Timed Process Calculi with Deterministic or Stochastic Delays: Commuting between Durational and Durationless Actions

Marco Bernardo[a], Flavio Corradini[b], Luca Tesei[b]

[a] *Dipartimento di Scienze Pure e Applicate, Università di Urbino, Italy*
[b] *Scuola di Scienze e Tecnologie, Università di Camerino, Italy*

Abstract

Several deterministically/stochastically timed process calculi have been proposed in the literature that, apart from their synchronization mechanism, mainly differ for the way in which actions and delays are represented. In particular, a distinction is made between integrated-time calculi, in which actions are durational, and orthogonal-time calculi, in which actions are instantaneous and delays are expressed separately. In a previous work on deterministic time, the two approaches have been shown to be reconcilable through an encoding from the integrated-time calculus CIPA to the orthogonal-time calculus TCCS, which preserves strong timed bisimilarity under certain conditions. In this paper, the picture is completed by first defining a reverse encoding from TCCS to CIPA, which requires slight modifications to both calculi and is shown to preserve only weak timed bisimilarity under conditions tighter than those for the direct encoding. Stochastic time is then addressed, by exhibiting an encoding from the integrated-time calculus MTIPP to the orthogonal-time calculus IML, together with a reverse encoding requiring slight modifications only to the former calculus, with both encodings being shown to preserve strong Markovian bisimilarity under suitable conditions. All the four encodings rely on the assumption that action execution is urgent. Variants are finally discussed in which action execution is delayable, in the sense that an enabled action can let time advance before starting its execution, or only the execution of internal actions is urgent, which is known as maximal progress.

Keywords: timed process calculi, deterministic time, stochastic time, bisimulation, expressiveness

1. Introduction

Computing systems are characterized not only by their functional behavior, but also by their quantitative features. In particular, *timing aspects* play a fundamental role, as they describe the temporal evolution of system activities. This is especially true for *real-time systems*, which are considered correct only if the execution of their activities fulfills certain *temporal constraints*, as well as *shared-resource systems*, in which resource contention determines stochastic fluctuations of the service level measured through suitable *performance indices*. On the modeling side, time is expressed through fixed numbers in the case of real-time systems, yielding a *deterministic* representation of time, or random variables in the case of shared-resource systems, originating a *stochastic* representation of time. In the following, we refer to *abstract time*, in the sense that we use time as a parameter for expressing constraints about instants of occurrences of actions. Unlike physical time, abstract time enables simplifications that allow tractable models to be obtained.

Many *timed process calculi* have appeared in the literature starting from the late 1980's. Among those with *deterministic delays*, we mention timed CSP [35], temporal CCS [29], timed CCS [39], real-time ACP [2], urgent LOTOS [10], CIPA [1], TPL [20], ATP [32], TIC [34], and PAFAS [14]. As observed in [31, 38, 12], these calculi differ on the basis of a number of *time-related options*, some of which are recalled below:

- *Durationless actions* versus *durational actions*. In the first case, actions are instantaneous events and time passes in between them; hence, functional behavior and time are *orthogonal*. In the second case, every action takes a certain amount of time to be performed and time passes only due to action execution; hence, functional behavior and time are *integrated*.

- *Relative time* versus *absolute time*. Assume that timestamps are associated with the events observed during system execution. In the first case, each timestamp refers to the time instant of the previous observation. In the second case, all timestamps refer to the starting time of the system execution.
- *Global clock* versus *local clocks*. In the first case, there is a single clock that governs time passing. In the second case, there are several clocks associated with the various system parts, which elapse independent of each other although they contribute to define a unique notion of global time.

Moreover, for these deterministically timed process calculi there are several different *interpretations of action execution*, in terms of whether and when the execution can be delayed, such as:

- *Eagerness*: actions must be performed as soon as they become enabled, i.e., without any delay, thereby implying that their execution is *urgent*.
- *Laziness*: after getting enabled, actions can be delayed arbitrarily long before they are executed.
- *Maximal progress*: enabled actions can be delayed arbitrarily long unless they are independent of the external environment, in which case their execution is urgent.

A comparative study was conducted in [12] by one of the authors, which focusses on two different deterministically timed process calculi obtained by suitably combining the time-related options above. One of the two calculi, TCCS [29], is inspired by the *two-phase functioning principle*, according to which actions are durationless, time is relative, and there is a single global clock. In contrast, the other calculus, CIPA [1], is inspired by the *one-phase functioning principle*, according to which actions are durational, time is absolute, and several local clocks are present. The two considered principles are among the richest combinations of time features [31, 38, 12], thus allowing for a deep investigation of their relative expressiveness, and the two considered calculi are well-known and paradigmatic instances of the two principles themselves.

In [12], it was shown that the multiple choices concerned with the time-related options are not irreconcilable under the various action execution interpretations. More precisely, a timed-bisimilarity-preserving encoding of CIPA into TCCS was developed for each action execution interpretation. Preservation turned out to depend on certain conditions under action eagerness and to be unconditional under action laziness and maximal progress, thus revealing when the two process calculi have a different expressive power.

In this paper, we complete the previous expressiveness study in two directions. First of all, we consider a reverse encoding from TCCS to CIPA under action eagerness. As pointed out at the end of [12], several issues need to be addressed before such a reverse encoding can be established. Our contribution consists of providing an answer to each of the various issues, which will lead to slight modifications of both calculi. The reverse encoding is proved to be fully abstract with respect to *weak* timed bisimilarity, as opposed to the direct one demonstrated to be fully abstract with respect to *strong* timed bisimilarity in [12].

The second direction that we explore is concerned with process calculi extended with *stochastic delays*, such as for instance MTIPP [18, 22], PEPA [25], MPA [11], EMPA_{gr} [8, 5], Sπ [33], IML [21], and PIOA [36]. In these calculi, delays are no longer expressed through nonnegative numbers, but through nonnegative random variables. The latter typically follow *exponential distributions*, each characterized by a positive real number called *rate*, so that the underlying stochastic processes turn out to be *continuous-time Markov chains*. These models are mathematically tractable [37] and fit well with the interleaving view of concurrency thanks to the *memoryless property* of the considered distributions.

The time-related options and action execution interpretations discussed for deterministically timed calculi apply to a large extent also to such stochastically timed calculi. This is especially true for the difference between durationless and durational actions. In contrast, the distinction between relative and absolute time and the concept of clock are not so important in a Markovian framework due to the memoryless property of exponential distributions, which establishes that the residual time to the termination of an event follows the same exponential distribution as the overall duration of the event.

However, there is a fundamental difference between deterministically timed calculi and stochastically timed calculi in terms of choices among alternative behaviors. In the former calculi, all choices are *nondeterministic* precisely as in classical process calculi, because time passing cannot affect choices according to

time determinism. In the latter calculi, choices can instead be *probabilistic* when they are based on delays or durational actions, because in that case the *race policy* is adopted, which means that the event sampling the least duration is the one that takes place. For example, in an orthogonal-time setting, the deterministically timed process $(t) \,.\, P_1 + (t) \,.\, P_2$ – where $+$ denotes the alternative composition operator – can let t time units pass and then evolves into $P_1 + P_2$ without resolving the choice, while the stochastically timed process $(\lambda_1) \,.\, P_1 + (\lambda_2) \,.\, P_2$ – where λ_1 and λ_2 are the rates of two exponentially distributed delays – evolves into either P_1 or P_2 with probabilities $\frac{\lambda_1}{\lambda_1+\lambda_2}$ and $\frac{\lambda_2}{\lambda_1+\lambda_2}$, respectively.

