodology to compute an element in $\mathcal{C}|_{\mathcal{C}_1}^{R(\tau),\alpha,x}$.

**Theorem 9** *Consider $\mathcal{C} = (V, A, G)$ and $\mathcal{C}_1 = (V_1, A_1, G_1)$ two probabilistic contracts in canonical form. Assume $\alpha$ and $x \in [0,1]$ and $\tau \in \mathbb{N}_\infty$. The contract $\mathcal{C}_2 = (V, \neg G \cap G_1, G \cup \neg G_1)$ is such that*

$$(\mathcal{C}_2, \alpha + 1 - x) \in \mathcal{C}|_{\mathcal{C}_1}^{R(\tau),\alpha,x}.$$

**Definition 13 (P-A-Quotient)** *Consider $\mathcal{C} = (V, A, G)$ and $\mathcal{C}_1 = (V_1, A_1, G_1)$, two probabilistic contracts in canonical form. Let $\alpha$ and $x \in [0,1]$, $\tau \in \mathbb{N}_\infty$ and a discount factor $d \in [0,1]$. Assume $V_1 \subseteq V$ and $G \subseteq G_1 \uparrow^V$. The set of P-A-Residuations of $\mathcal{C}$ by $\mathcal{C}_1$ with parameters $\alpha$,$x$ and $d$, denoted $\mathcal{C}|_{\mathcal{C}_1}^{A(\tau,d),\alpha,x}$, is the set of pairs $(\mathcal{C}', \beta)$ that satisfy the following relation*

$$(\mathcal{C}', \beta) \in \mathcal{C}|_{\mathcal{C}_1}^{A(\tau,d),\alpha,x} \iff$$
$$\forall S, S_1,\ (S \models_{d,\beta}^{A(\tau)} \mathcal{C}') \wedge\ (S_1 \models_{d,x}^{A(\tau)} \mathcal{C}_1) \Rightarrow S \cap S_1 \models_{d,\alpha}^{A(\tau)} \mathcal{C}.$$

Once again, there will be no notion of largest quotient linked to P-1-Satisfaction. However, the following theorem suggests a methodology to compute an element in $\mathcal{C}|_{\mathcal{C}_1}^{A(\tau,d),\alpha,x}$.

**Theorem 10** *Consider $\mathcal{C} = (V, A, G)$ and $\mathcal{C}_1 = (V_1, A_1, G_1)$ two probabilistic contracts in canonical form. Assume $\alpha$ and $x \in [0,1]$, $\tau \in \mathbb{N}_\infty$ and $d \in [0,1]$. The contract $\mathcal{C}_2 = (V, \neg G \cap G_1, G \cup \neg G_1)$ is such that*

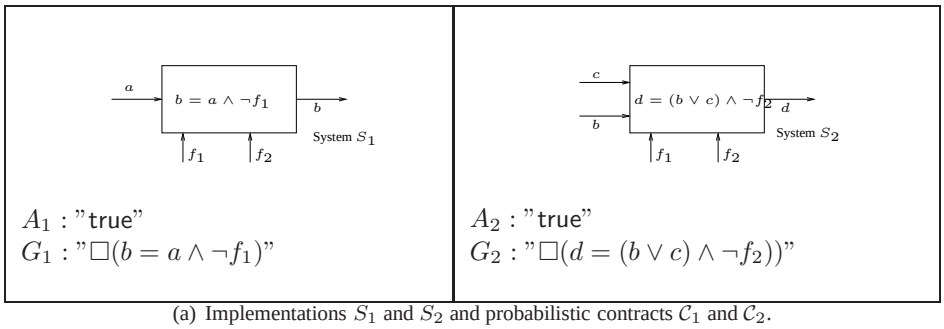
$$(\mathcal{C}_2, \alpha + 1 - x) \in \mathcal{C}|_{\mathcal{C}_1}^{A(\tau),\alpha,x,d}.$$

