



HAL
open science

Compositional Design Methodology with Constraint Markov Chains

Benoit Caillaud, Benoît Delahaye, Kim Guldstrand Larsen, Axel Legay,
Mikkel L. Pedersen, Andrzej Wasowski

► **To cite this version:**

Benoit Caillaud, Benoît Delahaye, Kim Guldstrand Larsen, Axel Legay, Mikkel L. Pedersen, et al..
Compositional Design Methodology with Constraint Markov Chains. [Research Report] RR-6993,
INRIA. 2009. inria-00404304v2

HAL Id: inria-00404304

<https://inria.hal.science/inria-00404304v2>

Submitted on 25 May 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Compositional Design Methodology with Constraint Markov Chains

Benoît Caillaud, INRIA / IRISA, France
— Benoît Delahaye, Université de Rennes 1 / IRISA, France
— Kim G. Larsen, Aalborg University, Denmark
— Axel Legay, INRIA / IRISA, France
— Mikkel L. Pedersen, Aalborg University, Denmark
— Andrzej Wařowski, IT University of Copenhagen, Denmark

N° 6993

Mai 2010

Thèmes COM et SYM

A large, light blue stylized 'R' logo is positioned to the left of the text 'Rapport de recherche'.

*Rapport
de recherche*

Compositional Design Methodology with Constraint Markov Chains

Benoît Caillaud, INRIA / IRISA, France
, Benoît Delahaye, Université de Rennes 1 / IRISA, France
, Kim G. Larsen, Aalborg University, Denmark
, Axel Legay, INRIA / IRISA, France
, Mikkel L. Pedersen, Aalborg University, Denmark
, Andrzej Wasowski, IT University of Copenhagen, Denmark

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

Rapport de recherche n° 6993 — Mai 2010 — 44 pages

Abstract: A specification theory combines notions of specification and implementation with a satisfaction relation, a refinement relation and a set of operators that together support stepwise design. We propose a new abstraction, Constraint Markov Chains, and use it to construct a specification theory for Markov Chains. Constraint Markov Chains generalize previously known abstractions by allowing arbitrary constraints on probability distributions. Our theory is the first specification theory for Markov Chains closed under conjunction, parallel composition and synchronization. Moreover, all the operators and relations introduced are computable.

Key-words: Compositional Reasoning, Probability, CMC

Méthode de Conception Compositionnelle avec les Chaînes de Markov avec Contraintes

Résumé : Une théorie de spécification combine les notions de spécification et implémentation avec des relations de satisfaction et raffinement, et un ensemble d'opérateurs qui permettent une conception incrémentale. Nous proposons une nouvelle abstraction, les chaînes de Markov avec contraintes, et les utilisons pour construire une théorie de spécification pour les chaînes de Markov. Les chaînes de Markov avec contraintes généralisent d'autres abstractions plus anciennes en autorisant des contraintes arbitraires sur les distributions de probabilité. Notre théorie est la première théorie de spécification pour les chaînes de Markov close sous conjonction, composition parallèle et synchronisation. De plus, tous les opérateurs et relations introduits sont calculables.

Mots-clés : Raisonnement Compositionnel, Probabilités, CMC

1 Introduction

Modern systems are big and complex, resulting from assembling multiple components. The components are designed by teams, working independently but with a common agreement on what the interface of each component should be. As a consequence, mathematical foundations that allow to reason at the abstract level of interfaces in order to infer global properties are an active research area known as *compositional design* [17]. Within this area specification theories provide a modeling language that allows designing, evolving and advisedly reusing components with formal guarantees. In a logical interpretation, interfaces are specifications and systems/components that implement a specification are models/implementations. There is an agreement that a good theory should support the following requirements:

1. *Consistency and Satisfaction.* It should be decidable whether a specification admits at least one implementation, and whether a system implements a specification.
2. *Refinement.* *Refinement* of specification expresses inclusion of sets of implementations, and therefore allows to compare richness and precision of specifications.
3. *Structural composition.* A theory should provide a combination operator on specifications, reflecting the standard composition of systems by, *e.g.* parallel product.
4. *Logical composition/conjunction.* Different aspects of systems are often specified by different teams. The issue of dealing with multiple aspects of multiple viewpoints is thus essential. It should be possible to represent several specifications (viewpoints) for the same system, and to combine them in a logical/conjunctive fashion.
5. *Incremental Design.* A theory should allow incremental design (composing / conjoining specifications in any order) and independent implementability (composable specifications can always be refined separately) [10].

For functional analysis of discrete-time non-probabilistic systems, the theory of Modal Transition Systems (MTS) [26] provides a specification formalism supporting refinement as well as conjunction and parallel composition. It has been recently applied to construct interface theories [30, 27], which are extensions of classical interface automata proposed by de Alfaro et al. [9, 12, 6].

As soon as systems include randomized algorithms, probabilistic protocols, or interact with physical environment, probabilistic models are required to reason about them. This is exacerbated by requirements for fault tolerance, when systems need to be analyzed quantitatively for the amount of failure they can tolerate, or for the delays that may appear. As Henzinger and Sifakis [17] point out, introducing probabilities into design theories allows assessing dependability of IT systems in the same manner as commonly practiced in other engineering disciplines.

Generalizing the notion of MTSs to the non-functional analysis of probabilistic systems, the formalism of Interval Markov Chains (IMCs) was introduced [23]; with notions of satisfaction and refinement generalizing probabilistic bisimulation. Informally, IMCs extend Markov Chains by labeling transitions with *intervals* of allowed probabilities rather than concrete probability values. Implementations of IMCs are Markov Chains (MCs) whose probably distributions match the constraints induced by

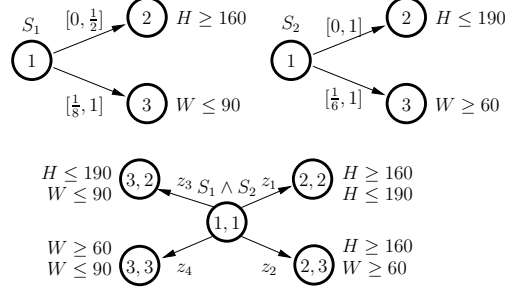


Figure 1: IMCs showing non-closure under conjunction

the intervals. IMCs is known to be an efficient model on which refinement and composition can be performed with efficient algorithms from linear algebra. Unfortunately, as we shall now see, the expressive power of IMCs is inadequate to support both logical and structural composition.

Consider two IMCs, S_1 and S_2 , in Figure 1 specifying different probability constraints related to the height H and weight W of a given person. Attempting to express the conjunction $S_1 \wedge S_2$ as an IMC by a simple intersection of bounds gives $z_1 \leq \frac{1}{2}$, $\frac{1}{6} \leq z_2 \leq \frac{1}{2}$, $\frac{1}{8} \leq z_3$ and $\frac{1}{6} \leq z_4$. However, this naive construction is too coarse: whereas $(z_1, z_2, z_3, z_4) = (\frac{1}{2}, \frac{1}{6}, \frac{1}{8}, \frac{5}{24})$ satisfies the constraints the resulting overall probability of reaching a state satisfying $H \geq 160$, i.e. $z_1 + z_2 = \frac{2}{3}$, violates the upper bound $\frac{1}{2}$ specified in S_1 . What is needed is the ability to express dependencies between the probabilities z_1, z_2, z_3, z_4 besides that of being a probability distribution ($z_1 + z_2 + z_3 + z_4 = 1$). The correct conjunctive combination is expressed by three following constraints, exceeding the expressive power of IMCs: $z_1 + z_2 \leq \frac{1}{2}$, $\frac{1}{8} \leq z_3 + z_4$, $\frac{1}{6} \leq z_2 + z_4$. A similar example shows that IMCs are also not closed under parallel composition, either.

One way to approach this problem could be to work with two types of specifications: IMCs for refinement and structural composition, and a probabilistic logic such as PCTL [16] on which a logical conjunction is naturally defined. Such a solution is clearly not satisfactory. Indeed, it is not clear how one can synthesize a MC (an implementation) that satisfies two PCTL formulas. It is also not possible to structurally compose two logical PCTL formulas.

The solution promoted in this paper is to enrich the model of IMCs. More precisely, we introduce *Constraint Markov Chains* (CMCs) as a foundation for component-based design of probabilistic systems. CMCs are a further extension of IMCs allowing rich constraints on the next-state probabilities from any state. Whereas linear constraints suffice for closure under conjunction, polynomial constraints are necessary for closure under parallel composition. We provide constructs for refinement, consistency checking, logical *and* structural composition of CMC specifications – all indispensable ingredients of a compositional design methodology.

In CMCs, each state is also labelled with a set of subsets of atomic propositions. Those propositions represent properties that should be satisfied by the implementation. The idea being that the satisfaction relation ensures that an implementation matches at least one of the subsets. This allows the specification to make additional assumptions on the behaviors of the implementation. Hence, at the level of specification, our model thus presents choices on subsets of actions. However these choices are independent from the probabilistic ones in the sense that any CMC whose states are labelled with a

set of subsets of atomic propositions can be turned to an equivalent (in terms of set of implementations) CMC whose states are labeled with a single subset of atomic propositions. There, choices between the subsets of actions disappear. It is thus not surprising that our notion of parallel composition is following the widely accepted *principle of separation of concerns*. The idea is to separate parallel composition of probability distributions from synchronization on sets of actions. This separation can be found in probabilistic specification theories that have probabilistic automata as an underlying semantic model [31, 15, 22, 20]. In fact, we show how probabilistic automata can be represented as CMCs, and how the traditional notions of parallel composition on such model can be derived in our framework with precongruence properties obtained for free. This latter result shows that CMCs capture computational structure of known models and operators, laying down a basis for studying shared properties of many probabilistic automata based languages. As already mentioned, we exemplify this by showing how precongruence properties for composition of probabilistic automata and known refinements can be obtained by reductions to CMCs.

The notions of satisfaction and strong/weak refinements for CMCs conservatively extend similar notions for IMCs [13, 23]. We characterize these relations in terms of implementation set inclusion. In particular, in the main theorem, we prove that for deterministic CMCs weak and strong refinements are complete with respect to implementation set inclusion. In addition, we provide a construction, which for any CMC S returns a deterministic CMC $\varrho(S)$ containing the models of S . Refinement relations are not complete for non-deterministic CMCs, but one can show that the weak refinement is more likely to coincide with implementation set inclusion in such a context. We show that refinement between CMCs with polynomial constraints can be decided in essentially single exponential time.

Structure of the paper. In Section 2, we introduce the concept of CMCs and a satisfaction relation with respect to Markov Chains. Consistency, refinement and conjunction are discussed in Section 3. Structural composition is introduced in Section 4. In Section 5, we introduce deterministic CMCs and show that, for this class of CMCs, strong and weak refinements coincide with inclusion of implementation sets. Section 6 discusses the class of polynomial CMCs, which is the smallest class of CMCs closed under all the compositional design operations. Section 8 concludes the paper with related and future work. Due to space constraints, some algorithms and proofs are given in an appendix.

2 Constraint Markov Chains

Let A, B be sets of propositions with $A \subseteq B$. The *restriction of $W \subseteq B$ to A* is given by $W \downarrow_A \equiv W \cap A$. If $T \subseteq 2^B$, then $T \downarrow_A \equiv \{W \downarrow_A \mid W \in T\}$. For $W \subseteq A$ define the *extension of W to B* as $W \uparrow^B \equiv \{V \subseteq B \mid V \downarrow_A = W\}$, so the set of sets whose restriction to A is W . Lift it to sets of sets as follows: if $T \subseteq 2^A$ then $T \uparrow^B \equiv \{W \subseteq B \mid W \downarrow_A \in T\}$. Let $M, \Delta \in [0, 1]^{n \times k}$ be two matrices and $x \in [0, 1]^{1 \times k}$ be a vector. We write M_{ij} for the cell in i th row and j th column of M , M_p for the p th row of M , and x_i for the i th element of x . Finally, Δ is a *correspondence matrix* iff $0 \leq \sum_{j=1}^k \Delta_{ij} \leq 1$ for all $1 \leq i \leq n$.

Definition 1 (Markov Chain) $P = \langle \{1, \dots, n\}, o, M, A, V \rangle$ is a Markov Chain if $\{1, \dots, n\}$ is a set of states containing the initial state o , A is a set of atomic propo-

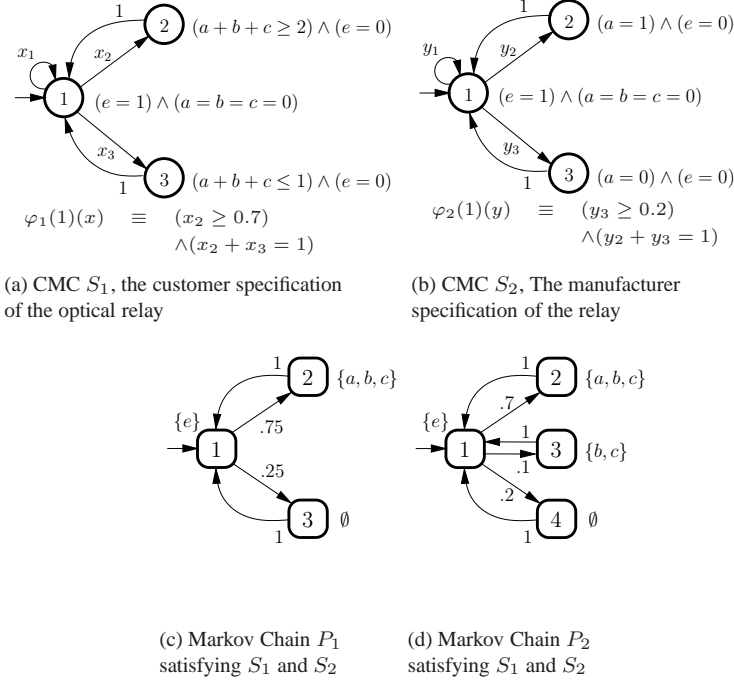


Figure 2: Two specifications (CMCs) and two implementations (MCs) of an optical relay

sitions, $V : \{1, \dots, n\} \rightarrow 2^A$ is a state valuation, and $M \in [0, 1]^{n \times n}$ is a probability transition matrix: $\sum_{j=1}^n M_{ij} = 1$ for $i = 1, \dots, n$.

We now introduce *Constraint Markov Chains* (CMCs for short), a finite representation for a possibly infinite set of MCs. Roughly speaking, CMCs generalize MCs in that, instead of specifying a concrete transition matrix, they only constrain probability values in the matrix. Constraints are modelled using a *characteristic function*, which for a given source state and a distribution of probabilities of leaving the state evaluates to 1 iff the distribution is permitted by the specification. Similarly, instead of a concrete valuation function for each state, a *constraint on valuations* is used. Here, a valuation is permitted iff it is contained in the set of admissible valuations of the specification.

Definition 2 (Constraint Markov Chain) A Constraint Markov Chain is a tuple $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$, where $\{1, \dots, k\}$ is a set of states containing the initial state o , A is a set of atomic propositions, $V : \{1, \dots, k\} \rightarrow 2^{2^A}$ is a set of admissible state valuations and $\varphi : \{1, \dots, k\} \rightarrow [0, 1]^k \rightarrow \{0, 1\}$ is a constraint function such that if $\varphi(j)(x) = 1$ then the x vector is a probability distribution: $x \in [0, 1]^k$ and $\sum_{i=1}^k x_i = 1$.

An *Interval Markov Chain* (IMC for short) [23] is a CMC whose constraint functions are represented by intervals, so for all $1 \leq i \leq k$ there exist constants α_i, β_i such that $\varphi(j)(x) = 1$ iff $\forall 1 \leq i \leq k, x_i \in [\alpha_i, \beta_i]$.

Example 1 Two parties, a customer and a vendor, are discussing a design of a relay for an optical telecommunication network. The relay is designed to amplify an optic

signal transmitted over a long distance over an optic fiber. The relay should have several modes of operation, modelled by four dynamically changing properties and specified by atomic propositions a , b , c , and e :

Atomic propositions in the optic relay specifications		
a	$\text{ber} \leq 10^{-9}$	bit error rate lower than 1 per billion bits transmitted
b	$\text{br} > 10\text{Gbits/s}$	The bit rate is higher than 10 Gbits/s.
c	$P < 10\text{W}$	Power consumption is less than 10 W.
e	Standby	The relay is not transmitting.

The customer presents CMC S_1 (Figure 2a) specifying the admissible behaviour of the relay from their point of view. States are labelled with formulas characterizing sets of valuations. For instance, " $(a + b + c \geq 2) \wedge (e = 0)$ " at state 2 of S_1 represents $V_1(2) = \{\{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}$, where a , b , c , and e range over Booleans. State 1 specifies a standby mode, where no signal is emitted and only marginal power is consumed. State 2 is the high power mode, offering a high signal/noise ratio, and hence a high bit-rate and low error rate, at the expense of a high power consumption. State 3 is the low power mode, with a low power consumption, low bit-rate and high error rate. The customer prescribes that the probability of the high power mode (state 2) is higher than 0.7. The vendor replies with CMC S_2 (Figure 2b), which represents possible relays that they can build. Because of thermal limitations, the low power mode has a probability higher than 0.2.

A state u of S is (directly) *reachable* from a state i if there exists a probability distribution $x \in [0, 1]^k$ with a nonzero probability x_u , which satisfies $\varphi(i)(x)$.

We relate CMC specifications to MCs implementing them, by extending the definition of satisfaction presented in [23] to observe the valuation constraints and the full-fledged constraint functions. Crucially, like [23], we abstract from syntactic structure of transitions—a single transition in the implementation MC can contribute to satisfaction of more than one transition in the specification, by distributing its probability mass against several transitions. Similarly many MC transitions can contribute to satisfaction of just one specification transition.

Definition 3 (Satisfaction Relation) Let $P = \langle \{1, \dots, n\}, o_P, M, A_P, V_P \rangle$ be a MC and $S = \langle \{1, \dots, k\}, o_S, \varphi, A_S, V_S \rangle$ be a CMC with $A_S \subseteq A_P$. Then $\mathcal{R} \subseteq \{1, \dots, n\} \times \{1, \dots, k\}$ is a satisfaction relation between states of P and S iff whenever $p \mathcal{R} u$ then

1. $V_P(p) \downarrow_{A_S} \in V_S(u)$, and
2. there exists a correspondence matrix $\Delta \in [0, 1]^{n \times k}$ such that
 - for all $1 \leq p' \leq n$ with $M_{pp'} \neq 0$, $\sum_{j=1}^k \Delta_{p'j} = 1$;
 - $\varphi(u)(M_p \times \Delta)$ holds, and if $\Delta_{p'u'} \neq 0$ then $p' \mathcal{R} u'$.

We write $P \models S$ iff there exists a satisfaction relation relating o_P and o_S , and call P an *implementation* of S . The set of all implementations of S is given by $\llbracket S \rrbracket \equiv \{P \mid P \models S\}$. Rows of Δ that correspond to reachable states of P always sum up to 1. This is to guarantee that the entire probability mass of implementation transitions is allocated. For unreachable states, we leave the corresponding rows in Δ unconstrained. P may have a richer alphabet than S , in order to facilitate abstract modelling: this way an implementation can maintain local information using internal variables. Algorithms to decide satisfaction are particular cases of algorithms to decide *refinement* between CMCs. See the next section.