This has an impact on the expressiveness of Markovian process calculi. The orthogonal-time ones are more expressive than the integrated-time ones, because the former can represent both action-based nondeterministic choices and time-based probabilistic choices, whereas the latter can represent only probabilistic choices based on action durations. In turn, this has an impact on the expressiveness of the synchronization discipline adopted in the considered calculi. Indeed, in the orthogonal-time case the time to the synchronization of two actions can be naturally described as the *maximum* of two exponentially distributed random variables, expressed as the interleaving of the two corresponding rates, whereas in the integrated-time case the duration of the synchronization of two exponentially timed actions can only be assumed to be exponentially distributed, with rate given by the application of an associative and commutative operation to the two original rates. This difference in expressiveness is compensated for by the fact that the states of the continuous-time Markov chain underlying an integrated-time process are in one-to-one correspondence with the states of the process itself – which is appreciated by many performance modelers – while a suitable algorithm has to be applied to derive the Markov chain in the case of an orthogonal-time process [21].

In spite of the different expressiveness they induce, we show that durationless and durational actions are reconcilable in a Markovian setting too. This is accomplished by concentrating on the orthogonal-time calculus IML and the integrated-time calculus MTIPP. For these two well-known exponentially timed calculi, we define a direct encoding and a reverse encoding under action eagerness, which are both proved to preserve strong Markovian bisimilarity under suitable conditions.

The paper, which is a revised, extended, and integrated version of [6] and [3], is organized as follows. In Sect. 2, we present syntax, operational semantics, and bisimilarity for TCCS, CIPA, IML, and MTIPP. In Sect. 3, we recall from [12] the direct encoding from CIPA to TCCS under action eagerness, together with the corresponding full abstraction result with respect to strong timed bisimilarity. In Sects. 4, 5, and 6, we respectively provide, under action eagerness, the reverse encoding from TCCS to CIPA and the direct and reverse encodings between MTIPP and IML and show the related full abstraction results with respect to bisimulation semantics. In Sect. 7, we discuss variants of the encodings in which action laziness or maximal progress is assumed in place of action eagerness. Finally, in Sect. 8 we provide some concluding remarks.

2. Timed Process Calculi: Syntax, Operational Semantics, and Bisimulation Equivalence

In this section, after introducing some notation, we recall syntax, operational semantics, and strong/weak bisimilarity for the two timed process languages with deterministic delays (TCCS and CIPA) and the two timed process languages with exponentially distributed delays (IML and MTIPP) considered in this paper.

2.1. Preliminaries for Calculi with Deterministic Time

Deterministically timed process calculi adopt a CCS-like, *binary* synchronization mechanism [28]. In this setting, we denote by A a nonempty set of *visible actions* – ranged over by a, b – and by $\bar{A} = \{\bar{a} \mid a \in A\}$ the set of corresponding *coactions*, such that $\bar{\bar{a}} = a$ for all $a \in A$. We use $Act = A \cup \bar{A} \cup \{\tau\}$ to indicate the set of all actions – ranged over by α, β – where τ is the *invisible action* resulting from the synchronization of a visible action with its coaction. We then let *Rel* be a set of *action relabeling functions*; each such function $\varphi : Act \to Act$ satisfies $\varphi(\tau) = \tau$ as well as $\varphi(a) \in Act \setminus \{\tau\}$ and $\overline{\varphi(a)} = \varphi(\bar{a})$ for all $a \in Act \setminus \{\tau\}$.

We denote by $\mathcal{T} = (T, \boxplus, \sqsubseteq)$ a *time domain* such that $T \cap Act = \emptyset$, equipped with a commutative and associative operation $\boxplus$ with neutral element 0_T and a total order relation $\sqsubseteq$ satisfying $t_1 \sqsubseteq t_2$ iff there exists $t' \in T$ such that $t_1 \boxplus t' = t_2$; typical choices are $(\mathbb{N}, +, \leq)$ and $(\mathbb{R}_{\geq 0}, +, \leq)$. We recall the following laws:

- If process P can evolve to both P'_1 and P'_2 in the same time $t \in T$, then $P'_1 = P'_2$ (*time determinism*).

- P can evolve to P' in time $t' \in T$ and P' can evolve to P'' in time $t'' \in T$ iff P can directly evolve to P'' in time $t = t' \boxplus t''$, with $P = P'$ when $t' = 0_T$ and $P' = P''$ when $t'' = 0_T$ (*time additivity*).

2.2. TCCS: Durationless Actions and Deterministic Delays

The process calculus TCCS [29] features instantaneous actions separated from fixed delays, motivated by the authors of that work on the basis of the fact that computation involves energy changes and that such changes and time cannot be measured simultaneously by a result of quantum mechanics. As in [12], from the syntax of TCCS we leave out the idling operator δ and the weak choice operator $\oplus$ because they have no counterpart in CIPA. In the following, we assume action execution to be *urgent* and we denote by *Var* a nonempty set of process variables – ranged over by X, Y – whose occurrences can be free or bound by the recursion constructor "rec".

Definition 2.1. The set $\mathcal{P}_{\mathrm{TCCS}}$ of TCCS process terms is generated by the following syntax:

P	::=	$\mathbf{0}$	stopped process
	|	$\alpha \,.\, P$	instantaneous action prefix
	|	$(t) \,.\, P$	fixed delay prefix
	|	$P + P$	alternative composition
	|	$P \mid P$	CCS-like parallel composition
	|	$P \setminus L$	restriction
	|	$P\,[\varphi]$	relabeling
	|	X	process variable
	|	$\mathrm{rec}\, X : P$	recursion

where $\alpha \in \mathit{Act}$, $t \in \mathbb{N}_{>0}$, $L \subseteq A$, $\varphi \in \mathit{Rel}$, and $X \in \mathit{Var}$. We denote by $\mathbb{P}_{\mathrm{TCCS}}$ the set of closed and guarded process terms of $\mathcal{P}_{\mathrm{TCCS}}$. ■

Process $\mathbf{0}$ can neither proceed with any action, nor proceed through time. Process $\alpha \,.\, P$ can perform the instantaneous action α and then evolves into process P; since the execution of α is urgent, time cannot progress before α is performed. Process $(t) \,.\, P$ evolves into process P after a delay equal to t.

Process $P_1 + P_2$ represents a *nondeterministic* choice between processes P_1 and P_2, with the choice being resolved depending on whether an action of P_1 or P_2 is executed first. According to time determinism, time passing cannot resolve choices, in the sense that any initial passage of time common to P_1 and P_2 must be allowed without making the choice. Process $P_1 \mid P_2$ describes the parallel composition of processes P_1 and P_2, where any two complementary actions may synchronize thereby resulting in a τ-action. Also in this case, any initial passage of time common to P_1 and P_2 must be permitted.

Process $P \setminus L$ behaves as process P except for actions belonging to the restriction set $L \cup \bar{L}$, whose execution is forbidden; this is useful to force synchronizations between complementary actions. Process $P\,[\varphi]$ behaves as process P, with the difference that every action is transformed via φ upon execution; this allows processes with different actions to communicate. Finally, $\mathrm{rec}\, X : P$ represents a recursive process, which behaves as process P in which every free occurrence of X is replaced by $\mathrm{rec}\, X : P$ itself; the resulting process will be denoted by $P\{\mathrm{rec}\, X : P \hookrightarrow X\}$.