### 4.5 An example

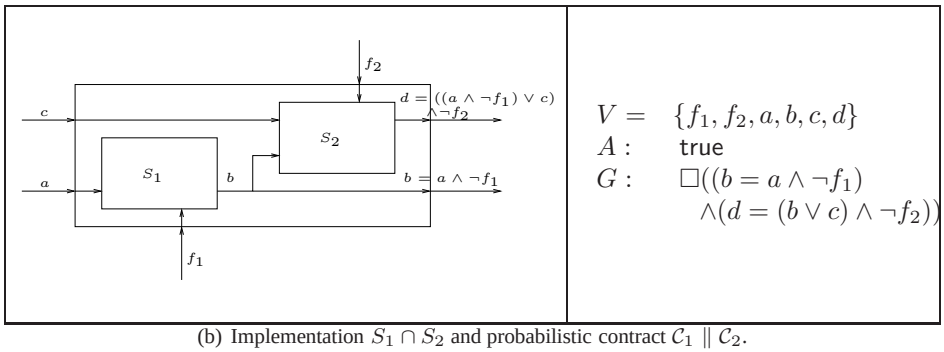
Consider the systems and contracts given in Figure 3. If we consider that the probabilistic variables are pairwise independant and such that $\forall i \in \mathbb{N}, \mathbb{P}(f_1(i) = 1) = 10^{-3}$ and $\mathbb{P}(f_2(i) = 1) = 2.10^{-3}$, then it is clear that $S_1 \models_{(1-10^{-3})^{50}}^{R(50)} \mathcal{C}_1$ and $S_2 \models_{(1-2.10^{-3})^{50}}^{R(50)} \mathcal{C}_2$. It would be more difficult to deduce the probability for which $S_1 \cap S_2$ satisfies the contract $\mathcal{C}_1 \parallel \mathcal{C}_2$ but, thanks to Theorem 5, we know for sure that this probability is at least $(0.999)^{50} + (0.998)^{50} - 1 = 0.86$. Considering $\mathcal{C}_3 = (\{f_1, f_2, a, c, d\}, \text{"}\mathsf{true}\text{"}, \text{"}\Box(d = ((a \wedge \neg f_1) \vee c) \wedge \neg f_2)\text{"})$, it is clear that $\mathcal{C}_1 \parallel \mathcal{C}_2 \preceq_1^{R(50)} \mathcal{C}_3$, which implies that $S_1 \cap S_2 \models_{0.86}^{R(50)} \mathcal{C}_3$.

## 5 Towards implementation : on effective Representations

We suggest symbolic and effective automata-based representations for contracts and systems. The latter is needed to handle possibly infinite sets of runs with a finite memory. Our representations allow to build on existing work when checking for (P-)R-Satisfaction. We will see that the case of (P-)A-Satisfaction can be checked with an extension of the work presented in [12]. Finally, we will also show how to perform operations between and on contracts using those representations.




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




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

Figure 3: Reliability : Example

The section is divided in three parts. In the first part, we recall basic concepts on automata-theory. In the two last parts, parts, we present the symbolic representations.

## 5.1 Background on automata theory and transition systems

We will be working with variables defined over a *finite* domain $D$. We assume the reader to be familiar with automata theory (see Appendix 7 for some definitions and notations). We recap the definition of automata. An automaton is a tuple $A = (\Sigma, Q, Q_0, \delta, F)$, where $\Sigma$ 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 $F$ is an acceptance condition.

We propose examples of effective symbolic representations for possibly infinite sets of runs. 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 we can check its reliability/availability.

We assume that systems that are not assumptions or guarantees are represented with *symbolic transition systems*.

**Definition 14** *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 $D$), $T : Q_s \times Q_s$ 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 $q_{s0}q_{s1}\dots$ 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 $(V, \Omega)$ is a symbolic transition system over $V$ whose set of runs is $\Omega$. Operations of (inverse) projection and intersection easily extend to symbolic transition systems. To simplify the presentation, we will assume that all runs of a symbolic transition system are infinite.

We now propose a symbolic representation for contracts.

**Definition 15** *Let $C = (V, A, G)$ be a contract. A symbolic contract for $C$ is 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 $L(\mathcal{B}_G) = G$.*

## 5.2 Non-probabilistic contracts

We first introduce the definition of *synchronous product* between automata and systems.