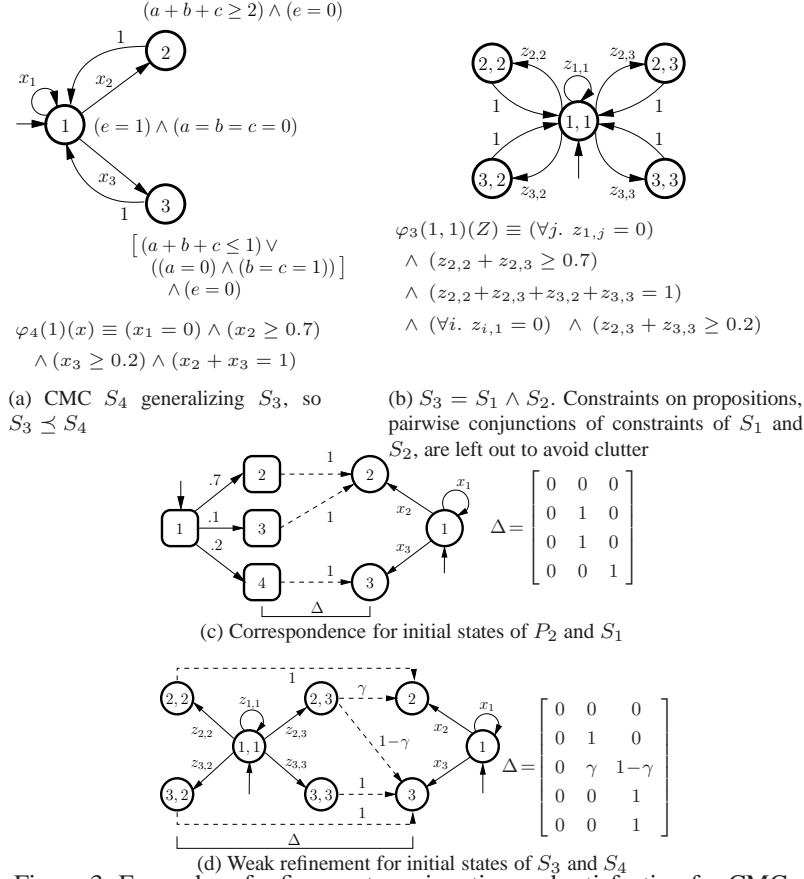


Figure 3: Examples of refinement, conjunction and satisfaction for CMCs

Example 2 We illustrate the concept of correspondence matrix between Specification S_1 (given in Figure 2a) and Implementation P_2 (given in Figure 2d). The CMC S_1 has three outgoing transitions from state 1 but, due to constraint function in 1, the transition labelled with x_1 cannot be taken (the constraint implies $x_1 = 0$). The probability mass going from state 1 to states 2 and 3 in P_2 corresponds to the probability allowed by S_1 from its state 1 to its state 2; The redistribution is done with the help of the matrix Δ given in Figure 3c. The i th column in Δ describes how big fraction of each transition probability (for transitions leaving 1) is associated with probability x_i in S_1 . Observe that the constraint function $\varphi_1(1)(0, 0.8, 0.2) = \varphi_1(1)((0, 0.7, 0.1, 0.2) \times \Delta)$ is satisfied.

CMC semantics follows the Markov Decision Process (MDP) tradition [32, 7]. The MDP semantics is typically opposed to the Uncertain Markov Chain semantics, where the probability distribution from each state is fixed a priori.

States of CMCs are labeled with set of subsets of atomic propositions. A single set of propositions represents properties that should be satisfied by the implementation. A set of sets models a choice of properties, with the idea being that the satisfaction relation ensures that an implementation matches at least one of the subsets. This allows the specification to make additional assumptions on the behaviors of the implementa-

tion. For an implementation, in each state the discrete choice of proposition set and the probabilistic choice of successor are independent.

It turns out that any CMC whose states are labelled with a set of subsets of atomic propositions can be turned into an equivalent (in terms of sets of implementations) CMC whose states are labeled with sets that contains a single subset of atomic propositions. Hence working with sets of subsets of valuations is a kind of modeling sugar that can be removed with a transformation to the *single valuation normal form*.

Definition 4 We say that a CMC is in a Single Valuation Normal Form if all its admissible valuation sets are singletons ($|V(i)| = 1$ for each $1 \leq i \leq k$).

More precisely every consistent CMC (except those that have more than one admissible valuation in the initial state) can be transformed into the normal form preserving its implementation set. A polynomial time normalization algorithm can be found in Appendix .9.

3 Consistency, Refinement and Conjunction

Consistency. A CMC S is *consistent* if it admits at least one implementation. We now discuss how to decide consistency. A state u of S is *valuation consistent* iff $V(u) \neq \emptyset$; it is *constraint consistent* iff there exists a probability distribution vector $x \in [0, 1]^{1 \times k}$ such that $\varphi(u)(x) = 1$. It is easy to see that if *each state* of S is both valuation and constraint consistent then S is also consistent. However, inconsistency of a state does not imply inconsistency of the specification. Indeed, an inconsistent state could be made unreachable by forcing the probabilities to reach it to zero. The operations presented later in this paper may introduce inconsistent states, leaving a question if a resulting CMC is consistent. In order to decide whether S is inconsistent, state inconsistencies are propagated throughout the entire state-space using a *pruning operator* β that removes inconsistent states from S . The result $\beta(S)$ is a new CMC, which may still contain some inconsistent states. The operator is applied iteratively, until a fixpoint is reached. S is consistent if the resulting CMC $\beta^*(S)$ contains at least one state. The formal definition is given in Appendix .2.

It can be shown (see Appendix .3) that pruning preserves the set of implementations.

Proposition 5 Let S be a CMC. We have that $\llbracket S \rrbracket = \llbracket \beta(S) \rrbracket$.

The fixpoint of β , and thus the entire consistency check, can be computed using a quadratic number of state consistency checks. The complexity of each check depends on the constraint language chosen.

Refinement. Comparing specifications is central to stepwise design methodologies. Systematic comparison enables simplification of specifications (abstraction) and adding details to specifications (elaboration). Usually specifications are compared using a *refinement* relation. Roughly, if S_1 refines S_2 , then any model of S_1 is also a model of S_2 .

We will now introduce two notions of refinement for CMCs that extend two well known refinements for IMCs [23, 13]. We not only generalize these refinements, but, unlike [23, 13], we also characterize them in terms of implementation set inclusion - also called *thorough refinement* - and computational complexity.

The strong refinement between IMCs, by Jonsson and Larsen [23], extends to CMCs in the following way:

Definition 6 (Strong Refinement) Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ be CMCs with $A_2 \subseteq A_1$. A relation $\mathcal{R} \subseteq \{1, \dots, k_1\} \times \{1, \dots, k_2\}$ is a strong refinement relation between states of S_1 and S_2 iff whenever $v \mathcal{R} u$ then

1. $V_1(v) \downarrow_{A_2} \subseteq V_2(u)$, and
2. there exists a correspondence matrix $\Delta \in [0, 1]^{k_1 \times k_2}$ such that for all probability distribution vectors $x \in [0, 1]^{1 \times k_1}$ if $\varphi_1(v)(x)$ holds then
 - For all $1 \leq i \leq k_1$, $x_i \neq 0 \Rightarrow \sum_{j=1}^{k_2} \Delta_{ij} = 1$;
 - $\varphi_2(u)(x \times \Delta)$ holds and
 - if $\Delta_{v'u'} \neq 0$ then $v' \mathcal{R} u'$.

We say that S_1 strongly refines S_2 iff $o_1 \mathcal{R} o_2$.

Strong refinement imposes a “fixed-in-advance” correspondence matrix regardless of the probability distribution satisfying the constraint function. In contrast, the *weak refinement*, which generalizes the one proposed in [13] for IMCs, allows choosing a different correspondence matrix for each probability distribution satisfying the constraint:

Definition 7 (Weak Refinement) Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ be CMCs with $A_2 \subseteq A_1$. The relation $\mathcal{R} \subseteq \{1, \dots, k_1\} \times \{1, \dots, k_2\}$ is a weak refinement relation iff $v \mathcal{R} u$ implies:

1. $V_1(v) \downarrow_{A_2} \subseteq V_2(u)$ and
2. for any distribution $x \in [0, 1]^{1 \times k_1}$ satisfying $\varphi_1(v)(x)$, there exists a matrix $\Delta \in [0, 1]^{k_1 \times k_2}$ such that
 - For all $1 \leq i \leq k_1$, $x_i \neq 0 \implies \sum_{j=1}^{k_2} \Delta_{ij} = 1$;
 - $\varphi_2(u)(x \times \Delta)$ holds and
 - $\Delta_{v'u'} \neq 0 \Rightarrow v' \mathcal{R} u'$.

CMC S_1 (weakly) refines S_2 , written $S_1 \preceq S_2$, iff $o_1 \mathcal{R} o_2$.

Example 3 Figure 3d illustrates a family of correspondence matrices parametrized by γ , witnessing the weak refinement between initial states of S_3 and S_4 (defined in Figures 3a–3b). The actual matrix used in proving the weak refinement depends on the probability distribution vector z that satisfies the constraint function φ_3 of state $(1, 1)$. Take $\gamma = \frac{0.7 - z_{22}}{z_{23}}$ if $z_{22} \leq 0.7$ and $\gamma = \frac{0.8 - z_{22}}{z_{23}}$ otherwise ($z_{22} \leq 0.8$ by definition). It is easy to see that $\varphi_3((1, 1))(z)$ implies $\varphi_4(1)(z \times \Delta)$.

Both weak and strong refinements imply implementation set inclusion (see Appendix .10). In Section 5, we shall see that the converse holds for a particular class of CMCs. However, this is not the case in general: strong refinement is strictly stronger than weak refinement, which is strictly stronger than implementation set inclusion. Formally, we have the following proposition.

Proposition 8 There exist CMCs S_a, S_b, S_c and S_d such that

- S_a weakly refines S_b , and S_a does not strongly refine S_b ;

- $\llbracket S_c \rrbracket \subseteq \llbracket S_d \rrbracket$, and S_c does not weakly refine S_d .

So our refinement relations for CMCs can be ordered from finest to coarsest: the strong refinement, the weak refinement, and the implementation set inclusion. As the implementation set inclusion is the *ultimate* refinement, checking finer refinements is used as a pragmatic syntax-driven, but sound, way of deciding it. Algorithms for checking strong and weak refinements are discussed in Appendix .0.1. Those algorithms are polynomial in the number of state, but the treatment of each state depends on the complexity of the constraints. Finally, let us mention that lower-bounds for the strong and weak refinement checking remain open problems.

Conjunction. *Conjunction*, also called *logical composition*, combines requirements of several specifications.

Definition 9 (Conjunction) Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ be two CMCs. The conjunction of S_1 and S_2 , written $S_1 \wedge S_2$, is the CMC $S = \langle \{1, \dots, k_1\} \times \{1, \dots, k_2\}, (o_1, o_2), \varphi, A, V \rangle$ with $A = A_1 \cup A_2$, $V((u, v)) = V_1(u) \uparrow^A \cap V_2(v) \uparrow^A$, and

$$\begin{aligned} \varphi((u, v))(x_{1,1}, x_{1,2}, \dots, x_{2,1}, \dots, x_{k_1, k_2}) &\equiv \\ \varphi_1(u)(\sum_{j=1}^{k_2} x_{1,j}, \dots, \sum_{j=1}^{k_2} x_{k_1,j}) &\wedge \\ \varphi_2(v)(\sum_{i=1}^{k_1} x_{i,1}, \dots, \sum_{i=1}^{k_1} x_{i,k_2}). & \end{aligned}$$

Conjunction may introduce inconsistent states and thus its use should normally be followed by applying the pruning operator β^* . As already stated in the introduction, the result of conjoining two IMCs is not an IMC in general, but a CMC whose constraint functions are systems of linear inequalities. Figure 3b depicts a CMC S_3 expressing the conjunction of IMCs S_1 and S_2 (see Figures 2a–2b). The constraint $z_{2,3} + z_{3,3} \geq 0.2$ in state $(1, 1)$ cannot be expressed as an interval.

As expected, conjunction of two specifications coincides with their greatest lower bound with respect to the weak refinement (also called *shared refinement*).

Theorem 10 Let S_1 , S_2 and S_3 be three CMCs. We have (a) $((S_1 \wedge S_2) \preceq S_1)$ and $((S_1 \wedge S_2) \preceq S_2)$ and (b) if $(S_3 \preceq S_1)$ and $(S_3 \preceq S_2)$, then $S_3 \preceq (S_1 \wedge S_2)$.

In fact, as follows from the later results of Section 5, the set of implementations of a conjunction of two *deterministic* specifications S_1 and S_2 coincides with the intersection of implementation sets of S_1 and S_2 (the greatest lower bound in the lattice of implementation sets).

4 Compositional Reasoning using the Principle of Separation of Concerns

Let us now turn to *structural* composition. In our theory, as we already said in the introduction and after presenting CMCs, choices regarding the set of valuations and stochastic choices are independent from each others. This property of the model naturally leads to a definition of the parallel composition operator based on the principle of *separation of concerns*. The idea is that probabilistic behaviours are composed separately from the synchronization of the sets of state valuations. This allows realizing probabilistic composition as a simple product of independent distributions.

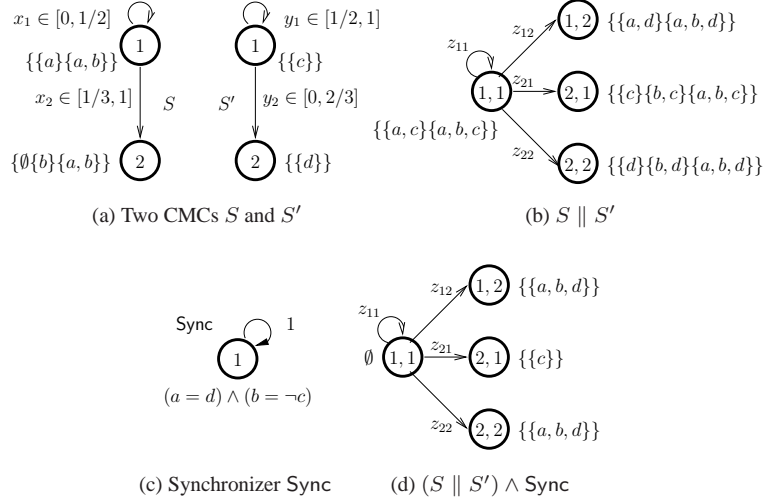


Figure 4: Parallel composition and synchronization of CMCs

Remark 1 *The principle of separation of concerns is intensively used in the definition of parallel composition for many systems that mix stochastic and non-deterministic choices. Among them, one can cite many theories for probabilistic process algebra [31, 22]. Similar principles are also applied for continuous time stochastic models, in a slightly different setting based on CTMCs [20]. In Section 7, we shall see that our structural composition covers the one of probabilistic automata.*

Following the separation of concerns principle, components are composed first into a product (or effectively just a vector of independent entities), and then synchronized by constraining their behaviour. This design is both simple and expressive: it allows applying diverse synchronization mechanisms, beyond just matching inputs to outputs. Moreover it elegantly exploits the prior knowledge on logical composition, as the synchronization operator turns out to be realizable using conjunction.

We start by discussing how systems and specifications can be composed in a non-synchronizing way, then we introduce a notion of synchronization. The non-synchronizing *independent* composition is largely just a product of two MCs (or CMCs).

Definition 11 (Parallel Composition of MCs) Let $P_1 = \langle \{1, \dots, n_1\}, o_1, M', A_1, V_1 \rangle$ and $P_2 = \langle \{1, \dots, n_2\}, o_2, M'', A_2, V_2 \rangle$ be two MCs with $A_1 \cap A_2 = \emptyset$. The parallel composition of P_1 and P_2 is the MC $P_1 \parallel P_2 = \langle \{1, \dots, n_1\} \times \{1, \dots, n_2\}, (o_1, o_2), M, A_1 \cup A_2, V \rangle$ where: $M \in [0, 1]^{(n_1 \times n_2) \times (n_1 \times n_2)}$ is such that $M_{(p,q)(r,s)} = M'_{pr} \cdot M''_{qs}$, and $V((p, q)) = V_1(p) \cup V_2(q)$.

This definition extends to the general case of CMCs.

Definition 12 (Parallel Composition of CMCs) Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ be CMCs with $A_1 \cap A_2 = \emptyset$. The parallel composition of S_1 and S_2 is the CMC $S_1 \parallel S_2 = \langle \{1, \dots, k_1\} \times \{1, \dots, k_2\}, (o_1, o_2), \varphi, A_1 \cup A_2, V \rangle$, where $\varphi((u, v))(z_{1,1}, z_{1,2}, \dots, z_{2,1}, \dots, z_{k_1, k_2}) = \exists x_1, \dots, x_{k_1}, y_1, \dots, y_{k_2} \in [0, 1]$ such that

$\forall (i, j) \in \{1, \dots, k_1\} \times \{1, \dots, k_2\}$ we have $z_{i,j} = x_i \cdot y_j$ and $\varphi_1(u)(x_1, \dots, x_{k_1}) = \varphi_2(v)(y_1, \dots, y_{k_2}) = 1$. Finally, $V((u, v)) = \{Q_1 \cup Q_2 \mid Q_1 \in V_1(u), Q_2 \in V_2(v)\}$.

It is worth mentioning that IMCs are not closed under composition. Consider IMCs S and S' given in Figure 4a and their composition $S \parallel S'$ given in Figure 4b. Assume first that $S \parallel S'$ is an IMC. As a variable $z_{i,j}$ is the product of two variables x_i and y_j , if $S \parallel S'$ is an IMC, then one can show that the interval for $z_{i,j}$ is obtained by computing the products of the bounds of the intervals over which x_i and y_j range. Hence, we can show that $z_{11} \in [0, 1/2]$, $z_{12} \in [0, 1/3]$, $z_{21} \in [1/6, 1]$, $z_{22} \in [0, 2/3]$. Let $[a, b]$ be the interval for the constraint $z_{i,j}$, it is easy to see that there exist implementations I_1 of S_1 and I_2 of S_2 such that $I_1 \parallel I_2$ satisfies the constraint $z_{i,j} = a$ (resp. $z_{i,j} = b$). However, while each bound of each interval can be satisfied independently, some points in the polytope defined by the intervals and the constraint $\sum z_{i,j} = 1$ cannot be reached. As an example, consider $z_{11} = 0$, $z_{12} = 1/3$, $z_{21} = 1/3$, $z_{22} = 1/3$. It is clearly inside the polytope, but one cannot find an implementation I of $S \parallel S'$ satisfying the constraints given by the parallel composition. Indeed, having $z_{11} = 0$ implies that $x_1 = 0$ and thus that $z_{12} = 0$.