Following [29], with every $\mathbb{P}_{\mathrm{TCCS}}$ process term is associated a labeled transition system defined according to Table 1. States correspond to terms, while transitions are labeled with actions or nonzero delays. The *action transition* relation $\longrightarrow$ on the left represents the functional behavior. The *delay transition* relation $-\!\!\rightsquigarrow$ on the right represents the timing behavior according to action eagerness (no delay rule for action prefix), time additivity (second and third rules), and time determinism (fourth and fifth rules); the second rule is necessary for the applicability of the fourth and fifth ones, while the third rule is necessary for achieving desirable identifications in the forthcoming equivalences. The presence of two distinct transition relations emphasizes that, in TCCS, action execution is orthogonal to time passing.

Strong bisimilarity for TCCS was defined in [29] by extending Milner's strong bisimilarity for purely nondeterministic processes [28].

$\dfrac{}{\alpha\,.\,P \xrightarrow{\alpha} P}$	$\dfrac{}{(t)\,.\,P \overset{t}{\rightsquigarrow} P}$
$\dfrac{P_1 \xrightarrow{\alpha} P_1'}{P_1 + P_2 \xrightarrow{\alpha} P_1'} \quad \dfrac{P_2 \xrightarrow{\alpha} P_2'}{P_1 + P_2 \xrightarrow{\alpha} P_2'}$	$\dfrac{}{(t' + t'')\,.\,P \overset{t'}{\rightsquigarrow} (t'')\,.\,P} \quad \dfrac{P \overset{t''}{\rightsquigarrow} P'}{(t')\,.\,P \overset{t'+t''}{\rightsquigarrow} P'}$
$\dfrac{P_1 \xrightarrow{\alpha} P_1'}{P_1 \mid P_2 \xrightarrow{\alpha} P_1' \mid P_2} \quad \dfrac{P_2 \xrightarrow{\alpha} P_2'}{P_1 \mid P_2 \xrightarrow{\alpha} P_1 \mid P_2'}$	$\dfrac{P_1 \overset{t}{\rightsquigarrow} P_1' \quad P_2 \overset{t}{\rightsquigarrow} P_2'}{P_1 + P_2 \overset{t}{\rightsquigarrow} P_1' + P_2'}$
$\dfrac{P_1 \xrightarrow{a} P_1' \quad P_2 \xrightarrow{\bar{a}} P_2'}{P_1 \mid P_2 \xrightarrow{\tau} P_1' \mid P_2'}$	$\dfrac{P_1 \overset{t}{\rightsquigarrow} P_1' \quad P_2 \overset{t}{\rightsquigarrow} P_2'}{P_1 \mid P_2 \overset{t}{\rightsquigarrow} P_1' \mid P_2'}$
$\dfrac{P \xrightarrow{\alpha} P' \quad \alpha \notin L \cup \bar{L}}{P \setminus L \xrightarrow{\alpha} P' \setminus L}$	$\dfrac{P \overset{t}{\rightsquigarrow} P'}{P \setminus L \overset{t}{\rightsquigarrow} P' \setminus L}$
$\dfrac{P \xrightarrow{\alpha} P'}{P\,[\varphi] \xrightarrow{\varphi(\alpha)} P'\,[\varphi]}$	$\dfrac{P \overset{t}{\rightsquigarrow} P'}{P\,[\varphi] \overset{t}{\rightsquigarrow} P'\,[\varphi]}$
$\dfrac{P\{\mathrm{rec}\,X : P \hookrightarrow X\} \xrightarrow{\alpha} P'}{\mathrm{rec}\,X : P \xrightarrow{\alpha} P'}$	$\dfrac{P\{\mathrm{rec}\,X : P \hookrightarrow X\} \overset{t}{\rightsquigarrow} P'}{\mathrm{rec}\,X : P \overset{t}{\rightsquigarrow} P'}$

Table 1: Structural operational semantic rules for TCCS

Definition 2.2. A symmetric relation $\mathcal{B}$ over $\mathbb{P}_{\mathrm{TCCS}}$ is a *strong timed bisimulation* iff, whenever $(P_1,P_2) \in \mathcal{B}$, then for all actions $\alpha \in Act$ and delays $t \in \mathbb{N}_{>0}$ it holds that:

- For each $P_1 \xrightarrow{\alpha} P_1'$ there exists $P_2 \xrightarrow{\alpha} P_2'$ such that $(P_1', P_2') \in \mathcal{B}$.
- For each $P_1 \overset{t}{\rightsquigarrow} P_1'$ there exists $P_2 \overset{t}{\rightsquigarrow} P_2'$ such that $(P_1', P_2') \in \mathcal{B}$.

We write $P_1 \sim_{\mathrm{TCCS}} P_2$ to denote that (P_1, P_2) is contained in a strong timed bisimulation. ■

A notion of weak bisimilarity for TCCS was subsequently studied in [30]. It is an extension of Milner's weak bisimilarity [28] that is capable of summing up delays while abstracting from τ-actions. For a smoother comparison with the weak bisimilarity for CIPA, and hence an appropriate formalization of the full abstraction result for the reverse encoding, we present the weak bisimilarity for TCCS in the so-called *delay bisimulation* style, by leaving out trailing τ-actions. We thus define TCCS weak transitions as follows:

- $\Longrightarrow = (\xrightarrow{\tau})^*$.
- $\overset{a}{\Longrightarrow} = \Longrightarrow \xrightarrow{a}$ for $a \in Act \setminus \{\tau\}$.
- $\overset{\hat{\alpha}}{\Longrightarrow} = \begin{cases} \Longrightarrow & \text{if } \alpha = \tau \\ \overset{\alpha}{\Longrightarrow} & \text{if } \alpha \neq \tau \end{cases}$.
- $\overset{t}{\Longrightarrow} = \Longrightarrow \overset{t_1}{\rightsquigarrow} \cdots \Longrightarrow \overset{t_n}{\rightsquigarrow}$ where $n \in \mathbb{N}_{\geq 1}$, $t_i \in \mathbb{N}_{>0}$ for $1 \leq i \leq n$, and $t = \sum_{1 \leq i \leq n} t_i$.

Definition 2.3. A symmetric relation $\mathcal{B}$ over $\mathbb{P}_{\mathrm{TCCS}}$ is a *weak timed bisimulation* iff, whenever $(P_1, P_2) \in \mathcal{B}$, then for all actions $\alpha \in Act$ and delays $t \in \mathbb{N}_{>0}$ it holds that:

- For each $P_1 \stackrel{\alpha}{\longrightarrow} P_1'$ there exists $P_2 \stackrel{\hat{\alpha}}{\Longrightarrow} P_2'$ such that $(P_1', P_2') \in \mathcal{B}$.
- For each $P_1 \stackrel{t}{\rightsquigarrow} P_1'$ there exists $P_2 \stackrel{t}{\Longrightarrow} P_2'$ such that $(P_1', P_2') \in \mathcal{B}$.

We write $P_1 \approx_{\mathrm{TCCS}} P_2$ to denote that (P_1, P_2) is contained in a weak timed bisimulation. ■

2.3. CIPA: Deterministically Timed Actions

The process calculus CIPA [1] associates fixed durations with actions, because the authors of that work wanted to show that concurrency information can be captured purely through timing while retaining the advantages of the classic interleaving approach, such as a simpler semantic model and the existence of an expansion law. As in [12], we add the relabeling operator to the syntax of CIPA. Like for TCCS, we assume action execution to be *urgent* and we make use of *Var*.