**Definition 16** *Let $V$ be a set of variables defined over a finite domain $D$ and $Symb = (V, Q_s, T, Q_{s0})$ be a symbolic transition system over $V$. Let $A = (\Sigma, Q, Q_0, \delta, F \subseteq Q)$ be an automaton such that $\Sigma$ is a mapping $V \rightarrow D$. The synchronous product between $A$ and $Symb$ is the automaton $A' = (\emptyset, Q', Q'_0, \delta', F')$, 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 $A$. If we do not take the information from the $A$ into account, a run of the product corresponds to a run of $Symb$.

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, which allows to consider logics such as LTL [19]. It is conceptually easy to decide whether $Symb$ R-satisfies $C$. Indeed, following results obtained for temporal logics [26, 27], implemented in the *SPIN* toolset [24], this amounts to check whether the Büchi automaton obtained by taking the synchronous product between $Symb$ and $\neg(G \cup \neg A)$ is empty. Observe that assumptions and guarantees can also be represented by logical formalisms that have a direct translation to Büchi automata, which includes *LTL* [19] and *ETL* [28]. The theory generalizes to other classes of infinite word automata closed under negation and union and other logical formalisms such as *CTL* [10] or *PSL* [14].
- **Availability with level m and discount factor d**. In [12], de Alfaro et al. proposed *DCTL*, a quantitative version of the CTL logic [10]. 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 $\varphi_1$ and $\varphi_2$ 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 $\varphi_1$ and $\varphi_2$. The value of $\forall\varphi_1$ (resp. $\exists\varphi_1$) is the minimum (resp. maximum) valuation of $\varphi_1$ 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 $G \cup \neg A$. 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 $Sym$ that satisfy $G \cup \neg A$. 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 $m$.

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 a symbolic contract. Moreover, since inclusion is decidable for those automata, we can always check whether refinement holds.

Systems that are not assumptions or guarantees could be represented by visibly pushdown automata[2] [2] whose language would be the set of runs of the system. In this context, R-Satisfaction can be checked with the technique introduced in [15]. There will be some efforts for A-satisfaction as there exists no algorithm for model checking DCTL on (visibly) pushdown automata. We could also model systems with *timed automata* [1]. The theory for R-Satisfaction and timed words has already been proposed in [5], but there exists no theory for A-Satisfaction.

## 5.3 Probabilistic contracts

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 : for a given state, the sum of the probability values for all outgoing transitions must be less or equal to one. A Markov Decision Process is a transition system with two types of states : the nonprobabilistic states that assign a value to variables in $D \setminus P$ and the probabilistic states that assign a value to variables in $P$. Transitions from nonprobabilistic states go to probabilistic states and are nondeterministic, transitions from probabilistic states go to nonprobabilistic states are labeled with probability values.

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

[2] Recap that visibly pushdown automata are closed under intersection.

assume that the distribution over $P$ is symbolically represented with a Markov Chain[3]. 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.

**Example 2** *The concept of representing $P$ with a Markov Chain is illustrated in Figure 5(a), where $P = \{b\}$ and $D = \{0, 1\}$. As an example, the probability that a run starts with $b = 0$ is $1/2$. The probability that a run starts with $b = 0, b = 1, b = 0$ is given by $(1/2) \times (1/4) \times (1/3)$.*

Observe now that each state of $Symb$ can be splited 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 3** *The split is illustrated in Figure 4. Consider the state $X = \{a = 1, b = 0, c = 1\}$ in the system given in Figure (a). This state can be splited into two states, $A = \{a = 1, c = 1\}$ and $E = \{b = 0\}$. The state $Y = \{a = 1, b = 1, c = 1\}$ can be splited 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).*