Theorem 13 *If S'_1, S'_2, S_1, S_2 are CMCs then $S'_1 \preceq S_1$ and $S'_2 \preceq S_2$ implies $S'_1 \parallel S'_2 \preceq S_1 \parallel S_2$, so the weak refinement is a precongruence with respect to parallel composition. Consequently, for any MCs P_1 and P_2 we have that $P_1 \models S_1 \wedge P_2 \models S_2$ implies $P_1 \parallel P_2 \models S_1 \parallel S_2$.*

As alphabets of composed CMCs have to be disjoint, the composition does not synchronize the components on state valuations like it is typically done for other (non-probabilistic) models. However, synchronization can be introduced by conjoining the composition with a *synchronizer*—a single-state CMC whose valuation function relates the atomic propositions of the composed CMCs.

Example 4 *CMC $S \parallel S'$ of Figure 4b is synchronized with the synchronizer Sync given in Figure 4c. Sync removes from $S \parallel S'$ all the valuations that do not satisfy $(a = d) \wedge (b = \neg c)$. The result is given in Figure 4d. Observe that an inconsistency appears in State $(1, 1)$. Indeed, there is no implementation of the two CMCs that can synchronize in the prescribed way. In general inconsistencies like this one can be uncovered by applying the pruning operator, which would return an empty specification. So synchronizers enable discovery of incompatibilities between component specifications in the same way as it is known for non-probabilistic specification models.*

Synchronization is associative with respect to composition, which means that the order of synchronization and composition is inessential for final functionality of the system.

Theorem 14 *Let S_1, S_2 and S_3 be three CMCs with pairwise disjoint sets of propositions A_1, A_2 and A_3 . Let Sync_{123} be a synchronizer over $A_1 \cup A_2 \cup A_3$ and let Sync_{12} be the same synchronizer with its set of propositions restricted to $A_1 \cup A_2$. The following holds $\llbracket ((S_1 \parallel S_2) \wedge \text{Sync}_{12}) \parallel S_3 \wedge \text{Sync}_{123} \rrbracket = \llbracket (S_1 \parallel S_2 \parallel S_3) \wedge \text{Sync}_{123} \rrbracket$.*

Finally, synchronized composition also supports component-based refinement in the style of Theorem 13:

Theorem 15 *If S'_1, S'_2, S_1, S_2 are CMCs, Sync is a synchronizer and $S'_1 \preceq S_1 \wedge S'_2 \preceq S_2$ then $(S'_1 \parallel S'_2) \wedge \text{Sync} \preceq (S_1 \parallel S_2) \wedge \text{Sync}$.*

Consequently, a modeller can continue independent refinement of specifications under synchronization, knowing that the original synchronized specification will not be violated.

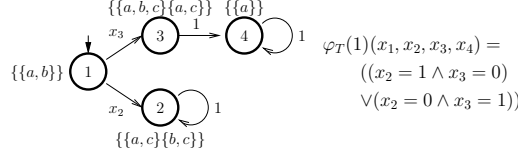


Figure 5: A CMC T whose set of implementations cannot be represented with a deterministic CMC

5 Deterministic CMCs

Clearly, if all implementations of a specification S_1 also implement a specification S_2 , then the former is a proper strengthening of the latter. Indeed, S_1 specifies implementations that break no assumptions that can be made about implementations of S_2 . Thus implementation set inclusion is a desirable refinement for specifications. Unfortunately, this problem is still open, and, as we have said, the weak and the strong refinement soundly approximate it. Had that approximation been complete, we would have had an effective decision procedure for implementation set inclusion. In this section, we argue that this indeed is the case for an important subclass of specifications: *deterministic CMCs*. A CMC S is *deterministic* iff for every state i , states reachable from i have pairwise disjoint admissible valuations:

Definition 16 *Let $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$ be a CMC. S is deterministic iff for all states $i, u, v \in \{1, \dots, k\}$, if there exists $x \in [0, 1]^k$ such that $(\varphi(i)(x) \wedge (x_u \neq 0))$ and $y \in [0, 1]^k$ such that $(\varphi(i)(y) \wedge (y_v \neq 0))$, then we have that $V(u) \cap V(v) = \emptyset$.*

In Figures 2a and 2b, both S_1 and S_2 are deterministic specifications. In particular states 2 and 3, reachable from 1 in both CMCs, have disjoint constraints on valuations. On the other hand, the CMC T given in Figure 5 is non-deterministic. Indeed, for States 2 and 3, which can both be reached from State 1, we have that $V_T(2) \cap V_T(3) = \{\{a, c\}\} \neq \emptyset$.

Deterministic CMCs are less expressive than non-deterministic ones, in the sense that the same implementation sets cannot sometimes be expressed. Consider again the CMC T given in Figure 5. It is such that its set of implementations cannot be represented by a deterministic CMC. Indeed, any merging of States 2 and 3 in T would result in a CMC that accepts models where one can loop on valuation $\{a, c\}$ and then accept valuation $\{a\}$ with probability 1. Such a model cannot be accepted by T .

Proposition 17 *Conjunction and composition preserve determinism.*

In Appendix .0.2, we present a determinization algorithm that can be applied to any CMC S whose initial state is a single valuation set. The result of the algorithm is a new CMC weakly refined by S . Consequently the implementation set of the result includes the one of S (see Appendix .8). This character of determinization resembles the known determinization algorithms for modal transition systems [3].

We now state one of the main theorems of the paper: the weak refinement is complete with respect to implementation set inclusion for deterministic CMCs:

Theorem 18 *Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ be two consistent single valuation normal form deterministic CMCs with $A_2 \subseteq A_1$. We have $\llbracket S_1 \rrbracket \subseteq \llbracket S_2 \rrbracket \Rightarrow S_1 \preceq S_2$.*

Proof:

We present a sketch of the proof and refer to Appendix .11 for details. We construct the refinement relation by relating all pairs of states of S_1 and S_2 for which implementation inclusion holds. Let $\mathcal{R} \subseteq \{1, \dots, k_1\} \times \{1, \dots, k_2\}$ such that $v \mathcal{R} u$ iff for all MC I and state p of I we have $p \models v \Rightarrow p \models u$. As we consider pruned CMCs, there exist implementations for all states. Then the usual, albeit complex and long in this case, coinductive proof technique is applied, showing that this relation is indeed a weak refinement relation. The crucial point of the argument lies in proving the closure property — i.e. that if a S_1 state u advances possibly to u' , then indeed the corresponding state v of S_2 can also advance to v' and the (u', v') pair is in \mathcal{R} . In other words that implementation inclusion of predecessors implies the implementation inclusion of successors. This is proven in an ad absurdum argument. Roughly, assume that there would exist an implementation I' of u' which is not an implementation of v' . Then one can construct an implementation I'' of u which evolves as I' . This implementation would not implement v' but could implement some other state of S_2 . This case is ruled out by requiring determinism and a normal form of S_2 . Then the only way for I'' to evolve is to satisfy v' which contradicts the assumption that I' is not an implementation of v' .

Since any consistent CMC with a single valuation in initial state can be normalized, Theorem 18 holds even if S_1 and S_2 are not in single valuation normal form. Thus, weak refinement and the implementation set inclusion coincide on the class of deterministic CMCs with at most single valuation in the initial state. Finally, Theorem 18 also holds for strong refinement. Indeed, the following theorem states that weak and strong refinements coincide on the class of deterministic CMCs.

Theorem 19 *Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A, V_2 \rangle$ be two deterministic CMCs in normal form. If there exists a weak refinement relation \mathcal{R} such that $S_1 \mathcal{R} S_2$, then \mathcal{R} is also a strong refinement relation.*

Finally, the above results on completeness for deterministic specifications carry over to refinements of [23] and [13], which are special cases of our refinements. Completeness properties for these refinements were open problems until now.

Discussion: A weaker Definition of Determinism. Our notion of determinism may look too strong. Indeed, it assumes that, from a given state i , one cannot reach two states u and v that share common sets of valuations. The assumption is made independently of the distributions used to reach the two states, i.e., it may be the case that there exists no distribution in where both u and v can be reached simultaneously. A natural way to solve the problem would be to consider a weaker version of determinism. More precisely, we say that a CMC $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$ is weakly deterministic if whenever there exists $x \in [0, 1]^k$ and states i, u, v such that $\varphi(i)(x)$ and $x_u > 0$ and $x_v > 0$, we have $V(u) \cap V(v) = \emptyset$. This version of determinism is weaker than the one given in Definition 16. Indeed, only states that can be reached by the same distribution should have disjoint sets of valuations. Though this notion seems reasonable, one can show (see Appendix .4 for a proof) that there exist two weakly deterministic CMCs S_c and S_d such that S_c thoroughly but not weakly refines S_d . Hence working with this weaker, but natural, version of determinism does not close the gap between weak and thorough refinements.

6 Polynomial CMCs

It is not surprising that CMCs are closed under both logical and structural compositions. Indeed, CMCs do not make any assumptions on constraint functions. There are however many classes of constraints that are practically intractable. While this paper is mainly concerned with the development of the theoretical foundations for CMCs, we now briefly study classes of CMCs for which operations on constraints required by our algorithms can be managed quite efficiently.

A first candidate could be linear constraints, which is the obvious generalization of interval constraints. Unfortunately, linear constraints CMCs are not closed under structural composition. Indeed, as we have seen in Section 4 the composition of two linear constraints leads to a polynomial constraint. However, what is more interesting is that polynomial constraints *are* closed under both logical and structural composition and that these operations do not increase the quantifier alternations since they only introduce existential quantifiers. Hence, one can claim that CMCs with polynomial constraints and only existential quantifiers are certainly the smallest extension of IMCs closed under all operations.

From the algorithmic point of view, working with polynomial constraints should not be seen as an obstacle. First, we observe that algorithms for logical and structural composition do not require any complex operations on polynomials. The refinements algorithms (presented in the appendix) are polynomial in the number of states, and each iteration requires a quantifier elimination. This procedure is known to be double exponential in general, but there exist efficient single exponential algorithms [4, 5] when quantifier alternations are fixed. Those algorithms are implemented in Maple [33]. The pruning operation is polynomial in the number of states, but each iteration also requires an exponential treatment as one has to decide whether the constraints have at least a solution. Again, such problem can be solved with efficient algorithms. Finally, determining a CMC can be performed with a procedure that is similar to the determinization procedure for finite-state automata. Such a procedure is naturally exponential in the number of states.

7 Discussion of Refinement and Composition

CMCs are a newcomer in a long series of probabilistic modeling languages and abstractions for them. Throughout the paper we have indicated that many of our results directly translate to simpler abstractions, like IMCs. We shall now further discuss this foundational aspect of CMCs, showing how they subsume a few established notions of refinement and composition for probabilistic automata (and for process algebra based on them).

Below we write $\text{Dist}(S)$ for the set of all probability distributions over a finite set S . Given two sets S and T and a probability distribution $\alpha \in \text{Dist}(S \times T)$, we denote the marginal distribution over S as $\alpha_{s,T} = \sum_{t \in T} \alpha_{s,t}$, and similarly for T . We say that φ is a *non-deterministic distribution constraint* over set I if all solutions x of φ are point distributions; so $\exists i. x_i = 1$. Write $[\overset{i}{S}]$ to denote a particular point distribution for which $[\overset{i}{S}]_i = 1$. Notice that non-deterministic distribution constraints model a non-deterministic choice of an element from S . They will be used to encode non-determinism in CMCs.

A probabilistic automaton (PA for short) [31] is a tuple $\mathbb{S} = (S, \text{Act}, \rightarrow, s_1)$, where S is a finite set of states, $\rightarrow \subseteq S \times \text{Act} \times \text{Dist}(S)$ is a finite transition relation and

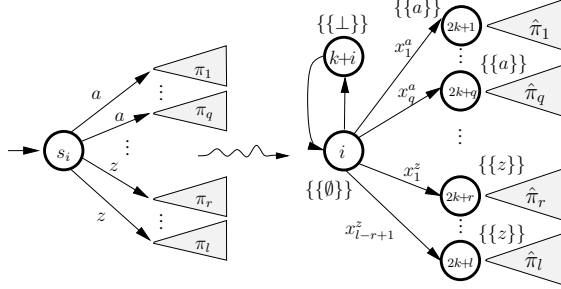


Figure 6: Reducing a PA to CMC. There $\hat{\pi}$ denotes a distribution constraint, which has a unique solution π .

$s_1 \in S$ is the initial state. The *derived combined transition relation* of \mathbb{S} is given by $\rightarrow_c \in S \times \text{Act} \times \text{Dist}(S)$. If $\pi \in \text{Dist}(S)$ and $\varrho \in \text{Dist}(T)$ then $\pi \otimes \varrho$ denotes the unique independent product distribution such that $(\pi \otimes \varrho)_{s,t} = \pi_s \cdot \varrho_t$.

We say that $t \xrightarrow{a}_c \varrho$ iff ϱ is a convex linear combination of vectors from $\varrho = \{\varrho_i \mid t \xrightarrow{a} \varrho_i\}$, so $\varrho = \varrho \times \lambda$, where λ is a distribution vector $\lambda \in [0, 1]^{|e|}$. We interpret ϱ as a matrix, where i th column is a distribution ϱ_i .

Consider two PA $\mathbb{S} = (S, \text{Act}, \rightarrow^S, s_0)$ and $\mathbb{T} = (T, \text{Act}, \rightarrow^T, t_0)$. For a binary relation $R \subseteq S \times T$ we define a derived relation $R^* \subseteq \text{Dist}(S) \times \text{Dist}(T)$ such that $\pi R^* \varrho$ iff there exists a distribution $\alpha \in \text{Dist}(S \times T)$ and (1) $\alpha_{q,T} = \pi_q$ for all $q \in S$, (2) $\alpha_{S,r} = \varrho_r$ for all $r \in T$ and (3) $\alpha_{s,t} \neq 0$ implies sRt .

Definition 20 (Simulation [31]) A relation $R \subseteq S \times T$ is a simulation iff $(s, t) \in R$ implies that whenever $s \xrightarrow{a} \pi$ for a distribution π , then $t \xrightarrow{a} \varrho$ for distribution ϱ such that $\pi R^* \varrho$.

R is a probabilistic simulation iff $(s, t) \in R$ implies that if $s \xrightarrow{a} \pi$ then $t \xrightarrow{a}_c \varrho$ for some distribution ϱ , and $\pi R^* \varrho$.

Let $A \subseteq \text{Act}$ be the subset of actions on which \mathbb{S} and \mathbb{T} should synchronize. The *parallel composition* of \mathbb{S} and \mathbb{T} is a PA $\mathbb{S} \parallel \mathbb{T} = (S \times T, \text{Act}, \rightarrow, (s_0, t_0))$, where \rightarrow is the largest transition relation such that $(s, t) \xrightarrow{a} \pi \otimes \varrho$ if:

- $a \in A$ and $s \xrightarrow{a}^S \pi$ and $t \xrightarrow{a}^T \varrho$, or
- $a \notin A$ and $s \xrightarrow{a}^S \pi$ and $\varrho = \begin{bmatrix} t \\ T \end{bmatrix}$, or
- $a \notin A$ and $\pi = \begin{bmatrix} s \\ S \end{bmatrix}$ and $t \xrightarrow{a}^T \varrho$.

We now propose a linear encoding of PAs into CMCs, which reduces simulation and composition of PAs to refinement and composition of CMCs (see Fig. 6). Let $\mathbb{S} = (\{s_1, \dots, s_k\}, \text{Act}, \rightarrow, s_0)$ be a PA. And let l be the number of reachable action-distribution pairs, so $\Omega_{\mathbb{S}} = \{(a_1, \pi_1), \dots, (a_l, \pi_l)\} = \{(a, \pi) \mid \exists s \in S. s \xrightarrow{a} \pi\}$. The corresponding CMC is $\hat{\mathbb{S}} = (\{1, \dots, 2k+l\}, 1, \hat{\varphi}, \text{Act} \cup \perp, \hat{V})$, where $\perp \notin \text{Act}$. $\hat{\mathbb{S}}$ has three kinds of states. Type-1 states, $1 \dots k$, correspond directly to states of \mathbb{S} . Distributions leaving these states model a non-deterministic choice. Type-2 states, $k+1, \dots, 2k$, model a possibility that a component remains idle in a state. Type-3 states, $2k+1, \dots, 2k+l$ model the actual distributions of \mathbb{S} .

\hat{V} assigns value $\{\emptyset\}$ to type-1 states and value $\{\{\perp\}\}$ to type-2 states. For type-3: $\hat{V}(2k+i') = \{\{a_{i'}\}\}$ for $1 \leq i' \leq l$. The distribution constraints are as follows:

$\widehat{\varphi}(i)(x)$ if i is type-1 and

$$x = [_{1..2k+l}^{k+i}] \text{ or } s_i \xrightarrow{a_{i'}} \pi_{i'} \wedge x = [_{1..2k+l}^{2k+i'}] \text{ for } 1 \leq i' \leq l.$$

$\widehat{\varphi}(k+i)(x)$ if $k+i$ is type-2 and $x = [_{1..2k+l}^i]$.

$\widehat{\varphi}(2k+i')(x)$ if $2k+i'$ is type-3 and $x = \pi_{i'}$

We can now relate simulation of PA to refinement of CMCs:

Theorem 21 \mathbb{T} simulates \mathbb{S} iff $\widehat{\mathbb{S}}$ strongly refines $\widehat{\mathbb{T}}$.

Another, very similar, but slightly more complicated, encoding exists, for which weak refinement coincides with *probabilistic* simulation. See Appendix .14 for details.

The same encoding is used to characterize parallel composition of PAs using parallel composition of CMCs.

Theorem 22 For two PAs \mathbb{S} and \mathbb{T} over the same set of synchronizing actions Act and a set $A \subseteq Act$ we have that $\widehat{\mathbb{S}} \parallel \widehat{\mathbb{T}}$ is isomorphic to

$$((\widehat{\mathbb{S}} \parallel \widehat{\mathbb{T}}[a'/a]_{a \in Act}) \wedge \mathbf{S}_A) [a/(a, a'); a/(a, \perp'); a/(\perp, a')]_{a \in Act}$$

where \mathbf{S}_A is a synchronizer over $Act_{\perp} \times Act'_{\perp}$, defined by

$$(\forall a \in A. a \iff a') \wedge (\forall a \notin A. (a \implies \perp') \wedge (a' \implies \perp))$$

Expression $S[a'_1/a_1; \dots; a'_n/a_n]_{a_1, \dots, a_n \in Act}$ denotes a substitution, substituting a primed version of name a_i for each occurrence in a_i , for all actions in Act .

Interestingly, the precongruence property for the parallel composition of PAs is obtained for free as a corollary of the above two reduction theorems and Thm. 13. Similarly, we obtain precongruence with probabilistic simulation using a suitable encoding—a good example how CMCs can be used to study properties of simpler languages in a generic way.

