Probabilistic Contracts for Component-based Design

Dana N. Xu, Gregor Goessler, Alain Girault

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Probabilistic Contracts for Component-based Design

Dana N. Xu — Gregor Gössler — Alain Girault

N° 7328

Juillet 2010

Embedded and Real Time Systems

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7328--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

centre de recherche
GRENOBLE - RHÔNE-ALPES

Probabilistic Contracts for Component-based Design*

Dana N. Xu , Gregor Gössler , Alain Girault

Theme : Embedded and Real Time Systems
Algorithmics, Programming, Software and Architecture
Équipes-Projets Pop Art, Gallium

Rapport de recherche n° 7328 — Juillet 2010 — 34 pages

Abstract: We define a probabilistic contract framework for describing and analysing component-based embedded systems, based on the theory of Interactive Markov Chains (IMC). A contract specifies the assumptions a component makes on its context and the guarantees it provides. Probabilistic transitions allow for uncertainty in the component behavior, e.g., to model observed black-box behavior (internal choice) or reliability. An interaction model specifies how components interact.

We provide the ingredients for a component-based design flow, including (1) contract satisfaction and refinement, (2) parallel composition of contracts over disjoint, interacting components, and (3) conjunction of contracts describing different requirements over the same component. Compositional design is enabled by congruence of refinement.

Key-words: component, probabilistic contract, refinement, composition

* supported by the European project COMBEST no. 215543

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Contrats probabilistes pour la conception à base de compostants

Résumé : Nous définissons un cadre formel de contrats probabilistes pour décrire et analyser des systèmes embarqués à base de composants. Ce cadre formel est fondé sur la théorie des chaînes de Markov interactives (IMC). Un contrat spécifie les hypothèses qu'un composant fait quant à son contexte et les garanties qu'il fournit. Des transitions probabilistes permettent de raisonner sur les incertitudes dans le comportement d'un composant, par exemple pour modéliser un comportement de type boîte noire (choix interne) ou sa fiabilité. Un modèle d'interaction spécifie la faҫn dont des composants interagissent.

Nous fournissons tous les ingrédients pour le flot de conception à base de composants, incluant (1) la satisfaction et le raffinement de contrat, (2) la composition parallèle de contrats portant sur des composants disjoints qui interagissent, et (3) la conjonction de contrats décrivant des comportements différents d'un même composant. Notre cadre formel permet de faire de la conception compositionnelle grâce à la congruence de l'opération de raffinement.

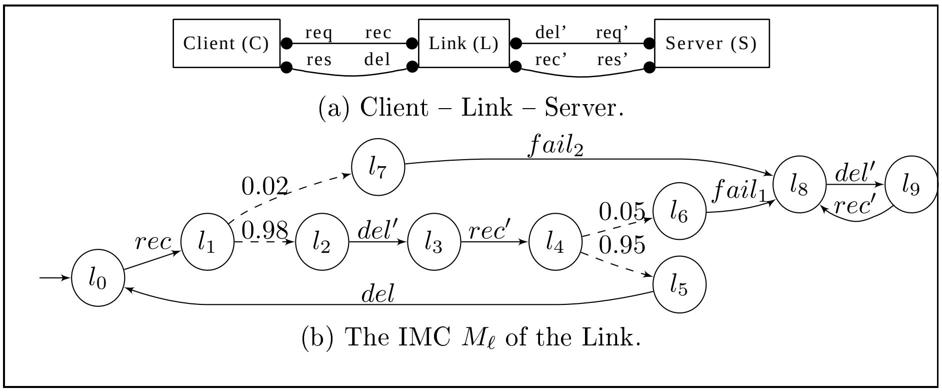
Mots-clés : composant, contrat probabiliste, raffinement, composition

Contents

1 Introduction

Typical embedded and distributed systems often encompass unreliable software or hardware components, as it may be technically or economically impossible to make a system entirely reliable. As a result, system designers have to deal with probabilistic specifications such as "the probability that this component fails at this point of its behavior is less than or equal to 10^{-4}". More generally, uncertainty in the observed behavior is introduced by abstraction of black-box — or simply too complex — behavior of components, the environment, or the execution platform. In this paper we introduce a framework for the design of correct systems from probabilistic, interacting components.

Figure 1(a) shows a Link system that transmits data between a Client and a Server. The Link receives a request from the Client and encodes the request before sending it to the Server. The encoding process fails with probability 0.02. After receiving a response from the Server, it decodes the data before delivering it to the Client. To model components, we adopt the discrete time Interactive Markov Chain (IMC) semantics model [8], which combines Labeled Transition System (LTS) and Markov Chain. Figure 1(b) shows an IMC describing the Link component of Figure 1(a). From its initial state l_0, the Link goes to state l_1 as soon as it receives (rec) a request from a Client; the probability that it delivers (del') this request to the Server is 0.98 and the probability that it fails to deliver it to the Server is 0.02. The Link goes to state l_4 immediately



(a) Client – Link – Server.

(b) The IMC M_ℓ of the Link.

Figure 1: An example of IMC: a Client-Link-Server.

after receiving a response (rec') from the Server; the probability that it delivers (del) the response to the Client is 0.95 and the probability of failing to do so is 0.05. In state l_8, the Link may still communicate with the Server regarding other services, but will not deliver any response to the Client.

Components communicate through interactions, that is, synchronized action transitions. Interactions are essential in component frameworks, as they allow the modeling of how components cooperate and communicate. We use the BIP framework [7] to model interactions between components.

Since the deploying context of a component is not known at design time, we use probabilistic *contracts* to specify and reason about correct behaviors of a component. Contracts were first introduced in [11]. They allow the designer to specify what a component can expect from its context, what it must guarantee, and explicitly limit the responsibilities of both.

The framework we propose here allows us to model components, their interactions, and the uncertainty in their observed behavior (§2). It supports different steps in a design flow: refinement, satisfaction, and abstraction (§3), parallel composition (§4.1), and conjunction (shared refinement) (§4.2). We prove that these operations satisfy the desired properties of *independent implementability* and *congruence* for parallel composition, and *soundness* for conjunction. Thus,

- refinement is compositional, that is, contracts over different components can be refined and implemented independently;

- the parallel composition of two contracts is satisfied by the parallel composition of any two implementations of the contracts; and

- several contracts C_i over the same component may be used to independently specify different requirements, possibly over different subsets of the component interactions. The conjunction is a common refinement of all C_i.

As pointed out in [2], conjunction of probabilistic specifications is non trivial, as a straight-forward approach would introduce spurious behaviors.

2 Components and Contracts

We give a formal definition to the discrete-time Interactive Markov Chains described in [8], used to model the behavior of components.

Definition 1 (Probability distribution)**.** *A* probability distribution *over a set X is a function $f : X \to [0,1]$ such that $\sum_{x \in X} f(x) = 1$.*

Definition 2 (Interactive Markov Chain (IMC))**.** *An IMC is a tuple $(\mathcal{Q}, \mathcal{A}, \rightarrow, \pi, s_0)$ where:*

- *$\mathcal{Q}$ is a nonempty finite set of states, partitioned into $\mathcal{Q}^{\mathsf{p}}$, the set of probabilistic states, and $\mathcal{Q}^{\mathsf{a}}$, the set of action states;*
- *A is a finite alphabet of actions;*
- *$\rightarrow \subseteq \mathcal{Q}^{\mathsf{a}} \times \mathcal{A} \times \mathcal{Q}$ is an action transition relation;*
- *$\pi : \mathcal{Q}^{\mathsf{p}} \to (\mathcal{Q} \to [0,1])$ is a transition probability function such that, for each $s \in \mathcal{Q}^{\mathsf{p}}$, $\pi(s)$ is a probability distribution over $\mathcal{Q}$;*
- *s_0 is the initial state.*

IMCs may interact with each other by synchronizing on action transitions (details in §4). Each action state in $\mathcal{Q}^{\mathsf{a}}$ has outgoing action transitions like those in an LTS. Each probabilistic state in $\mathcal{Q}^{\mathsf{p}}$ has outgoing probabilistic transitions like those in a Markov Chain. Probability distributions on states are memoryless, i.e., the future of an IMC depends only on the current state, not on past choices. For example, in Figure 1(b), the probabilistic choice that the Link delivers the response to the Client (i.e., $\pi(l_4)(l_5) = 0.95$) is independent of the probabilistic choice of delivering a request to the Server (i.e., $\pi(l_1)(l_2) = 0.98$).

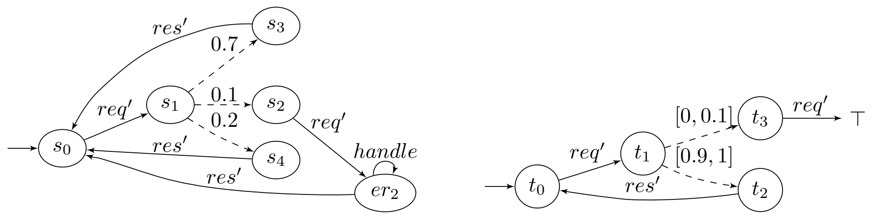
Notation: For convenience, we sometimes write the transition probability function π as a transition relation $\dashrightarrow \subseteq \mathcal{Q}^{\mathsf{p}} \times [0,1] \times \mathcal{Q}$ such that

$$\dashrightarrow = \{(s, p, s') \mid s \in \mathcal{Q}^{\mathsf{p}} \land s' \in \mathcal{Q} \land p = \pi(s)(s')\}$$

We introduce *contracts* as a finite specification for a possibly infinite number of IMCs. In contrast to IMCs, the probabilistic transitions of a contract are labeled with probability *intervals*, similar to [9, 14]. Moreover, a distinct $\top$ state is used to distinguish assumptions on the use of the component from the guarantees it provides.

Definition 3 (Contract)**.** *A contract is a tuple $(\mathcal{Q}, \mathcal{A}, \rightarrow, \sigma, t_0)$ where:*

- *$\mathcal{Q}$ is a nonempty finite set of states, partitioned into $\mathcal{Q} = \mathcal{Q}^{\mathsf{p}} \cup \mathcal{Q}^{\mathsf{a}} \cup \{\top\}$, where $\mathcal{Q}^{\mathsf{p}}$ is the set of probabilistic states, $\mathcal{Q}^{\mathsf{a}}$ is the set of action states, and $\top$ is a distinct state without any outgoing transitions;*
- *A is a finite alphabet of actions;*
- *$\rightarrow \subseteq \mathcal{Q}^{\mathsf{a}} \times \mathcal{A} \times \mathcal{Q}$ is the action transition relation;*
- *$\sigma : \mathcal{Q}^{\mathsf{p}} \to (\mathcal{Q} \to 2^{[0,1]})$ is a transition probability predicate, associating with each pair of states $(s, s') \in \mathcal{Q}^{\mathsf{p}} \times \mathcal{Q}$ an interval of probabilities;*
- *t_0 is the initial state.*



(a) IMC M_s for Server (b) Contract C_s for Server

Figure 2: Contract Examples

Notations: We also write σ as a transition relation $\dashrightarrow \subseteq \mathcal{Q}^{\mathsf{p}} \times 2^{[0,1]} \times \mathcal{Q}$ such that $\dashrightarrow = \{(s, P, s') \mid s \in \mathcal{Q}^{\mathsf{p}} \wedge s' \in \mathcal{Q} \wedge P = \sigma(s)(s')\}$. We write $q \overset{>0}{\dashrightarrow} q'$ if $\exists p > 0 : p \in \sigma(q, q')$, and $\overset{>0}{\dashrightarrow}^{+}$ for the transitive closure of $\overset{>0}{\dashrightarrow}$.

The meaning of a contract over a component C is the following:

- a transition $s \xrightarrow{a} \top$ specifies the *assumption* of the component C that an interaction involving action a does not occur in state s;
- in an action state s, an action a labeling a transition not leading to $\top$ specifies the *guarantee* of the component C that a is enabled in s; conversely, the absence of any outgoing transition labeled with a specifies the guarantee that an interaction involving a will not occur;
- the $\top$ state represents the fact that the assumption has been violated, and henceforth, the component C can show arbitrary, uncontrollable behavior;
- a transition $s \overset{[a,b]}{\dashrightarrow} t$ specifies an interval of allowed transition probabilities.

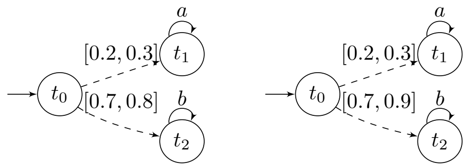
Example 1. *The contract C_s in Figure 2(b) specifies that, after the Server receives a request req', the probability that it reaches state t_3 is within* $[0, 0.1]$*; in state t_3, it* assumes *that the environment does not give req' again; if this occurs, its implementation is not bound by C_s any more; the probability that it reaches t_2 from t_1 is within* $[0.9, 1]$*; in state t_2, it* guarantees *to send a response (res'). In §3, we show how to check that the IMC M_s (in Figure 2(a)) satisfies the contract C_s.*

From the definitions of IMC and contract, we can see that an IMC can be trivially converted into a contract. For this, we define a lifting operator $\lfloor . \rfloor$ (Figure 3 (c)). For the sake of simplicity, we use the same notation $\dashrightarrow$ to represent both kinds of probabilistic transitions (i.e., those in an IMC and in a contract).

The following definition, borrowed from [5], states that, for any probability chosen in the interval of any probabilistic transition, it is always possible to chose probabilities in the intervals of all the remaining transitions outgoing from the same state such that the sum is 1.

Definition 4 (Delimited contract)**.** *A contract $C = (\mathcal{Q}, \mathcal{A}, \rightarrow, \sigma, t_0)$ is delimited [5] iff $\forall s \in \mathcal{Q}^{\mathsf{p}}\ \forall s' \in \mathcal{Q}\ \forall p \in \sigma(s)(s') : 1 - p \in \sum_{s'' \in \mathcal{Q} \setminus \{s'\}} \sigma(s)(s'')$.*

Example 2. *Figure 3(a) shows a delimited contract: for all $p \in [0, 2, 0.3]$, we can find $p' \in [0.7, 0.8]$ such that $p + p' = 1$ and vice versa. Figure 3(b) shows a contract that is not delimited. However, we can* cut *[5] the redundant sub-interval [0.8,0.9] from the interval [0.7,0.9] to obtain the delimited contract of Figure 3(a).*



$$\lfloor s_1 \xrightarrow{\alpha} s_2 \rfloor = s_1 \xrightarrow{\alpha} s_2$$

$$\lfloor s_1 \overset{p}{\dashrightarrow} s_2 \rfloor = s_1 \overset{[p,p]}{\dashrightarrow} s_2$$

(a) Delimited. (b) Non-delimited. (c) Lifting rules.

Figure 3: Delimited contract and rules for lifting IMC to contract.

We define some useful operations related to the probability-interval in Figure 4. Regarding summing up lower bounds and upper bounds, by Definition 4 [Delimited contract], the case that summation of the lower bounds greater than 1 cannot occur. When summing up the upper bounds, the ceiling for a probability value is 1, so if the summation is greater than 1, we let the result be 1. The k is a constant scalar.

$$\begin{array}{rcll}
\lceil n \rceil & = & \text{if } n > 1 \text{ then } 1 \text{ else } n & \\
[l_1, u_1] + [l_2, u_2] & = & [l_1 + l_2, \lceil u_1 + u_2 \rceil] & \text{[F1]} \\
[l_1, u_1] * [l_2, u_2] & = & [l_1 * l_2, u_1 * u_2] & \text{[F2]} \\
k * [l, u] & = & [k * l, k * u] & \text{[F3]} \\
[l, u] * k & = & [l * k, u * k] & \text{[F4]}
\end{array}$$

Figure 4: Operations on Probability Interval(s)

3 Contract Refinement

System synthesis involves refining a contract several times until an implementation is obtained. We therefore define formally the notion of contract refinement.

3.1 Refinement and Satisfaction

We first define contract refinement, and give thereafter some explanations.