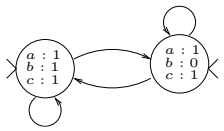
It is easy to see that we can use the Markov Chain for the probability distribution to "transform" the transitions from a non probabilistic variable state of $Symb'$ into a probability distribution over the probabilistic variable states simply by synchronizing the two systems. Hence $Symb''$ becomes a *turn-based Markov Decision Process* (MDP). Recall that a turn-based MDPs mixes both nondeterminism and probability. In our setting, nondeterminism will come from the choice of the values for the non-probabilistic variables, while probability will come when evaluating variables in $P$. The transitions from states that are labeled with probability variables are thus nondeterministic (since one has to pick up the next values for the nonprobabilistic variables). Transitions from states that are labeled with nonprobabilistic 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 nonprobabilistic and the probabilistic variables. A scheduler for a Markov Decision Process [9] is a mechanism that, in a non deterministic state, selects the successor state without taking predecessors into account. This definition particularizes the one we proposed in Definition 8.

**Example 4** *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 5. 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.*

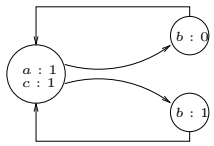
Assuming that the combination of the system with the distribution can be represented with a MDP, we now briefly discuss P-R-Satisfaction and P-A-Satisfaction.

- **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 [25, 11] (which requires a determinization step from Büchi to deterministic Rabin [21]) and implemented in

[3]Roughly speaking, a Markov Chain is a transition system where transitions are labeled with probability values. For a given state, the sum of the values for all outgoing transitions must be less or equal to one.

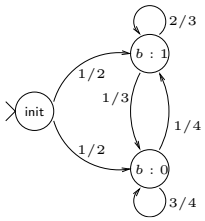



(a) A symbolic transition system $Symb$ for $V = \{a, b, c\}$, where the domain of $a, c$ is 1 and the domain of $b$ is $\{0, 1\}$, $b$ is the probabilistic input and the set of runs is given by $((a = 1, b = 1, c = 1) \cup (a = 1, b = 0, c = 1))^{\omega}$.

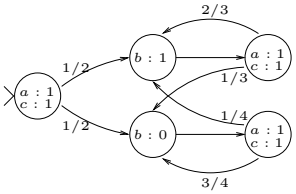



(b) The split $Symb''$ for $Symb$.

Figure 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 5(a) and the transition system in Figure 4(b).

Figure 5: The product of a splitted symbolic transition system with a Markov Chain.

the *LIQUOR* toolset [8]. Indeed, this technique allows to compute the minimal[4] probability for a Markov decision process to satisfy a property which is represented 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 [28].

- **A-Satisfaction with level m and discount factor d**. The DCTL logic can also be interpreted over MDPs. The synchronous product of Definition 16 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 nonprobabilistic case (assuming that the states of the automaton have also been splitted (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".

We observe that probabilistic refinement and quotient can be checked with a technique similar to the one we propose for P-R-Satisfaction.

[4] With respect to a given scheduler.

# 6 Conclusion

We have proposed a new theory for (probabilistic) contracts, which extends the one we developed for the European project *SPEEDS* [23]. The new contributions are : (1) a theory for quotients and availability, (2) a treatment of the probabilistic aspects and (3) a discussion on effective symbolic representations.

We are currently implementing the non probabilistic approach in the SPIN toolset [24] and we plan to implement the probabilistic approach in the LIQUOR toolset [8]. To this purpose, we will have to implement algorithms from [12] and enrich PROMELA [20] and PROBMELA [4] languages with compositional reasoning operators.

In addition to implementation, there are various other directions for future research. A first direction is to develop a notion of qualitative refinement that is compatible with A-satisfaction. We also plan to consider other symbolic representations such as visibly pushdown systems [15]. Considering such representations will require new DCTL model checking algorithms. Finally, we will extend our results to the timed setting.

# References

[1] R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

[2] R. Alur and P. Madhusudan. Visibly pushdown languages. In *Proc. 36th Int. ACM Symposium on Theory of Computing (STOC)*, pages 202–211. ACM, 2004.

[3] S. Andova. Process algebra with probabilistic choice. In *ARTS*, volume 1601 of *Lecture Notes in Computer Science*, pages 111–129. Springer, 1999.