8 Related Work and Concluding Remarks

We have presented CMCs—a new model for representing a possibly infinite family of MCs. Unlike the previous attempts [23, 13], our model is closed under many design operations, including composition and conjunction. We have studied these operations as well as several classical compositional reasoning properties, showing that, among others, the CMC specification theory is equipped with a complete refinement relation (for deterministic specifications), which naturally interacts with parallel composition, synchronization and conjunction. We have also demonstrated how our framework can be used to obtain properties for less expressive languages, by using reductions.

Two recent contributions [13, 25] are related to our work. Fecher et al. [13] propose a model checking procedure for PCTL [8] and Interval Markov Chains (other procedures recently appear in [7, 14]), which is based on weak refinement. However, our objective is not to use CMCs within a model checking procedure for probabilistic systems, but rather as a specification theory.

Very recently Katoen and coauthors [25] have extended Fecher's work to *Interactive* Markov Chains, a model for performance evaluation [18, 21]. Their abstraction uses the continuous time version of IMCs [24] augmented with may and must transitions, very much in the spirit of [26]. Parallel composition is defined and studied for this abstraction, however conjunction has been studied neither in [13] nor in [25].

Over the years process algebraic frameworks have been proposed for describing and analyzing probabilistic systems based on Markov Chains (MCs) and Markov Decision Processes [20, 1, 29]. Also a variety of probabilistic logics have been developed for expressing properties of such systems, e.g., PCTL [16]. Both traditions support refinement between specifications using various notions of probabilistic simulation [13, 23] and, respectively, logical entailment [19]. Whereas the process algebraic approach favors structural composition (parallel composition), the logical approach favors logical composition (conjunction). Neither of the two supports *both* structural and logical composition.

As a future work, it would be of interest to design, implement and evaluate efficient algorithms for procedures outlined in this paper. We will also study the decidability of the set inclusion problem. We would also like to define a quotient relation for CMCs, presumably building on results presented in [28]. The quotienting operation is of particular importance for component reuse. One could also investigate applicability of our approach in model checking procedures, in the same style as Fecher and coauthors have used IMCs for model checking PCTL [13]. Finally, it would be interesting to extend our composition operation by considering products of dependent probability distributions in the spirit of [11].

Acknowledgements

This work was supported by the European STREP-COMBEST project no. 215543, by VKR Centre of Excellence MT-LAB, and by an “Action de Recherche Collaborative” ARC (TP)I.

References

- [1] S. Andova. Process algebra with probabilistic choice. In *ARTS*, volume 1601 of *LNCS*. Springer, 1999.
- [2] S. Basu. New results on quantifier elimination over real closed fields and applications to constraint databases. *Journal of the ACM*, 46(4):537–555, July 1999.
- [3] N. Benes, J. Kretinsky, K. G. Larsen, and J. Srba. On determinism in modal transition systems. To appear in *TCS*.
- [4] C. W. Brown. Simple CAD construction and its applications. *Journal of Symbolic Computation*, 31(5), 2001.
- [5] C. W. Brown and J. H. Davenport. The complexity of quantifier elimination and cylindrical algebraic decomposition. In *SSAC*, pages 54–60, Waterloo, ON, Canada, 2007.
- [6] A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and F. Y. C. Mang. Synchronous and bidirectional component interfaces. In *CAV*, volume 2404 of *LNCS*, pages 414–427. Springer, 2002.
- [7] K. Chatterjee, K. Sen, and T. A. Henzinger. Model-checking omega-regular properties of interval Markov chains. In *FoSSaCS*, volume 4962 of *LNCS*. Springer, 2008.

- [8] F. Ciesinski and M. Größer. On probabilistic computation tree logic. In *VSS*, volume 2925 of *LNCS*. Springer, 2004.
- [9] L. de Alfaro and T. A. Henzinger. Interface automata. In *FSE*, pages 109–120. ACM Press, 2001.
- [10] L. de Alfaro and T. A. Henzinger. Interface-based design. In *Engineering Theories of Software-intensive Systems*, volume 195 of *NATO Science Series: Mathematics, Physics, and Chemistry*, pages 83–104. Springer, 2005.
- [11] L. de Alfaro, T. A. Henzinger, and R. Jhala. Compositional methods for probabilistic systems. In *CONCUR*, volume 2154 of *lncs*, pages 351–365. Springer, 2001.
- [12] L. Doyen, T. A. Henzinger, B. Jobstmann, and T. Petrov. Interface theories with component reuse. In *EMSOFT*, pages 79–88. ACM Press, 2008.
- [13] H. Fecher, M. Leucker, and V. Wolf. Don’t Know in probabilistic systems. In *SPIN*, volume 3925 of *LNCS*, 2006.
- [14] S. Haddad and N. Pekergin. Using stochastic comparison for efficient model checking of uncertain Markov chains. In *QEST*, pages 177–186. IEEE Computer Society Press, 2009.
- [15] H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *IEEE Real-Time Systems Symposium*, pages 278–287, 1990.
- [16] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Asp. Comput.*, 6(5), 1994.
- [17] T. A. Henzinger and J. Sifakis. The embedded systems design challenge. In *FM*, volume 4085 of *lncs*, pages 1–15. Springer, 2006.
- [18] H. Hermans, U. Herzog, and J. Katoen. Process algebra for performance evaluation. *Theor. Comput. Sci.*, 274(1-2):43–87, 2002.
- [19] H. Hermans, B. Wachter, and L. Zhang. Probabilistic CEGAR. In *CAV*, volume 5123 of *LNCS*. Springer, 2008.
- [20] H. Hermans. *Interactive Markov Chains*. Springer, 2002.
- [21] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
- [22] B. Jonsson, K. Larsen, and W. Yi. Probabilistic extensions of process algebras. In *Handbook of Process Algebra*, pages 681–710. Elsevier, 2001.
- [23] B. Jonsson and K. G. Larsen. Specification and refinement of probabilistic processes. In *LICS*. IEEE Computer, 1991.
- [24] J. Katoen, D. Klink, M. Leucker, and V. Wolf. Three-valued abstraction for continuous-time Markov chains. In *CAV*, volume 4590 of *LNCS*, pages 311–324. Springer, 2007.

-
- [25] J. Katoen, D. Klink, and M. R. Neuhäuser. Compositional abstraction for stochastic systems. In *FORMATS*, volume 5813 of *LNCS*, pages 195–211. Springer, 2009.
 - [26] K. G. Larsen. Modal specifications. In *AVMS*, volume 407 of *LNCS*, pages 232–246. Springer, 1989.
 - [27] K. G. Larsen, U. Nyman, and A. Wasowski. Modal I/O automata for interface and product line theories. In *ESOP*, *LNCS*, pages 64–79. Springer, 2007.
 - [28] K. G. Larsen and A. Skou. Compositional verification of probabilistic processes. In *CONCUR*, volume 630 of *LNCS*, pages 456–471. Springer, 1992.
 - [29] N. López and M. Núñez. An overview of probabilistic process algebras and their equivalences. In *VSS*, volume 2925 of *LNCS*, pages 89–123. Springer, 2004.
 - [30] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay, and R. Passerone. Modal interfaces: Unifying interface automata and modal cifications. In *EMSOFT*, 2009.
 - [31] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In *CONCUR*, volume 836 of *LNCS*, pages 481–496. springer, 1994.
 - [32] K. Sen, M. Viswanathan, and G. Agha. Model-checking Markov chains in the presence of uncertainties. In *TACAS*, volume 3920 of *LNCS*, pages 394–410. Springer, 2006.
 - [33] H. Yanami and H. Anai. SyNRAC: a Maple toolbox for solving real algebraic constraints. *ACM Communications in Computer Algebra*, 41(3):112–113, September 2007.

The following appendix contains proofs of the most essential claims. It also contains algorithms that had to be removed from the main text of the paper. The appendix is to be reviewed at the discretion of the programme committee.

.0.1 An Algorithm for Checking Refinement

We now briefly discuss algorithms for checking weak and strong refinements between two CMCs $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ with $k_1, k_2 \leq n$. Checking whether a relation $\mathcal{R} \subseteq \{1, \dots, k_1\} \times \{1, \dots, k_2\}$ is a strong (resp. weak) refinement relation reduces to checking, for all $(i, j) \in \mathcal{R}$, the validity of the following *refinement formulas*: $\exists \Delta, \forall x, \varphi_1(i)(x) \Rightarrow \varphi_2(j)(x \times \Delta) \wedge \bigwedge_{i'} (\sum_{j'} \Delta_{i'j'} = 1) \wedge \bigwedge_{i', j'} (i' \mathcal{R} j' \vee \Delta_{i'j'} = 0)$ for the strong refinement, and $\forall x, \varphi_1(i)(x) \Rightarrow \exists \Delta, \varphi_2(j)(x \times \Delta) \wedge \bigwedge_{i'} (\sum_{j'} \Delta_{i'j'} = 1) \wedge \bigwedge_{i', j'} (i' \mathcal{R} j' \vee \Delta_{i'j'} = 0)$ for the weak refinement. Strong and weak refinements can be decided by iterated strengthening of \mathcal{R} with refinement formulas, starting from $\mathcal{R}_0 = \{(i, j) \mid V_1(i) \downarrow_{A_2} \subseteq V_2(j)\}$, until either $(o_1, o_2) \notin \mathcal{R}$, in which case S_1 does not strongly (resp. weakly) refine S_2 , or \mathcal{R} is found to be a strong (resp. weak) refinement.

The exact complexity of the algorithm depends on the type of constraints that are used in the specifications. As an example, consider that all the constraints in S_1 and S_2 are polynomial of degree d with less than k bound variables – we shall see that polynomial constraints is the least class under which CMCs are closed. There, deciding refinement formulas can be done by quantifier elimination. When the number of quantifier alternations is constant, the *cylindrical algebraic decomposition algorithm* [4, 5], implemented in Maple [33], performs this quantifier elimination in time double exponential in the number of variables. Consequently, refinement can be checked in $O(n^2 2^{2^{n^2}})$ time.

However, considering constraints φ contain only existential quantifiers, quantifier alternation is either one or two for strong refinement and exactly one for weak refinement. There are quantifier elimination algorithms that have a worst case complexity single exponential only in the number of variables, although they are double exponential in the number of quantifier alternations [2]. Thanks to these algorithms, deciding whether \mathcal{R} is a strong (resp. weak) refinement relation can be done in time single exponential in the number of states n and k , the number of bound variables appearing in the constraints: $O(n^2 s^{P(n,k)} d^{P(n,k)})$ where P is a polynomial.

.0.2 A Determinisation Algorithm

The determinization algorithm in Section 5 is presented in the following definition.

Definition 23 *Let $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$ be a consistent CMC in the single valuation normal form. Let $m < k$ and $h : \{1, \dots, k\} \rightarrow \{1, \dots, m\}$ be a surjection such that (1) $\{1, \dots, k\} = \cup_{v \in \{1, \dots, m\}} h^{-1}(v)$ and (2) for all $1 \leq i \neq j \leq k$, if there exists $1 \leq u \leq k$ and $x, y \in [0, 1]^k$ such that $(\varphi(u)(x) \wedge x_i \neq 0)$ and $(\varphi(u)(y) \wedge y_j \neq 0)$, then $(h(i) = h(j) \iff V(i) = V(j))$; otherwise $h(i) \neq h(j)$. A deterministic CMC for S is the CMC $\varrho(S) = \langle \{1, \dots, m\}, o', \varphi', A, V' \rangle$ where $o' = h(o)$, $\forall 1 \leq i \leq k$, $V'(h(i)) = V(i)$, and for each $1 \leq i \leq m$,*

$$\begin{aligned} \varphi'(i)(y_1, \dots, y_m) &= \exists x_1, \dots, x_k, \\ \bigvee_{u \in h^{-1}(i)} [& (\forall 1 \leq j \leq m, y_j = \sum_{v \in h^{-1}(j)} x_v) \\ & \wedge \varphi(u)(x_1, \dots, x_k)]. \end{aligned}$$

.1 Correspondence matrices

Definition 24 Define the following operations:

1. If $\Delta \in [0, 1]^{k \times q}$ and $\Delta' \in [0, 1]^{k \times r}$ are two correspondence matrices, we define $\Delta'' = \Delta \otimes \Delta'$ by $\Delta'' \in [0, 1]^{k \times (q \cdot r)}$ and $\Delta''_{i(j,n)} = \Delta_{ij} \cdot \Delta'_{in}$;
2. If $\Delta \in [0, 1]^{k \times q}$ and $\Delta' \in [0, 1]^{r \times s}$ are two correspondence matrices, we define $\Delta'' = \Delta \odot \Delta'$ by $\Delta'' \in [0, 1]^{(k \cdot r) \times (q \cdot s)}$ and $\Delta''_{(i,j)(n,p)} = \Delta_{in} \cdot \Delta'_{jp}$.

Lemma 25 1. Let $\Delta \in [0, 1]^{k \times q}$ and $\Delta' \in [0, 1]^{q \times r}$ be two correspondence matrices. The matrix $\Delta'' = \Delta \times \Delta'$ is a correspondence matrix;

2. Let $\Delta \in [0, 1]^{k \times q}$ and $\Delta' \in [0, 1]^{k \times r}$ be two correspondence matrices. The matrix $\Delta'' = \Delta \otimes \Delta'$ is a correspondence matrix;
3. Let $\Delta \in [0, 1]^{k \times q}$ and $\Delta' \in [0, 1]^{r \times s}$ be two correspondence matrices. The matrix $\Delta'' = \Delta \odot \Delta'$ is a correspondence matrix;

Proof:

1. Let $1 \leq i \leq k$ and $1 \leq j \leq r$. We have $\Delta''_{ij} = \sum_{n=1}^q \Delta_{in} \cdot \Delta'_{nj}$. Thus,

$$\begin{aligned} \sum_{j=1}^r \Delta''_{ij} &= \sum_{j=1}^r \sum_{n=1}^q \Delta_{in} \cdot \Delta'_{nj} = \sum_{n=1}^q \sum_{j=1}^r \Delta_{in} \cdot \Delta'_{nj} \\ &= \sum_{n=1}^q \Delta_{in} \cdot \left(\sum_{j=1}^r \Delta'_{nj} \right) \\ &= \sum_{n=1}^q \Delta_{in} \cdot 1 \leq 1. \end{aligned}$$

2. Let $1 \leq i \leq k$ and $(j, n) \in \{1, \dots, q\} \times \{1, \dots, r\}$. We have $\Delta''_{i(j,n)} = \Delta_{ij} \cdot \Delta'_{in}$. Thus,

$$\begin{aligned} \sum_{(j,n) \in \{1, \dots, q\} \times \{1, \dots, r\}} \Delta''_{i(j,n)} &= \sum_{j=1}^q \sum_{n=1}^r \Delta_{ij} \cdot \Delta'_{in} \\ &= \sum_{j=1}^q \Delta_{ij} \cdot \sum_{n=1}^r \Delta'_{in} \\ &= \sum_{j=1}^q \Delta_{ij} \cdot 1 \leq 1. \end{aligned}$$

3. Let $(i, j) \in \{1, \dots, k\} \times \{1, \dots, r\}$ and $(n, p) \in \{1, \dots, q\} \times \{1, \dots, s\}$. We have $\Delta''_{(i,j)(n,p)} = \Delta_{in} \cdot \Delta'_{jp}$. Thus,

$$\begin{aligned}
\sum_{(n,p) \in \{1, \dots, q\} \times \{1, \dots, s\}} \Delta''_{(i,j)(n,p)} &= \sum_{n=1}^q \sum_{p=1}^s \Delta_{in} \cdot \Delta'_{jp} \\
&= \left(\sum_{n=1}^q \Delta_{in} \right) \cdot \left(\sum_{p=1}^s \Delta'_{jp} \right) \\
&\leq 1.
\end{aligned}$$

.2 Formal Definition of the Pruning Operator

We define β formally. Let $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$.

- If o is locally inconsistent then let $\beta(S) = \emptyset$.
- If S does not contain locally inconsistent states then $\beta(S) = S$.
- Else proceed in two steps. First for $k' < k$ define a function $\nu : \{1, \dots, k\} \rightarrow \{\perp, 1, \dots, k'\}$, which will remove inconsistent states. All inconsistent states are mapped to \perp . For all $1 \leq i \leq k$ take $\nu(i) = \perp$ iff $[(V(i) = \emptyset) \vee (\forall x \in [0, 1]^k, \varphi(i)(x) = 0)]$. All remaining states are mapped injectively into $\{1, \dots, k'\}$: $\nu(i) \neq \perp \implies \forall j \neq i, \nu(j) \neq \nu(i)$. Then let $\beta(S) = \langle \{1, \dots, k'\}, \nu(o), \varphi', A, V' \rangle$, where $V'(i) = V(\nu^{-1}(i))$ and for all $1 \leq j \leq k'$ the constraint $\varphi'(j)(y_1, \dots, y_{k'})$ is: $\exists x_1, \dots, x_k$ such that

$$\begin{aligned}
[\nu(i) = \perp \implies x_i = 0] \wedge [\forall 1 \leq l \leq k' : y_l = x_{\nu^{-1}(l)}] \\
\wedge [\varphi(\nu^{-1}(j))(x_1, \dots, x_k)]
\end{aligned}$$

The constraint makes the inconsistent states unreachable, and then \perp is dropped as a state.

.3 Pruning Preserves Implementations

On page 9 we claim that that pruning preserves the set of implementations. Below we formalize and prove that claim.

Theorem 26 *Let $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$ be a CMC and $\beta^*(S) = \lim_{n \rightarrow \infty} \beta^n(S)$ be the fixpoint of β . For any MC P , we have (1) $P \models S \iff P \models \beta(S)$ and (2) $\llbracket S \rrbracket = \llbracket \beta^*(S) \rrbracket$.*

Proof:

Let $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$ be a CMC (with at least an inconsistent state) and $P = \langle \{1, \dots, n\}, o_P, M, A_P, V_P \rangle$ be a MC. Let $S' = \langle \{1, \dots, k'\}, o', \varphi', A, V' \rangle = \beta(S)$. If $\beta(S)$ is empty, then both S and $\beta(S)$ are inconsistent.

Consider a function ν for removing inconsistent states (one exists because there are inconsistent states), such that $k' < k$ and for all $1 \leq i \leq k$, $\nu(i) = \perp \iff [(V(i) = \emptyset) \vee (\forall x \in [0, 1]^k, \neg \varphi(i)(x))]$ and $\nu(i) \neq \perp \implies \forall j \neq i, \nu(j) \neq \nu(i)$. We first prove that $P \models S \iff P \models \beta(S)$.

\implies Suppose that $P \models S$. Then there exists a satisfaction relation \mathcal{R} such that $o_P \mathcal{R} o$. Define the relation $\mathcal{R}' \subseteq \{1, \dots, n\} \times \{1, \dots, k'\}$ such that $p \mathcal{R}' v$

iff there exists $u \in \{1, \dots, k\}$ such that $p \mathcal{R} u$ and $\nu(u) = v$. It is clear that $o_P \mathcal{R}' o'$. We prove that \mathcal{R}' is a satisfaction relation. Let p, u, v such that $p \mathcal{R} u$ and $\nu(u) = v$.