Definition 2.4. The set $\mathcal{P}_{\mathrm{CIPA}}$ of CIPA process terms is generated by the following syntax:

Q	::=	nil	inactive process
	|	$a\,.\,Q$	durational action prefix
	|	$\mathrm{wait}\,t\,.\,Q$	fixed waiting prefix
	|	$Q + Q$	alternative composition
	|	$Q \mid Q$	CCS-like parallel composition
	|	$Q \setminus L$	restriction
	|	$Q\,[\varphi]$	relabeling
	|	X	process variable
	|	$\mathrm{rec}\,X : Q$	recursion

where $a \in Act \setminus \{\tau\}$, $t \in \mathbb{N}_{>0}$, $L \subseteq A$, $\varphi \in Rel$, $X \in Var$, and the duration of each visible action is established by function $\Delta : (Act \setminus \{\tau\}) \to \mathbb{N}_{>0}$ such that $\Delta(\bar{a}) = \Delta(a)$. We denote by $\mathbb{P}_{\mathrm{CIPA}}$ the set of closed and guarded process terms of $\mathcal{P}_{\mathrm{CIPA}}$. ■

Process nil cannot proceed with any action, but can let time pass unlike process $\mathbf{0}$ of TCCS. Process $a\,.\,Q$ can perform the eager action a and evolves into process Q after $\Delta(a)$ time units; note that all the occurrences of a visible action have the same duration. Process $\mathrm{wait}\,t\,.\,Q$ waits for t time units and then becomes process Q; observe that different occurrences of the waiting prefix may have different durations. We also point out that in $Q_1 + Q_2$ and $Q_1 \mid Q_2$ action durations are not considered at all to determine which action is executed first, i.e., the choice is nondeterministic rather than based on a race.

All the other operators work as in TCCS, with three differences. There are no common initial passages of time to deal with for alternative and parallel composition. Any pair of actions a and $\bar{a}$ can synchronize only if they start at the same time, yielding a τ-action with the same duration as the two original actions. Each relabeling function φ must preserve durations, i.e., $\Delta(\varphi(a)) = \Delta(a)$ for all $a \in Act \setminus \{\tau\}$.

Following [1], in the labeled transition system associated with a $\mathbb{P}_{\mathrm{CIPA}}$ process term, the set of states correspond to process terms *augmented with local clocks*, so to keep track of the time elapsed in the various sequential subprocesses. The extended syntax for the set $\mathcal{KP}$ of augmented process terms – with $\mathbb{KP}$ denoting the set of closed and guarded elements of $\mathcal{KP}$ – is as follows:

$$K \;::=\; k \Rightarrow \mathrm{nil} \mid k \Rightarrow a\,.\,Q \mid k \Rightarrow \mathrm{wait}\,t\,.\,Q \mid k \Rightarrow X \mid k \Rightarrow \mathrm{rec}\,X : Q \mid K + K \mid K \mid K \mid K \setminus L \mid K\,[\varphi]$$

We use the shorthand $k \Rightarrow Q$ to indicate that the clock value $k \in \mathbb{N}$ is distributed over all subprocesses of $Q \in \mathcal{P}_{\mathrm{CIPA}}$ according to the following laws:

$$\begin{aligned} k \Rightarrow (Q_1 + Q_2) &= (k \Rightarrow Q_1) + (k \Rightarrow Q_2) \\ k \Rightarrow (Q_1 \mid Q_2) &= (k \Rightarrow Q_1) \mid (k \Rightarrow Q_2) \\ k \Rightarrow (Q \setminus L) &= (k \Rightarrow Q) \setminus L \\ k \Rightarrow (Q\,[\varphi]) &= (k \Rightarrow Q)\,[\varphi] \end{aligned}$$

$$\frac{}{k \Rightarrow (a\,.\,Q) \xrightarrow[\Delta(a)]{a@k} (k+\Delta(a)) \Rightarrow Q} \qquad \frac{}{k \Rightarrow (\mathrm{wait}\,t\,.\,Q) \xrightarrow[t]{\tau@k} (k+t) \Rightarrow Q}$$

$$\frac{K_1 \xrightarrow[t]{\alpha@k} K_1' \quad \neg(K_2 \xrightarrow[t']{\alpha'@k'} K_2' \wedge k' < k)}{K_1 + K_2 \xrightarrow[t]{\alpha@k} K_1'} \qquad \frac{K_2 \xrightarrow[t]{\alpha@k} K_2' \quad \neg(K_1 \xrightarrow[t']{\alpha'@k'} K_1' \wedge k' < k)}{K_1 + K_2 \xrightarrow[t]{\alpha@k} K_2'}$$

$$\frac{K_1 \xrightarrow[t]{\alpha@k} K_1' \quad \neg(K_2 \xrightarrow[t']{\alpha'@k'} K_2' \wedge k' < k)}{K_1 \mid K_2 \xrightarrow[t]{\alpha@k} K_1' \mid K_2} \qquad \frac{K_2 \xrightarrow[t]{\alpha@k} K_2' \quad \neg(K_1 \xrightarrow[t']{\alpha'@k'} K_1' \wedge k' < k)}{K_1 \mid K_2 \xrightarrow[t]{\alpha@k} K_1 \mid K_2'}$$

$$\frac{K_1 \xrightarrow[\Delta(a)]{a@k} K_1' \quad K_2 \xrightarrow[\Delta(a)]{\bar{a}@k} K_2'}{K_1 \mid K_2 \xrightarrow[\Delta(a)]{\tau@k} K_1' \mid K_2'}$$

$$\frac{K \xrightarrow[t]{\alpha@k} K' \quad \alpha \notin L \cup \bar{L}}{K \setminus L \xrightarrow[t]{\alpha@k} K' \setminus L} \qquad \frac{K \xrightarrow[t]{\alpha@k} K'}{K\,[\varphi] \xrightarrow[t]{\varphi(\alpha)@k} K'\,[\varphi]} \qquad \frac{k \Rightarrow (Q\{\mathrm{rec}\,X : Q \hookrightarrow X\}) \xrightarrow[t]{\alpha@k} K'}{k \Rightarrow (\mathrm{rec}\,X : Q) \xrightarrow[t]{\alpha@k} K'}$$

Table 2: Structural operational semantic rules for CIPA

In this setting, any transition is of the form $K \xrightarrow[t]{\alpha@k} K'$, meaning that $K \in \mathbb{KP}$ performs an action of name $\alpha \in Act$ that starts at time $k \in \mathbb{N}$ and has duration $t \in \mathbb{N}_{>0}$, after which it evolves to $K' \in \mathbb{KP}$. The transition relation is defined in Table 2, where neither time determinism nor time additivity is taken into account. Negative premises are present as in [12]. Those in the rules for alternative composition enforce action eagerness. Those in the rules for parallel composition avoid the generation of *ill-timed paths*, i.e., computations along which the starting time of some transitions is lower than that of preceding transitions.

Strong bisimilarity for CIPA can be defined in a way analogous to $\sim_{\mathrm{TCCS}}$.

Definition 2.5. A symmetric relation $\mathcal{B}$ over $\mathbb{KP}$ is a *strong timed bisimulation* iff, whenever $(K_1, K_2) \in \mathcal{B}$, then for all actions $\alpha \in Act$, starting times $k \in \mathbb{N}$, and durations $t \in \mathbb{N}_{>0}$ it holds that:

- For each $K_1 \xrightarrow[t]{\alpha@k} K_1'$ there exists $K_2 \xrightarrow[t]{\alpha@k} K_2'$ such that $(K_1', K_2') \in \mathcal{B}$.