[4] C. Baier, F. Ciesinski, and M. Größer. Probmela and verification of markov decision processes. *SIGMETRICS Performance Evaluation Review*, 32(4):22–27, 2005.

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

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

[7] K. Chatterjee, L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. Compositional quantitative reasoning. In *QEST*, pages 179–188. IEEE Computer Society, 2006.

[8] 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.

[9] F. Ciesinski and M. Größer. On probabilistic computation tree logic. In *Validation of Stochastic Systems*, volume 2925 of *Lecture Notes in Computer Science*, pages 147–188. Springer, 2004.

[10] 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 *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[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 *Lecture Notes in Computer Science*, pages 77–92. Springer, 2004.

[13] B. Delahaye, B. Caillaud, and A. Legay. Compositional reasoning on (probabilistic) contracts (long version). Technical report. Available at http://perso.bretagne.ens-cachan.fr/˜delahaye.

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

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

[16] 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.

[17] A. Høyland and M. Rausand. *System Reliability Theory: Models and Statistical Methods*. J. Wiley & Sons, New York, 1994.

[18] N. López and M. Núñez. An overview of probabilistic process algebras and their equivalences. In *Validation of Stochastic Systems*, volume 2925 of *Lecture Notes in Computer Science*, pages 89–123. Springer, 2004.

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

[20] The promela language. available at http://spinroot.com/spin/Man/promela.html.

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

[22] R. Sinnamon and J. Andrews. Fault tree analysis and binary decision diagrams. pages 215–222, Jan 1996.

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

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

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

[26] 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.

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

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

# 7 Preliminaries on finite automata

Let $\Sigma$ be an alphabet. A finite word over $\Sigma$ is a mapping $w : \{0, \ldots, n-1\} \to \Sigma$. An *infinite word* (or $\omega$-*word*) $w$ over $\Sigma$ is a mapping $w : \mathbb{N} \to \Sigma$. An automaton is a tuple $A = (\Sigma, Q, Q_0, \delta, F)$, where $\Sigma$ 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 finite word $w : \{0, \ldots, n-1\} \to \Sigma$ is a labeling $\rho : \{0, \ldots, n\} \to Q$ 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 $\rho$ 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 $(\forall 0 \leq i)(\rho(i+1) \in \delta(w(i), \rho(i))$. An infinite run $\rho$ 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 $\rho$.

We distinguish between finite-word automata that are finite automata accepting finite words, and Büchi automata [6] 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, union, and complementation. Inclusion and emptiness are also decidable.

# 8 Properties common to all proofs

In this section, we recap 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:*

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

**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:*

$$(E \uparrow^{V'}) \uparrow^{V''} = E \uparrow^{V''}; \tag{2}$$

$$(E \uparrow^{V''}) \downarrow_{V'} = E \uparrow^{V'}; \tag{3}$$

$$(E'' \downarrow_{V'}) \downarrow_{V} = E \downarrow_{V}; \tag{4}$$

$$w \in E'' \Rightarrow w \downarrow_V \in E'' \downarrow_V; \tag{5}$$

$$w \in E \Rightarrow w \uparrow^{V'} \subseteq E \uparrow^{V'} . \tag{6}$$

**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. We have:*

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

Proof :
*Let $f' = f \downarrow_{U'}$. By definition, $f' : [P]^* \to S \downarrow_{U'}$. 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*

$$\forall f \in \mathsf{Sched}(S),\ \exists f' \in \mathsf{Sched}(S') \text{ s.t. } \forall w \in [P]^*,\ f(w) \in S' \Rightarrow f'(w) = f(w).$$

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

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

*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, $f'(w') < f'(w) < f'(w).\sigma'$.*

# 9 Proof of Theorem 2

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 $C = (V, A, G) = C_1 \parallel C_2$. 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,*

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

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

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

   *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,*

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

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,*

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

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

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

*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,*

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

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*

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

*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,*

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

# 10 Proof of Theorem 3

Proof :