- As $\nu(u) \neq \perp$, we have by definition that $V'(v) = V(u)$, thus $V_P(p) \downarrow_A \in V'(v)$.
- Let $\Delta \in [0, 1]^{n \times k}$ be the correspondence matrix witnessing $p \mathcal{R} u$. Let $\Delta' \in [0, 1]^{n \times k'}$ such that $\Delta'_{qv} = \Delta_{q\nu^{-1}(v)}$. It is clear that Δ' is a correspondence matrix. We first show that

$$\forall u' \in \{1, \dots, k\}, (\nu(u') = \perp) \Rightarrow (\forall q \in \{1, \dots, n\}, \Delta_{qu'} = 0). \quad (1)$$

Let $u' \in \{1, \dots, k\}$ such that $\nu(u') = \perp$, and suppose that there exists $q \in \{1, \dots, n\}$, $\Delta_{qu'} \neq 0$. As Δ is a correspondence matrix, we have $q \mathcal{R} u'$. Thus $V_P(q) \downarrow_A \in V(u')$, which means that $V(u') \neq \emptyset$, and there exists Δ'' such that $\varphi(u')(M_q \times \Delta'')$. Thus, there exists $x \in [0, 1]^{1 \times k}$ such that $\varphi(u')(x)$. As a consequence, we cannot have $\nu(u') = \perp$, which is a contradiction, thus (1).

We now prove that \mathcal{R}' satisfies the axioms of a satisfaction relation.

1. Let $p' \in \{1, \dots, n\}$ such that $M_{pp'} \neq 0$. This implies, by definition, that $\sum_{j=1}^k \Delta_{p'j} = 1$. We have $\sum_{j=1}^{k'} \Delta'_{p'j} = \sum_{r \in \{1, \dots, k\} \mid \nu(r) \neq \perp} \Delta_{p'r}$. By (1), $\sum_{r \in \{1, \dots, k\} \mid \nu(r) \neq \perp} \Delta_{p'r} = \sum_{r=1}^k \Delta_{p'r} = 1$.
2. Let $y = M_p \times \Delta' \in [0, 1]^{1 \times k'}$ and $x = M_p \times \Delta \in [0, 1]^{1 \times k}$. We know that $\varphi(u)(x)$ holds. Moreover, by (1), if $\nu(q) = \perp$, then $x_q = 0$, and for all $l \in \{1, \dots, k'\}$, $y_l = x_{\nu^{-1}(l)}$. Clearly, this implies that $\varphi'(v)(M_p \times \Delta')$ holds.
3. Let $p', v' \in \{1, \dots, n\} \times \{1, \dots, k'\}$ such that $\Delta'_{p'v'} \neq 0$. We have $\Delta'_{p'v'} = \Delta_{p'\nu^{-1}(v')} \neq 0$, thus there exists $u' \in \{1, \dots, k\}$ such that $p' \mathcal{R} u'$ and $\nu(u') = v'$. Finally $p' \mathcal{R}' v'$.

Finally, \mathcal{R}' is a satisfaction relation such that $o_P \mathcal{R}' o'$, thus $P \models \beta(S)$.

\Leftarrow Conversely, the reasoning is the same, except that we now build Δ from Δ' saying that $\Delta_{qv} = 0$ if $\nu(v) = \perp$ and $\Delta_{qv} = \Delta'_{q\nu(v)}$ otherwise.

We have proved that β is implementations-conservative, thus the fixpoint of β verifies the same property.

4 Proof of Proposition 8

In section 3, we claim that there is a strict ordering between the different refinement relations in the general case. We now give constructions that prove this fact:

There exist CMCs S_a, S_b, S_c and S_d such that

- S_a weakly refines S_b , and S_a does not strongly refine S_b ;
- $\llbracket S_c \rrbracket \subseteq \llbracket S_d \rrbracket$, and S_c does not weakly refine S_d .

$$\Delta_x = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \gamma & (1-\gamma) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Delta = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & a & (1-a) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 7: Correspondence matrices for $S_a \preceq S_b$

Proof:

- Consider the CMCs S_a and S_b given in Figures 8a and 8b respectively. Call X_a (resp. X_b) State X in S_a (resp. S_b). We first show that there exists a weak refinement relation \mathcal{R} such that $S_a \preceq S_b$, with $1_a \mathcal{R} 1_b$. We then show that there exists no strong refinement relation between S_a and S_b .

1. Let $\mathcal{R} = \{(1_a, 1_b), (2_a, 2_b), (3_a, 3_b), (3_a, 4_b), (4_a, 5_b)\}$. We show that \mathcal{R} is a weak refinement relation. We first focus on building the correspondence matrix for the couple $(1_a, 1_b)$. Let x be a “valid” valuation of the outgoing transitions of 1_a . Let $\gamma = \frac{0.7-x_2}{x_3}$ if $x_2 \leq 0.7$ and $\frac{0.8-x_2}{x_3}$ otherwise. As x satisfies $\varphi_a(1_a)$, we have $0 \leq \gamma \leq 1$. Consider the correspondence matrix Δ_x given in Figure 7

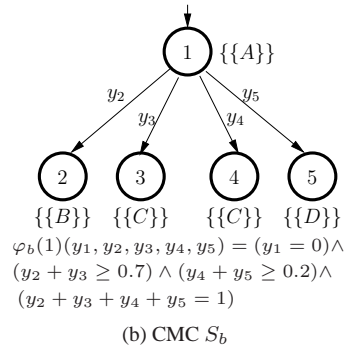
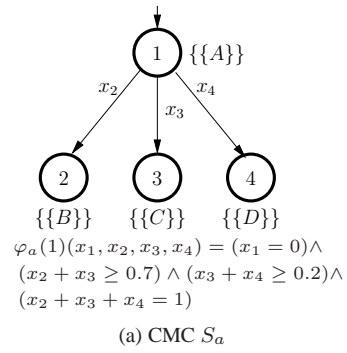
It is easy to see that for all valuation x satisfying $\varphi_a(1_a)$, $\varphi_b(1_b)(x \times \Delta_x)$ also holds. The correspondence matrices for the other pairs in \mathcal{R} are trivial since there are no outgoing transitions from those states. Thus \mathcal{R} is a weak refinement relation between S_a and S_b .

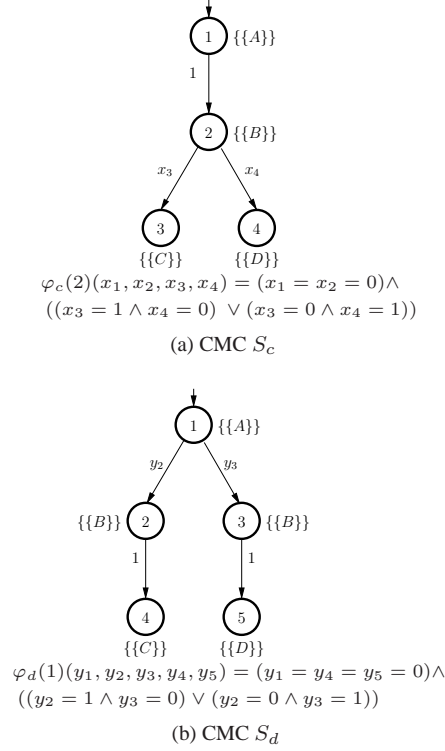
2. Suppose that there exists a strong refinement relation \mathcal{R}' such that $1_a \mathcal{R}' 1_b$. Let Δ be the correspondence matrix associated to $1_a \mathcal{R}' 1_b$. Since $2_a, 3_a$ and 4_a can all be reached from 1_a with an admissible transition, the sum of the elements in the corresponding rows in Δ must be one. From the valuations of the states, we obtain that Δ is of the type given in Figure 7, with $a \geq 0$.

Moreover, if \mathcal{R}' is a strong refinement relation, then we have that for all valuation x satisfying $\varphi_a(1_a)$, $\varphi_b(1_b)(x \times \Delta)$ also holds.

Let $x^1 = (0, 0.6, 0.1, 0.3)$ and $x^2 = (0, 0.8, 0.1, 0.1)$. Both x^1 and x^2 satisfy $\varphi_a(1)$. If there exists a strong refinement, this implies that $\varphi_b(1)(x^1 \times \Delta)$ and $\varphi_b(1)(x^2 \times \Delta)$ also hold. However, $\varphi_b(1)(x^1 \times \Delta) = 1$ implies that $a \geq 1$ and $\varphi_b(1)(x^2 \times \Delta)$ implies that $a \leq 0$.

It is thus impossible to find a unique correspondence matrix working for all the “valid” valuations of the outgoing transitions of 1_a . As a consequence, there cannot exist a strong refinement relation \mathcal{R}' such that $1_a \mathcal{R}' 1_b$.

Figure 8: CMCs S_a and S_b

Figure 9: CMCs S_c and S_d

- Consider the CMCs S_c and S_d given in Figures 9a and 9b. It is easy to see that S_c and S_d share the same set of implementations. However, due to the constraints, State 2 of S_c cannot refine any state of S_d . As a consequence, S_c cannot refine S_d .

5 Proof of Theorem 10

Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$, $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ and $S_3 = \langle \{1, \dots, k_3\}, o_3, \varphi_3, A_3, V_3 \rangle$ be three CMCs. We want to prove that

1. $((S_1 \wedge S_2) \preceq S_1) \wedge ((S_1 \wedge S_2) \preceq S_2)$;
2. $(S_3 \preceq S_1) \wedge (S_3 \preceq S_2) \Rightarrow S_3 \preceq (S_1 \wedge S_2)$.

Proof:

We separately prove the two items of the theorem.

1. Let $S_1 \wedge S_2 = S = \langle \{1, \dots, k_1\} \times \{1, \dots, k_2\}, o, \varphi, A, V \rangle$.

Let $\mathcal{R} \subseteq (\{1, \dots, k_1\} \times \{1, \dots, k_2\}) \times \{1, \dots, k_1\}$ such that $(u, v) \mathcal{R} w \iff u = w$. We will prove that \mathcal{R} is a **strong** refinement relation. Let $u \in \{1, \dots, k_1\}$

and $v \in \{1, \dots, k_2\}$. We have $(u, v) \mathcal{R} u$. By definition of S , we also have $V((u, v)) \downarrow_{A_1} = (V_1(u) \uparrow^A \cap V_2(v) \uparrow^A) \downarrow_{A_1} \subseteq V_1(u)$.

Let $\Delta \in [0, 1]^{k_1 \cdot k_2 \times k_1}$ such that $\Delta_{(i,j),i} = 1$ and $\Delta_{(i,j),k} = 0$ if $k \neq i$. By definition, we have $\forall(i, j), \sum_{k=1}^{k_1} \Delta_{(i,j),k} = 1$. As a consequence, Δ is correspondence matrix. We now prove that it satisfies the axioms of a satisfaction relation for $(u, v) \mathcal{R} u$.

- (a) If $x \in [0, 1]^{1 \times k_1 \cdot k_2}$ is such that $\varphi((u, v))(x)$, it implies by definition that $\varphi_1(u)(\sum_{j=1}^{k_2} x_{1,j}, \dots, \sum_{j=1}^{k_2} x_{k_1,j}) = \varphi_1(u)(x \times \Delta)$ holds.
- (b) If $\Delta_{(u',v'),w'} \neq 0$, we have by definition $u' = w'$ and $(u', v') \mathcal{R} u'$.

From (a) and (b), we conclude that \mathcal{R} is a **strong** refinement relation.

Since $(o_1, o_2) \mathcal{R} o_1$, we have $S_1 \wedge S_2 \preceq S_1$. By symmetry, we also have $S_1 \wedge S_2 \preceq S_2$.

2. Suppose that $S_3 \preceq S_1$ and $S_3 \preceq S_2$. By definition, there exist two refinement relations $\mathcal{R}_1 \subseteq \{1, \dots, k_3\} \times \{1, \dots, k_1\}$ and $\mathcal{R}_2 \subseteq \{1, \dots, k_3\} \times \{1, \dots, k_2\}$ such that $o_3 \mathcal{R}_1 o_1$ and $o_3 \mathcal{R}_2 o_2$. Let $S_1 \wedge S_2 = S = \langle \{1, \dots, k_1\} \times \{1, \dots, k_2\}, o, \varphi, A, V \rangle$.

Let $\mathcal{R} \subseteq \{1, \dots, k_3\} \times (\{1, \dots, k_1\} \times \{1, \dots, k_2\})$ such that $u \mathcal{R}(v, w) \iff u \mathcal{R}_1 v$ and $u \mathcal{R}_2 w$. We now prove that \mathcal{R} is a weak refinement relation.

Consider u, v, w such that $u \mathcal{R}(v, w)$.

- (a) By definition, we have $V_3(u) \downarrow_{A_1} \subseteq V_1(v)$ and $V_3(u) \downarrow_{A_2} \subseteq V_2(w)$. As a consequence, $V_3(u) \downarrow_A \subseteq V((v, w))$.
- (b) Let $x \in [0, 1]^{1 \times k_3}$ such that $\varphi_3(u)(x)$. Consider the correspondence matrices $\Delta \in [0, 1]^{k_3 \times k_1}$ and $\Delta' \in [0, 1]^{k_3 \times k_2}$ given by $u \mathcal{R}_1 v$ and $u \mathcal{R}_2 w$ for the transition vector x . Let $\Delta'' \in [0, 1]^{k_3 \times k_1 \cdot k_2} = \Delta \otimes \Delta'$. By Lemma 25, Δ'' is a correspondence matrix. We now prove that it satisfies the axioms of a refinement relation for $u \mathcal{R}(v, w)$.

- i. Let $1 \leq i \leq k_3$ such that $x_i \neq 0$. By definition of Δ and Δ' , we have $\sum_{j=1}^{k_1} \Delta_{ij} = 1$ and $\sum_{q=1}^{k_2} \Delta'_{iq} = 1$. By construction,

$$\sum_{(j,q) \in \{1, \dots, k_1\} \times \{1, \dots, k_2\}} \Delta''_{i(j,q)} = (\sum_{j=1}^{k_1} \Delta_{ij}) \cdot (\sum_{q=1}^{k_2} \Delta'_{iq}) = 1.$$

- ii. By definition of Δ and Δ' , both $\varphi_1(v)(x \times \Delta)$ and $\varphi_2(w)(x \times \Delta')$ hold.

Let $x' = x \times \Delta''$. It is clear that $x \times \Delta = (\sum_{j=1}^{k_2} x'_{1,j}, \dots, \sum_{j=1}^{k_2} x'_{k_1,j})$

and $x \times \Delta' = (\sum_{i=1}^{k_1} x'_{i,1}, \dots, \sum_{i=1}^{k_1} x'_{i,k_2})$. As a consequence, $\varphi((v, w))(x \times \Delta'')$ holds.

- iii. Let u', v', w' such that $\Delta''_{u'(v',w')} \neq 0$. By construction, this implies $\Delta_{u'v'} \neq 0$ and $\Delta'_{u'w'} \neq 0$. As a consequence, $u' \mathcal{R}_1 v'$ and $u' \mathcal{R}_2 w'$, thus $u' \mathcal{R}(v', w')$.

From (i) - (iii), we conclude that \mathcal{R} is a weak refinement relation. Since $o_3 \mathcal{R}(o_1, o_2)$, we have $S_3 \preceq (S_1 \wedge S_2)$.

.6 Proof of Theorem 13

Let $S'_1 = \langle \{1, \dots, k'_1\}, o'_1, \varphi'_1, A'_1, V'_1 \rangle$, $S'_2 = \langle \{1, \dots, k'_2\}, o'_2, \varphi'_2, A'_2, V'_2 \rangle$, $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$, $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ be four CMCs. Suppose $S'_1 \preceq S_1 \wedge S'_2 \preceq S_2$. We prove that $S'_1 \parallel S'_2 \preceq S_1 \parallel S_2$.

Proof:

Let $S = \langle \{1, \dots, k_1\} \times \{1, \dots, k_2\}, (o_1, o_2), \varphi, A, V \rangle = S_1 \parallel S_2$ and $S' = \langle \{1, \dots, k'_1\} \times \{1, \dots, k'_2\}, (o'_1, o'_2), \varphi', A', V' \rangle = S'_1 \parallel S'_2$.

By definition, there exist two weak refinement relations \mathcal{R}_1 and \mathcal{R}_2 such that $o'_1 \mathcal{R}_1 o_1$ and $o'_2 \mathcal{R}_2 o_2$. Define \mathcal{R} such that $(u', v') \mathcal{R} (u, v) \iff u' \mathcal{R}_1 u$ and $v' \mathcal{R}_2 v$. Consider now such (u', v') and (u, v) . We prove that \mathcal{R} satisfies the axioms of a refinement relation between (u', v') and (u, v) .

1. We have $(V'((u', v')))\downarrow_A = \{Q \subseteq 2^{A'} \mid \exists Q_1 \in V'_1(u'), Q_2 \in V'_2(v'), Q = Q_1 \cup Q_2\}\downarrow_A = \{Q \subseteq 2^A \mid \exists Q_1 \in V_1(u'), Q_2 \in V_2(v'), Q = Q_1 \downarrow_{A_1} \cup Q_2 \downarrow_{A_2}\}$. Thus $(V'((u', v')))\downarrow_A \subseteq V((u, v))$.
2. Let $z' \in [0, 1]^{1 \times k'_1 \cdot k'_2}$ such that $\varphi'(u', v')(z')$. We now build the correspondence matrix Δ witnessing $(u', v') \mathcal{R} (u, v)$. Consider the correspondence matrices Δ_1 and Δ_2 given by $u' \mathcal{R}_1 u$ and $v' \mathcal{R}_2 v$ for the transition vector z' . Define $\Delta = \Delta_1 \odot \Delta_2 \in [0, 1]^{k'_1 \cdot k'_2 \times k_1 \cdot k_2}$. By Lemma 25, Δ is a correspondence matrix. Moreover, since $\varphi'(u', v')(z')$ holds, there exists $x' \in [0, 1]^{1 \times k'_1}$ and $y' \in [0, 1]^{1 \times k'_2}$ such that $\forall i, j, z'_{(i,j)} = x'_i \cdot y'_j$ and $\varphi'_1(u')(x')$ and $\varphi'_2(v')(y')$.

- (a) Let $(u'', v'') \in \{1, \dots, k'_1\} \times \{1, \dots, k'_2\}$ such that $z_{(u'', v'')} \neq 0$. By definition of x' and y' , this implies that $x'_{u''} \neq 0$ and $y'_{v''} \neq 0$. Thus $\sum_{j=1}^{k_1} \Delta_{1u''j} = 1$ and $\sum_{j=1}^{k_2} \Delta_{2v''j} = 1$.

$$\begin{aligned} & \sum_{(r,s) \in \{1, \dots, k_1\} \times \{1, \dots, k_2\}} \Delta_{(u'', v'')(r,s)} = \\ & \sum_{(r,s) \in \{1, \dots, k_1\} \times \{1, \dots, k_2\}} \Delta_{1u''r} \cdot \Delta_{2v''s} \\ & = \sum_{r=1}^{k_1} \sum_{s=1}^{k_2} \Delta_{1u''r} \cdot \Delta_{2v''s} \\ & = \left(\sum_{r=1}^{k_1} \Delta_{1u''r} \right) \cdot \left(\sum_{s=1}^{k_2} \Delta_{2v''s} \right) = 1. \end{aligned}$$

- (b) Let $z = z' \times \Delta \in [0, 1]^{1 \times k_1 \cdot k_2}$. Remark that $z = (x' \times \Delta_1) \otimes (y' \times \Delta_2)$.