Definition 5 (Contract refinement)**.** *Let $C_1 = (\mathcal{Q}_1, \mathcal{A}, \rightarrow_1, \sigma_1, s_0)$ and $C_2 = (\mathcal{Q}_2, \mathcal{A}, \rightarrow_2, \sigma_2, t_0)$ be two contracts. C_1 refines C_2 (written $C_1 \leq C_2$) iff $s_0 \leq t_0$, where $\leq \subseteq \mathcal{Q}_1 \times \mathcal{Q}_2$ is the greatest relation s.t. for all $s \leq t$ we have:*

1. $s = \top \implies t = \top$;

2. If $(s, t) \in \mathcal{Q}_1^{\mathsf{a}} \times (\mathcal{Q}_2^{\mathsf{a}} \cup \{\top\})$ then

(a) $\forall t' \neq \top \in \mathcal{Q}_2,\ (t \xrightarrow{\alpha}_2 t') \implies (\exists s' \in \mathcal{Q}_1,\ s \xrightarrow{\alpha}_1 s' \wedge s' \leq t')$;

(b) $\forall s' \in \mathcal{Q}_1,\ (s \xrightarrow{\alpha}_1 s') \implies (t = \top \ \vee\ \exists t' \in \mathcal{Q}_2,\ t \xrightarrow{\alpha}_2 t' \wedge s' \leq t')$.

3. *If* $(s,t) \in \mathcal{Q}_1^{\mathsf{p}} \times \mathcal{Q}_2^{\mathsf{p}}$ *then there exists a function* $\delta : \mathcal{Q}_1 \times \mathcal{Q}_2 \to [0,1]$*, which, for each* $s' \in \mathcal{Q}_1$*, gives a probability distribution* $\delta(s')$ *over* $\mathcal{Q}_2$*, such that for every probability distribution* f *over* $\mathcal{Q}_1$ *with* $f(s') \in \sigma_1(s)(s')$ *and* $\forall t' \in \mathcal{Q}_2$*,*

$$\sum_{s' \in \mathcal{Q}_1} f(s') {*} \delta(s')(t') \in \sigma_2(t)(t') \quad \text{and} \quad \forall s' \in \mathcal{Q}_1 : \big(\delta(s')(t') > 0 \implies s' \leq t'\big)$$

4. *If* $(s,t) \in \mathcal{Q}_1^{\mathsf{a}} \times \mathcal{Q}_2^{\mathsf{p}}$ *then* $\exists t^{\mathsf{a}} \in \mathcal{Q}_2^{\mathsf{a}} : t \overset{>0}{\dashrightarrow}{}_2^{+}\ t^{\mathsf{a}} \wedge s \leq t^{\mathsf{a}}$ *and* $\forall t' \in \mathcal{Q}_2$*,* $\big(t \overset{>0}{\dashrightarrow}_2 t' \implies s \leq t'\big)$.

5. *If* $(s,t) \in \mathcal{Q}_1^{\mathsf{p}} \times \mathcal{Q}_2^{\mathsf{a}}$ *then* $\exists s^{\mathsf{a}} \in \mathcal{Q}_1^{\mathsf{a}} : s \overset{>0}{\dashrightarrow}{}_1^{+}\ s^{\mathsf{a}} \wedge s^{\mathsf{a}} \leq t$ *and* $\forall s' \in \mathcal{Q}_1$*,* $\big(s \overset{>0}{\dashrightarrow}_1 s' \implies s' \leq t\big)$.

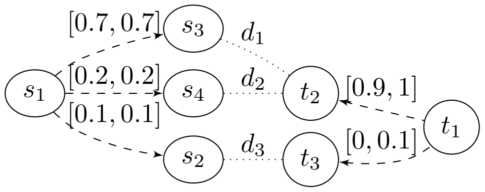
In Definition 5, condition (1) ensures that C_1 makes no stronger assumptions on the context than C_2. Condition (2a) says that any transition accepted by C_2 must also be accepted by C_1. However, unexpected transitions (i.e. transitions leading to $\top$) do not need to be present in the refinement. That is why we have $\forall t' \neq \top$ in Condition (2a). On the other hand, condition (2b) says that each action transition of C_1 must also be enabled in C_2, unless C_2 is in the $\top$ state. Condition (3), adapted from [9], deals with refinement among probabilistic states. Intuitively, $s \leq t$ if there exists a function δ which distributes the probabilities of transitions from s to s' onto the transitions from t to t', such that the sum of the probability fractions (i.e., $f(s') * \delta(s')(t')$) is in the range $\sigma_2(t)(t')$, as illustrated in Example 3 below. Condition (4) says that an action state s refines a probabilistic state t if it refines all action states reachable with a path of positive probability from t. Finally, condition (5) is symmetrical to condition (4).

Before giving an example of refinement, we define the satisfaction of a contract by an implementation (an IMC) as the refinement of the contract by the lifted IMC (i.e., written in the form of a contract).

Definition 6 (Contract satisfaction)**.** *An IMC* M *satisfies a contract* C *(written* $M \models C$*) iff* $\lfloor M \rfloor \leq C$*.*

Example 3. *We illustrate in Figure 5 how to check* $\lfloor M_s \rfloor \leq C_s$*, in particular,* $s_1 \leq t_1$*. It is easy to check* $s_3 \leq t_2$*,* $s_4 \leq t_2$*, and* $s_2 \leq t_3$*. In Figure 5, dashed lines stand for non-negative distribution* δ*. Condition (3) in Definition 5 states that* $s_1 \leq t_1$ *if, for each successor of* s_1*, there is a function* δ *(i.e., three real numbers* d_1*,* d_2*, and* d_3*) such that, for each tuple* (p_2, p_3, p_4) *satisfying the constraints (1) to (4) in Figure 5, the constraints (5) and (6) are implied. Condition (3) can be checked efficiently by requiring the set inclusion to hold for the bounds of interval* $\sigma(s)(s')$*, using a linear programming solver. As* $\delta(s')$ *is a probability distribution, we obtain for our example* $d_1 = d_2 = d_3 = 1$*. (Note that if we had* $s_2 \leq t_2$ *as well, say, we had* d_4 *from* s_2 *to* t_2*, we would have another constraint* $d_3 + d_4 = 1$*.)*

Lemma 1 (Reflexivity of refinement)**.** *For all contracts* $C = (\mathcal{Q}, \mathcal{A}, \to, \sigma, s_0)$*,* $C \leq C$ *and for all states* $s \in \mathcal{Q}$*,* $s \leq s$*.*



(1) $p_2 \in [0.1, 0.1]$
(2) $p_3 \in [0.7, 0.7]$
(3) $p_4 \in [0.2, 0.2]$
(4) $p_2 + p_3 + p_4 = 1$
(5) $p_3 * d_1 + p_4 * d_2 \in [0.9, 1]$
(6) $p_2 * d_3 \in [0, 0.1]$

Figure 5: Left: Contract refinement $s_1 \leq t_1$. Right: Constraints to be checked.

Proof. Definition 5 (1) and (2) are satisfied. Definition 5 (3) is satisfied with $\delta(\_)(\_) = 1$. □

Lemma 2 (Transitivity of refinement)**.** *For all contracts C_1, C_2 and C_3, if $C_1 \leq C_2$ and $C_2 \leq C_3$, then $C_1 \leq C_3$.*

Proof. See appendix A.1. □

Lemma 3 (Refinement ($\leq$) is a preorder)**.** *The relation $\leq$ over two contracts is a partial order.*

Proof. The relation $\leq$ is reflexive (Lemma 1) and transitive (Lemma 2). □

Definition 7 (Models of contracts)**.** *The set of models of a contract C (written $\mathcal{M}(C)$) is the set of IMCs that satisfy C: $\mathcal{M}(C) = \{M \mid M \models C\}$.*

Definition 8 (Semantical equivalence)**.** *Contracts C_1 and C_2 are semantically equivalent (written $C_1 \equiv C_2$) iff $\mathcal{M}(C_1) = \mathcal{M}(C_2)$.*

Lemma 4 (Monotonicity of satisfaction)**.** *For all IMC M and contracts C_1 and C_2, if $M \models C_1$ and $C_1 \leq C_2$, then $M \models C_2$.*

Proof.

$$\begin{array}{cl}
 & M \models C_1 \text{ and } C_1 \leq C_2 \\
\Longleftrightarrow & \text{(By Definition 6 [Contract satisfaction] } (\models)\text{)} \\
 & \lfloor M \rfloor \leq C_1 \text{ and } C_1 \leq C_2 \\
\Rightarrow & \text{(By Lemma 2 [Transitivity of } \leq\text{])} \\
 & \lfloor M \rfloor \leq C_2 \\
\Longleftrightarrow & \text{(By Definition 6 [Contract satisfaction] } (\models)\text{)} \\
 & M \models C_2
\end{array}$$

□

Lemma 5 (Refinement and model inclusion)**.** *For all contracts C_1 and C_2, $C_1 \leq C_2 \implies \mathcal{M}(C_1) \subseteq \mathcal{M}(C_2)$.*

Proof. ($\Rightarrow$) We prove it by contradiction. Suppose that $C_1 \leq C_2 \not\Rightarrow \mathcal{M}(C_1) \subseteq \mathcal{M}(C_2)$. Then there exists an IMC M such that $M \models C_1$ and $C_1 \leq C_2$, but $M \not\models C_2$. By Lemma 4, for all IMCs $M \models C_1$ and $C_1 \leq C_2$, then $M \models C_2$. We reach a contradiction. So our assumption is false. Thus, we have the desired result. □

Remark: the converse of Lemma 5 does not hold, as shown by the counter example in Figure 6.

We can see that there is no model for C_1, i.e. $\mathcal{M}(C_1) = \emptyset$, while there are models for C_2. Thus, we have $\mathcal{M}(C_1) \subseteq \mathcal{M}(C_2)$. It is obvious that $C_1 \not\leq C_2$.

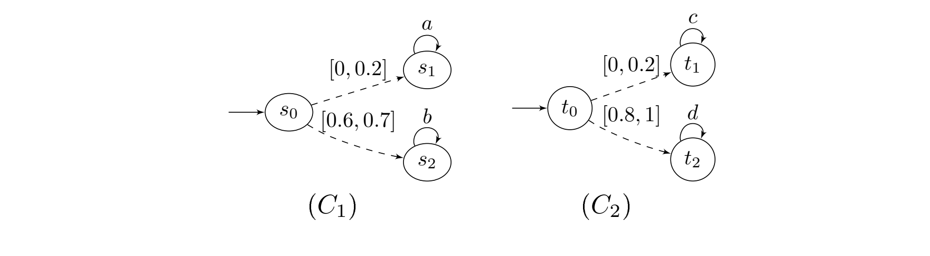


Figure 6: Counter example for Lemma 5.

3.2 Bisimulation

We adapt the usual notion of bisimulation to contracts, and define reduction of a contract with respect to bisimulation.

Definition 9 (Bisimulation $\simeq$)**.** *Given a contract $C = (\mathcal{Q}, \mathcal{A}, \rightarrow, \dashrightarrow, s_0)$, let $\simeq \subseteq \mathcal{Q} \times \mathcal{Q}$ be the greatest relation such that if $s \simeq t$ then:*

1. $s = \top \iff t = \top$;

2. If $(s, t) \in \mathcal{Q}^{\mathsf{a}} \times \mathcal{Q}^{\mathsf{a}}$ *then*

  (a) $\forall \alpha \in \mathcal{A} \quad \forall s' \in \mathcal{Q}, (s \xrightarrow{\alpha} s' \implies \exists t' \in \mathcal{Q}, (t \xrightarrow{\alpha} t' \wedge s' \simeq t'))$

  (b) $\forall \alpha \in \mathcal{A} \quad \forall t' \in \mathcal{Q}, (t \xrightarrow{\alpha} t' \implies \exists s' \in \mathcal{Q}, (s \xrightarrow{\alpha} s' \wedge s' \simeq t'))$

3. If $(s, t) \in \mathcal{Q}^{\mathsf{p}} \times \mathcal{Q}^{\mathsf{p}}$ *then*

  (a) there is a function $\delta : \mathcal{Q} \times \mathcal{Q} \rightarrow [0, 1]$, which for each $s' \in \mathcal{Q}$ gives a probability distribution $\delta(s')$ on $\mathcal{Q}$, s.t. for every probability distribution f over $\mathcal{Q}$ with $f(s') \in \sigma(s)(s')$ and $\forall t' \in \mathcal{Q}$

$$\sum_{s' \in \mathcal{Q}} f(s') * \delta(s', t') \in \sigma(t)(t') \quad \text{and} \quad \forall s' \in \mathcal{Q} : \big(\delta(s', t') > 0 \implies s' \simeq t'\big)$$

  (b) symmetric to (3a);

4. If $(s, t) \in \mathcal{Q}^{\mathsf{a}} \times \mathcal{Q}^{\mathsf{p}}$ then $\exists t^{\mathsf{a}} \in \mathcal{Q}^{\mathsf{a}} : t \overset{>0}{\dashrightarrow}{}^{+} t^{\mathsf{a}} \wedge s \simeq t^{\mathsf{a}}$ and $\forall t' \in \mathcal{Q}$, $t \overset{>0}{\dashrightarrow} t' \implies s \simeq t'$;

5. If $(s, t) \in \mathcal{Q}^{\mathsf{p}} \times \mathcal{Q}^{\mathsf{a}}$ then $\exists s^{\mathsf{a}} \in \mathcal{Q}^{\mathsf{a}} : s \overset{>0}{\dashrightarrow}{}^{+} s^{\mathsf{a}} \wedge s^{\mathsf{a}} \simeq t$ and $\forall s' \in \mathcal{Q}$, $s \overset{>0}{\dashrightarrow} s' \implies s' \simeq t$.

In Definition 9, condition (2) is the standard definition for bisimulation. Conditions (3a) and (3b) deal with the probabilistic transitions. Finally, conditions (4) and (5) say that an action state is bisimilar with a probabilistic state if it is bisimilar with all its successors with non-zero probability, and there exists at least one action state that is reachable from this probabilistic state.

Definition 10 (Reduction modulo $\simeq$)**.** *Let $C = (\mathcal{Q}, \mathcal{A}, \rightarrow, \sigma, s_0)$ be a contract. For all $s \in \mathcal{Q}$, let $\mathcal{C}_s = \{q \in \mathcal{Q} \mid s \simeq q\}$ be the equivalence class of s. Let $\mathcal{C} = \{\mathcal{C}_s \mid s \in \mathcal{Q}\}$. The reduced contract, written $\overline{C}$, is $(\mathcal{C}, \mathcal{A}, \rightarrow_{\simeq}, \sigma_{\simeq}, \mathcal{C}_{s_0})$ such that, $\forall s = \{s_1, \ldots, s_m\}, t = \{t_1, \ldots, t_n\} \in \mathcal{C}$, we have: (1) $s \xrightarrow{a}_{\simeq} t$ iff $\exists i, j : s_i \xrightarrow{a} t_j$, and (2) $\sigma_{\simeq}(s, t) = \sum_{1 \leq j \leq n} \sigma(s_1, t_j))$.*

Notice that an equivalence class may contain both action and probabilistic states. By Definition 9, except for probabilistic transitions with probability interval $[0, 0]$, either all transitions leaving an equivalence class are action transitions and Definition 9 (2) applies, or they are all probabilistic transitions and Definition 9 (3) applies as follows. For each probabilistic state $s_i \in s$, the probabilities of transitions to states $t_j \in t$ are summed up (it does not matter which of the transitions is taken since all the successors t_j are equivalent). This sum is the transition probability from s_i to some state in t. By definition of $\simeq$, the sum is the same for all $s_i \in s$, thus we pick $\sigma(s_1, t_j)$.