We write $K_1 \sim_{\mathrm{CIPA}} K_2$ to denote that (K_1, K_2) is contained in a strong timed bisimulation. ■

A notion of weak bisimilarity for CIPA capable of summing up consecutive waitings was studied in [1]. It was developed in the branching style of Van Glabbeek and Weijland [16]. To make it comparable with $\approx_{\mathrm{TCCS}}$, we present the weak bisimilarity for CIPA in the same delay bisimulation style used for $\approx_{\mathrm{TCCS}}$, by removing the branching constraint according to which the states traversed by a sequence of τ-actions should be equivalent to each other. We thus define CIPA weak transitions as follows:

- $\Longrightarrow = \xrightarrow[t_1]{\tau@k_1} \cdots \xrightarrow[t_n]{\tau@k_n}$ for $n \in \mathbb{N}$, where $k_1 \leq \cdots \leq k_n$ due to the well timedness of computations.
- $\overset{a@k}{\underset{t}{\Longrightarrow}} = \Longrightarrow \xrightarrow[t]{a@k}$ for $a \in Act \setminus \{\tau\}$, where k is $\geq$ than the starting time of the last transition in $\Longrightarrow$.

- $\stackrel{\widehat{\alpha@k}}{\stackrel{\Longrightarrow}{t}} = \begin{cases} \Longrightarrow & \text{if } \alpha = \tau \\ \stackrel{\alpha@k}{\stackrel{\Longrightarrow}{t}} & \text{if } \alpha \neq \tau \end{cases}$.

Definition 2.6. A symmetric relation $\mathcal{B}$ over $\mathbb{KP}$ is a *weak timed bisimulation* iff, whenever $(K_1, K_2) \in \mathcal{B}$, then for all actions $\alpha \in Act$, starting times $k \in \mathbb{N}$, and durations $t \in \mathbb{N}_{>0}$ it holds that:

- For each $K_1 \stackrel{\alpha@k}{\stackrel{\longrightarrow}{t}} K_1'$ there exists $K_2 \stackrel{\widehat{\alpha@k}}{\stackrel{\Longrightarrow}{t}} K_2'$ such that $(K_1', K_2') \in \mathcal{B}$.

We write $K_1 \approx_{\mathrm{CIPA}} K_2$ to denote that (K_1, K_2) is contained in a weak timed bisimulation. ■

Notice that, for $\alpha = \tau$, process K_2 may simply idle or perform a sequence of τ-actions with an arbitrary duration because of the definition of $\Longrightarrow$, in which timing information is lost. As a consequence, in both cases, on K_2 side there may not be any weak τ-transition starting at time k and having duration t that matches the τ-transition on K_1 side. However, in order for the equivalence to hold, possible *subsequent visible* actions must start their execution at the same time in both processes.

Finally, we say that $Q_1, Q_2 \in \mathbb{P}_{\mathrm{CIPA}}$ are related by $\sim_{\mathrm{CIPA}}$ or $\approx_{\mathrm{CIPA}}$ iff so are $0 \Rightarrow Q_1, 0 \Rightarrow Q_2 \in \mathbb{KP}$.

2.4. Preliminaries for Calculi with Stochastic Time

Stochastically timed process calculi adopt a CSP-like, *multiway* synchronization mechanism [26], such that only identical actions can synchronize. In this setting, we denote by A a nonempty set of *visible actions* – ranged over by a, b – and we use $Act = A \cup \{\tau\}$ to indicate the set of all actions – ranged over by α, β – where τ is the *invisible action*. We then let *Rel* be a set of *action relabeling functions*; each such function $\varphi : Act \to Act$ satisfies $\varphi(\tau) = \tau$ as well as $\varphi(a) \in Act \setminus \{\tau\}$ for all $a \in Act \setminus \{\tau\}$.

Time is *continuous* and described by elements of $\mathbb{R}_{>0}$ – ranged over by λ, μ – each of which is called *rate* and uniquely identifies an *exponentially distributed random variable*. A variable V_λ of this kind is defined according to the probability distribution function $\Pr\{V_\lambda \leq t\} = 1 - \mathrm{e}^{-\lambda \cdot t}$ for all $t \in \mathbb{R}_{>0}$, which is the only continuous distribution enjoying the *memoryless property*, i.e., $\Pr\{V_\lambda \leq t + t' \mid V_\lambda > t'\} = \Pr\{V_\lambda \leq t\}$ for all $t, t' \in \mathbb{R}_{>0}$. The expected value of V_λ is $1/\lambda$, while its variance is $1/\lambda^2$.

Unlike the deterministic-time setting, where time does not resolve choices, here the *race policy* is adopted. This means that, whenever several activities are taking place, the one that terminates first is the one that samples the least completion time from the associated exponentially distributed random variable, with the distributions of the residual time to completion of the other activities being equal to the corresponding original distributions by virtue of the memoryless property. If $V_{\lambda_1}, \ldots, V_{\lambda_n}$ are the considered random variables, enforcing the race policy in the state in which those activities start their execution implies that:

- The *state sojourn time* is given by the exponentially distributed random variable $\min_{1 \leq i \leq n} V_{\lambda_i} = V_{\sum_{1 \leq i \leq n} \lambda_i}$, hence the *expected sojourn time* is $1/\sum_{1 \leq i \leq n} \lambda_i$.
- The *execution probability* of activity j, i.e., the probability that it terminates first, is $\lambda_j / \sum_{1 \leq i \leq n} \lambda_i$.

2.5. MTIPP: Exponentially Timed Actions

The process calculus MTIPP [18, 22] associates exponentially distributed durations with actions, so to support performance evaluation via the solution of continuous-time Markov chain models [37] and, at the same time, exploit the memoryless property of exponential distributions to fit well with the interleaving view of concurrency. As for deterministically timed calculi, we assume action execution to be *urgent* and we denote by *Var* a nonempty set of process variables – ranged over by X, Y – whose occurrences can be free or bound by the recursion constructor "rec".

$$\frac{}{<\alpha,\lambda>.\,G \stackrel{\alpha,\lambda}{\longrightarrow} G}$$

$$\frac{G_1 \stackrel{\alpha,\lambda}{\longrightarrow} G_1'}{G_1 + G_2 \stackrel{\alpha,\lambda}{\longrightarrow} G_1'} \qquad \frac{G_2 \stackrel{\alpha,\lambda}{\longrightarrow} G_2'}{G_1 + G_2 \stackrel{\alpha,\lambda}{\longrightarrow} G_2'}$$

$$\frac{G_1 \stackrel{\alpha,\lambda}{\longrightarrow} G_1' \quad \alpha \notin S}{G_1 \parallel_S G_2 \stackrel{\alpha,\lambda}{\longrightarrow} G_1' \parallel_S G_2} \qquad \frac{G_2 \stackrel{\alpha,\lambda}{\longrightarrow} G_2' \quad \alpha \notin S}{G_1 \parallel_S G_2 \stackrel{\alpha,\lambda}{\longrightarrow} G_1 \parallel_S G_2'}$$

$$\frac{G_1 \stackrel{\alpha,\lambda_1}{\longrightarrow} G_1' \quad G_2 \stackrel{\alpha,\lambda_2}{\longrightarrow} G_2' \quad \alpha \in S}{G_1 \parallel_S G_2 \stackrel{\alpha,\lambda_1 \cdot \lambda_2}{\longrightarrow} G_1' \parallel_S G_2'}$$

$$\frac{G \stackrel{\alpha,\lambda}{\longrightarrow} G' \quad \alpha \notin H}{G \,/\, H \stackrel{\alpha,\lambda}{\longrightarrow} G' \,/\, H} \qquad \frac{G \stackrel{\alpha,\lambda}{\longrightarrow} G' \quad \alpha \in H}{G \,/\, H \stackrel{\tau,\lambda}{\longrightarrow} G' \,/\, H}$$

$$\frac{G \stackrel{\alpha,\lambda}{\longrightarrow} G'}{G\,[\varphi] \stackrel{\varphi(\alpha),\lambda}{\longrightarrow} G'\,[\varphi]}$$

$$\frac{G\{\mathrm{rec}\, X : G \hookrightarrow X\} \stackrel{\alpha,\lambda}{\longrightarrow} G'}{\mathrm{rec}\, X : G \stackrel{\alpha,\lambda}{\longrightarrow} G'}$$