1. $C_2 \in C|_{C_1}^{R(\tau)}$*:*

   *Consider* $S_1$ *and* $S_2$ *two systems such that* $S_1 \models^{R(\tau)} C_1$ *and* $S_2 \models^{R(\tau)} C_2$*. By theorem 1, we have* $S_1 \cap S_2 \models^{R(\tau)} C_1 \parallel C_2 = C'$*. After simplifications,* $C' = (V, \neg G \cup \neg G_1, G \cap G_1)$*. By definition,* $(S_1 \cap S_2)|^{(\leq\tau)} \subseteq G \cap G_1 \cup \neg(\neg G \cup \neg G_1) = G \cap G_1 \subseteq G \cup \neg A$*. Thus* $S_1 \cap S_2 \models^{R(\tau)} C$*.*

2. $\forall C' \in C|_{C_1}^{R(\tau)},\ C' \preceq^{(\leq\tau)} C_2$*:*

   *Let* $C' = (V', A', G') \in C|_{C_1}^{R(\tau)}$*. Consider* $S' = (V', G')$*,* $S_1 = (V_1, G_1)$ *and* $S_2 = (V', \neg A')$*. We have* $S' \models^{R(\tau)} C'$ *and* $S_1 \models^{R(\tau)} C_1$*. By definition, we thus have* $S' \cap S_1 \models^{R(\tau)} C$*, and as a consequence,* $(G' \uparrow^{V_1} \cap G_1)|^{\leq\tau} \subseteq G$*. Thus* $(G' \uparrow^{V_1})|^{\leq\tau} \subseteq G \cup \neg G_1$*.*

   *Moreover, since* $S_2 \models^{R(\tau)} C'$*, we have* $[(\neg A') \uparrow^{V_1} \cap G_1]|^{\leq\tau} \subseteq G$*. This implies* $[(\neg A') \uparrow^{V_1}]|^{\leq\tau} \subseteq G \cup \neg G_1$*, and hence* $[\neg G \cap G_1]|^{\leq\tau} \subseteq A' \uparrow^{V_1}$*.*

# 11 Proof of Theorem 4

Proof :
*Consider two systems* $S_1$ *and* $S_2$ *such that* $S_1 \models_{d,x}^{A(\tau)} C_1$ *and* $S_2 \models_{d,\alpha+1-x}^{A(\tau)} C_2$*. By theorem 2, we have* $S_1 \cap S_2 \models_{d,\alpha}^{A(\tau)} C_1 \parallel C_2 = C'$*. After simplifications,* $C' = (V, \neg G_1 \cup \neg G, G_1 \cap G)$*.*

*By definition,* $\forall w \in ((S_1 \cap S_2) \uparrow^V)|^\tau,\ \forall i \leq t \leq \tau,\ \varphi_w^{C'}(i) = 1 \Rightarrow w_{[0,i]} \in (G_1 \cap G) \Rightarrow w_{[0,i]} \in (G \cup \neg A) \Rightarrow \varphi_w^C(i) = 1$*. As a consequence,*

$$\begin{aligned}&\forall w \in ((S_1 \cap S_2) \uparrow^V)|^\tau, \forall i \leq \tau,\ \varphi_w^C \geq \varphi_w^{C'}\\ \Rightarrow &\forall t \leq \tau,\ \forall w \in ((S_1 \cap S_2) \uparrow^V)|^\tau,\ D_C^{t,d}(w) \geq D_{C'}^{t,d}(w)\\ \Rightarrow &S_1 \cap S_2 \models_{d,\alpha}^{A(\tau)} C.\end{aligned}$$

# 12 Proof of Theorem 5

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 $(f \in \mathsf{Sched}((S_1 \uparrow^{U_1 \cup U_2}) \uparrow^{U \cup V})) \wedge (f \in \mathsf{Sched}((S_2 \uparrow^{U_1 \cup U_2}) \uparrow^{U \cup V}))$.*

$$\Rightarrow (f \in \mathsf{Sched}(S_1 \uparrow^{U \cup V})) \wedge (f \in \mathsf{Sched}(S_2 \uparrow^{U \cup V}))\ \textit{by (2)}.$$