Example 4. *We can reduce the contract C_2 of Figure 9(c) by combining the bisimilar states t_2 and t_3 into one: $t_1 \overset{[0.2,0.6]}{\dashrightarrow} \{t_2, t_3\}$.*

Lemma 6 (Reduction and refinement)**.** *For all delimited contracts C, $\overline{C} \leq C$ and $C \leq \overline{C}$.*

Proof. See appendix A.2. □

Lemma 7 (Refinement and equivalence)**.** *For all contracts C_1 and C_2, $C_1 \leq C_2$ and $C_2 \leq C_1$ implies $C_1 \equiv C_2$.*

Proof.

$$\begin{array}{ll}
 & C_1 \leq C_2 \text{ and } C_2 \leq C_1 \\
\Rightarrow & \text{(By Lemma 5 [Refinement and model inclusion])} \\
 & \forall M,\ M \models C_1 \Rightarrow M \models C_2 \text{ and } \forall M,\ M \models C_2 \Rightarrow M \models C_1 \\
\iff & \text{(By Logic: } (\forall m, P(m) \Rightarrow Q(m) \wedge \forall m, Q(m) \Rightarrow P(m)) \\
 & \equiv \forall m, P(m) \iff Q(m)) \\
 & \forall M,\ M \models C_1 \iff M \models C_2 \\
\iff & \text{(By Definition 8 [Semantical equivalence] } (\equiv)) \\
 & C_1 \equiv C_2
\end{array}$$

□

Lemma 8 (Model equivalence)**.** *For all delimited contracts C, $\overline{C} \equiv C$.*

Proof. By Lemma 6 and Lemma 7 [Refinement and Equivalence]. □

3.3 Contract Abstraction

The need of abstraction arises naturally in contract frameworks. We abstract actions in $\mathcal{A} \setminus \mathcal{B}$ that are not relevant by renaming them into internal τ actions. The contract over the alphabet $\mathcal{B} \cup \{\tau\}$ is then projected on the sub-alphabet $\mathcal{B}$ by using the standard determinization algorithm (see e.g. [1]).

Definition 11 (Projection)**.** *Let $C = (\mathcal{Q}, \mathcal{A}, \rightarrow_1, \sigma, s_0)$ be a contract and $\mathcal{B} \subseteq \mathcal{A}$ such that for any $q \in \mathcal{Q}^{\mathsf{a}}$ and $a \in \mathcal{A}$, if $q \xrightarrow{a}_1 \top$ then $a \in \mathcal{B}$. Let $C' = (\mathcal{Q}, \mathcal{B} \cup \{\tau\}, \rightarrow_2, \sigma, s_0)$ be the contract where all transition labels in $\mathcal{A} \setminus \mathcal{B}$ are replaced with τ. The projection of C on $\mathcal{B}$ (written $\pi_{\mathcal{B}}(C)$) is obtained by τ-elimination (determinization) of C'.*

Example 5. *In Figure 2, if we do not care how the implementation handles failure cases, we can check that $\pi_{\mathcal{A}_s \setminus \{handle\}}(M_s) \models C_s$.*

Lemma 9 (Abstraction and refinement)**.** *For all contracts* $C_1 = (\mathcal{Q}_1, \mathcal{A}, \rightarrow_1, \dashrightarrow_1, s_0)$*,* $C_2 = (\mathcal{Q}_2, \mathcal{A}, \rightarrow_2, \dashrightarrow_2, t_0)$ *and* $\mathcal{B} \subseteq \mathcal{A}$*, if* $C_1 \leq C_2$*, then* $\pi_{\mathcal{B}}(C_1) \leq \pi_{\mathcal{B}}(C_2)$*.*

Proof. See appendix A.3. □

4 Contract Composition

We introduce two composition operations for contracts: parallel composition $||$, parametrized with an interaction set $\mathcal{I}$, and conjunction $\wedge$ (shared refinement).

4.1 Parallel Composition of Contracts

Parallel composition allows the designer to build complex models from simpler components in a stepwise and hierarchical manner. In order to reason about the composition of components at the contract level, we introduce the parallel composition of contracts. As in the BIP component framework [7], parallel composition is parametrized with a set of interactions, where each interaction is a set of component actions occurring simultaneously. For instance, an interaction set $\{a, a|b, c\}$ says that action a can interleave or synchronize with b; action b must synchronize with a; action c is a singleton interaction that always interleaves. The symbol "|" is commutative, which means that $a|b$ is identical to $b|a$. In Figure 7, the interactions α and β are of the form c, $a|b$, or $a|b|d$, and so on.

Definition 12 (Parallel composition of contracts)**.** *Let* $C_1 = (\mathcal{Q}_1, \mathcal{A}_1, \rightarrow_1, \dashrightarrow_1, s_0)$ *and* $C_2 = (\mathcal{Q}_2, \mathcal{A}_2, \rightarrow_2, \dashrightarrow_2, t_0)$ *be two contracts. The parallel composition of* C_1 *and* C_2 *on interaction set* $\mathcal{I}$ *(written* $C_1||_{\mathcal{I}}C_2$*) is the contract* $\big(\mathcal{Q}, \mathcal{I}, \rightarrow, \dashrightarrow, (s_0, t_0)\big)$ *where:*

1. $\mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2$ *with* $\top = (\mathcal{Q}_1 \times \{\top_2\}) \cup (\{\top_1\} \times \mathcal{Q}_2)$ *— that is,* $\top$ *of* $C_1||_{\mathcal{I}}C_2$ *is an aggregate state reached as soon as* C_1 *or* C_2 *reaches its* $\top_i$ *state —,* $\mathcal{Q}^{\mathsf{a}} = \mathcal{Q}_1^{\mathsf{a}} \times \mathcal{Q}_2^{\mathsf{a}}$*, and* $\mathcal{Q}^{\mathsf{p}} = \mathcal{Q} \setminus (\mathcal{Q}^{\mathsf{a}} \cup \top)$*;*

2. $\rightarrow$ *is the least relation satisfying the rules* [R1]–[R3] *in Figure 7; and*

3. $\dashrightarrow$ *is the least relation satisfying the rules* [R4]–[R6] *in Figure 7.*

$$\frac{q_1 \xrightarrow{\alpha}_1 q_1' \quad \alpha \in \mathcal{I} \qquad q_2 \in \mathcal{Q}_2^a}{(q_1, q_2) \xrightarrow{\alpha} (q_1', q_2)}\ [\mathrm{R1}] \qquad \frac{q_2 \xrightarrow{\alpha}_2 q_2' \quad \alpha \in \mathcal{I} \qquad q_1 \in \mathcal{Q}_1^a}{(q_1, q_2) \xrightarrow{\alpha} (q_1, q_2')}\ [\mathrm{R2}]$$

$$\frac{q_1 \xrightarrow{\alpha}_1 q_1' \quad q_2 \xrightarrow{\beta}_2 q_2' \quad \alpha|\beta \in \mathcal{I}}{(q_1, q_2) \xrightarrow{\alpha|\beta} (q_1', q_2')}\ [\mathrm{R3}] \qquad \frac{q_1 \overset{[p1,p2]}{\dashrightarrow}{}_1\, q_1' \quad q_2 \overset{[p3,p4]}{\dashrightarrow}{}_2\, q_2'}{(q_1, q_2) \overset{[p_1*p_3,p_2*p_4]}{\dashrightarrow} (q_1', q_2')}\ [\mathrm{R4}]$$

$$\frac{q_1 \overset{P}{\dashrightarrow}_1 q_1' \qquad q_2 \in \mathcal{Q}_2^a}{(q_1, q_2) \overset{P}{\dashrightarrow} (q_1', q_2)}\ [\mathrm{R5}] \qquad \frac{q_2 \overset{P}{\dashrightarrow}_2 q_2' \qquad q_1 \in \mathcal{Q}_1^a}{(q_1, q_2) \overset{P}{\dashrightarrow} (q_1, q_2')}\ [\mathrm{R6}]$$

Figure 7: Rules for the parallel composition of contracts.

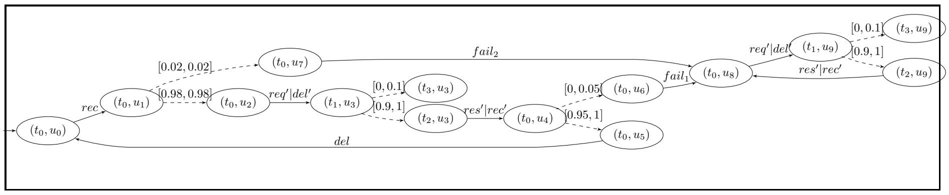


Figure 8: Parallel composition of C_s and C_ℓ.

Rules [R1] to [R3] are the usual parallel composition rules for interactive processes, while Rule [R4] is similar to the typical parallel composition for Markov Chains but on probability intervals. Finally, Rules [R5] and [R6] state that probabilistic transitions, usually modeling hidden internal behavior, have priority over action transitions.

Example 6. *Figure 8 illustrates the parallel composition of contracts C_s (from Figure 2(b)) and $C_\ell = \lfloor M_\ell \rfloor$ (where M_ℓ is given in Figure 1(b)), with $\mathcal{I} = \{rec, del, req'|del', res'|rec', fail_1, fail_2\}$. The composed contract $C_s \;||_\mathcal{I}\; C_\ell$ states that a failure in the Link component does not prevent it from continuing to deliver the request req' to the Server, and receiving the response res' from the Server, but the failure prevents it from delivering the response res' back to the Client.*

We end the section on parallel composition with some nice properties.

Lemma 10 (Commutativity of $||_\mathcal{I}$). *For all contracts C_1, C_2 and interaction set $\mathcal{I}$, $C_1||_\mathcal{I}C_2 = C_2||_\mathcal{I}C_1$.*

Proof. It is obvious as rules for parallel composition are symmetrically defined. □

Theorem 1 (Congruence of refinement for $||_\mathcal{I}$). *For all contracts C_1, C_2, C_3, C_4 and an interaction set $\mathcal{I}$, if $C_1 \leq C_2$ and $C_3 \leq C_4$, then $C_1||_\mathcal{I}\; C_3 \;\leq\; C_2||_\mathcal{I}\; C_4$.*

Proof. See appendix B.1. □

Theorem 2 (Independent implementability). *For all IMCs M, N, contracts C_1, C_2, and interaction set $\mathcal{I}$, if $M \models C_1$ and $N \models C_2$, then $M||_\mathcal{I}N \;\models\; C_1||_\mathcal{I}C_2$.*

Proof.

$$
\begin{array}{cl}
 & M \models C_1 \text{ and } N \models C_2 \\
\iff & \text{(By definition of } \models\text{)} \\
 & \lfloor M \rfloor \leq C_1 \text{ and } \lfloor N \rfloor \leq C_2 \\
\Rightarrow & \text{(By Theorem 1 [Congruence for parallel composition])} \\
 & \lfloor M \rfloor ||_\mathcal{I} \lfloor N \rfloor \leq C_1||_\mathcal{I}C_2
\end{array}
$$

□

Theorem 3 (Reduction and parallel composition). *For all delimited contracts C_1 and C_2, $\overline{C_1}\; ||_\mathcal{I}\; \overline{C_2} \equiv C_1 \;||_\mathcal{I}\; C_2$.*

Proof.

$$\begin{array}{ll}
 & \text{(By Lemma 6 [Reduction and Refinement])} \\
 & \overline{C_1} \leq C_1 \text{ and } \overline{C_2} \leq C_2 \text{ and } C_1 \leq \overline{C_1} \text{ and } C_2 \leq \overline{C_2} \\
\Rightarrow & \text{(By Theorem 1 [Congruence for parallel composition])} \\
 & \overline{C_1}||_{\mathcal{I}}\overline{C_2} \leq C_1||_{\mathcal{I}}C_2 \text{ and } C_1||_{\mathcal{I}}C_2 \leq \overline{C_1}||_{\mathcal{I}}\overline{C_2} \\
\Rightarrow & \text{(By Lemma 7 [Refinement and Equivalence])} \\
 & \overline{C_1}||_{\mathcal{I}}\overline{C_2} \equiv C_1||_{\mathcal{I}}C_2
\end{array}$$

□

4.2 Conjunction of contracts

A single component may have to satisfy several contracts that are specified independently, each of them specifying different requirements on the component, such as safety, reliability, and quality of service aspects. Therefore, the contracts may use different, possibly overlapping, sub-alphabets of the component. The *conjunction* of contracts computes a common refinement of all contracts. Prior to conjunction, we define *similarity* of contracts as a test whether a common refinement exists.

Definition 13 (Similarity ($\sim$)). *Let $C_1 = (\mathcal{Q}_1, \mathcal{A}_1, \rightarrow_1, \dashrightarrow_1, s_0)$ and $C_2 = (\mathcal{Q}_2, \mathcal{A}_2, \rightarrow_2, \dashrightarrow_2, t_0)$ be two contracts. $\sim \subseteq Q_1 \times Q_2$ is the largest relation such that $\forall (s,t) \in \mathcal{Q}_1 \times \mathcal{Q}_2$, $s \sim t$ iff $(s = \top \vee t = \top)$ or conditions (1) to (4) below hold:*

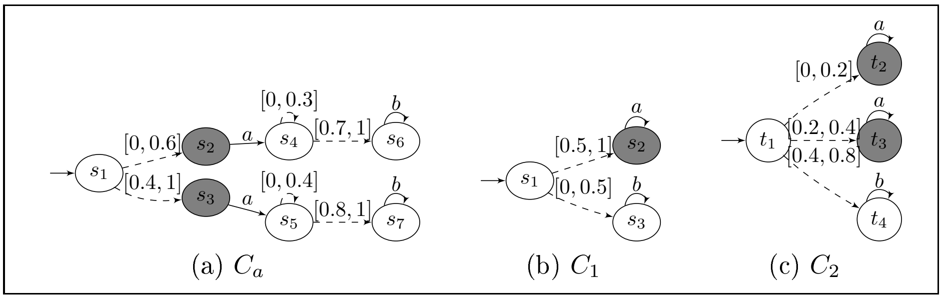
1. *If $(s,t) \in \mathcal{Q}_1^{\mathsf{a}} \times \mathcal{Q}_2^{\mathsf{a}}$ then*
   (a) *for all $s' \in \mathcal{Q}_1$, if $s \xrightarrow{a} s'$, then either $t \xrightarrow{a} t'$ for some $t' \in \mathcal{Q}_2$ and $s' \sim t'$, or $a \notin \mathcal{A}_2$ and $s' \sim t$; and*
   (b) *for all $t' \in \mathcal{Q}_2$, if $t \xrightarrow{a} t'$, then either $s \xrightarrow{a} s'$ for some $s' \in \mathcal{Q}_1$ and $s' \sim t'$, or $a \notin \mathcal{A}_1$ and $s \sim t'$;*
2. *If $(s,t) \in \mathcal{Q}_1^{\mathsf{p}} \times \mathcal{Q}_2^{\mathsf{p}}$ then*
   (a) *for all $s' \in \mathcal{Q}_1$, if $s \overset{P_1}{\dashrightarrow} s'$, then $t \overset{P_2}{\dashrightarrow} t'$ for some $t' \in \mathcal{Q}_2$ with $P_1 \cap P_2 \neq \emptyset$ and $(s' \sim t' \ \vee \ 0 \in P_1 \cap P_2)$;*
   (b) *for all $t' \in \mathcal{Q}_2$, if $t \overset{P_2}{\dashrightarrow} t'$, then $s \xrightarrow{P_1} s'$ for some $s' \in \mathcal{Q}_1$ with $P_1 \cap P_2 \neq \emptyset$ and $(s' \sim t' \ \vee \ 0 \in P_1 \cap P_2)$;*
3. *If $(s,t) \in \mathcal{Q}_1^{\mathsf{a}} \times \mathcal{Q}_2^{\mathsf{p}}$ then for all $t' \in \mathcal{Q}_2$ with $t \overset{P}{\dashrightarrow}_2 t'$, $(s \sim t' \quad \vee \quad 0 \in P)$.*
4. *If $(s,t) \in \mathcal{Q}_1^{\mathsf{p}} \times \mathcal{Q}_2^{\mathsf{a}}$ then for all $s' \in \mathcal{Q}_1$ with $s \overset{P}{\dashrightarrow}_1 s'$, $(s' \sim t \quad \vee \quad 0 \in P)$;*

Finally, C_1 and C_2 are similar, written $C_1 \sim C_2$, iff $s_0 \sim t_0$.