Table 3: Structural operational semantic rules for MTIPP

Definition 2.7. The set $\mathcal{P}_{\mathrm{MTIPP}}$ of MTIPP process terms is generated by the following syntax:

G	::=	$\underline{0}$	inactive process
	|	$<\alpha,\lambda>.\,G$	exponentially timed action prefix
	|	$G + G$	alternative composition
	|	$G \parallel_S G$	CSP-like parallel composition
	|	$G \,/\, H$	hiding
	|	$G\,[\varphi]$	relabeling
	|	X	process variable
	|	$\mathrm{rec}\, X : G$	recursion

where $\alpha \in Act$, $\lambda \in \mathbb{R}_{>0}$, $S, H \subseteq Act \setminus \{\tau\}$, $\varphi \in Rel$, and $X \in Var$. We denote by $\mathbb{P}_{\mathrm{MTIPP}}$ the set of closed and guarded process terms of $\mathcal{P}_{\mathrm{MTIPP}}$. ■

Process $\underline{0}$ cannot perform any action. Process $<\alpha,\lambda>.\,G$ can perform the eager action α at rate λ and then evolves into process G; note that different occurrences of the same action may have different rates.

Process $G_1 + G_2$ represents a rate-based *probabilistic* choice between G_1 and G_2 governed by the race policy, hence each action enabled in G_1 or G_2 has an execution probability proportional to its rate. Process $G_1 \parallel_S G_2$ describes the parallel composition of processes G_1 and G_2, where only occurrences of the same visible action belonging to the synchronization set S can synchronize, and generate another occurrence of that action whose rate is the product of the rates of the two original occurrences (see [24] for an overview of rate synchronization policies). Also parallel composition is subject to the race policy.

Process $G \,/\, H$ behaves as process G except for actions in the hiding set H, which become τ when they are executed; this is useful for preventing synchronizations between identical visible actions. Process $G\,[\varphi]$

behaves as process G, with the difference that every action is transformed via φ upon execution; this allows processes with different actions to communicate. Finally, $\mathrm{rec}\, X : G$ represents a recursive process, which behaves as process G in which every free occurrence of X is replaced by $\mathrm{rec}\, X : G$ itself; the resulting process will be denoted by $G\{\mathrm{rec}\, X : G \hookrightarrow X\}$.

Following [18, 22], with every $\mathbb{P}_{\mathrm{MTIPP}}$ process term is associated a labeled transition system defined according to Table 3, in which all the possible ways of deriving a transition have to be taken into account. The reason is that, e.g., process $<\alpha, \lambda> . G + <\alpha, \lambda> . G$ must have two outgoing α-transitions with rate λ – not just one as it would be in a deterministically timed setting – so that the corresponding expected sojourn time is $1/(2 \cdot \lambda)$. The interested reader is referred to [15] for a complete treatment of this issue.

Strong bisimilarity for MTIPP was defined in [22] by extending Larsen and Skou's strong bisimilarity for probabilistic processes [27] (weak bisimilarity was later defined in [4]). As a preliminary step, we introduce the *action exit rate* of a process $G \in \mathbb{P}_{\mathrm{MTIPP}}$ with respect to action $\alpha \in Act$ and destination process class $\mathcal{C} \subseteq \mathbb{P}_{\mathrm{MTIPP}}$ by letting:

$$rate_{\mathrm{a}}(G, \alpha, \mathcal{C}) \;=\; \sum\{\!|\, \lambda \in \mathbb{R}_{>0} \mid \exists G' \in \mathcal{C}.\, G \xrightarrow{\alpha,\lambda} G' \,|\!\}$$

where $\{\!|$ and $|\!\}$ are multiset delimiters and the summation is taken to be zero if the multiset is empty.

Definition 2.8. An equivalence relation $\mathcal{B}$ over $\mathbb{P}_{\mathrm{MTIPP}}$ is a *strong Markovian bisimulation* iff, whenever $(G_1, G_2) \in \mathcal{B}$, then for all actions $\alpha \in Act$ and equivalence classes $\mathcal{C} \in \mathbb{P}_{\mathrm{MTIPP}}/\mathcal{B}$ it holds that:

$$rate_{\mathrm{a}}(G_1, \alpha, \mathcal{C}) \;=\; rate_{\mathrm{a}}(G_2, \alpha, \mathcal{C})$$

We write $G_1 \sim_{\mathrm{MTIPP}} G_2$ to denote that (G_1, G_2) is contained in a strong Markovian bisimulation. ■

2.6. IML: Durationless Actions and Exponential Delays

The process calculus IML [21] features instantaneous actions separated from exponentially distributed delays, taking inspiration from previous calculi such as those of [29, 19] in which quantitative aspects are orthogonal to functional behavior. The aim of the author was that of supporting not only performance evaluation as in MTIPP, but also nondeterminism intended as implementation/scheduling freedom or lack of information. Like for MTIPP, we assume action execution to be *urgent* and we make use of *Var*.

Definition 2.9. The set $\mathcal{P}_{\mathrm{IML}}$ of IML process terms is generated by the following syntax:

F	::=	$\underline{0}$	inactive process
	|	$\alpha\,.\,F$	instantaneous action prefix
	|	$(\lambda)\,.\,F$	exponential delay prefix
	|	$F + F$	alternative composition
	|	$F \parallel_S F$	CSP-like parallel composition
	|	$F \,/\, H$	hiding
	|	$F\,[\varphi]$	relabeling
	|	X	process variable
	|	$\mathrm{rec}\, X : F$	recursion

where $\alpha \in Act$, $\lambda \in \mathbb{R}_{>0}$, $S, H \subseteq Act \setminus \{\tau\}$, $\varphi \in Rel$, and $X \in Var$. We denote by $\mathbb{P}_{\mathrm{IML}}$ the set of closed and guarded process terms of $\mathcal{P}_{\mathrm{IML}}$. ■

Process $\underline{0}$ cannot perform any action as in MTIPP. Process $\alpha\,.\,F$ can perform the instantaneous action α and then evolves into process F. Process $(\lambda)\,.\,F$ evolves into process F after a delay sampled from an exponentially distributed random variable of rate λ.

All the other operators work as in MTIPP, with two differences. In the case of alternative and parallel composition, the choice among actions is nondeterministic, while the choice among exponential delays is governed by the race policy. No synchronization is possible for exponential delays, hence they can only interleave; this enables the description of the time to the beginning of the execution of an action as the maximum of several exponentially distributed random variables (which is not exponentially distributed, as opposed to the minimum of the same variables).