   *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*

$$\begin{aligned}&\begin{cases} \wedge & (f_1 \in \mathsf{Sched}((S_1 \uparrow^{U \cup V}) \downarrow_{U_1 \cup V_1}))\\ & (f_2 \in \mathsf{Sched}((S_2 \uparrow^{U \cup V}) \downarrow_{U_2 \cup V_2})) \end{cases}\\ \Rightarrow &(f_1 \in \mathsf{Sched}(S_1 \uparrow^{U_1 \cup V_1}) \wedge (f_2 \in \mathsf{Sched}(S_2 \uparrow^{U_2 \cup V_2}))\ \textit{by (3)}.\end{aligned}$$

   *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*

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

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*

$$\begin{array}{l} \left\{ \begin{array}{l} (f_1 \in \mathsf{Sched}((S \uparrow^{U \cup V}) \downarrow_{U \cup V_1})) \\ \wedge \\ (f_2 \in \mathsf{Sched}((S \uparrow^{U \cup V}) \downarrow_{U \cup V_2})) \end{array} \right. \\ \Rightarrow (f_1 \in \mathsf{Sched}(S \uparrow^{U \cup V_1}) \wedge (f_2 \in \mathsf{Sched}(S \uparrow^{U_2 \cup V_2})) \text{ by (3).} \end{array}$$

   *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,*

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

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

$$\left\{ \begin{array}{l} \mathbb{P}(E_1) \geq \alpha \\ \mathbb{P}(E_2) \geq \beta. \end{array} \right.$$

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

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

# 13 Proof of Theorem 6

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* $(f \in \mathsf{Sched}((S_1 \uparrow^{U_1\cup U_2}) \uparrow^{U\cup V})) \wedge (f \in \mathsf{Sched}((S_2 \uparrow^{U_1\cup U_2}) \uparrow^{U\cup V}))$.

$$\Rightarrow (f \in \mathsf{Sched}(S_1 \uparrow^{U\cup V})) \wedge (f \in \mathsf{Sched}(S_2 \uparrow^{U\cup V})) \text{ by (2).}$$

*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*

$$\begin{aligned}\Rightarrow & \begin{cases} \wedge & (f_1 \in \mathsf{Sched}((S_1 \uparrow^{U\cup V}) \downarrow_{U_1\cup V_1})) \\ & (f_2 \in \mathsf{Sched}((S_2 \uparrow^{U\cup V}) \downarrow_{U_2\cup V_2})) \end{cases} \\ & \Rightarrow (f_1 \in \mathsf{Sched}(S_1 \uparrow^{U_1\cup V_1}) \wedge (f_2 \in \mathsf{Sched}(S_2 \uparrow^{U_2\cup V_2})) \text{ by (3).}\end{aligned}$$

*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* $[(f_1(w)_{[0,i]} \notin G_1 \uparrow^{U_1\cup V_1}) \vee (f_2(w)_{[0,i]} \notin G_2 \uparrow^{U_2\cup V_2})]$. *As a consequence,*

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

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

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

*As a consequence,*

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

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

*By hypothesis, we have*

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

*Thus,*

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

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

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*

$$\begin{aligned} &\Rightarrow \begin{cases} \wedge & (f_1\in\mathsf{Sched}((S\uparrow^{U\cup V})\downarrow_{U\cup V_1})) \\ & (f_2\in\mathsf{Sched}((S\uparrow^{U\cup V})\downarrow_{U\cup V_2})) \end{cases} \\ &\Rightarrow (f_1\in\mathsf{Sched}(S\uparrow^{U\cup V_1})\wedge(f_2\in\mathsf{Sched}(S\uparrow^{U\cup V_2}))\ \textit{by (3).} \end{aligned}$$