P_i in Definition 13 refers to a probabilistic interval in the form of $[\ell_i, u_i]$. Any state is similar to the top state $\top$ (where the contract does not constrain the implementation in any way). Two action states are similar if they agree on the enabled actions in the common alphabet, and the successor states are similar again. Two probabilistic states are similar if the probabilistic transitions can be matched such that the intervals overlap, and the successor states are either similar, or can be made unreachable by refining the probability interval to $[0,0]$.

Definition 14 (Unambiguous contract). *A contract $C = (\mathcal{Q}, \mathcal{A}, \rightarrow, \dashrightarrow, s_0)$ is unambiguous iff for all $r, s, t \in \mathcal{Q}$, if $r \overset{>0}{\dashrightarrow} s \wedge r \overset{>0}{\dashrightarrow} t \wedge s \sim t$, then $s = t$.*

A contract is *unambiguous* if the reachable successor states of any probabilistic state are pairwise non-similar.



(a) C_a (b) C_1 (c) C_2

Figure 9: (a) An ambiguous contract C_a; (b,c) Two non-similar contracts C_1 and C_2.

Example 7. *In Figure 9(a), the contract C_a is ambiguous because $s_2 \sim s_3$ (highlighted in gray) but $s_2 \neq s_3$.*

We are now ready to define the conjunction of contracts.

Definition 15 (Conjunction of contracts ($\wedge$)). *For unambiguous contracts $C_1 = (\mathcal{Q}_1, \mathcal{A}_1, \rightarrow_1, \dashrightarrow_1, s_0)$ and $C_2 = (\mathcal{Q}_2, \mathcal{A}_2, \rightarrow_2, \dashrightarrow_2, t_0)$ such that C_1 and C_2 are similar, let $C_1 \wedge C_2$ be the contract $\big(\mathcal{Q}_1 \times \mathcal{Q}_2, \mathcal{A}_1 \cup \mathcal{A}_2, \rightarrow, \dashrightarrow, (s_0, t_0)\big)$ where $\top = (\top_1, \top_2)$ and*

1. *$\rightarrow$ is the least relation satisfying the rules* [C1] – [LiftR] *in Figure 10, and*
2. *$\dashrightarrow$ is the least relation satisfying the rules* [C3] – [C4R] *in Figure 10 (where for all other probabilistic transitions $(q_1, q_2) \overset{P}{\dashrightarrow} (q_1', q_2')$, $P =$* $[0, 0]$*).*

Rule [C1] requires the contracts to agree on action transitions over the common alphabet. According to rule [C2L] (resp. [C2R]), the conjunction behaves like the first (resp. second) contract as soon as the other contract is in $\top$. Rules [LiftL] and [LiftR] allow the interleaving of action transitions that are not in the common alphabet. Rules [C3] – [C4R] define the probabilistic transitions whose successor states are similar. For non-similar successor states, the probability interval is refined to $[0, 0]$, according to Definition 15.

Example 8. *Figure 11 shows three contracts for the Link component: $C_{\ell 1}$ specifies that the implementation should receive a request (req) from the Client and deliver it to the Server (del′); $C_{\ell 2}$ specifies that the implementation should receive a response (rec′) from the Server and deliver it to the Client (del); $C_{\ell 3}$ requires the response (rec′) received from the Server to occur after the request (del′) delivered to the Server. We can verify that $M_\ell \models (C_{\ell 1} \wedge C_{\ell 3}) \wedge (C_{\ell 2} \wedge C_{\ell 3})$ (where M_ℓ is in Figure 1(b)).*

$$\frac{q_1 \xrightarrow{\alpha}_1 q_1' \quad q_2 \xrightarrow{\alpha}_2 q_2'}{(q_1, q_2) \xrightarrow{\alpha} (q_1', q_2')} \quad [\text{C1}]$$

$$\frac{q_1 \xrightarrow{\alpha}_1 q_1'}{(q_1, \top) \xrightarrow{\alpha} (q_1', \top)} \quad [\text{C2L}] \qquad \frac{q_2 \xrightarrow{\alpha}_1 q_2'}{(\top, q_2) \xrightarrow{\alpha} (\top, q_2')} \quad [\text{C2R}]$$

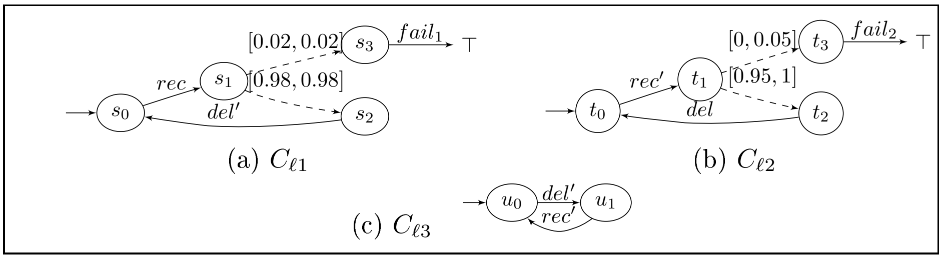
$$\frac{q_1 \xrightarrow{\alpha}_1 q_1' \quad \alpha \notin \mathcal{A}_2 \quad q_2 \in \mathcal{Q}_2^a}{(q_1, q_2) \xrightarrow{\alpha} (q_1', q_2)} \quad [\text{LiftL}]$$

$$\frac{q_2 \xrightarrow{\alpha}_2 q_2' \quad \alpha \notin \mathcal{A}_1 \quad q_1 \in \mathcal{Q}_1^a}{(q_1, q_2) \xrightarrow{\alpha} (q_1, q_2')} \quad [\text{LiftR}]$$

$$\frac{q_1 \overset{P_1}{\dashrightarrow}_1 q_1' \quad q_2 \overset{P_2}{\dashrightarrow}_2 q_2' \quad q_1' \sim q_2'}{(q_1, q_2) \overset{P_1 \cap P_2}{\dashrightarrow} (q_1', q_2')} \quad [\text{C3}]$$

$$\frac{\begin{array}{c} q_1 \overset{P}{\dashrightarrow}_1 q_1' \quad q_2 \in \mathcal{Q}_2^a \cup \{\top\} \\ q_1' \sim q_2 \end{array}}{(q_1, q_2) \overset{P}{\dashrightarrow} (q_1', q_2)} \quad [\text{C4L}] \qquad \frac{\begin{array}{c} q_2 \overset{P}{\dashrightarrow}_2 q_2' \quad q_1 \in \mathcal{Q}_1^a \cup \{\top\} \\ q_1 \sim q_2' \end{array}}{(q_1, q_2) \overset{P}{\dashrightarrow} (q_1, q_2')} \quad [\text{C4R}]$$

Figure 10: Rules for conjunction of contracts.

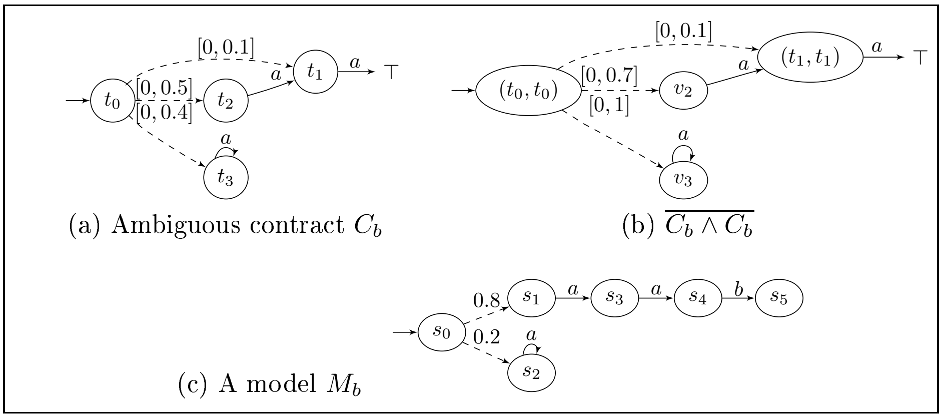


(a) $C_{\ell 1}$

(b) $C_{\ell 2}$

(c) $C_{\ell 3}$

Figure 11: Example: Conjunction of Contracts



(a) Ambiguous contract C_b

(b) $\overline{C_b \wedge C_b}$

(c) A model M_b

Figure 12: Example where $M_b \models C_b \wedge C_b$ but $M_b \not\models C_b$.

Example 9. *As a contract that is not in reduced form is not unambiguous, contracts should be reduced before performing conjunction. In Figure 9 (left), contract C_2 is non unambiguous, but $t_2 \simeq t_3$. If we reduce C_2 by applying*

Definition 10, we get $t_1 \overset{[0.2,0.6]}{\dashrightarrow} \{t_2, t_3\} \xrightarrow{a} \{t_2, t_3\}$. *The reduced contract is unambiguous and* $s_1 \sim t_1$, *such that conjunction yields a common refinement of* C_1 *and* C_2.

Theorem 4 (Conjunction is a common refinement)**.** *For all contracts* C_1 *and* C_2, $\pi_{\mathcal{A}_i}(C_1 \wedge C_2) \leq C_i$ *for* $i = 1, 2$.

Proof. See appendix B.2. □

Theorem 5 (Soundness of conjunction)**.** *For any IMC* M *and unambiguous contracts* C_i *with alphabets* $\mathcal{A}_i$, $i = 1, 2$, *such that* $C_1 \sim C_2$, *if* $M \models C_1 \wedge C_2$ *then* $\pi_{\mathcal{A}_i}(M) \models C_i$, $i = 1, 2$.

Proof. See appendix B.2. □

Example 10. *Figure 12 motivates the requirement of conjunction (Definition 15) for unambiguous contracts. The resulting contract* $C_b \wedge C_b$ *is reduced such that the model relation can be seen easily. The node* v_2 *denotes the equivalent class* $\{(s_1, s_2), (s_2, s_1), (s_2, s_2)\}$; *the node* v_3 *denotes the equivalent class* $\{(s_1, s_3), (s_2, s_3), (s_3, s_1), (s_3, s_2), (s_3, s_3)\}$. *As* $t_1 \sim t_2 \sim t_3$, *duplicated intervals lead to an unsound result.*

It is interesting to note the similarity of conjunction with discrete controller synthesis [13], in the sense that conjunction is a refinement of both contracts making bad states (i.e., pairs of states where both contracts are contradictory) unreachable. In this analogy, both action transitions and probabilistic transitions with strictly positive intervals amount to uncontrollable transitions, whereas transitions whose probability interval contains 0 amount to controllable transitions that can be refined to $[0, 0]$ so as to make bad states unreachable.

5 Case Study

We study a dependable computing system with time redundancy. The system specification is expressed by the contract C_S of Figure 13 (top left), which specifies that the computation *comp* should have a success probability of at least 0.999. If the computation fails, then nothing is specified (state $\top$).

The processor P the system is running on is specified by the contract C_P of Figure 13 (top right). Following an execution request *exe*, either the processor succeeds and replies with *ok* (with a probability at least p), or fails and replies with *nok* (with a probability at most $1 - p$). The failure rates for successive executions are independent. The probability p is a *parameter* of the contract.

We place ourselves in a setting where the reliability level guaranteed by C_P alone (as expressed by p) cannot fulfill the requirement of C_S (that is, 0.999), and hence some form of redundancy must be used. We propose to use time redundancy, as expressed by the contract C_T of Figure 13 (bottom). Each computation *comp* is first launched on the processor P (*exe*$'$), either followed by a positive (*ok*$'$) or negative (*nok*$'$) answer from P. In the latter case, the execution is launched a second time, therefore implementing time redundancy. The contract C_T finally answers with *success* if *either* execution is followed by *ok*$'$, or with $fail$ is *both* executions are followed by *nok*$'$.

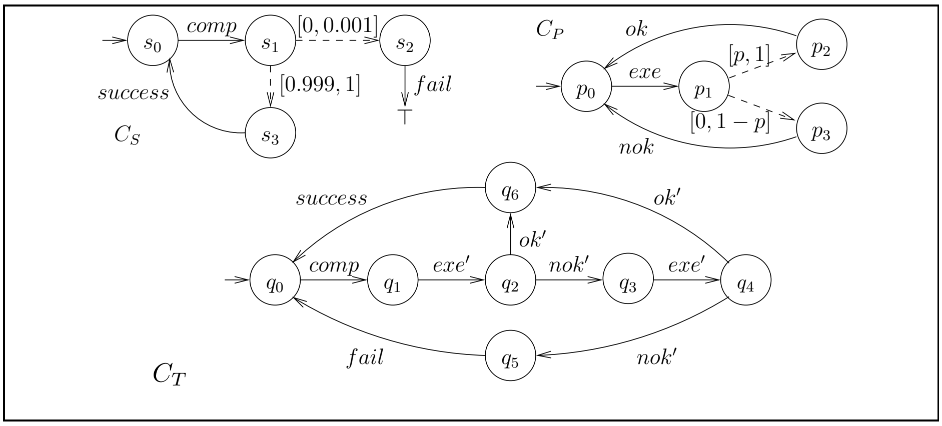


Figure 13: (top left) Specification C_S; (top right) Processor contract C_P; (bottom) Time redundancy contract C_T.

In terms of component-based design for reliability, we wonder what is the minimum value of p that guarantees the reliability level of C_S. To compute this minimum value, we first compute the parallel composition $C_T||_{\mathcal{I}}C_P$, with the interaction set $\mathcal{I} = \{comp, exe|exe', ok|ok', nok|nok', success, fail\}$. The reduction modulo bisimulation of this parallel composition is shown in Figure 14 (top), where the interactions $exe|exe'$, $ok|ok'$, and $nok|nok'$ have been replaced for conciseness by **exe**, **ok**, and **nok**, respectively. We call this new contract $C_{T||P}$. We then compute the projection of $C_{T||P}$ onto the set $\mathcal{B} = \{comp, success, fail\}$. The result $C_\pi = \pi_{\mathcal{B}}(C_{T||P})$ is shown in Figure 14 (bottom left).

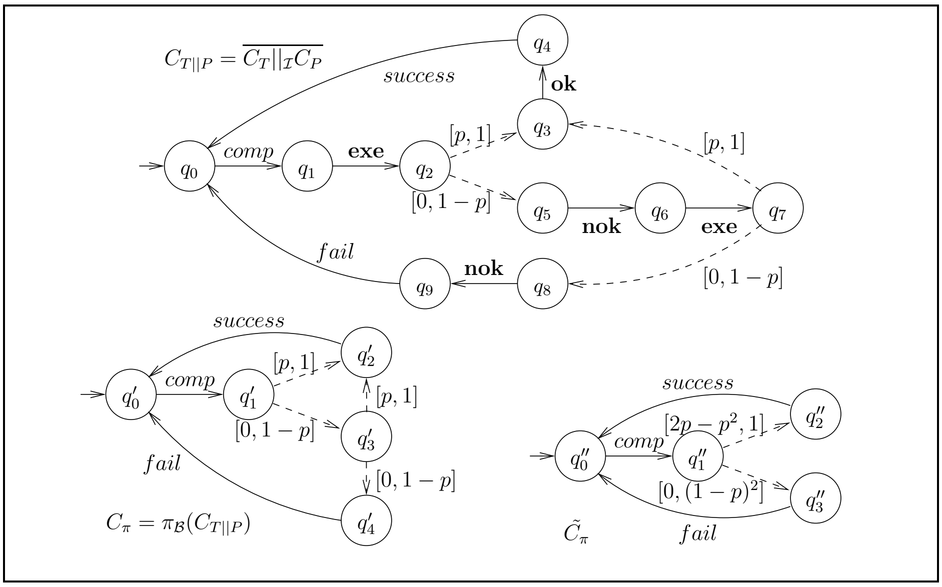


Figure 14: Parallel composition $C_{T||P}$; Projection C_π; Transitive closure $\tilde{C}_\pi$.