$\dfrac{}{\alpha \,.\, F \xrightarrow{\alpha} F}$	$\dfrac{}{(\lambda) \,.\, F \overset{\lambda}{\rightsquigarrow} F}$
$\dfrac{F_1 \xrightarrow{\alpha} F_1'}{F_1 + F_2 \xrightarrow{\alpha} F_1'} \quad \dfrac{F_2 \xrightarrow{\alpha} F_2'}{F_1 + F_2 \xrightarrow{\alpha} F_2'}$	$\dfrac{F_1 \overset{\lambda}{\rightsquigarrow} F_1'}{F_1 + F_2 \overset{\lambda}{\rightsquigarrow} F_1'} \quad \dfrac{F_2 \overset{\lambda}{\rightsquigarrow} F_2'}{F_1 + F_2 \overset{\lambda}{\rightsquigarrow} F_2'}$
$\dfrac{F_1 \xrightarrow{\alpha} F_1' \quad \alpha \notin S}{F_1 \parallel_S F_2 \xrightarrow{\alpha} F_1' \parallel_S F_2} \quad \dfrac{F_2 \xrightarrow{\alpha} F_2' \quad \alpha \notin S}{F_1 \parallel_S F_2 \xrightarrow{\alpha} F_1 \parallel_S F_2'}$	$\dfrac{F_1 \overset{\lambda}{\rightsquigarrow} F_1'}{F_1 \parallel_S F_2 \overset{\lambda}{\rightsquigarrow} F_1' \parallel_S F_2} \quad \dfrac{F_2 \overset{\lambda}{\rightsquigarrow} F_2'}{F_1 \parallel_S F_2 \overset{\lambda}{\rightsquigarrow} F_1 \parallel_S F_2'}$
$\dfrac{F_1 \xrightarrow{\alpha} F_1' \quad F_2 \xrightarrow{\alpha} F_2' \quad \alpha \in S}{F_1 \parallel_S F_2 \xrightarrow{\alpha} F_1' \parallel_S F_2'}$	
$\dfrac{F \xrightarrow{\alpha} F' \quad \alpha \notin H}{F \,/\, H \xrightarrow{\alpha} F' \,/\, H} \quad \dfrac{F \xrightarrow{\alpha} F' \quad \alpha \in H}{F \,/\, H \xrightarrow{\tau} F' \,/\, H}$	$\dfrac{F \overset{\lambda}{\rightsquigarrow} F'}{F \,/\, H \overset{\lambda}{\rightsquigarrow} F' \,/\, L}$
$\dfrac{F \xrightarrow{\alpha} F'}{F\,[\varphi] \xrightarrow{\varphi(\alpha)} F'\,[\varphi]}$	$\dfrac{F \overset{\lambda}{\rightsquigarrow} F'}{F\,[\varphi] \overset{\lambda}{\rightsquigarrow} F'\,[\varphi]}$
$\dfrac{F\{\mathrm{rec}\, X : F \hookrightarrow X\} \xrightarrow{\alpha} F'}{\mathrm{rec}\, X : F \xrightarrow{\alpha} F'}$	$\dfrac{F\{\mathrm{rec}\, X : F \hookrightarrow X\} \overset{\lambda}{\rightsquigarrow} F'}{\mathrm{rec}\, X : F \overset{\lambda}{\rightsquigarrow} F'}$

Table 4: Structural operational semantic rules for IML

Following [21], with every $\mathbb{P}_{\mathrm{IML}}$ process term is associated a labeled transition system defined according to Table 4. The *action transition* relation $\longrightarrow$ on the left represents the functional behavior. The *delay transition* relation $\rightsquigarrow$ on the right represents the timing behavior according to the race policy; as in MTIPP, all the possible ways of deriving a delay transition have to be taken into account. The presence of two distinct transition relations emphasizes that, in IML, action execution is orthogonal to time passing.

Strong (resp. weak) bisimilarity for IML was defined in [21] by combining Milner's strong (resp. weak) bisimilarity for purely nondeterministic processes [28] and strong bisimilarity for MTIPP [22]. Therefore, we introduce the *exit rate* of a process $F \in \mathbb{P}_{\mathrm{IML}}$ with respect to destination process class $\mathcal{C} \subseteq \mathbb{P}_{\mathrm{MTIPP}}$ by considering only delay transitions as follows:

$$rate(F, \mathcal{C}) \;=\; \sum\{\!|\, \lambda \in \mathbb{R}_{>0} \mid \exists F' \in \mathcal{C}.\, F \overset{\lambda}{\rightsquigarrow} F' \,|\!\}$$

To enforce action eagerness, the exit rate equality check is carried out only for processes in which no action transition can be performed.

Definition 2.10. An equivalence relation $\mathcal{B}$ over $\mathbb{P}_{\mathrm{IML}}$ is a *strong Markovian bisimulation* iff, whenever $(F_1, F_2) \in \mathcal{B}$, then for all actions $\alpha \in Act$ and equivalence classes $\mathcal{C} \in \mathbb{P}_{\mathrm{MTIPP}}/\mathcal{B}$ it holds that:

- For each $F_1 \xrightarrow{\alpha} F_1'$ there exists $F_2 \xrightarrow{\alpha} F_2'$ such that $(F_1', F_2') \in \mathcal{B}$.
- If F_1 and F_2 have no action transitions, then:
$$rate(F_1, \mathcal{C}) \;=\; rate(F_2, \mathcal{C})$$

We write $F_1 \sim_{\mathrm{IML}} F_2$ to denote that (F_1, F_2) is contained in a strong Markovian bisimulation. ■

3. Direct Encoding from CIPA to TCCS

The first encoding that we examine for timed process calculi is the one defined in [12] in a deterministically timed setting under action eagerness. This encoding translates CIPA processes into TCCS processes by proceeding by induction on the syntactical structure of CIPA processes.

Definition 3.1. The direct encoding $[\![\_]\!] : \mathcal{P}_{\mathrm{CIPA}} \to \mathcal{P}_{\mathrm{TCCS}}$ under action eagerness is defined as follows:

$$\begin{array}{rcl}
[\![\mathrm{nil}]\!] & = & \mathrm{rec}\, X : (1) \,.\, X \\
[\![a \,.\, Q]\!] & = & a \,.\, (\Delta(a)) \,.\, [\![Q]\!] \\
[\![\mathrm{wait}\, t \,.\, Q]\!] & = & \tau \,.\, (t) \,.\, [\![Q]\!] \\
[\![Q_1 + Q_2]\!] & = & [\![Q_1]\!] + [\![Q_2]\!] \\
[\![Q_1 \mid Q_2]\!] & = & [\![Q_1]\!] \mid [\![Q_2]\!] \\
[\![Q \setminus L]\!] & = & [\![Q]\!] \setminus L \\
[\![Q\,[\varphi]]\!] & = & [\![Q]\!]\,[\varphi] \\
[\![X]\!] & = & X \\
[\![\mathrm{rec}\, X : Q]\!] & = & \mathrm{rec}\, X : [\![Q]\!]
\end{array}$$

■

We now comment on the first three clauses, as the others are obvious. Process nil cannot be mapped to $\mathbf{0}$, because the former can let any amount of time pass, while the latter cannot; time passing is encoded through one unit at a time in this discrete-time setting (otherwise process $\mathbf{0}$ should be replaced as in Sect. 4.1). A durational action a is translated into an identically named instantaneous action followed by a delay equal to the duration of the original action; the delay cannot precede the instantaneous action because what is observed in CIPA operational semantics is the starting time of actions, not their completion time as, e.g., in [17]. Likewise, a waiting is translated into an instantaneous τ-action followed by a delay equal to the duration of the waiting itself, although the presence of the τ-action is not necessary in principle.