Let $x = x' \times \Delta_1$ and $y = y' \times \Delta_2$. Since $u' \mathcal{R}_1 u$ and $v' \mathcal{R}_2 v$, we have $\varphi_1(u)(x)$ and $\varphi_2(v)(y)$. Thus $\varphi(u, v)(z \times \Delta)$.

- (c) Let u'', v'', u''', v''' such that $\Delta_{(u'', v'')(u''', v''')} \neq 0$. By definition, it implies that $\Delta_{1u''u'''} \neq 0$ and $\Delta_{2v''v'''} \neq 0$, and as a consequence $(u'', v'') \mathcal{R} (u''', v''')$.

From (a),(b),(c), we conclude that \mathcal{R} is a weak refinement relation. Since $(o'_1, o'_2) \mathcal{R} (o_1, o_2)$, we have $S' \preceq S$.

The proof of the second part of the theorem is similar, and left to the reader.

7 Proof of Theorem 14

Let S_1, S_2 and S_3 be three CMCs with disjoint sets of atomic propositions A_1, A_2 and A_3 . Let $\text{Sync}_{123} = \langle \{1\}, 1, \lambda x.x = 1, A_1 \cup A_2 \cup A_3, V_{\text{Sync}} \rangle$ be a synchronizer between A_1, A_2 and A_3 . Consider $\text{Sync}_{12} = \langle \{1\}, 1, \lambda x.x = 1, A_1 \cup A_2, V_{\text{Sync}} \downarrow_{A_1 \cup A_2} \rangle$. We want to prove that $\llbracket ((S_1 \parallel S_2) \wedge \text{Sync}_{12}) \parallel S_3 \wedge \text{Sync}_{123} \rrbracket = \llbracket [S_1 \parallel S_2 \parallel S_3] \wedge \text{Sync}_{123} \rrbracket$.

Proof:

We first prove the following statement. Let S_1 and S_2 be two CMCs with disjoint sets of atomic propositions A_1 and A_2 . Let Sync_1 be a synchronizing vector on A_1 . We have $(S_1 \parallel S_2) \wedge \text{Sync}_1 = (S_1 \wedge \text{Sync}_1) \parallel S_2$.

First, remember that synchronizers are single state CMCs, with a single transition taken with probability 1. As a consequence, computing the conjunction with a synchronizer preserves the structure of any CMC. The only change lies in the sets of valuations.

Let p be a state of S_1 and q be a state of S_2 . We have $(V_1(p) \cup V_2(q)) \cap V_{\text{Sync}_1} \uparrow^{A_1 \cup A_2} = (V_1(p) \cap V_{\text{Sync}_1}) \cup V_2(q)$. As a consequence, the valuations of $(S_1 \wedge \text{Sync}_1) \parallel S_2$ are the same as the valuations of $(S_1 \parallel S_2) \wedge \text{Sync}_1$.

By monotony of conjunction, we have $(S_1 \parallel S_2) \wedge \text{Sync}_{12} \preceq (S_1 \parallel S_2)$. By Theorem 13, it implies that $\llbracket ((S_1 \parallel S_2) \wedge \text{Sync}_{12}) \parallel S_3 \wedge \text{Sync}_{123} \rrbracket \preceq \llbracket [S_1 \parallel S_2 \parallel S_3] \wedge \text{Sync}_{123} \rrbracket$, and finally $\llbracket ((S_1 \parallel S_2) \wedge \text{Sync}_{12}) \parallel S_3 \wedge \text{Sync}_{123} \rrbracket \subseteq \llbracket [S_1 \parallel S_2 \parallel S_3] \wedge \text{Sync}_{123} \rrbracket$.

We now prove that $\llbracket [S_1 \parallel S_2 \parallel S_3] \wedge \text{Sync}_{123} \rrbracket \preceq \llbracket ((S_1 \parallel S_2) \wedge \text{Sync}_{12}) \parallel S_3 \wedge \text{Sync}_{123} \rrbracket$. By monotony of conjunction, we have $\llbracket [S_1 \parallel S_2 \parallel S_3] \wedge \text{Sync}_{123} \rrbracket \preceq \llbracket [S_1 \parallel S_2 \parallel S_3] \wedge \text{Sync}_{12} \wedge \text{Sync}_{123} \rrbracket$. Moreover, by the statement proved above, we have $\llbracket [S_1 \parallel S_2 \parallel S_3] \wedge \text{Sync}_{12} \rrbracket \preceq \llbracket ((S_1 \parallel S_2) \wedge \text{Sync}_{12}) \parallel S_3 \rrbracket$. As a consequence, we have $\llbracket [S_1 \parallel S_2 \parallel S_3] \wedge \text{Sync}_{123} \rrbracket \preceq \llbracket ((S_1 \parallel S_2) \wedge \text{Sync}_{12}) \parallel S_3 \wedge \text{Sync}_{123} \rrbracket$, and thus $\llbracket [S_1 \parallel S_2 \parallel S_3] \wedge \text{Sync}_{123} \rrbracket \subseteq \llbracket ((S_1 \parallel S_2) \wedge \text{Sync}_{12}) \parallel S_3 \wedge \text{Sync}_{123} \rrbracket$.

8 Determinization is Weakening

On page 14 we claimed that the determinization algorithm increases the set of the implementations of the transformed CMC. Below comes the proof of this fact.

Theorem 27 *Let S be a CMC in single valuation normal form, we have $S \preceq \varrho(S)$.*

Proof:

Let $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$ be a CMC in single valuation normal form. Let $\varrho(S) = \langle \{1, \dots, m\}, o', \varphi', A, V' \rangle$ be a determinization of S and $h : \{1, \dots, k\} \rightarrow \{1, \dots, m\}$ the associated projection.

Define $\mathcal{R} \subseteq \{1, \dots, k\} \times \{1, \dots, m\}$ such that $u \mathcal{R} v \iff h(u) = v$. We will show that \mathcal{R} is a strong refinement relation. Let u, v such that $u \mathcal{R} v$.

1. *By definition, we have $h(u) = v$, thus $V'(v) = V(u)$.*
2. *Let $\Delta \in [0, 1]^{k \times m}$ such that $\Delta_{i,j} = 1$ if $h(i) = j$ and 0 else. Δ is clearly a correspondence matrix.*

- (a) Let $x \in [0, 1]^k$ such that $\varphi(u)(x)$. For all $1 \leq j \leq m$, we have $y_j = \sum_{i \in h^{-1}(j)} x_i$ and $\varphi(u)(x)$, thus $\varphi'(v)(x \times \Delta)$. Moreover, for all $1 \leq i \leq k$, $\sum_{j=1}^m \Delta_{i,j} = 1$ by construction.
- (b) If $\Delta_{u',v'} \neq 0$, then $h(u') = v'$ and thus $u' \mathcal{R} v'$.

Finally, \mathcal{R} is a strong refinement relation and $o \mathcal{R} o'$, thus S strongly refines $\varrho(S)$. As strong refinement implies weak refinement, we also have $S \preceq \varrho(S)$.

9 Normalization

The normalization algorithm basically separates each state u with m possible valuations into m states u_1, \dots, u_m , each with a single admissible valuation. Then the constraint function is adjusted, by substituting sums of probabilities going to the new states in place of the old probabilities targeting u . Finally, a mutual exclusion constraint is added so that it is not allowed to have positive probability of reaching more than one of u_i states from the same source state. The transformation is local and syntax based. It can be performed in polynomial time and it only increases the size of the CMC polynomially. We will write $\mathcal{N}(S)$ for a result of normalization of S .

Definition 28 (Normalization) Let $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$ be a CMC. If there exists a function $\mathcal{N} : \{1, \dots, k\} \rightarrow 2^{\{1, \dots, m\}}$ such that

1. $\{1, \dots, m\} = \cup_{i \in \{1, \dots, k\}} \mathcal{N}(i)$;
2. For all $1 \leq i \neq j \leq k$, $\mathcal{N}(i) \cap \mathcal{N}(j) = \emptyset$;
3. $\forall 1 \leq i \leq k$, $|\mathcal{N}(i)| = |V(i)|$;

If, moreover, $|V(o)| = 1$, the normalization of S is the CMC $\mathcal{N}(S) = \langle \{1, \dots, m\}, o', \varphi', A, V' \rangle$ such that $\mathcal{N}(o) = o'$ and

1. $\forall 1 \leq j \leq m$, $|V'(j)| = 1$;
2. $\forall 1 \leq i \leq k$, $V(i) = \cup_{u \in \mathcal{N}(i)} V'(u)$;
3. $\forall 1 \leq i \leq k$, $\forall u, v \in \mathcal{N}(i)$, $u \neq v \iff V'(u) \neq V'(v)$;
4. $\forall 1 \leq j \leq m$,

$$\varphi'(j)(x_1, \dots, x_m) = \varphi(\mathcal{N}^{-1}(j))\left(\sum_{u \in \mathcal{N}(1)} x_u, \dots, \sum_{u \in \mathcal{N}(k)} x_u\right).$$

By construction, $\mathcal{N}(S)$ is in single valuation normal form. Moreover, if S is consistent, then a function \mathcal{N} satisfying the conditions above exists.

Theorem 29 Let $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$ be a consistent CMC. If $|V(o)| = 1$, then for all MC P , we have $P \models S \iff P \models \mathcal{N}(S)$.

Proof:

Let $S = \langle \{1, \dots, k\}, o, \varphi, A, V \rangle$ be a consistent CMC such that $|V(o)| = 1$. Let $S' = \mathcal{N}(S) = \langle \{1, \dots, m\}, o', \varphi', A, V' \rangle$ and $\mathcal{N} : \{1, \dots, k\} \rightarrow 2^{\{1, \dots, m\}}$ the associated function.

\Rightarrow Let $P = \langle \{1, \dots, n\}, o_P, M, A_P, V_P \rangle$ be a MC such that $P \models S$. Let \mathcal{R} be the associated satisfaction relation. Let $\mathcal{R}' \subseteq \{1, \dots, n\} \times \{1, \dots, m\}$ such that $p \mathcal{R}' u \iff V_P(p) \in V'(u)$ and $p \mathcal{R}' \mathcal{N}^{-1}(u)$. We will show that \mathcal{R}' is a satisfaction relation. Let p, u such that $p \mathcal{R}' u$.

1. By definition, we have $V_P(p) \in V'(u)$.
2. We have $p \mathcal{R}' \mathcal{N}^{-1}(u)$. Let $\Delta \in [0, 1]^{n \times k}$ be the associated correspondence matrix. Define $\Delta' \in [0, 1]^{n \times m}$ such that $\Delta'_{q,v} = \Delta_{q, \mathcal{N}^{-1}(v)}$ if $V_P(q) \in V'(v)$ and 0 else. As every coefficient of Δ appears once and only once in the same row of Δ' , it is clear that Δ' is a correspondence matrix. Moreover,
 - (a) If q is such that $M_{pq} \neq 0$, then $\sum_{j=1}^m \Delta'_{q,j} = \sum_{i=1}^k \Delta_{q,i} = 1$;
 - (b) For all $1 \leq i \leq k$, $\sum_{j \in \mathcal{N}(i)} ([M_p \times \Delta']_j) = [M_p \times \Delta]_i$. As a consequence, $\varphi'(u)(M_p \times \Delta') = \varphi(\mathcal{N}^{-1}(u))(M_p \times \Delta)$ holds.
 - (c) If q, v are such that $\Delta'_{q,v} \neq 0$, then $\Delta_{q, \mathcal{N}^{-1}(v)} \neq 0$ and $V_P(q) \in V'(v)$, thus $q \mathcal{R}' v$.

Finally, \mathcal{R}' is a satisfaction relation. It is easy to see that $o_p \mathcal{R}' o'$. As a consequence, we have $P \models \mathcal{N}(S)$.

\Leftarrow Let $P = \langle \{1, \dots, n\}, o_P, M, A_P, V_P \rangle$ be a MC such that $P \models \mathcal{N}(S)$. Let \mathcal{R} be the associated satisfaction relation. Let $\mathcal{R}' \subseteq \{1, \dots, n\} \times \{1, \dots, k\}$ such that $p \mathcal{R}' u \iff \exists j \in \mathcal{N}(u)$ s.t. $p \mathcal{R} j$. We will show that \mathcal{R}' is a satisfaction relation. Let p, u such that $p \mathcal{R}' u$.

1. We have $V_P(p) \in V(u) = \cup_{j \in \mathcal{N}(u)} V'(j)$.
2. Let $j \in \mathcal{N}(u)$ such that $p \mathcal{R} j$, and let $\Delta \in [0, 1]^{n \times m}$ be the associated correspondence matrix. Define $\Delta' \in [0, 1]^{n \times k}$ such that $\Delta'_{q,v} = \sum_{i \in \mathcal{N}(v)} \Delta_{q,i}$. It is clear that for all q , $\sum_{v=1}^k \Delta'_{q,v} = \sum_{r=1}^m \Delta_{q,r}$. Thus Δ' is a correspondence matrix. Moreover,
 - (a) If q is such that $M_{pq} \neq 0$, then $\sum_{i=1}^k \Delta'_{q,i} = \sum_{r=1}^m \Delta_{q,r} = 1$;
 - (b) For all $1 \leq i \leq k$, $[M_p \times \Delta']_i = \sum_{r \in \mathcal{N}(i)} ([M_p \times \Delta]_r)$. As a consequence, $\varphi(u)(M_p \times \Delta) = \varphi'(j)(M_p \times \Delta')$ holds.
 - (c) If q, v are such that $\Delta'_{q,v} \neq 0$, then there exists $r \in \mathcal{N}(v)$ such that $\Delta_{q,r} \neq 0$, thus $q \mathcal{R}' v$.

Finally, \mathcal{R}' is a satisfaction relation. It is easy to see that $o_P \mathcal{R}' o$. As a consequence, we have $P \models S$.

It is easy to see that normalization preserves determinism.

.10 Soundness of weak refinement

Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ be two CMCs. Assume $S_1 \preceq S_2$, we prove that $\llbracket S_1 \rrbracket \subseteq \llbracket S_2 \rrbracket$.

Proof:

Since $S_1 \preceq S_2$, there exists a weak refinement relation $\mathcal{R} \subseteq \{1, \dots, k_1\} \times \{1, \dots, k_2\}$ such that $o_1 \mathcal{R} o_2$. Consider $P = \langle \{1, \dots, n\}, o_P, M, A_P, V_P \rangle$ such that $P \models S_1$. By definition, we have $o_P \models o_1$ and there exists a satisfaction relation $\mathcal{R}' \subseteq \{1, \dots, n\} \times \{1, \dots, k_1\}$ such that $o_P \mathcal{R}' o_1$.

Let $\mathcal{R}'' \subseteq \{1, \dots, n\} \times \{1, \dots, k_2\}$ such that $p \mathcal{R}'' u \iff \exists v \in \{1, \dots, k_1\}$ with $p \mathcal{R}' v$ and $v \mathcal{R} u$. Let's show that \mathcal{R}'' is a satisfaction relation. First, it is clear that $A_2 \subseteq A_1 \subseteq A_P$.

Now, consider p, u such that $p \mathcal{R}'' u$. By definition, there exists v such that $p \mathcal{R}' v$ and $v \mathcal{R} u$. Since $V_P(p) \downarrow_{A_1} \in V_1(v)$ and $V_1(v) \downarrow_{A_2} \in V_2(u)$, we have $V_P(p) \downarrow_{A_2} \in V_2(u)$.

We now build a correspondence matrix Δ'' that satisfies the axioms of Definition 3. Let $x = M_p \in [0, 1]^{1 \times n}$ and $\Delta' \in [0, 1]^{n \times k_1}$ be a correspondence matrix witnessing $p \models v$. Let $y = x \times \Delta' \in [0, 1]^{1 \times k_1}$. By definition of Δ' , we have $\varphi_1(v)(y)$. Let $\Delta \in [0, 1]^{k_1 \times k_2}$ be the correspondence matrix witnessing $v \preceq u$ and define $\Delta'' = \Delta' \times \Delta \in [0, 1]^{n \times k_2}$. By Lemma 25, Δ'' is also a correspondence matrix. We prove that Δ'' satisfies the axioms of Definition 3.

1. Let $1 \leq p' \leq n$ such that $M_{pp'} \neq 0$. As a consequence, $\sum_{j=1}^{k_1} \Delta'_{p'j} = 1$. We want to prove that $\sum_{j=1}^{k_2} \Delta''_{p'j} = 1$.

$$\begin{aligned} \sum_{j=1}^{k_2} \Delta''_{p'j} &= \sum_{j=1}^{k_2} \left(\sum_{q=1}^{k_1} \Delta'_{p'q} \cdot \Delta_{qj} \right) \\ &= \sum_{q=1}^{k_1} \Delta'_{p'q} \cdot \left(\sum_{j=1}^{k_2} \Delta_{qj} \right) \end{aligned}$$

Let q such that $\Delta'_{p'q} \neq 0$. It is then clear that $y_q \geq M_{pp'} \cdot \Delta'_{p'q} > 0$. As Δ is a witness of $v \preceq u$, we have $\sum_{j=1}^{k_2} \Delta_{qj} = 1$. Finally, this implies that $\sum_{j=1}^{k_2} \Delta''_{p'j} = 1$.

2. By construction, $\varphi_2(u)(M_p \times \Delta'')$ holds.
3. Let p', u' such that $\Delta''_{p'u'} \neq 0$. By construction, it is clear that there exists v' such that $\Delta'_{p'v'} \neq 0$ and $\Delta_{v'u'} \neq 0$. By definition of Δ' and Δ , this implies that $p' \mathcal{R}' v'$ and $v' \mathcal{R} u'$, thus $p' \mathcal{R}'' u'$.

From 1-3, we can conclude that \mathcal{R}'' is a satisfaction relation. Since $o_P \mathcal{R}'' o_2$, we have $P \in \llbracket S_2 \rrbracket$ and $\llbracket S_1 \rrbracket \subseteq \llbracket S_2 \rrbracket$.

.11 Completeness of Weak Refinement (Thm. 18)

We suppose that the CMCs we consider in this proof are pruned. Moreover we only consider CMCs in single valuation normal form. Given two CMCs S_1 and S_2 such that $\llbracket S_1 \rrbracket \subseteq \llbracket S_2 \rrbracket$, we prove that $S_1 \preceq S_2$. The proof is structured as following.

1. • We define the relation \mathcal{R} between S_1 and S_2 .

$$R = \{(v, u) \mid \forall I, \forall p \in I, p \models v \Rightarrow p \models u\}$$

We consider u and v such that $v \mathcal{R} u$ and prove that \mathcal{R} satisfies Axiom (1) of the refinement relations.