We are thus faced with a contract C_π having *sequences* of probabilistic transitions; more precisely, since some probabilistic states have several outgoing transitions, we have DAGs of probabilistic transitions. We therefore compute the transitive closure for each such DAG: that is, for each sequence of probabilistic transitions from the initial state of the DAG (e.g., q'_1 in C_π) to one of its final states (e.g., q'_2 and q'_4 in C_π), we compute the equivalent probabilistic transition. Without entering into the details of this computation, we show the resulting contract $\tilde{C}_\pi$ in Figure 14 (bottom right).

The last step involves checking under which condition on p the contract $\tilde{C}_\pi$ refines the specification C_S. We have $\tilde{C}_\pi \leq C_S \Leftrightarrow (1-p)^2 \leq 0.001 \Leftrightarrow p \geq 0.968$. This means that, with time redundancy and a processor with a reliability level of at least 0.969, we are able to ensure an overall reliability level of 0.999.

6 Discussion

We have introduced a design framework based on probabilistic contracts, and proved essential properties for the use in component-based design. Our definition of contracts is based on the ideas from [9, 14, 5], although the frameworks in [9, 5] do not support interactions between contracts.

Shared refinement of interfaces, and conjunction of modal specifications over possibly different alphabets have been defined in [4, 12]. A framework over modal assume/guarantee-contracts is introduced in [6], for which both parallel composition and conjunction are defined. Probabilistic assume/guarantee-contracts have also been introduced in [3] in terms of traces. [10] introduces a compositional framework based on continuous time IMCs, adopting a similar interaction model as done in this paper. This framework supports abstraction, parallel and symmetric composition, but not conjunction. The recently introduced Constraint Markov Chains (CMC) [2] generalize Markov Chains by introducing constraints on state valuations and transition probability distributions, aiming at a similar goal of providing a probabilistic component-based design framework. Whereas CMCs do not support explicit interactions among components, they allow the designer to expressively specify constraints on probability distributions. Conjunction is shown to be sound and complete in this framework.

Future work will encompass implementing the framework and carrying out case studies. A particularly interesting application would be the design of adaptive systems where the probabilistic behavior of components may change over time, while the overall system must at any time satisfy a set of safety, reliability, and quality of service contracts.

References

[1] A.V. Aho, R. Sethi, and J.D. Ullman. *Compilers – Principles, Techniques, and Tools*. Addison Wesley, 1986.

[2] B. Caillaud, B. Delahaye, K.G. Larsen, A. Legay, M. Pedersen, and A. Wasowski. Compositional design methodology with constraint markov chains. Research Report 6993, INRIA, 2009.

[3] B. Delahaye and B. Caillaud. A model for probabilistic reasoning on assume/guarantee contracts. Research Report 6719, INRIA, 2008.

[4] L. Doyen and T. Petrov T.A. Henzinger, B. Jobstmann. Interface theories with component reuse. In *Proc. EMSOFT'08*, pages 79–88. ACM, 2008.

[5] H. Fecher, M. Leucker, and V. Wolf. Don't know in probabilistic systems. In *Model Checking Software*, LNCS, pages 71–88. Springer, 2006.

[6] G. Gössler and J.-B. Raclet. Modal contracts for component-based design. In *Proc. SEFM'09*, pages 295–303. IEEE, 2009.

[7] G. Gössler and J. Sifakis. Composition for component-based modeling. *Science of Computer Programming*, 55(1-3):161–183, 3 2005.

[8] H. Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *LNCS*. Springer, 2002.

[9] B. Jonsson and K.G. Larsen. Specification and refinement of probabilistic processes. In *LICS*, pages 266–277. IEEE Computer Society, 1991.

[10] J.-P. Katoen, D. Klink, and M.R. Neuhäußer. Compositional abstraction for stochastic systems. In *Proc. FORMATS'09*, pages 195–211, 2009.

[11] B. Meyer. *Advances in Object-Oriented Software Engineering*, chapter Design by Contract, pages 1–50. Prentice Hall, 1991.

[12] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, and R. Passerone. Why modalities are good for interface theories? In *Proc. ACSD'09*. IEEE, 2009.

[13] P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control and Optimization*, 25(1), 1987.

[14] W. Yi. Algebraic reasoning for real-time probabilistic processes with uncertain information. In *FTRTFT*, volume 863 of *LNCS*, pages 680–693. Springer, 1994.

A Contract Refinement

A.1 Transitivity of Refinement

Lemma 2 [Transitivity of $\leq$] For all contracts C_1, C_2 and C_3, if $C_1 \ \leq \ C_2$ and $C_2 \leq C_3$, then $C_1 \ \leq \ C_3$.

Proof. Let

$$\begin{array}{rcl} C_1 & = & (\mathcal{Q}_1, \mathcal{A}_1, \rightarrow_1, \sigma_1, r_0) \\ C_2 & = & (\mathcal{Q}_2, \mathcal{A}_2, \rightarrow_2, \sigma_2, s_0) \\ C_3 & = & (\mathcal{Q}_3, \mathcal{A}_3, \rightarrow_3, \sigma_3, t_0) \end{array}$$

To show $C_1 \leq C_2$ and $C_2 \leq C_3$ implies $C_1 \leq C_3$, by Definition 5, we want to show $r_0 \leq s_0$ and $s_0 \leq t_0$ implies $r_0 \leq t_0$.

That is, for all $r \in \mathcal{Q}_1, s \in \mathcal{Q}_2, t \in \mathcal{Q}_3$, we want to show that

$$r \leq s \wedge s \leq t \Rightarrow r \leq t$$

Let us first consider conditions (1) and (2) in Definition 5 [Contract Refinement]. We have the following co-induction hypothesis: for all r', s', t' which are next states of r, s, t respectively,

$$r' \leq s' \wedge s' \leq t' \Rightarrow r' \leq t' \qquad \text{[H1]}$$

(1)

$$\begin{array}{rl} & r = \top \\ \Rightarrow & \text{(By Definition 5 (1))} \\ & s = \top \\ \Rightarrow & \text{(By Definition 5 (1))} \\ & t = \top \end{array}$$

(2a) For all $t' \neq \top \in \mathcal{Q}_3$, we have:

$$\begin{array}{rl} & t \xrightarrow{a}_3 t' \\ \Rightarrow & \text{(By Definition 5 (2a))} \\ & s \xrightarrow{a}_2 s' \text{ for some } s' \text{ and } s' \leq t' \\ \Rightarrow & (s' \leq t' \text{ implies } s' \neq \top, \text{so by Definition 5 (2a))} \\ & r \xrightarrow{a}_1 r' \text{ for some } r' \text{ and } r' \leq s' \text{ and } s' \leq t' \\ \Rightarrow & \text{(By co-induction hypothesis [H1])} \\ & r \xrightarrow{a}_1 r' \text{ and } r' \leq t' \end{array}$$

(2b) For all $r' \neq \top \in \mathcal{Q}_1$, we have:

$$\begin{array}{rl} & r \xrightarrow{a}_1 r' \\ \Rightarrow & \text{(By Definition 5 (2b))} \\ & s = \top \text{ or } s \xrightarrow{a}_2 s' \text{ for some } s' \text{ and } r' \leq s' \end{array}$$

There are two cases to consider:

- Case $s = \top$.

$$\begin{array}{rl} & s = \top \\ \Rightarrow & \text{(By Definition 5 (1))} \\ & t = \top \end{array}$$

Since any state refines $\top$, we have $r \leq \top$.

- Case $s \neq \top$.

$$\begin{array}{rl} & s \xrightarrow{a}_2 s' \text{ for some } s' \text{ and } r' \leq s' \\ \Rightarrow & \text{(By Definition 5 (2b))} \\ & t = \top \text{ or } t \xrightarrow{a}_3 t' \text{ for some } t' \text{ and } s' \leq t' \text{ and } r' \leq s' \end{array}$$

There are two subcases to consider:

* Subcase $t = \top$. Since any state refines $\top$, we have $r \leq \top$.
* Subcase $t \neq \top$.

$$\begin{array}{rl} & t \xrightarrow{a}_3 t' \text{ for some } t' \text{ and } s' \leq t' \text{ and } r' \leq s' \\ \Rightarrow & \text{(By co-induction hypothesis [H1])} \\ & t \xrightarrow{a}_3 t' \text{ for some } t' \text{ and } r' \leq t' \end{array}$$

(3) Now, let us consider the Definition 5 [Contract Refinement] (3). Given $C_1 \leq C_2$, by Definition 5 (3), we know there is a probability distribution $\delta_{12} \subset \mathcal{Q}_1 \times \mathcal{Q}_2 \times [0,1]$, such that, $\forall f_1 \in \sigma_1(r), s' \in \mathcal{Q}_2$,

$$(A) \quad \textstyle\sum_{r' \in \mathcal{Q}_1} (f_1(r') * \delta_{12}(r')(s')) \in \sigma_2(s)(s'), \text{and } \delta(r')(s') > 0 \Rightarrow r' \leq s'$$

Given $C_2 \leq C_3$, by Definition 5 [Contract Refinement] (3), we know there is a probability distribution $\delta_{23} \subset \mathcal{Q}_2 \times \mathcal{Q}_3 \times [0,1]$, such that, $\forall f_2 \in \sigma_2(s), t' \in \mathcal{Q}_3$,

$$(B) \quad \textstyle\sum_{s' \in \mathcal{Q}_1} (f_2(s') * \delta_{23}(s')(t')) \in \sigma_3(t)(t'), \text{and } \delta(s')(t') > 0 \Rightarrow s' \leq t'$$

We want to establish a $\delta_{13} \subset \mathcal{Q}_1 \times \mathcal{Q}_3 \times [0,1]$ such that Definition 5 (3) holds. Let δ_{13} be

$$\delta_{13}(r')(t') = \sum_{s' \in \mathcal{Q}_2} \delta_{12}(r')(s') * \delta_{23}(s')(t')$$

We want to check that δ_{13} satisfies the condition Definition 5 (3) for all $f_1 \in \delta_1(r), t' \in \mathcal{Q}_3$.

$$\begin{array}{ll}
 & \sum_{r' \in \mathcal{Q}_1} (f_1(r') * \delta_{13}(r')(t)) \\
= & \text{(By definition of } \delta_{13}) \\
 & \sum_{r' \in \mathcal{Q}_1} (f_1(r') * \sum_{s' \in \mathcal{Q}_2} \delta_{12}(r')(s') * \delta_{23}(s')(t')) \\
= & \text{(By distribution of } * \text{ over } +) \\
 & \sum_{r' \in \mathcal{Q}_1} \sum_{s' \in \mathcal{Q}_2} f_1(r') * \delta_{12}(r')(s') * \delta_{23}(s')(t') \\
= & \text{(By commutativity and associativity of } +) \\
 & \sum_{s' \in \mathcal{Q}_2} \sum_{r' \in \mathcal{Q}_1} f_1(r') * \delta_{12}(r')(s') * \delta_{23}(s')(t') \\
= & \text{(By (A), } \exists f_2 \in \sigma_2(s), f_2(s') = \sum_{r' \in \mathcal{Q}_1} f_1(r') * \delta_{12}(r')(s')) \\
 & \sum_{s' \in \mathcal{Q}_2} f_2(s') * \delta_{23}(s')(t') \\
\in & \text{(By (B), which holds for all } f_2 \in \sigma_2(s)) \\
 & \sigma_3(t)(t')
\end{array}$$

So we have the desired result $\sum_{r' \in \mathcal{Q}_1} (f_1(r') * \delta_{13}(r')(t)) \in \sigma_3(t)(t')$.

(4) If $r \in \mathcal{Q}_1^a$ and $t \in \mathcal{Q}_3^p$ and $r \leq s$ and $s \leq t$, then there are two subcases to consider: $s \in \mathcal{Q}_2^a$ and $s \in \mathcal{Q}_2^p$.

- Subcase $s \in \mathcal{Q}_2^a$.

$$\begin{array}{ll}
 & r \leq s \text{ and } s \leq t \\
\Longleftrightarrow & \text{(By Definition 5 [Contract refinement] (4))} \\
 & r \leq s \text{ and } \exists t^a \in \mathcal{Q}_3^a : t \overset{>0}{\dashrightarrow}{}_3^{+} t^a \wedge s \leq t^a \text{ and} \\
 & \forall t' \in \mathcal{Q}_3, \big(t \overset{>0}{\dashrightarrow}_3 t' \implies s \leq t'\big) \\
\Longleftrightarrow & \text{(By co-induction hypothesis [H1] } r' = r, s' = s, t' = t^a) \\
 & r \leq s \text{ and } \exists t^a \in \mathcal{Q}_3^a : t \overset{>0}{\dashrightarrow}{}_3^{+} t^a \wedge r \leq t^a \text{ and} \\
 & \forall t' \in \mathcal{Q}_3, \big(t \overset{>0}{\dashrightarrow}_3 t' \implies s \leq t'\big) \\
\Longleftrightarrow & \text{(By co-induction hypothesis [H1] } r' = r, s' = s, t' = t') \\
 & \exists t^a \in \mathcal{Q}_3^a : t \overset{>0}{\dashrightarrow}{}_3^{+} t^a \wedge r \leq t^a \text{ and} \\
 & \forall t' \in \mathcal{Q}_3, \big(t \overset{>0}{\dashrightarrow}_3 t' \implies r \leq t'\big) \\
\Longleftrightarrow & \text{(By Definition 5 [Contract refinement] (4))} \\
 & r \leq t
\end{array}$$

– Subcase $s \in \mathcal{Q}_2^p$.

$$\begin{array}{ll}
 & r \leq s \text{ and } s \leq t \\
\iff & \text{(By Definition 5 [Contract refinement] (4))} \\
 & \exists s^a \in \mathcal{Q}_2^a : s \overset{>0}{\dashrightarrow}_2^{+} s^a \wedge r \leq s^a \text{ and} \\
 & \forall s' \in \mathcal{Q}_2, \big(s \overset{>0}{\dashrightarrow}_2 s' \implies r \leq s'\big) \text{ and } s \leq t \\
\iff & \text{(By Definition 5 [Contract refinement] (3))} \\
 & (1)\ \exists s^a \in \mathcal{Q}_2^a : s \overset{>0}{\dashrightarrow}_2^{+} s^a \wedge r \leq s^a \text{ and} \\
 & (2)\ \forall s' \in \mathcal{Q}_2, \big(s \overset{>0}{\dashrightarrow}_2 s' \implies r \leq s'\big) \text{ and} \\
 & (\exists \delta : \mathcal{Q}_2 \times \mathcal{Q}_3 \rightarrow [0,1], \forall f \in \sigma_3(s) \text{ and} \\
 & (3)\ \forall t' \in \mathcal{Q}_3, \sum_{s' \in \mathcal{Q}_2} (f(s') * \delta(s')(t')) \subseteq \sigma_3(t)(t') \\
 & \text{and } \forall s' \in \mathcal{Q}_2 : \big(\delta(s')(t') > 0 \implies s' \leq t'\big)) \\
\iff & \text{(By (3) and Definition 5 (4,5), we have (4); by (2), (3) and} \\
 & \text{co-induction hypothesis [H1] where } r' = r, s' = s', t' = t', \\
 & \text{we have (5))} \\
 & (4)\ \exists t^a \in \mathcal{Q}_3^a : t \overset{>0}{\dashrightarrow}_3^{+} t^a \wedge s^a \leq t^a \text{ and} \\
 & (5)\ \forall t' \in \mathcal{Q}_3, \big(t \overset{>0}{\dashrightarrow}_3 t' \implies r \leq t'\big) \\
\iff & \text{(By co-induction hypothesis [H1] where } r' = r, s' = s^a, t' = t^a) \\
 & \exists t^a \in \mathcal{Q}_3^a : t \overset{>0}{\dashrightarrow}_3^{+} t^a \wedge r \leq t^a \text{ and} \\
 & \forall t' \in \mathcal{Q}_3, \big(t \overset{>0}{\dashrightarrow}_3 t' \implies r \leq t'\big) \\
\iff & \text{(By Definition 5 [Contract refinement] (4))} \\
 & r \leq t
\end{array}$$

(5) Similar to the proof in (4).