*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,*

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

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

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

*As a consequence,*

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

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

*By hypothesis, we have*

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

*Thus,*

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

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

# 14 Proof of Theorem 7

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,*

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

*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*

$$\begin{aligned} &\forall f \in \mathsf{Sched}(S \uparrow^{U \cup V_2}),\ \mathbb{P}([f([P]^k) \cap (G_2 \cup \neg A_2) \uparrow^{U \cup V_2}] \downarrow_P) \geq \alpha + \beta - 1 \\ &\Rightarrow \inf_{f \in \mathsf{Sched}(S \uparrow^{U \cup V_2})} \mathbb{P}([f([P]^k) \cap (G_2 \cup \neg A_2) \uparrow^{U \cup V_2}] \downarrow_P) \geq \alpha + \beta - 1. \end{aligned}$$

# 15 Proof of Theorem 8

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*

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

*Thus,*

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

*By hypothesis,*

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

*As a consequence,*

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

*Finally,*

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

# 16 Proof of Theorem 9

Proof :
*Consider two systems $S_1$ and $S_2$ such that $S_1 \parallel\!\models^{R(\tau)}_{x} \mathcal{C}_1$ and $S_2 \parallel\!\models^{R(\tau)}_{\alpha+1-x} \mathcal{C}_2$. By theorem 5, we have $S_1 \cap S_2 \parallel\!\models^{R(\tau)}_{\alpha} \mathcal{C}_1 \parallel \mathcal{C}_2 = \mathcal{C}'$. After simplifications, $\mathcal{C}' = (V, \neg G_1 \cup \neg G, G_1 \cap G)$.*

*Let $f \in \mathsf{Sched}(S_1 \cap S_2 \uparrow^V)$, we have by definition*

$$\mathbb{P}([f([P]^k) \cap (G_1 \cap G)\uparrow^V]\downarrow_P) \geq \alpha.$$

*Moreover, $G_1 \cap G \subseteq G \cup \neg A$. As a consequence,*

$$\mathbb{P}([f([P]^k) \cap (G \cup \neg G_1)\uparrow^V]\downarrow_P) \geq \mathbb{P}([f([P]^k) \cap (G_1 \cap G)\uparrow^V]\downarrow_P)$$
$$\Rightarrow \mathbb{P}([f([P]^k) \cap (G \cup \neg G_1)\uparrow^V]\downarrow_P) \geq \alpha.$$

# 17 Proof of Theorem 10

Proof :
*Consider two systems $S_1$ and $S_2$ such that $S_1 \parallel\!\models^{A(\tau)}_{d,x} \mathcal{C}_1$ and $S_2 \parallel\!\models^{A(\tau)}_{d,\alpha+1-x} \mathcal{C}_2$. By theorem 6, we have $S_1 \cap S_2 \parallel\!\models^{A(\tau)}_{d,\alpha} \mathcal{C}_1 \parallel \mathcal{C}_2 = \mathcal{C}'$. After simplifications, $\mathcal{C}' = (V, \neg G_1 \cup \neg G, G_1 \cap G)$.*

*By definition, $\forall w \in [P]^\tau,\ \forall f \in \mathsf{Sched}((S_1 \cap S_2)\uparrow^V),\ \forall i \leq t \leq \tau$, $\varphi^{\mathcal{C}'}_{f(w)}(i) = 1 \Rightarrow f(w)_{[0,i]} \in (G_1 \cap G) \Rightarrow f(w)_{[0,i]} \in (G \cup \neg A) \Rightarrow \varphi^{\mathcal{C}}_{f(w)}(i) = 1$. As a consequence,*

$$\forall w \in [P]^k,\ \forall f \in \mathsf{Sched}((S_1 \cap S_2)\uparrow^V), \forall i \leq \tau,\ \varphi^{\mathcal{C}}_{f(w)} \geq \varphi^{\mathcal{C}'}_{f(w)}$$
$$\Rightarrow \forall t \leq \tau,\ \forall w \in [P]^k,\ \forall f \in \mathsf{Sched}((S_1 \cap S_2)\uparrow^V),\ D^{t,d}_{\mathcal{C}}(f(w)) \geq D^{t,d}_{\mathcal{C}'}(f(w))$$
$$\Rightarrow S_1 \cap S_2 \parallel\!\models^{A(\tau)}_{d,\alpha} \mathcal{C}.$$

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399