- Axiom (2) of the weak refinement relations : Given a distribution X on the outgoing transitions of v , we must find a correspondence matrix Δ satisfying Axioms 2(a), 2(b) and 2(c) of the refinement relation :
 - We consider a distribution X on the outgoing transitions from v and we build a MC I satisfying S_1 such that the outgoing probabilities of the state v_I are exactly X .
 - This leads to $v_I \models u$ and gives a correspondence matrix Δ_2 , which we will take as our correspondence matrix Δ .
 - By definition, Δ satisfies the axioms 2(a) and 2(b) of the weak refinement relations.
2. As Δ comes from a satisfaction relation, the axiom 2(c) of the refinement relation is not so immediate. It tells us that if a coefficient $\Delta_{v'u'}$ is not 0, then there exists an implementation I and a state v'_I such that $v'_I \models v'$ and $v'_I \models u'$. What we need is that for all implementations I' and state p' such that $p' \models v'$, we have $p' \models u'$. **The rest of the proof is dedicated to proving that this statement being false leads to a contradiction.**

Assuming there exists I' and p' such that $p' \models v'$ and $p' \not\models u'$, we build an implementation \widehat{I} from I and I' such that the state v' of \widehat{I} is syntactically equivalent to the state p' . We then prove that this state v' of \widehat{I} still satisfies the state u' of S_2 because it is a successor of v and S_2 is deterministic. As the state v' of \widehat{I} is syntactically equivalent to the state p' of I' , this means that $p' \models u'$, which is a contradiction.

We now go through the mathematical foundations of this proof.

Proof:

Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_1, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A_2, V_2 \rangle$ be two consistent and deterministic CMCs in single valuation normal form such that $A_2 \subseteq A_1$ and $\llbracket S_1 \rrbracket \subseteq \llbracket S_2 \rrbracket$.

First, remark that $S_1 \preceq S_2 \iff S'_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A_2, V_1 \downarrow_{A_2} \rangle \preceq S_2$. It is thus safe to suppose that $A_1 = A_2$. Similarly, if $I = \langle \dots, A_I, V_I \rangle$ is a MC, we have $I \models S_1 \iff I' = \langle \dots, A_I, V_I \downarrow_{A_1} \rangle \models S_1$. As a consequence, it is also safe to suppose that implementations have the same set of atomic propositions as S_1 and S_2 .

1. Let $\mathcal{R} \subseteq \{1, \dots, k_1\} \times \{1, \dots, k_2\}$ such that $v \mathcal{R} u$ iff for all MC I and state p of I , $p \models v \Rightarrow p \models u$. As we consider pruned CMCs, there exist implementations for all states.

Consider v and u such that $v \mathcal{R} u$.

- (a) By definition of \mathcal{R} , there exists a MC I and a state p of I such that $p \models v$ and $p \not\models u$. Thus $V_I(p) \in V_1(v)$ and $V_I(p) \in V_2(u)$. As S_1 and S_2 are in single valuation normal form, $V_1(v)$ and $V_2(u)$ are singletons, so $V_1(v) = V_2(u)$.
- (b) Consider $x \in [0, 1]^{1 \times k_1}$ such that $\varphi_1(v)(x)$ and build the MC $I = \langle \{1, \dots, k_1\}, o_1, M, A_1, V'_1 \rangle$ such that for all $1 \leq w \leq k_1$,

- $V'_1(w)$ is the only valuation T such that $V_1(w) = \{T\}$;
- If $w \neq v$, the line M_w is any solution of $\varphi_1(w)$. One exists because S_1 is pruned;
- $M_v = x$.

When necessary, we will address state w of I as w_I to differentiate it from state w of S_1 . We will now build the correspondence matrix Δ .

I clearly satisfies S_1 with a satisfaction relation $\mathcal{R}_1 = \text{Identity}$, and $v_I \models v$. By hypothesis, we thus have $v_I \models u$. Consider \mathcal{R}_2 the satisfaction relation such that $v_I \mathcal{R}_2 u$ and Δ_2 the corresponding correspondence matrix. Let $\Delta = \Delta_2$.

(c) As a consequence,

- i. $\forall 1 \leq i \leq k_1, x_i \neq 0 \Rightarrow \sum_{j=1}^{k_2} \Delta_{ij} = 1$;
- ii. $\varphi_2(u)(x \times \Delta)$ holds;

2. Let v' be a state of S_1 such that If $x_{v'} \neq 0$ and $\Delta_{v'u'} \neq 0$. By definition of I and Δ , we have $v'_I \models v'$ and $v'_I \models u'$. We want to prove that for all implementations I' and state p' in I' , $p' \models v'$ implies $p' \models u'$.

Suppose this is not the case. There exists an implementation $I' = \langle \{1, \dots, n\}, o', M', A_1, V' \rangle$ and a state p' of I' such that $p' \models v'$ and $p' \not\models u'$. Let \mathcal{R}' be the correspondence matrix witnessing $p' \models v'$.

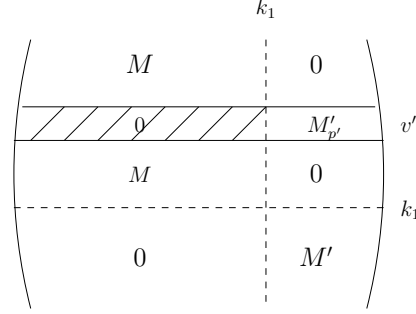
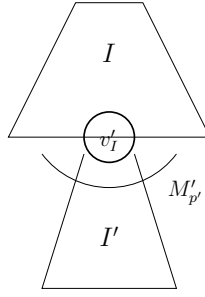
Consider the MC $\widehat{I} = \langle \{1, \dots, k_1, k_1 + 1, \dots, k_1 + n\}, o_I, \widehat{M}, A_1, \widehat{V} \rangle$. Intuitively, the first k_1 states correspond to I and the next n states to I' . The state v'_I will be the link between the two and its outgoing transitions will be the ones of p' . Define

- $\widehat{M}_{i,j} = M_{i,j}$ if $1 \leq i, j \leq k_1$ and $i \neq v'$
- $\widehat{M}_{v',j} = 0$ if $1 \leq j \leq k_1$
- $\widehat{M}_{i,j} = 0$ if $1 \leq i \leq k_1$ and $i \neq v'$ and $j > k_1$
- $\widehat{M}_{v',j} = m'_{p',j-k_1}$ if $j > k_1$
- $\widehat{M}_{i,j} = 0$ if $i > k_1$ and $1 \leq j \leq k_1$
- $\widehat{M}_{i,j} = m'_{i-k_1,j-k_1}$ if $i > k_1$ and $j > k_1$.
- $\widehat{V}(i) = V'_1(i)$ if $i \leq k_1$
- $\widehat{V}(i) = V'(i - k_1)$ if $i > k_1$

We want to prove that v'_I satisfies u' . This should imply that p'_I , also satisfies u' , which is absurd.

Consider the relation $\widehat{\mathcal{R}}$ between the states of \widehat{I} and the states of S_1 defined as follows :

$$\begin{aligned} \widehat{\mathcal{R}} = & \{(q, w) \in R_1 \mid q \neq v'\} \cup \\ & \{(q, w) \mid (q - k_1) \mathcal{R}' w\} \cup \\ & \{(v', w) \mid p' \mathcal{R}' w\} \end{aligned}$$

(a) The transition matrix \widehat{M} (b) The MC \widehat{I}

Intuitively, $\widehat{\mathcal{R}}$ is equal to \mathcal{R}_1 for the states $q \leq k_1$, except v' , and equal to \mathcal{R}' for the states $q > k_1$. The states related to v'_I are the ones that were related to p' with \mathcal{R}' .

We will show that $\widehat{\mathcal{R}}$ is a satisfaction relation between \widehat{I} and S_1 .

Let q, w such that $q \widehat{\mathcal{R}} w$. For all the pairs where $q \neq v'_I$, the conditions of the satisfaction relation obviously still hold because they held for \mathcal{R}_1 if $q \leq k_1$ and for \mathcal{R}' otherwise. It remains to check the conditions for the pairs where $q = v'_I$.

Consider w such that $v'_I \widehat{\mathcal{R}} w$.

- (a) Because (v'_I) and (p'_I) are both implementations of v' , it is clear that $\widehat{V}(v'_I) = \widehat{V}(p'_I)$. As $p'_I \mathcal{R}' w$, we know that $V'(p'_I) \in V_1(w)$. Thus, $\widehat{V}(v'_I) \in V_1(w)$.
- (b) Consider the correspondence matrix Δ' given by $p'_I \mathcal{R}' w$. Let $\widehat{\Delta} \in [0, 1]^{(k_1+n) \times k_1}$ such that $\widehat{\Delta}_{ij} = 0$ if $i \leq k_1$, and $\widehat{\Delta}_{ij} = \Delta'_{(i-k_1)j}$ otherwise.
 - i. We want to show that if $\widehat{M}_{(v'_I)(w')} \neq 0$, then $\sum_{j=1}^{k_1} \widehat{\Delta}_{w'j} = 1$. We know that $\widehat{M}_{(v'_I)(w')} = 0$ if $w' \leq k_1$. Take $w' > k_1$ such that $\widehat{M}_{(v'_I)(w')} \neq 0$. Then we know that $\widehat{M}_{(v'_I)(w')} = M'_{p'_I(w'-k_1)}$. Because \mathcal{R}' is a satisfaction relation, it implies that $\sum_{j=1}^{k_1} \Delta'_{(w'-k_1)j} = 1$. Thus, $\sum_{j=1}^{k_1} \widehat{\Delta}_{w'j} = \sum_{j=1}^{k_1} \Delta'_{(w'-k_1)j} = 1$.
 - ii. We want to show now that $\varphi_1(w)(\widehat{M}_{v'_I} \times \widehat{\Delta})$ holds. Let $1 \leq j \leq k_1$. We have

$$\begin{aligned}
M\widehat{M}_{v'_i} \times \widehat{\Delta}t_j &= \sum_{l=1}^{k_1+n} \widehat{M}_{(v'_i)l} \cdot \widehat{\Delta}l_j \\
&= 0 + \sum_{l=k_1+1}^{k_1+n} \widehat{M}_{(v'_i)l} \cdot \widehat{\Delta}l_j \\
&= \sum_{l=1}^n M'_{p'l} \cdot \Delta'_{l_j} = [M'_{p'} \times \Delta']_j
\end{aligned}$$

As a consequence, $\widehat{M}_{v'_i} \times \widehat{\Delta} = M'_{p'} \times \Delta'$. Since Δ' is a witness of $p' \mathcal{R}' w$, $\varphi_1(w)(M'_{p'} \times \Delta')$ holds. So does $\varphi_1(w)(\widehat{M}_{v'_i} \times \widehat{\Delta})$.

- iii. We want to show that if $\widehat{M}_{(v'_i)q} \neq 0$ and $\widehat{\Delta}_{qw'} \neq 0$, then $q\widehat{\mathcal{R}}w'$. We only need to consider $q > k_1$ (since otherwise $\widehat{M}_{(v'_i)q} = 0$) and w' such that $\widehat{\Delta}_{qw'} \neq 0$. In this case, $\widehat{M}_{(v'_i)q} = M'_{p'(q-k_1)} \neq 0$ and $\Delta'_{(q-k_1)w'} \neq 0$. As Δ' is a witness of $p' \mathcal{R}' w$, it has to be that $(q - k_1) \mathcal{R}' w'$, which implies, by definition of $\widehat{\mathcal{R}}$, that $q\widehat{\mathcal{R}}w'$.

Finally \widehat{I} satisfies S_1 , and in particular, $v_{\widehat{I}} \models v$. As $v \mathcal{R} u$, it implies that $v_{\widehat{I}} \models u$. As a consequence, there exists $\Delta'' \in [0, 1]^{(k_1+n) \times k_2}$ such that $\varphi_2(u)(\widehat{M}_{v_{\widehat{I}}} \times \Delta'')$.

- (A) Consider $u'' \neq u'$ such that $V_2(u'') = V_2(u')$. Due to determinism of S_2 , and to the fact that u' is accessible from u , we have $[\widehat{M}_{v_{\widehat{I}}} \times \Delta'']_{u''} = 0$. Since $\widehat{M}_{(v_{\widehat{I}})(v'_i)} \neq 0$ and $\widehat{M}_{(v_{\widehat{I}})(v'_i)} \cdot \Delta''_{(v'_i)u''}$ is part of $[\widehat{M}_{v_{\widehat{I}}} \times \Delta'']_{u''}$, we must have $\Delta''_{(v'_i)u''} = 0$.
- (B) Consider u''' such that $V(u''') \neq V(u')$. It is clear that $\Delta''_{(v'_i)u'''} = 0$ since Δ'' is witnessing satisfaction between \widehat{I} and S_2 .
- (C) Moreover, we know that $\widehat{M}_{(v_{\widehat{I}})(v'_i)} \neq 0$. Thus, $\sum_{j=1}^{k_2} \Delta''_{v'_i j} = 1$.

According to (A) and (B), the only non-zero value in the sum in (C) must be $\Delta''_{(v'_i)u'}$. Since Δ'' is witnessing $\widehat{I} \models S_2$, this means that $v'_{\widehat{I}} \models u'$.

By construction, $v'_{\widehat{I}}$ and p' only differ by state names. This contradicts the assumption that $p' \not\models u'$. Thus $v' \mathcal{R} u'$, and \mathcal{R} is a weak refinement relation.

Finally, we have by hypothesis that $\llbracket S_1 \rrbracket \subseteq \llbracket S_2 \rrbracket$, which implies that $o_1 \mathcal{R} o_2$.

.12 Proof of Theorem 19

We start with the following lemma, which is a direct consequence of the notion of determinism. It states that correspondence matrices associated to a satisfaction relation for a deterministic CMC have at most one non-zero value per row.

Lemma 30 *Let $S = \langle \{1, \dots, k\}, o_S, \varphi, A, V_S \rangle$ be a deterministic CMC in single valuation normal form. Let $P = \langle \{1, \dots, n\}, o_P, M, A, V_P \rangle \in \llbracket S \rrbracket$ and a satisfaction*

relation \mathcal{R} such that $o_P \mathcal{R} o_S$. Let $p \in \{1, \dots, n\}$ and $u \in \{1, \dots, k\}$ such that $p \mathcal{R} u$, and let Δ be the associated correspondence matrix. We have

$$\forall p' \in \{1, \dots, n\}, M_{pp'} \neq 0 \Rightarrow |\{u' \in \{1, \dots, k\} \mid \Delta_{p'u'} \neq 0\}| = 1.$$

Let $S_1 = \langle \{1, \dots, k_1\}, o_1, \varphi_1, A, V_1 \rangle$ and $S_2 = \langle \{1, \dots, k_2\}, o_2, \varphi_2, A, V_2 \rangle$ be two deterministic CMCs in normal form such that $S_1 \preceq S_2$ with a weak refinement relation \mathcal{R} . We prove that \mathcal{R} is in fact a strong refinement relation.

Proof:

Let $v \in \{1, \dots, k_1\}$ and $u \in \{1, \dots, k_2\}$ such that $v \mathcal{R} u$.

1. By hypothesis, $V_1(v) \subseteq V_2(u)$;
2. We know that for all $x \in [0, 1]^{k_1}$ satisfying $\varphi_1(u)$, there exists a correspondence matrix Δ^x satisfying the axioms of a (weak) refinement relation. We will build a correspondence matrix Δ^0 that will work for all x . Let $p \in \{1, \dots, k_1\}$.

- If for all $x \in [0, 1]^{k_1}$, $\varphi_1(v)(x) \Rightarrow x_p = 0$, then let $\Delta_p^0 = (0, \dots, 0)$.
- Else, consider $x \in [0, 1]^{k_1}$ such that $\varphi_1(v)(x)$ and $x_p \neq 0$. By hypothesis, there exists a correspondence matrix Δ^x associated to $v \mathcal{R} u$. Let $\Delta_p^0 = \Delta_p^x$. By Lemma 30, there is a single $u' \in \{1, \dots, k_2\}$ such that $\Delta_{pu'}^x \neq 0$. Moreover, by definition of Δ^x , we know that $\sum_{r=1}^{k_2} \Delta_{pr}^x = 1$, thus $\Delta_{pu'}^x = 1$.

Suppose there exists $y \neq x \in [0, 1]^{k_1}$ such that $\varphi_1(v)(y)$ and $y_p \neq 0$. Let Δ^y be the associated correspondence matrix. As for x , there exists a unique $u'' \in \{1, \dots, k_2\}$ such that $\Delta_{pu''}^y \neq 0$. Moreover $\Delta_{pu''}^y = 1$. Let $x' = x \times \Delta^x$ and $y' = y \times \Delta^y$. By definition, both $\varphi_2(v)(x')$ and $\varphi_2(v)(y')$ hold, $x'_{u'} \neq 0$ and $y'_{u''} \neq 0$. As $\Delta_{pu'}^x = \Delta_{pu''}^y = 1$, we have $V_2(u') \cap V_2(u'') \neq \emptyset$. By hypothesis, S_2 is deterministic, thus $u' = u''$.

As a consequence, we have $\Delta_p^x = \Delta_p^y$, so $\forall z \in [0, 1]^{k_1}$, $(\varphi_1(v)(z) \wedge (z_p \neq 0)) \Rightarrow \Delta_p^z = \Delta_p^0$.

Finally, consider Δ^0 defined as above. Let $x \in [0, 1]^{k_1}$ such that $\varphi_1(u)(x)$. We have

- (a) $x_i \neq 0 \Rightarrow \Delta_i^0 = \Delta_i^x \Rightarrow \sum_{j=1}^{k_2} \Delta_{ij}^0 = 1$;
- (b) $x \times \Delta^0 = x \times \Delta^x$, thus $\varphi_2(v)(x \times \Delta^0)$ holds;
- (c) If $\Delta_{v'u'}^0 \neq 0$, then there exists $y \in [0, 1]^{k_1}$ such that $\varphi_1(v)(y)$ and $\Delta_{v'u'}^0 = \Delta_{v'u'}^y$, thus $v' \mathcal{R} u'$.

Finally, \mathcal{R} is a strong refinement relation.

.13 Reduction from Simulation (Sec. 7)

We will now prove Theorem 21. This section contains sketches of proofs, with information sufficient to reconstruct them without diligent work. In the next section we present the second encoding, which is actually slightly reacher, and present a complete correctness proof for it. We have chosen to present that proof in detail, because due to its use of linear combinations, it is much harder to reconstruct during review.

We begin by demonstrating a lemma about non-deterministic distribution constraints.

We say that a constraint is a single-point constraint, if it is only satisfied by a unique distribution. Observe that all constraints in the encoding presented in Section 7 are non-deterministic distribution constraints or single-point constraints.

Lemma 31 *Let φ and ψ be single-point constraints. If for each $x \in [0, 1]^{1 \times k_1}$ such that $\varphi(x)$ holds, there exists a correspondence matrix $\Delta_x \in [0, 1]^{k_1 \times k_2}$ such that $\psi(x \times \Delta_x)$ holds then there exists a correspondence matrix $\Delta \in [0, 1]^{k_1 \times k_2}$ such that for all $x \in [0, 1]^{1 \times k_1}$ we have that $\varphi(x) \implies \psi(x \times \Delta)$.*

The lemma holds trivially because there is only one distribution satisfying φ .