□

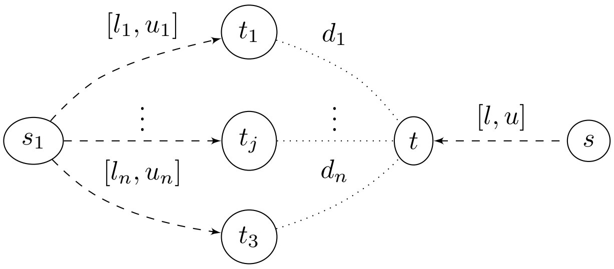
A.2 Reduction

Lemma 11 (Bisimulation and Equivalence)**.** *For all delimited contracts C_1 and C_2, $C_1 \simeq C_2 \Rightarrow C_1 \equiv C_2$.*

Proof. By inspecting Definition 5 and Definition 9, we can see $C_1 \simeq C_2$ implies $C_1 \leq C_2$ and $C_2 \leq C_1$. By Lemma 7 [Refinement and Equivalence], we have $C_1 \simeq C_2 \Rightarrow C_1 \equiv C_2$. □

Lemma 6 [Reduction and refinement] For all delimited contracts C, $\overline{C} \leq C$ and $C \leq \overline{C}$.

Proof. Let $C = (\mathcal{Q}, \mathcal{A}, \rightarrow, \pi, s_0)$. The reduction combines all bisimilar states into one. Definition 10 (1) is from literature so we omit its correctness proof here. We now prove the correctness of Definition 10 (2), $\sigma_{\simeq}(s,t) = \sum_{1 \leq j \leq n} \sigma(s_1, t_j))$, which combines probabilistic transitions leading to bisimilar states into one as shown below.



Definition 10 (2) computes $[l, u]$ as follows.

$$l = l_1 + \cdots + l_n \qquad u = \lceil u_1 + \cdots + u_n \rceil$$

By Lemma 1 [Reflexivity of $(\leq)$], $s \leq s$. Since $t_1 \simeq \ldots \simeq t_j \simeq \ldots \simeq t_n$, by Lemma 11 [Bisimulation and equivalence] and Lemma 7 [Refinement and equivalence], $t_1 \leq t_j, t_j \leq t_1, t_j \leq t_n, t_n \leq t_j$ for all $1 \leq j \leq n$. So we have the mapping d_i from t_j for all $1 \leq i = j \leq n$.

Case ($\overline{C} \leq C$). We want to show that for all $s, t \in \mathcal{Q}$ there exists a probability distribution $\delta_1 \subset \mathcal{Q} \times \mathcal{Q} \times [0, 1]$ such that $s_1 \leq s$. Let s' denote any next state of s.

Let $\delta_1(s')(t') = 1$. For any $f \in \sigma(s)$, we have

$$\begin{array}{ll} & \sum_{i=1}^{n}(f(s') * \delta_1(s')(s_i)) \\ = & \text{(By definition of } \delta_1\text{)} \\ & \sum_{i=1}^{n} f(s') \\ \subseteq & \text{(By Definition 4 [Delimited contract], } \sum_{i=1}^{n} f(s') \leq 1\text{)} \\ & [l_1 + \cdots + l_n, \lceil u_1 + \cdots + u_n \rceil] \\ = & \text{(By definition of } l \text{ and } u\text{)} \\ & [l, u] \end{array}$$

Case ($C \leq \overline{C}$). We want to show that for all $s, t \in \mathcal{Q}$ there exists a probability distribution $\delta_2 \subset \mathcal{Q} \times \mathcal{Q} \times [0, 1]$ such that $s \leq s_1$.

(a) Since $\delta_2(s_i)$ is a probability distribution, we need to make sure that sum of d_i is 1.

Let $d_i \in [l_i/l, u_i/u]$ for $1 \leq i \leq n$. We show that there exists a vector of d_i such that $\sum_{i=1}^{n} d_i = 1$ as follows.

$$\begin{array}{ll} & \sum_{i=1}^{n} d_i \\ \in & \text{(By definition of } d_i\text{)} \\ & [\sum_{i=1}^{n} l_i/l, \sum_{i=1}^{n} u_i/u] \\ = & [l_1/l + \cdots + l_n/l, u_1/u + \cdots + u_n/u] \\ & [(\sum_{i=1}^{n} l_i)/l, (\sum_{i=1}^{n} u_i)/u] \\ = & \text{(By definition of } l \text{ and } u\text{)} \\ & [l/l, u/u] \\ = & [1, 1] \end{array}$$

(b) We need to check that for all $1 \leq i \leq n$, $[l, u] * d_i \subseteq [l_i, u_i]$. Here is the proof.

$$\begin{array}{ll} & d_i \in [l_i/l, u_i/u] \\ \iff & \text{(By set theory and math)} \\ & l * d_i \in [l_i, (u_i * l)/u] \text{ and } u * d_i \in [(l_i * u)/l, u_i] \\ \iff & \text{(Since } l \leq u \text{, we know } l/u \leq 1 \text{ and } u/l \geq 1.\text{)} \\ & [l_i, (u_i * l)/u] \subseteq [l_i, u_i] \text{ and } [(l_i * u)/l, u_i] \subseteq [l_i, u_i] \\ \iff & \text{(Since } l \leq u \text{, we know } l * d_i \leq u * d_i.\text{)} \\ & [l * d_i, u * d_i] \subseteq [l_i, u_i] \\ \iff & [l, u] * d_i \subseteq [l_i, u_i] \end{array}$$

□

A.3 Contract Abstraction

Lemma 9 [Abstraction and refinement] For all contracts $C_1 = (\mathcal{Q}_1, \mathcal{A}, \rightarrow_1, \dashrightarrow_1, s_0)$, $C_2 = (\mathcal{Q}_2, \mathcal{A}, \rightarrow_2, \dashrightarrow_2, t_0)$ and $\mathcal{B} \subseteq \mathcal{A}$, if $C_1 \leq C_2$, then $\pi_{\mathcal{B}}(C_1) \leq \pi_{\mathcal{B}}(C_2)$.

Proof. Let

$$\begin{array}{rcl} \pi_{\mathcal{B}}(C_1) & = & (\mathcal{Q}_3, \mathcal{A}, \rightarrow_3, \sigma_3, \mathbf{s_0}) \\ \pi_{\mathcal{B}}(C_2) & = & (\mathcal{Q}_4, \mathcal{A}, \rightarrow_4, \sigma_4, \mathbf{t_0}) \end{array}$$

Given a state s and its next state s' in a contract before projection, we use bold font $\mathbf{s}$ and $\mathbf{t}$ to represent a set of states in $\mathcal{Q}_1$ and $\mathcal{Q}_2$ respectively. The next state of $\mathbf{s}$ is represented by $\mathbf{s}'$. To show that if $s_0 \leq t_0$ then $\mathbf{s_0} \leq \mathbf{t_0}$, we show the general case: for all $s \in \mathcal{Q}_1, t \in \mathcal{Q}_2$, if $s \leq t$, then $\mathbf{s} \leq \mathbf{t}$. We prove this lemma by structural induction. We have the following induction hypothesis: for all $s' \in \mathcal{Q}_1, t' \in \mathcal{Q}_2, \mathbf{s}' \in \mathcal{Q}_3, \mathbf{t}' \in \mathcal{Q}_4$, such that $s' \in \mathbf{s}$ and $t' \in \mathbf{t}$,

$$s' \leq t' \implies \mathbf{s}' \leq \mathbf{t}' \qquad \text{[H]}$$

We have the following cases to consider.

- Case $s = \top$. Since $s \leq t$, by Definition 5 (1), we have $t = \top$. Actions leading to $\top$ are kept in the projection. There is no state in the projection containing other states than $\top$. Therefore, $\mathbf{s} = \mathbf{t} = \top$.

- Case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^a \cup \{\top\}$. There are three cases to consider:

  (a) $\forall t' \neq \top \in \mathcal{Q}_2$, $(t \xrightarrow{\alpha}_2 t') \implies (\exists s' \in \mathcal{Q}_1,\ s \xrightarrow{\alpha}_1 s' \wedge s' \leq t')$.
  If $\alpha \in \mathcal{B}$, this action transition is kept in $\pi_{\mathcal{B}}(C_1)$ and $\pi_{\mathcal{B}}(C_1)$. So we have $\mathbf{s} \xrightarrow{\alpha}_3 \mathbf{s}'$ and $\mathbf{t} \xrightarrow{\alpha}_4 \mathbf{t}'$. From $s' \leq t'$, by induction hypothesis [H], we have (1) $\mathbf{s}' \leq \mathbf{t}'$. So we have $\forall \mathbf{t}' \neq \top \in \mathcal{Q}_4$, $(\mathbf{t} \xrightarrow{\alpha}_4 \mathbf{t}') \implies (\exists \mathbf{s}' \in \mathcal{Q}_3,\ \mathbf{s} \xrightarrow{\alpha}_3 \mathbf{s}' \wedge \mathbf{s}' \leq \mathbf{t}')$.
  If $\alpha \notin \mathcal{B}$, this action transition does not appear in $\pi_{\mathcal{B}}(C_1)$ and $\pi_{\mathcal{B}}(C_1)$. We have $\{s, s'\} \subseteq \mathbf{s}$ and $\{t, t'\} \subseteq \mathbf{t}$. By induction hypothesis [H], we have $\mathbf{s} \leq \mathbf{t}$.

  (b) $\forall s' \in \mathcal{Q}_1$, $(s \xrightarrow{\alpha}_1 s') \implies (t = \top \vee \exists t' \in \mathcal{Q}_2,\ t \xrightarrow{\alpha}_2 t' \wedge s' \leq t')$. For the case $t = \top$, As actions leading to $\top$ are kept in the projection, there is no state in the projection containing other states than $\top$.

Therefore, $\mathbf{t} = \top$. By Definition 5 (1) and (2), any state refines $\top$, so we have $\mathbf{s} \leq \mathbf{t}$.

For the case $\exists t' \in \mathcal{Q}_2,\ t \xrightarrow{\alpha}_2 t' \wedge s' \leq t'$, we have two subcases to consider.

- $*$ If $\alpha \in \mathcal{B}$, this action transition is kept in $\pi_{\mathcal{B}}(C_1)$ and $\pi_{\mathcal{B}}(C_1)$. So we have $\mathbf{s} \xrightarrow{\alpha}_3 \mathbf{s}'$ and $\mathbf{t} \xrightarrow{\alpha}_4 \mathbf{t}'$. From $s' \leq t'$, by induction hypothesis [H], we have $\mathbf{s}' \leq \mathbf{t}'$. So we have (2) $\forall \mathbf{s}' \in \mathcal{Q}_3,\ (\mathbf{s} \xrightarrow{\alpha}_3 \mathbf{s}') \implies \exists \mathbf{t}' \in \mathcal{Q}_4,\ \mathbf{t} \xrightarrow{\alpha}_2 \mathbf{t}' \wedge \mathbf{s}' \leq \mathbf{t}'$.
- $*$ If $\alpha \notin \mathcal{B}$, this action transition does not appear in $\pi_{\mathcal{B}}(C_1)$ and $\pi_{\mathcal{B}}(C_1)$. We have $\{s, s'\} \subseteq \mathbf{s}$ and $\{t, t'\} \subseteq \mathbf{t}$. By induction hypothesis [H], we have $\mathbf{s} \leq \mathbf{t}$.

From (1) and (2), by Definition 5 (2), we have $\mathbf{s} \leq \mathbf{t}$.

- Case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^p$. By Definition 5 (4), $\exists t^{\mathsf{a}} \in \mathcal{Q}_2^{\mathsf{a}} : t \overset{>0}{\dashrightarrow}{}_2^{+}\ t^{\mathsf{a}} \wedge s \leq t^{\mathsf{a}}$ and $\forall t' \in \mathcal{Q}_2,\ \big(t \overset{>0}{\dashrightarrow}_2 t' \implies s \leq t'\big)$. If we have $t' \in \mathcal{Q}_2^p$, we have $s \leq t'$. Projection does not have effect on probabilistic transitions, by induction hypothesis [H], we are done. If we have $t' \in \mathcal{Q}_2^a$, we have $s \leq t^a$. Since $s \in \mathcal{Q}_1^a$, this falls into the case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^a$, which has been proved above.
- Case $s \in \mathcal{Q}_1^p, t \in \mathcal{Q}_2^a$. Similar reasoning as the case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^p$.
- Case $s \in \mathcal{Q}_1^p, t \in \mathcal{Q}_2^p$. By Definition 5 (3), we know $s \overset{P_1}{\dashrightarrow}_1 s'$, $t \overset{P_2}{\dashrightarrow}_2 t'$ and $s' \leq t'$. Projection only has effect on action states, the probabilistic transitions remain the same (up to their target states). That is, we have (1) $\mathbf{s} \overset{P_3}{\dashrightarrow}_3 \mathbf{s}'$ and (2) $\mathbf{t} \overset{P_4}{\dashrightarrow}_4 \mathbf{t}'$. From $s' \leq t'$, by induction hypothesis [H], we have (3) $\mathbf{s}' \leq \mathbf{t}'$. From (1), (2), (3), by Definition 5 (3), we have $\mathbf{s} \leq \mathbf{t}$.

□

B Contract Composition

B.1 Congruence of Refinement for Parallel Composition

Lemma 12 (Congruence of refinement for $||_{\mathcal{I}}$). *For all contracts C_1, C_2, C_3, and interaction set $\mathcal{I}$, if $C_1 \leq C_2$, then $C_1||_{\mathcal{I}}C_3 \leq C_2||_{\mathcal{I}}C_3$.*

Proof. Let

$$\begin{array}{rcl} C_1 & = & (\mathcal{Q}_1, \mathcal{A}_1, \rightarrow_1, \sigma_1, s_0) \\ C_2 & = & (\mathcal{Q}_2, \mathcal{A}_2, \rightarrow_2, \sigma_2, t_0) \\ C_3 & = & (\mathcal{Q}_3, \mathcal{A}_3, \rightarrow_3, \sigma_3, u_0) \\ C_1||_{\mathcal{I}}\, C_3 & = & (\mathcal{Q}_{13}, \mathcal{A}_{13}, \rightarrow_{13}, \sigma_{13}, (s_0, u_0)) \\ C_2||_{\mathcal{I}}\, C_3 & = & (\mathcal{Q}_{23}, \mathcal{A}_{23}, \rightarrow_{23}, \sigma_{23}, (t_0, u_0)) \end{array}$$

Let $\theta \subseteq \mathcal{Q}_1 \times \mathcal{Q}_2$ be the refinement relation stating that $s \leq t$. Let $\theta' \subseteq \mathcal{Q}_{13} \times \mathcal{Q}_{23}$ be a binary relation such that $((s, u), (t, u)) \in \theta'$ if $(s, t) \in \theta$. We now prove that θ' allows to establish that $(s, u) \leq (t, u)$.

Notation: For all σ, let $\underline{\sigma}$ and $\overline{\sigma}$ be the lower bound and upper bound of σ respectively.

We first consider 3 cases involving the state $\top$.

(a) Case $s = \top$. Since $s \leq t$, by Definition 5 [Refinement] (1), $t = \top$. By Definition 12 [Parallel composition] (1), if one state is in $\top$ state, the composed state is in $\top$ state, we have $(s, u) = \top$ and $(t, u) = \top$. Thus, $(s, u) \leq (t, u)$.

(b) Case $t = \top$. By Definition 12 [Parallel composition] (1), we have $(t, u) = \top$. Since any state refines $\top$ state, we have $(s, u) \leq (t, u)$.

(c) Case $u = \top$. By Definition 12 [Parallel composition] (1), we have $(s, u) = \top$ and $(t, u) = \top$. Thus, $(s, u) \leq (t, u)$.

Now, we consider cases where $s \neq \top$, $t \neq \top$ and $u \neq \top$. We have the following co-induction hypothesis: for all s', t', u' such that s', t', u' are the next states of s, t and u respectively, and $((s', u'), (t', u')) \in \theta'$,

$$s' \leq t' \Rightarrow (s', u') \leq (t', u') \qquad \text{[H]}$$

Given $((s, u), (t, u)) \in \theta'$, we have the following cases to consider.

- Case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^a, u \in \mathcal{Q}_3^a$. Since $s \leq t$, we have (1) $s \xrightarrow{\alpha}_1 s'$; (2) $t \xrightarrow{\alpha}_2 t'$; (3) $u \xrightarrow{\beta}_3 u'$; (4) $s' \leq t'$. There are three cases to consider:

  (a) Subcase $\alpha | \beta \in \mathcal{I}$.
  By (1), (3) and rule [R3], we have (5) $(s, u) \xrightarrow{\alpha|\beta}_{12} (s', u')$.
  By (2), (3) and rule [R3], we have (6) $(t, u) \xrightarrow{\alpha|\beta}_{23} (t', u')$.
  From (4), by induction hypothesis [H], we have (7) $(s', u') \leq (t', u')$.
  By Definition 5 (2), we have $(s, u) \leq (t, u)$.

  (b) Subcase $\alpha \in \mathcal{I}$.
  By (1), (3) and rule [R1], we have (5) $(s, u) \xrightarrow{\alpha}_{12} (s', u)$.
  By (2), (3) and rule [R1], we have (6) $(t, u) \xrightarrow{\alpha}_{23} (t', u)$.
  From (4), by induction hypothesis [H], we have (7) $(s', u) \leq (t', u)$.
  By Definition 5 (2), we have $(s, u) \leq (t, u)$.

  (c) Subcase $\beta \in \mathcal{I}$.
  By (1), (3) and rule [R2], we have (5) $(s, u) \xrightarrow{\alpha}_{12} (s, u')$.
  By (2), (3) and rule [R2], we have (6) $(t, u) \xrightarrow{\alpha}_{23} (t, u')$.
  From (4), by induction hypothesis [H], we have (7) $(s, u') \leq (t, u')$.

  For each subcase, from (5), (6), (7), by Definition 5 (2), we have $(s, u) \leq (t, u)$.

- Case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^a, u \in \mathcal{Q}_3^p$. Since $s \leq t$, we have (1) $s \xrightarrow{\alpha}_1 s'$; (2) $t \xrightarrow{\alpha}_2 t'$; (3) $u \overset{P_3}{\dashrightarrow}_3 u'$; (4) $s' \leq t'$.
  By (1), (3) and rule [R6], we have (5) $(s, u) \overset{P_3}{\dashrightarrow}_{12} (s, u')$.
  By (2), (3) and rule [R6], we have (6) $(t, u) \overset{P_3}{\dashrightarrow}_{23} (t, u')$.
  From (4), by induction hypothesis [H], we have $(s, u') \leq (t, u')$. By Definition 5 (3), we have $(s, u) \leq (t, u)$.

- Case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^p, u \in \mathcal{Q}_3^a$. Since $s \leq t$, we have (1) $s \xrightarrow{\alpha}_1 s'$; (2) $t \overset{P_2}{\dashrightarrow}_2 t'$; (3) $u \xrightarrow{\beta}_3 u'$. (4) $\exists t^{\mathsf{a}} \in \mathcal{Q}_2^{\mathsf{a}} : t \overset{>0}{\dashrightarrow}_2^{+} t^{\mathsf{a}} \wedge s \leq t^{\mathsf{a}}$; $\forall t' \in \mathcal{Q}_2$, $\big(t \overset{>0}{\dashrightarrow}_2 t' \implies s \leq t'\big)$. There are three cases:

(a) Subcase $\alpha|\beta \in \mathcal{I}$.
By (1), (3) and rule [R3], we have $(s, u) \xrightarrow{\alpha|\beta}_{12} (s', u')$.

(b) Subcase $\alpha \in \mathcal{I}$.
By (1), (3) and rule [R1], we have $(s, u) \xrightarrow{\alpha}_{12} (s', u')$.

(c) Subcase $\beta \in \mathcal{I}$.
By (1), (3) and rule [R2], we have $(s, u) \xrightarrow{\alpha}_{12} (s', u')$.

By (2), (3) and rule [R6], we have $(t, u) \overset{P_2}{\dashrightarrow}_{23} (t', u)$.
From (5), by induction hypothesis [H], we have $(s, u) \leq (t', u)$. By Definition 5 (4), we have $(s, u) \leq (t', u)$.

- Case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^p, u \in \mathcal{Q}_3^p$. Since $s \leq t$, we have (1) $s \xrightarrow{\alpha}_1 s'$; (2) $t \overset{[p_1,p_2]}{\dashrightarrow}_2 t'$; (3) $u \overset{[p_3,p_4]}{\dashrightarrow}_3 u'$. (4) $\exists t^{\mathsf{a}} \in \mathcal{Q}_2^{\mathsf{a}} : t \overset{>0}{\dashrightarrow}_2^{+} t^{\mathsf{a}} \wedge t \leq t^{\mathsf{a}}$; (5) $\forall t' \in \mathcal{Q}_2$, $\big(t \overset{>0}{\dashrightarrow}_2 t' \implies s \leq t'\big)$.
By (1), (3) and rule [R6], we have $(s, u) \xrightarrow{[p_3,p_4]}_{12} (s, u')$.
By (2), (3) and rule [R4], we have $(t, u) \overset{[p_1*p_3,p_2*p_4]}{\dashrightarrow}_{12} (t', u')$.
That means

$$\begin{array}{rl} (\dagger_1) & \sigma_{23}(t,u)(t',u') \\ = & [\underline{\sigma_{23}}(t,u)(t',u'), \overline{\sigma_{23}}(t,u)(t',u')] \\ = & [\underline{\sigma_2}(t)(t') * \underline{\sigma_3}(u)(u'), \overline{\sigma_2}(t)(t') * \overline{\sigma_3}(u)(u')] \end{array}$$

By Lemma 1 [Reflexivity of refinement], $u \leq u$. That means there exists a δ that satisfies the condition Definition 5 (3) for all $f_u(u') \in \sigma_3(u)(u')$ and $u' \in \mathcal{Q}_3$. By definition of f_u, we have

$$\begin{array}{rl} (\dagger_2) & \sum_{u' \in \mathcal{Q}_3} f_u(u') * \delta_u(u')(u') \in \sigma(u)(u') \\ \iff & \sum_{u' \in \mathcal{Q}_3} \sigma_3(u)(u') * \delta_u(u')(u') \subseteq \sigma_3(u)(u') \end{array}$$

We want to check that there exists a δ that satisfies the condition Definition 5 (3) for all $f(s, u') \in \sigma_{12}(s, u)(s, u')$ and $(t', u') \in \mathcal{Q}_{23}$. Let

$\delta((s,u'))((t',u')) \in \sigma(t)(t') * \delta_u(u')(u')$

$$\begin{array}{ll}
 & \text{(By definition [F2] in Figure 4: } [a,b]*[c,d]=[a*c,b*d]) \\
 & \sigma(t)(t') * \sigma_3(u)(u') \\
 & \subseteq [\underline{\sigma_2}(t)(t') * \underline{\sigma_3}(u)(u'), \overline{\sigma_2}(t)(t') * \overline{\sigma_3}(u)(u')] \\
\Rightarrow & \text{(By } \dagger_2 \text{ and by set theory} \\
 & [a,b]*[c,d] \subseteq [e,f] \wedge [c_1,d_1] \subseteq [c,d] \implies [a,b]*[c_1,d_1] \subseteq [e,f]) \\
 & \sum_{u' \in \mathcal{Q}_3} \sigma(t)(t') * \sigma_3(u)(u') * \delta_u(u')(u') \\
 & \subseteq [\underline{\sigma_2}(t)(t') * \underline{\sigma_3}(u)(u'), \overline{\sigma_2}(t)(t') * \overline{\sigma_3}(u)(u')] \\
\Rightarrow & \text{(By definition of } \delta \text{ and commutativity of } *) \\
 & \sum_{u' \in \mathcal{Q}_3} (\sigma_3(u)(u') * \delta(s,u')(t',u')) \\
 & \subseteq [\underline{\sigma_2}(t)(t') * \underline{\sigma_3}(u)(u'), \overline{\sigma_2}(t)(t') * \overline{\sigma_3}(u)(u')] \\
\Longleftrightarrow & \text{(By (1), (3), rule [R6], } \sum_{(s,u') \in \mathcal{Q}_{13}} \sigma_{13}(s,u)(s,u') = \sum_{u' \in \mathcal{Q}_3} \sigma_3(u)(u')) \\
 & \sum_{(s,u') \in \mathcal{Q}_{13}} (\sigma_{13}(s,u)(s,u') * \delta'(s,u')(t',u')) \\
 & \subseteq [\underline{\sigma_2}(t)(t') * \underline{\sigma_3}(u)(u'), \overline{\sigma_2}(t)(t') * \overline{\sigma_3}(u)(u')] \\
\Longleftrightarrow & \text{(By } (\dagger_1)) \\
 & \sum_{(s,u') \in \mathcal{Q}_{13}} (\sigma_{13}(s,u)(s',u) * \delta'(s,u')(t',u')) \subseteq \sigma_{23}(t,u)(t',u'), \\
\Longleftrightarrow & \text{(By definition of } f) \\
 & \sum_{(s,u') \in \mathcal{Q}_{13}} (f(s,u') * \delta(s,u')(t',u')) \in \sigma_{23}(t,u)(t',u')
\end{array}$$

- Case $s \in \mathcal{Q}_1^p, t \in \mathcal{Q}_2^a, u \in \mathcal{Q}_3^a$. Similar to the case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^p, u \in \mathcal{Q}_3^a$.

- Case $s \in \mathcal{Q}_1^p, t \in \mathcal{Q}_2^a, u \in \mathcal{Q}_3^p$. Similar to the case $s \in \mathcal{Q}_1^a, t \in \mathcal{Q}_2^p, u \in \mathcal{Q}_3^p$.

- Case $s \in \mathcal{Q}_1^p, t \in \mathcal{Q}_2^p, u \in \mathcal{Q}_3^a$. We have (1) $s \overset{P_1}{\dashrightarrow}_1 s'$; (2) $t \overset{P_2}{\dashrightarrow}_2 t'$; (3) $u \xrightarrow{\alpha}_3 u'$. By (1), (3) and rule [R5], we have (5) $(s,u) \overset{P_1}{\dashrightarrow}_{12} (s',u)$.
  By (2), (3) and rule [R5], we have (6) $(s,u) \overset{P_1}{\dashrightarrow}_{12} (s',u)$.
  We know there is a probability distribution $\delta \subset \mathcal{Q}_1 \times \mathcal{Q}_2 \times [0,1]$, such that, $\forall f \in \sigma(s), t' \in \mathcal{Q}_2$,

  $$\begin{array}{ll} (\dagger) & \sum_{s' \in \mathcal{Q}_1} (f(s') * \delta(s')(t')) \in \sigma_2(t)(t') \text{ and} \\ & \forall s' \in \mathcal{Q}_1, \delta(s')(t') > 0 \Rightarrow s' \leq t' \end{array}$$

  We want to check that δ' satisfies the condition Definition 5 (3) for all $f(s',u') \in \sigma_{12}(s,u)(s',u)$ and $(t',u) \in \mathcal{Q}_{23}$.

  $$\begin{array}{ll}
  = & \text{(By definition of } \delta') \\
   & \sum_{(s',u) \in \mathcal{Q}_{13}} (f(s',u) * \delta(s')(t')) \\
  = & \text{(By (3) and rule [R5]}, \sum_{(s',u) \in \mathcal{Q}_{13}} f(s',u) = \sum_{s' \in \mathcal{Q}_1} f(s')) \\
   & \sum_{(s',u) \in \mathcal{Q}_{13}} (f(s') * \delta(s')(t')) \\
  \in & \text{(By } (\dagger)) \\
   & \sigma_2(t)(t') \\
  = & \text{(By (3) and rule [R5]}, \sigma_{23}(t,u)(t',u) = \sigma_2(t)(t')) \\
   & \sigma_{23}(t,u)(t',u),
  \end{array}$$

- Case $s \in \mathcal{Q}_1^p, t \in \mathcal{Q}_2^p, u \in \mathcal{Q}_3^p$. We have (1) $s \overset{[p_1,p_2]}{\dashrightarrow}_1 s'$ and (2) $u \overset{[p_3,p_4]}{\dashrightarrow}_3 u'$. From (1), (2), by rule [R4], we have $(s,u) \overset{[p_1*p_3,p_2*p_4]}{\dashrightarrow}_{13} (s',u')$. That means:

  $$(\dagger_1) \quad \sigma_{13}(s,u)(s',u') = \sigma_1(s)(s') * \sigma_3(u)(u')$$

Since $s \leq t$, by Definition 5 [Contract Refinement] (3), we know $t \overset{[p_5,p_6]}{\dashrightarrow}_2 t'$ for some t', p_5, p_6 and $s' \leq t'$. By $u \overset{[p_3,p_4]}{\dashrightarrow}_3 u'$ and rule [R4], we know $(t,u) \overset{[p_5*p_3,p_6*p_4]}{\dashrightarrow}_{23} (t',u')$. That means:

$$(\dagger_2) \quad \sigma_{23}(t,u)(t',u') = \sigma_2(t)(t') * \sigma_3(u)(u')$$

By Definition 5 (3), We also know there is a probability distribution $\delta \subset \mathcal{Q}_1 \times \mathcal{Q}_2 \times [0,1]$, such that, $\forall f \in \sigma(s), t' \in \mathcal{Q}_2$,

$$\sum_{s' \in \mathcal{Q}_1} (f(s') * \delta(s')(t')) \in \sigma_2(t)(t'), \text{ and } s' \leq t' \text{ if } \delta(s')(t') > 0$$

We know:

$$\begin{array}{ll} (\dagger_3) & \forall f \in \sigma_1(s) \sum_{s' \in \mathcal{Q}_1} (f(s') * \delta(s')(t')) \in \sigma_2(t)(t') \\ \iff & \sum_{s' \in \mathcal{Q}_1} (\sigma_1(s)(s') * \delta(s')(t')) \subseteq \sigma_2(t)(t') \end{array}$$

We want to show that there is a probability distribution $\delta' \subset \mathcal{Q}_{13} \times \mathcal{Q}_{23} \times [0,1]$, such that Definition 5 (3) holds. Let δ' be

$$\delta'(s',u'')(t',u') = \begin{cases} \delta(s')(t'), & \text{if } u'' = u' \\ 0, & \text{otherwise} \end{cases}$$

We want to check that δ' satisfies the condition Definition 5 (3) for all $f' \in \sigma_{13}(s,u))$ and $(t',u') \in \mathcal{Q}_{23}$. We prove it for all $t' \in \mathcal{Q}_2$ as follows.

$$\begin{array}{ll}
 & \text{(By } (\dagger_3)) \\
 & \sum_{s' \in \mathcal{Q}_1} \sigma_1(s)(s') * \delta(s')(t') \subseteq \sigma_2(t)(t') \\
\iff & \text{(By arithmetic, if } [a,b],[c,d],[e,f] \subseteq [0,1], \text{ then} \\
 & [a,b] \subseteq [c,d] \iff [a,b]*[e,f] \subseteq [c,d]*[e,f]. \\
 & \text{We also know } \sigma_3(u)(u') \subseteq [0,1]) \\
 & \forall u' \in \mathcal{Q}_3, \sum_{s' \in \mathcal{Q}_1} \sigma_1(s)(s') * \sigma_3(u)(u') * \delta(s')(t'), \\
 & \subseteq \sigma_2(t)(t') * \sigma_3(u)(u') \\
\iff & \text{(By } (\dagger_1) \text{ and } (\dagger_2)) \\
 & \forall u' \in \mathcal{Q}_3, \sum_{s' \in \mathcal{Q}_1} \sigma_{13}(s,u)(s',u') * \delta'(s')(t') \subseteq \sigma_{23}(t,u)(t',u') \\
\iff & \text{(For } u'' \neq u', \sum_{(s',u'') \in \mathcal{Q}_{13}} \text{ does not add any non-zero term.} \\
 & \text{Also by definition of } \delta') \\
 & \forall u' \in \mathcal{Q}_3, \sum_{(s',u'') \in \mathcal{Q}_{13}} \sigma_{13}(s,u)(s',u'') * \delta'(s',u'')(t',u') \subseteq \sigma_{23}(t,u)(t',u') \\
\iff & \text{(By definition of } f') \\
 & \forall u' \in \mathcal{Q}_3, \sum_{(s',u'') \in \mathcal{Q}_{13}} (f'(s',u'') * \delta'(s',u'')(t',u')) \in \sigma_{23}(t,u)(t',u')
\end{array}$$

□

Theorem 1 [Congruence of refinement for $||_{\mathcal{I}}$] For all contracts C_1, C_2, C_3, C_4 and an interaction set $\mathcal{I}$, if $C_1 \leq C_2$ and $C_3 \leq C_4$, then $C_1 ||_{\mathcal{I}}\, C_3 \;\leq\; C_2 ||_{\mathcal{I}}\, C_4$.