Lemma 32 *Let φ (respectively ψ) is a non-deterministic distribution constraint over $\{1, \dots, k_1\}$ (respectively $\{1, \dots, k_2\}$). Then if for each distribution vector x satisfying φ there exists a correspondence matrix $\Delta_x \in [0, 1]^{k_1 \times k_2}$ such that $\psi(x \times \Delta_x)$ holds then there exists a correspondence matrix $\Delta \in [0, 1]^{k_1 \times k_2}$ such that for all $x \in [0, 1]^{1 \times k_1}$ we have that $\varphi(x) \implies \psi(x \times \Delta)$.*

Proof:

Let x be such that $\varphi(x)$ holds (then there exists $1 \leq i \leq k_1$ such that $x_i = 1$). There is a finite number of such vectors. Let x^i denote the one that has 1 on the i th position. Take Δ such that $\Delta_i = (\Delta_{x^i})_i$ (the witness from the lemma assumption) if x^i satisfies φ and $\Delta_i = 0^{1 \times k_2}$ otherwise.

Now for each x^i satisfying φ we have that $x^i \times \Delta = x^i \times \Delta_{x^i}$ and then $\varphi(x^i) \implies \psi(x^i \times \Delta_{x^i}) \iff \psi(x^i \times \Delta)$.

Corollary 33 *For any two probabilistic automata \mathbb{S} and \mathbb{T} we have that $\widehat{\mathbb{S}}$ strongly refines $\widehat{\mathbb{T}}$ iff $\widehat{\mathbb{S}}$ weakly refines $\widehat{\mathbb{T}}$.*

Lemma 34 *For any two probabilistic automata \mathbb{S} and \mathbb{T} such that \mathbb{T} simulates \mathbb{S} we have that $\widehat{\mathbb{S}}$ weakly refines $\widehat{\mathbb{T}}$.*

Proof:

(sketch) Let $R \subset S \times T$ be the relation witnessing the simulation of \mathbb{S} by \mathbb{T} . Consider a relation Q as follows:

$$\begin{aligned} Q_1 &= \{(i, j) \mid i \in \{1, \dots, k_1\}, j \in \{1, \dots, k_2\}, \\ &\quad (s_i, t_j) \in R\} \\ Q_2 &= \{(k_1 + i, k_2 + j) \mid i \in \{1, \dots, k_1\}, j \in \{1, \dots, k_2\}, \\ &\quad (s_{i-k_1}, t_{j-k_2}) \in R\} \\ Q_3 &= \{(2k_1 + i', 2k_2 + j') \mid i' \in \{1, \dots, l_1\}, \\ &\quad j' \in \{1, \dots, l_2\}, (a_i, \pi_i) \in \Omega_S, (a_j, \varrho_j) \in \Omega_T, \\ &\quad a_i = a_j, (\pi_i, \varrho_i) \in R^*\} \\ Q &= Q_1 \cup Q_2 \cup Q_3 \end{aligned}$$

It is easy to show that Q is a weak refinement. First observe that valuations always match for pairs in Q . The valuation is empty for both S and T in Q_1 , it is $\{\perp\}$ in Q_2 , and $\{a_i\}$ in Q_3 .

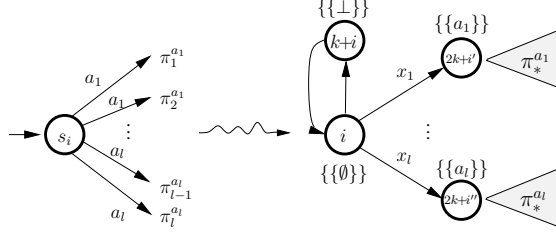


Figure 10: An attempt to visualize the second encoding. π_*^a denotes a constraint expressing a probability vector that is a linear combination of all probability distributions labeled by a . Below this is formalized as $\varphi(2k + i')(x)$.

For a pair in $(i, j) \in Q_1$ a distribution vector x satisfying the constraint of S is always a point distribution. If $x_{k_1+i} = 1$, take $\Delta_{k_1+i, k_2+j} = 1$ and zero otherwise. If $x_{2k_1+i'} = 1$ take $\Delta_{2k_1+i', 2k_2+j'} = 1$ and zero otherwise, where j' is such that $t_{j'} \xrightarrow{a_{i'}} \varrho_{j'}$ and $\pi_{i'} R^* \varrho_{j'}$.

For a pair $(k_1 + i, k_2 + j) \in Q_2$ take $\Delta_{i,j} = 1$, and zero otherwise.

For a pair $(2k_1 + i', 2k_2 + j') \in Q_3$ take Δ such that for $(i, j) \in \{1, \dots, k_1\} \times \{1, \dots, k_2\}$ we have $\Delta_{ij} = \alpha_{ij}/x_i$, or zero if $x_i = 0$, where α is the distribution witnessing $\pi_{i'} R^* \varrho_{j'}$.

Lemma 35 For any two probabilistic automata \mathbb{S} and \mathbb{T} such that $\widehat{\mathbb{S}}$ strongly refines $\widehat{\mathbb{T}}$ we have that \mathbb{T} simulates \mathbb{S} .

Proof:

(sketch) Assume that $\widehat{\mathbb{S}}$ strongly refines $\widehat{\mathbb{T}}$ is witnessed by a relation $R \subseteq \{1, \dots, 2k_1 + l_1\} \times \{1, \dots, 2k_2 + l_2\}$. Show that a relation $Q = \{(s_i, t_j) \in S \times T \mid (i, j) \in R, i \in \{1, \dots, k_1\}, j \in \{1, \dots, k_2\}\}$ is a simulation relation.

In the crucial point of the proof consider $\alpha_{s_i, t_j} = \Delta_{i,j} \cdot \pi_{i'}(s_i)$, where $\pi_{i'}$ is a distribution being the only solution of a point constraint for state $i' \in \{2k_1, \dots, 2k_2 + l_1\}$.

Theorem 21 follows as a corollary from the above two lemma and the Corollary 33.

.14 Encoding Probabilistic Simulation

We now present another encoding of PAs into CMCs, which aims at capturing probabilistic simulation (as opposed to simulation).

Consider a PA $\mathbb{S} = (S, Act, \rightarrow, s_1)$, where $S = \{s_1, \dots, s_k\}$. Let $\{(s^1, a_1), \dots, (s^l, a_l)\} = \{(s, a) \mid s \in S \wedge a \in Act\}$. The corresponding CMC is

$$\check{\mathbb{S}} = (\{1, \dots, 2k + l\}, 1, \check{\varphi}, Act \cup \perp, \check{V}) ,$$

where \perp is a fresh symbol not in Act . We have three types of states (see Figure 10). Type-1 states, $\{1, \dots, k\}$, correspond directly to states $\{s_1, \dots, s_k\}$ —their distribution constraints encode the non-deterministic choice of action. Type-2 states, $\{k + 1, \dots, 2k\}$, represent ability of a state to be idle. We will use them in parallel composition. Type-3 states, $\{2k + 1, \dots, 2k + l\}$, encode choice of a probability distribution as a linear combination of distributions allowed by the automaton.

The valuation functions are given by:

$$\begin{aligned} \check{V}(i) &= \{\emptyset\} && \text{for } 1 \leq i \leq k \\ \check{V}(k+i) &= \{\{\perp\}\} && \text{for } 1 \leq i \leq k \\ \check{V}(2k+i') &= \{\{a_{i'}\}\} && \text{for } 1 \leq i' \leq l \end{aligned}$$

and

$$\begin{aligned} \check{\varphi}(i)(x) &\text{ is } x_{k+i} = 1 \text{ or } \exists 1 \leq i' \leq l. x_{2k+i'} = 1 \wedge s^{i'} = s_i \\ &\text{ for } 1 \leq i \leq k \quad (\text{type-1 states}) \\ \check{\varphi}(k+i)(x) &\text{ is } x_i = 1 \\ &\text{ for } 1 \leq i \leq k \quad (\text{type-2 states}) \\ \check{\varphi}(2k+i')(x) &\text{ is } \exists \lambda \in \text{Dist}(1, \dots, |\pi|). x = \pi \lambda \\ &\text{ for } 1 \leq i' \leq l \quad (\text{type-3 states}) \end{aligned}$$

where $\pi = \{\pi \mid s^j \xrightarrow{a_j} \pi\}$. Technically speaking π is a matrix, whose columns are distributions π . We write $|\pi|$ for the number of columns in π . Additionally x is implicitly required to be a probability distribution over $\{1, \dots, 2k+l\}$.

Observe that $\widehat{\mathbb{S}}$ is only polynomially larger than \mathbb{S} .

Lemma 36 (Soundness) *For any two probabilistic automata \mathbb{S} and \mathbb{T} such that \mathbb{S} weakly refines \mathbb{T} . We have that $\widehat{\mathbb{T}}$ probabilistically simulates $\widehat{\mathbb{S}}$.*

Proof:

Let $\mathbb{S} = (S, \text{Act}, \rightarrow^S, s_1)$ and $\mathbb{T} = (T, \text{Act}, \rightarrow^T, t_1)$, with $S = \{s_1, \dots, s_{k_1}\}$ and $T = \{t_1, \dots, t_{k_2}\}$. In the proof we write $\check{\varphi}$ to refer to the constraint function of $\widehat{\mathbb{S}}$, and $\check{\varrho}$ to refer to the constraint function of $\widehat{\mathbb{T}}$. Also l_1 and l_2 are used to refer to the number of combinations of state-action of respectively $\widehat{\mathbb{S}}$ and $\widehat{\mathbb{T}}$. Finally q_i and r_j are used to range over states in S (respectively in T), when s_i and t_j are bound to some concrete value.

Let $R \in \{1, \dots, 2k_1+l_1\} \times \{1, \dots, 2k_2+l_2\}$ be a weak refinement relation between $\widehat{\mathbb{S}}$ and $\widehat{\mathbb{T}}$, witnessing the assumption of the lemma. The proof proceeds by showing that

$$Q = \{(s_i, t_j) \mid (i, j) \in R \wedge 1 \leq i \leq k_1 \wedge 1 \leq j \leq k_2\}$$

is a probabilistic simulation relation between \mathbb{S} and \mathbb{T} .

We apply the usual coinductive proof technique. Take $(s_i, t_j) \in Q$. Let $\pi \in \text{Dist}(S)$ be such that $s_i \xrightarrow{a} \pi$, and $(s^{i'}, a_{i'}) = (s_i, a)$.¹

By construction of the encoding we know that any probability distribution x satisfying $\varphi(i)(x)$ is a point distribution, and x such that $x_{2k+i'} = 1$ is possible. So consider such a distribution x . Since $(i, j) \in R$ we know that there exists a correspondence matrix $\Delta \in [0, 1]^{2k_1+l_1 \times 2k_2+l_2}$ such that $\psi(j)(x \times \Delta)$ holds. Moreover $x \times \Delta$ must be a point distribution by construction of the encoding. So $(x \times \Delta)_{2k_2+j'} = 1$ for some $1 \leq j' \leq l_2$. And, by refinement again, we get that valuation functions for both $2k_1+i'$ and for $2k_2+j'$ both return $\{\{a\}\}$ and that $(2k_1+i', 2k_2+j') \in R$.

¹The equality binds i' to be the index of (s_i, a) on the list of state-action pairs in the encoding of \mathbb{S} .

But $\check{\mathbb{T}}$ is also constructed using the encoding, so it necessarily is that $t_j \xrightarrow{a} \varrho$ for some $\varrho \in \text{Dist}(T)$.

Observe that $\varphi(2k_1 + i')$ holds, because π is always a convex linear combination of a set of vectors containing it. Since $(2k_1 + i', 2k_2 + j') \in R$, there exists a correspondence matrix $\Delta' \in [0, 1]^{2k_1+l_1 \times 2k_2+l_2}$ such that $\psi(2k_2 + j')(\pi \times \Delta')$ holds. The latter implies that $\pi \times \Delta'$ is a linear combinations of vectors in $\varrho = \{\varrho \mid t_j \xrightarrow{a} \varrho\}$.

It remains to show that $\pi R^*(\pi \times \Delta')$. Take $\alpha_{q_i, q_j} = \pi_i \cdot \Delta'_{ij}$. We first argue that $\alpha \in \text{Dist}(S \times T)$. Clearly $\pi_i \Delta'_{ij} \in [0, 1]$ for all i, j . Also $\sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \pi_i \Delta'_{ij} = \sum_{i=1}^{k_1} \pi_i = 1$ (the former because each row of a correspondence matrix sums up to 1).

Consider $\alpha_{q_i, T} = \sum_{j=1}^{k_2} \alpha_{q_i, t_j} = \sum_{j=1}^{k_2} \pi_i \cdot \Delta'_{ij} = \pi_i \sum_{j=1}^{k_2} \Delta'_{ij} = \pi_i$ as required by $\pi R^*(\pi \times \Delta')$.

Now consider $\alpha_{S, r_j} = \sum_{i=1}^{k_1} \alpha_{s_i, r_j} = \sum_{i=1}^{k_1} \pi_i \cdot \Delta'_{ij} = (\pi \times \Delta')_j$ as required by $\pi R^*(\pi \times \Delta')$.

Now if $\alpha_{q_i, r_j} \neq 0$ then $\Delta'_{ij} \neq 0$, which in turn with refinement of $2k_2 + j'$ by $2k_1 + i'$ implies that $(i, j) \in R$, and furthermore $(s_i, s_j) \in Q$ by construction, as required by $\pi R^*(\pi \times \Delta')$. This finishes the proof.

Lemma 37 (Completeness) For any two probabilistic automata \mathbb{S} and \mathbb{T} such that $\check{\mathbb{T}}$ probabilistically simulates $\check{\mathbb{S}}$, we have that \mathbb{S} weakly refines \mathbb{T} .

Proof:

Let $\mathbb{S} = (S, \text{Act}, \rightarrow^S, s_1)$ and $\mathbb{T} = (T, \text{Act}, \rightarrow^T, t_1)$, with $S = \{s_1, \dots, s_{k_1}\}$ and $T = \{t_1, \dots, t_{k_2}\}$. Let $Q \subseteq S \times T$ be the probabilistic simulation relation between \mathbb{S} and \mathbb{T} , witnessing the assumption of the lemma.

The proof proceeds by showing that a relation $R \subseteq \{1, \dots, 2k_1+l_1\} \times \{1, \dots, 2k_2+l_2\}$ is a weak refinement relation between $\check{\mathbb{S}}$ and $\check{\mathbb{T}}$.

Take the following candidate for R :

$$\begin{aligned} R_1 &= \{(i, j) \mid (s_i, t_j) \in Q\} \\ R_2 &= \{(k_1 + i, k_2 + j) \mid (s_i, t_j) \in Q\} \\ R_3 &= \{(2k_1 + i', 2k_2 + j') \mid (s_i, t_j) \in R \wedge \\ &\quad s_i = s^{i'} \wedge t_j = t^{j'}\} \\ R &= R_1 \cup R_2 \cup R_3 \end{aligned}$$

We apply the usual coinductive proof technique.

Case 1. Take $(i, j) \in R_1$ and x satisfying $\varphi(i)(x)$. We know that x can only be a point-distribution. If $x_{k_1+i} = 1$ then we take Δ such that $\Delta_{k_1+i, k_2+j} = 1$ (and Δ is zero for all other cells). Clearly Δ is a correspondence matrix. Moreover $x \times \Delta$ is a point distribution with 1 on $(k_2 + j)$ th position, so $\psi(j)(x \times \Delta)$ holds by construction of the encoding (see first case in encoding of constraints). Also $(k_1 + i, k_2 + j) \in R_2$ since $(s_i, t_j) \in Q$.

If $x_{2k_1+i'} = 1$ then it means that $s_i \xrightarrow{\check{V}(i)} \pi$ for some π and action $\check{V}(i)$. But then, since $(s_i, t_j) \in Q$, it is possible that $t_j \xrightarrow{\check{V}(i)} \varrho$ for some distribution ϱ . Let j' be such that $t_j = t^{j'}$ and $a_{j'} = \check{V}(i)$. Take a correspondence matrix Δ such that $\Delta_{2k_1+i', 2k_2+j'} = 1$ (and Δ is zero for all other cells). We have that $x \times \Delta$ is a point distribution with 1 on $2k_2 + j'$ th position, so $\psi(j)(x \times \Delta)$ holds by construction of encoding resulting in j (see first case in encoding of constraints). Also $(2k_1 + i', 2k_2 + j') \in R_3 \subseteq R$ by definition of R_3 .

Case 2. Take $(k_1 + i, k_2 + j) \in R_2$. The argument is almost identical to the first subcase in Case 1. We omit it here.

Case 3. Take $(2k_1 + i', 2k_2 + j') \in R_3$ and x satisfying $\varphi(2k_1 + i')(x)$. Let $s_i = s^{i'}$ and $t_j = t^{j'}$. By R_3 we know that $(s_i, t_j) \in Q$. By construction of the encoding $s_i \xrightarrow{V(2k_1+i')} x$ and furthermore $t_j \xrightarrow{V(2k_1+i')} \varrho$, where $\varrho = \underline{\varrho} \times \lambda$ for some probability distribution $\lambda \in \text{Dist}(1, \dots, |\underline{\varrho}|)$. Clearly $\psi(2k_2 + j')(\varrho) = 1$. It remains to check that π can be correspondence to ϱ .

To this end consider a correspondence matrix Δ such that

$$\Delta_{ij} = \begin{cases} \alpha_{s_i, t_j} / x_i & \text{if } x_i \neq 0 \text{ and } i \leq k_1, j \leq k_2 \\ 0 & \text{if otherwise} \end{cases}$$

Now $(x \times \Delta)_j = \sum_{i=1}^{2k_1+l_1} x_i \Delta_{ij} = \sum_{i=1}^{k_1} x_i \cdot \alpha_{s_i, t_j} / x_i = \sum_{i=1}^{k_1} \alpha_{s_i, t_j} = \alpha_{S, t_j} = \varrho_j$ by $xR^*\varrho$ (this discussion only holds for $j \leq k_2$, but the remaining cells are zero, which is easy to argue for. Also somewhat sloppily we ignored the possibility of division by zero – indeed it cannot happen since for $x_i = 0$ we said that Δ_{ij} is simply zero). Effectively $x \times \Delta = \varrho$, so it satisfies $\psi(2k_2 + j')$. Valuations obviously match.

Moreover if $\Delta_{ij} \neq 0$ then $\alpha_{s_i, t_j} \neq 0$. then $(s_i, t_j) \in Q$ and then $(i, j) \in R_1 \subseteq R$, which finishes the proof.

Theorem 38 is a corollary from the following two lemmas.

Theorem 38 \mathbb{T} probabilistically simulates \mathbb{S} iff $\widehat{\mathbb{S}}$ weakly refines $\widehat{\mathbb{T}}$.



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