Proof.

$$\begin{array}{cl}
 & C_1 \leq C_2 \text{ and } C_3 \leq C_4 \\
\Rightarrow & \text{(By Lemma 12 [Congruence of } \leq \text{])} \\
 & C_1||_{\mathcal{I}}C_3 \leq C_2||_{\mathcal{I}}C_3 \text{ and } C_3 \leq C_4 \\
\Longleftrightarrow & \text{(By Lemma 10 [Commutativity of } ||_{\mathcal{I}}\text{])} \\
 & C_1||_{\mathcal{I}}C_3 \leq C_3||_{\mathcal{I}}C_2 \text{ and } C_3 \leq C_4 \\
\Rightarrow & \text{(By Lemma 12, } C_3 \leq C_4 \Rightarrow C_3||_{\mathcal{I}}C_2 \leq C_4||_{\mathcal{I}}C_2) \\
 & C_1||_{\mathcal{I}}C_3 \leq C_3||_{\mathcal{I}}C_2 \text{ and } C_3||_{\mathcal{I}}C_2 \leq C_4||_{\mathcal{I}}C_2) \\
\Rightarrow & \text{(By Lemma 2 [Transitivity of } \leq \text{])} \\
 & C_1||_{\mathcal{I}}C_3 \leq C_4||_{\mathcal{I}}C_2 \\
\Longleftrightarrow & \text{(By Lemma 10 [Commutativity of } ||_{\mathcal{I}}\text{])} \\
 & C_1||_{\mathcal{I}}C_3 \leq C_2||_{\mathcal{I}}C_4
\end{array}$$

□

B.2 Conjunction of Contracts

Theorem 4 [Conjunction is a common refinement] For all contracts C_1 and C_2, $\pi_{\mathcal{A}_i}(C_1 \wedge C_2) \leq C_i$ for $i = 1, 2$.

Proof. We only show the proof for $\pi_{\mathcal{A}_1}(C_1 \wedge C_2) \leq C_1$ as the proof for $\pi_{\mathcal{A}_2}(C_1 \wedge C_2) \leq C_2$ is similar.

Let

$$\begin{array}{rcl}
C_1 & = & (\mathcal{Q}_1, \mathcal{A}_1, \rightarrow_1, \dashrightarrow_1, s_0) \\
C_2 & = & (\mathcal{Q}_2, \mathcal{A}_2, \rightarrow_2, \dashrightarrow_2, t_0) \\
\pi_{\mathcal{A}_1}(C_1 \wedge C_2) & = & (\mathcal{Q}_{12}, \mathcal{A}_1, \rightarrow_{12}, \dashrightarrow_{12}, (s_0, t_0))
\end{array}$$

Let $\theta \subseteq \mathcal{Q}_{12} \times \mathcal{Q}_1$ be a binary relation such that

$$\{((s,t),s) \mid s \in \mathcal{Q}_1, t \in \mathcal{Q}_2, (s,t) \in \mathcal{Q}_{12}\}$$

We want to show that θ allows us to establish $\theta \subseteq \leq$.

There are 8 cases to consider when we perform $C_1 \wedge C_2$. Since projection is only done for action transitions where the action is in $\mathcal{A}_2$ and not in $\mathcal{A}_1$, it only has effect for the case [LiftR].

- Rule [C1]. If we have $(s,t) \xrightarrow{\alpha}_{12} (s',t')$, by rule [C1], we have $s \xrightarrow{\alpha}_1 s'$ (and $\alpha \in \mathcal{A}_1$) and $t \xrightarrow{\alpha}_2 t'$. We have the following co-induction hypothesis:

  $$(s',t') \leq s' \qquad \text{[HC1]}$$

  Since we have $s \xrightarrow{\alpha}_1 s'$ and $(s,t) \xrightarrow{\alpha}_{12} (s',t')$ and [HC1], it is easy to check that Definition 5 ($\leq$) (2a) and (2b) are satisfied and since $(s,t) \neq \top$, Definition 5 ($\leq$) (1) is vacuously true.

- Rule [C2L]. If we have $(s,\top) \xrightarrow{\alpha}_{12} (s',\top)$, by rule [C2L], we have $s \xrightarrow{\alpha}_1 s'$. We have the following co-induction hypothesis:

  $$(s',\top) \leq s' \qquad \text{[HC2L]}$$

  Since we have $s \xrightarrow{\alpha}_1 s'$ and $(s,\top) \xrightarrow{\alpha}_{12} (s',\top)$ and [HC2L], it is easy to check that Definition 5 ($\leq$) (2a) and (2b) are satisfied and since $(s,t) \neq \top$, Definition 5 ($\leq$) (1) is vacuously true.

- Rule [C2R]. Since C_1 has reached $\top$ state and any state refines $\top$ state, we are done.

- Rule [C3]. If we have $(s,t) \overset{[p5,p6]}{\dashrightarrow}_{12} (s',t')$ where $p_5 = max(p_1,p_3)$ and $p_6 = min(p_2,p_4)$ and $s' \sim t'$, by rule [C3], we have $s \overset{[p_1,p_2]}{\dashrightarrow}_1 s'$ and $t \overset{[p_3,p_4]}{\dashrightarrow}_2 t'$. We have the following co-induction hypothesis:

  $$(s',t') \leq s' \qquad \text{[HC3]}$$

  Since $\leq$ is reflexive (by Lemma 1), we have $s \leq s$. we know there is a probability distribution $\delta \subset \mathcal{Q}_1 \times \mathcal{Q}_1 \times [0,1]$, such that, $\forall f \in \sigma(s), s' \in \mathcal{Q}_1$,

  $$(\dagger) \quad \sum_{s' \in \mathcal{Q}_1}(f(s') * \delta(s')(s')) \in \sigma_1(s)(s'), \text{and } \delta(s')(s') > 0 \implies s' \leq s'$$

  We want to establish a δ' such that Definition 5 (3) holds. Let $\delta' \subset \mathcal{Q}_{12} \times \mathcal{Q}_1 \times [0,1]$ be

  $$\delta'(s',t')(s') = \delta(s')(s')$$

  We want to check that δ' satisfies the condition Definition 5 (3) for all $f' \in \delta_{12}(s,t)$.

  $$\begin{array}{ll}
  & \text{(By } (\dagger)) \\
  & \sum_{s' \in \mathcal{Q}_1}(f(s') * \delta(s')(s')) \in \sigma_1(s)(s') \\
  \iff & \text{(By definition of } f) \\
  & \sum_{s' \in \mathcal{Q}_1}([\underline{\sigma_1}(s)(s'), \overline{\sigma_1}(s)(s')] * \delta(s')(s')) \subseteq \sigma_1(s)(s') \\
  \iff & \text{(By rule [C3], } [\underline{\sigma_{12}}(s',t'), \overline{\sigma_{12}}(s',t')] \subseteq [\underline{\sigma_1}(s)(s'), \overline{\sigma_1}(s)(s')]) \\
  & \sum_{s' \in \mathcal{Q}_1}([\underline{\sigma_{12}}(s',t'), \overline{\sigma_{12}}(s',t')] * \delta(s')(s')) \subseteq \sigma_1(s)(s'), \\
  \iff & \text{(By Definition 14 [Unambiguous contract], the similarity between} \\
  & s' \text{ and } t' \text{ is a bijection, so the number of } (s',t') \text{ states is the same as} \\
  & \text{the number of } s' \text{ states.)} \\
  & \sum_{(s',t') \in \mathcal{Q}_{12}}([\underline{\sigma_{12}}(s',t'), \overline{\sigma_{12}}(s',t')] * \delta(s')(s')) \subseteq \sigma_1(s)(s'), \\
  \iff & \text{(By definition of } \delta') \\
  & \sum_{(s',t') \in \mathcal{Q}_{12}}([\underline{\sigma_{12}}(s',t'), \overline{\sigma_{12}}(s',t')] * \delta'(s',t')(s')) \subseteq \sigma_1(s)(s'), \\
  \iff & \text{(By definition of } f') \\
  & \sum_{(s',t') \in \mathcal{Q}_{12}}(f'(s',t') * \delta'(s',t')(s')) \in \sigma_1(s)(s')
  \end{array}$$

  Together with co-induction hypothesis [HC3], we have the desired result.

- Rule [C4L]. If we have $(s,t) \overset{P}{\dashrightarrow}_{12} (s',t)$ and $P \neq [0,0]$, by rule [C4L], we have $s \overset{P}{\dashrightarrow}_1 s', t \in \mathcal{Q}^a$ and $s' \sim t$. We have the following co-induction hypothesis:

  $$(s',t) \leq s' \qquad \text{[HC4L]}$$

  We know there is a probability distribution $\delta \subset \mathcal{Q}_1 \times \mathcal{Q}_1 \times [0,1]$, such that, $\forall f \in \sigma(s), s' \in \mathcal{Q}_1$,

  $$(1) \quad \sum_{s' \in \mathcal{Q}_1}(f(s') * \delta(s')(s')) \in \sigma_1(s)(s'), \text{and } \delta(s')(s') > 0 \implies s' \leq s'$$

  We want to establish a δ' such that Definition 5 (3) holds. Let $\delta' \subset \mathcal{Q}_{12} \times \mathcal{Q}_1 \times [0,1]$ be

  $$\delta'(s',t')(s') = \delta(s')(s')$$

We want to check that δ' satisfies the condition Definition 5 (3) for all $f' \in \delta_{12}(s,t)$.

$$\begin{array}{ll}
 & \text{(By (1))} \\
 & \sum_{s' \in \mathcal{Q}_1}(f(s') * \delta(s')(s')) \in \sigma_1(s)(s') \\
\Longleftrightarrow & \text{(By definition of } f\text{)} \\
 & \sum_{s' \in \mathcal{Q}_1}([\underline{\sigma_1}(s)(s'), \overline{\sigma_1}(s)(s')] * \delta(s')(s')) \subseteq \sigma_1(s)(s') \\
\Longleftrightarrow & \text{(By rule [C4L], } [\underline{\sigma_{12}}(s',t'), \overline{\sigma_{12}}(s',t')] = [\underline{\sigma_1}(s)(s'), \overline{\sigma_1}(s)(s')]\text{)} \\
 & \sum_{(s',t') \in \mathcal{Q}_{12}}([\underline{\sigma_{12}}(s',t'), \overline{\sigma_{12}}(s',t')] * \delta(s')(s')) \subseteq \sigma_1(s)(s'), \\
\Longleftrightarrow & \text{(By definition of } \delta'\text{)} \\
 & \sum_{(s',t') \in \mathcal{Q}_{12}}([\underline{\sigma_{12}}(s',t'), \overline{\sigma_{12}}(s',t')] * \delta'(s',t')(s')) \subseteq \sigma_1(s)(s'), \\
\Longleftrightarrow & \text{(By definition of } f'\text{)} \\
 & \sum_{(s',t') \in \mathcal{Q}_{12}}(f'(s',t') * \delta'(s',t')(s')) \in \sigma_1(s)(s')
\end{array}$$

Together with co-induction hypothesis [HC4L], we have the desired result.

- Rule [C4R]. Similar to the proof for Rule [C4L].

- Rule [LiftL]. If we have $(s,t) \xrightarrow{\alpha}_{12} (s',t)$, by rule [LiftL], we have $s \xrightarrow{\alpha}_1 s', \alpha \notin \mathcal{A}_2$ and $q_2 \in \mathcal{Q}_2^{\alpha}$. We have the following co-induction hypothesis:

  $$(s',t) \leq s' \qquad \text{[HLiftL]}$$

  Since we have $s \xrightarrow{\alpha}_1 s'$ and $(s,t) \xrightarrow{\alpha}_{12} (s',t)$ and [HLiftL], it is easy to check that Definition 5 ($\leq$) (2a) and (2b) are satisfied and since $(s,t) \neq \top$, (1) is vacuously true.

- Rule [LiftR]. If we have $(s,t) \xrightarrow{\alpha}_{12} (s,t')$, by rule [LiftR], we have $t \xrightarrow{\alpha}_1 t', \alpha \notin \mathcal{A}_1$ and $q_2 \in \mathcal{Q}_2^{\alpha}$. We have the following co-induction hypothesis:

  $$(s,t') \leq s \qquad \text{[HLiftR]}$$

  After projection on $\mathcal{A}_1$, we have $(s,t) = (s,t')$. By [HLiftR], we know $(s,t) \leq s$ so we are done.

□

Lemma 13 (Abstraction and lifting)**.** *For all contracts C, $\lfloor \pi_{\mathcal{A}}(C) \rfloor \equiv \pi_{\mathcal{A}}(\lfloor C \rfloor)$*

Proof. By inspecting definition of $\lfloor . \rfloor$ and Definition 11 [Projection], lifting is like an identity operation except converting a single probability p to an interval $[p,p]$, so it is obvious that doing lifting before or after projection have the same effect. □

Theorem 5 [Soundness of conjunction] For any IMC M and unambiguous contracts C_i with alphabets $\mathcal{A}_i$, $i = 1,2$, such that $C_1 \sim C_2$, if $M \models C_1 \wedge C_2$ then $\pi_{\mathcal{A}_i}(M) \models C_i$, $i = 1,2$.

Proof.

$$
\begin{array}{cl}
 & M \models C_1 \wedge C_2 \\
\Longleftrightarrow & \text{(By Definition 6 } \models) \\
 & \lfloor M \rfloor \leq C_1 \wedge C_2 \\
\Rightarrow & \text{(By Lemma 9 [Abstraction and refinement])} \\
 & \pi_{\mathcal{A}_i}(\lfloor M \rfloor) \leq \pi_{\mathcal{A}_i}(C_1 \wedge C_2) \text{ for } i = 1, 2 \\
\Rightarrow & \text{(By Lemma 2 [Transitivity of } \leq \text{] and} \\
 & \text{by Theorem 4 [Conjunction is a common refinement])} \\
 & \pi_{\mathcal{A}_1}(\lfloor M \rfloor) \leq C_1 \text{ and } \pi_{\mathcal{A}_2}(\lfloor M \rfloor) \leq C_2 \\
\Longleftrightarrow & \text{(By Lemma 13 [Abstraction and lifting])} \\
 & \lfloor \pi_{\mathcal{A}_1}(M) \rfloor \leq C_1 \text{ and } \lfloor \pi_{\mathcal{A}_2}(M) \rfloor \leq C_2 \\
\Longleftrightarrow & \text{(By Definition 6 } \models) \\
 & \pi_{\mathcal{A}_1}(M) \models C_1 \text{ and } \pi_{\mathcal{A}_2}(M) \models C_2
\end{array}
$$

□

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
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 Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes 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