Theorem 3.2. Let $Q_1, Q_2 \in \mathbb{P}_{\mathrm{CIPA}}$ be restriction free. Then:

$$Q_1 \sim_{\mathrm{CIPA}} Q_2 \iff [\![Q_1]\!] \sim_{\mathrm{TCCS}} [\![Q_2]\!]$$

Proof See [12]. ■

The full abstraction result with respect to strong timed bisimilarity does not hold for processes including occurrences of the restriction operator, thereby pinpointing the source of the different expressive power of CIPA and TCCS under action eagerness. For example, given the two processes $Q_1 = (a \,.\, \mathrm{nil}) \setminus \{a\} \mid (b \,.\, c \,.\, \mathrm{nil})$ and $Q_2 = (a \,.\, \mathrm{nil}) \setminus \{a\} \mid (b \,.\, \mathrm{nil})$, we have that $Q_1 \not\sim_{\mathrm{CIPA}} Q_2$ because Q_1 can perform a durational b-action followed by a durational c-action while Q_2 can only perform a durational b-action. On the other hand, $[\![Q_1]\!] \sim_{\mathrm{TCCS}} [\![Q_2]\!]$ because both of them can only perform an instantaneous b-action after which the passage of $\Delta(b)$ time units is blocked by the deadlocked common subprocess $[\![(a \,.\, \mathrm{nil}) \setminus \{a\}]\!]$. The same problem would arise if the restriction on a were applied to the entire processes instead of their common subprocess enabling an a-action.

4. Reverse Encoding from TCCS to CIPA

A number of issues were raised in [12] about the existence of a reverse encoding from TCCS to CIPA. In this section, we recall those issues, discuss how to address them, define a reverse encoding under action eagerness, and show that it can only preserve *weak* timed bisimilarity due to the different way in which time additivity is supported in TCCS and CIPA.

4.1. Adapting Syntax and Semantics of TCCS and CIPA

In order for a translation of TCCS processes into CIPA processes to be possible, it is necessary to slightly revise syntax and semantics of both languages.

The first issue that we discuss is *timelock*. In a TCCS process, time does not resolve choices; indeed, the operational rules for alternative and parallel composition allow time to pass only if all the subprocesses do so.

As a consequence, a local timelock always implies a global timelock, which may in turn determine a deadlock. By contrast, in CIPA timelock cannot occur unless there is a deadlock, because time passing is associated with action execution and explicit waiting. Consider the TCCS process $\mathbf{0} + (t) \,.\, \mathbf{0}$ and the ideally corresponding CIPA process $\mathrm{nil} + \mathrm{wait}\, t \,.\, \mathrm{nil}$; the former process cannot let time pass, while the latter process can. The same would happen with $(a \,.\, \mathbf{0}) \setminus \{a\} + (t) \,.\, \mathbf{0}$ and $(a \,.\, \mathrm{nil}) \setminus \{a\} + \mathrm{wait}\, t \,.\, \mathrm{nil}$.

To avoid timelocks due to the stopped process $\mathbf{0}$, we replace it with the inactive process $\underline{\mathbf{0}}$ introduced in [30], which lets time pass according to the additional operational semantic rule $\underline{\mathbf{0}} \stackrel{t}{\longrightarrow} \underline{\mathbf{0}}$ where $t \in \mathbb{N}_{>0}$. We denote by $\mathcal{P}'_{\mathrm{TCCS}}$ the resulting set of process terms and by $\mathbb{P}'_{\mathrm{TCCS}}$ the subset of closed and guarded process terms. To avoid timelocks caused by restriction, we have to limit our attention to restriction-free processes as in Thm. 3.2. This is different from resorting, for both calculi, to a two-level syntax featuring restriction only at the top level, as we did in [6], which unfortunately does not eliminate timelocks completely.

The second issue is related to the range of values that can be used in CIPA to express action durations. In TCCS, it is possible to describe both timed processes and *untimed* ones. Consider for example the untimed TCCS process $a \,.\, b \,.\, \underline{\mathbf{0}}$. This cannot be translated into a reasonably corresponding CIPA process for the very simple reason that actions a and b are instantaneous, but CIPA does not allow zero durations. Moreover, due to instantaneous actions, TCCS processes may exhibit *Zeno behaviors*, which are not possible in CIPA. For instance, the timed TCCS process $(t_1) \,.\, a \,.\, \mathrm{rec}\, X : (b \,.\, X + c \,.\, (t_2) \,.\, \underline{\mathbf{0}})$ may perform, after time t_1 and action a, an arbitrary, even infinite, number of b-actions at the same time. These problems can be straightforwardly solved by admitting in CIPA *zero durations* through an extended duration function $\Delta' : (Act \setminus \{\tau\}) \to \mathbb{N}$, as well as *zero waitings* for dealing with instantaneous τ-actions. We denote by $\mathcal{P}'_{\mathrm{CIPA}}$ the resulting set of process terms and by $\mathcal{KP}'$ the related set of augmented process terms, with $\mathbb{P}'_{\mathrm{CIPA}}$ and $\mathbb{KP}'$ being the corresponding subsets of closed and guarded process terms.

4.2. From TCCS Delays to CIPA Durations

One of the major design decisions about the translation of modified TCCS into modified CIPA is how to assign durations to actions – directly via Δ' for visible actions, indirectly through waitings for τ-actions. In principle, it is desirable to be able to associate a suitable *nonzero duration* with every action occurring in a TCCS process that is *not untimed*. Unfortunately, in most cases this is not possible or may compromise full abstraction, as we now show.

Consider the TCCS process $a \,.\, (t_1) \,.\, b \,.\, (t_2) \,.\, \underline{\mathbf{0}}$. In this case, it is natural to interpret delay t_1 as the duration of a and delay t_2 as the duration of b, thus regarding the occurrence of an instantaneous action of TCCS as the *beginning* of the corresponding durational action of CIPA (*initial view*). In the TCCS process $(t_1) \,.\, a \,.\, (t_2) \,.\, b \,.\, \underline{\mathbf{0}}$, the durations are as before, provided that the occurrence of an instantaneous action is regarded as the *end* of the corresponding durational action (*final view*). Notice that, if $a = b$ but $t_1 \neq t_2$, the translation into a reasonably corresponding CIPA process would not be possible, unless, as noted in [12], we further extend the duration function for CIPA by admitting that different occurrences of the same action may have different durations.

Let us now examine the case in which there is not a precise pairing between actions and delays, like, e.g., in the TCCS process $(t_1) \,.\, a \,.\, (t_2) \,.\, \underline{\mathbf{0}}$. In this scenario, the duration of a can be either t_1 or t_2, but in any case a waiting is necessary to account for the delay that is not associated with a. The situation is even more complicated if we consider the TCCS process $a \,.\, (t_1) \,.\, (t_2) \,.\, b \,.\, \underline{\mathbf{0}}$. One option is to interpret $t_1 + t_2$ as the duration of a (initial view), with b having duration 0. The dual option is to interpret $t_1 + t_2$ as the duration of b (final view), with a having duration 0. In any case, the definition of the encoding would become technically involved, especially in the presence of recursion, due to the necessity of performing some lookahead. Moreover, there seems not to be any strong reason for choosing one option rather than the other.

Yet another option is to interpret t_1 as the duration of a (initial view) and t_2 as the duration of b (final view). This *mixed* option should be discarded because it would disrupt full abstraction of the encoding. Indeed, the considered TCCS process is equivalent to the TCCS process $a \,.\, (t_2) \,.\, (t_1) \,.\, b \,.\, \underline{\mathbf{0}}$ while, under the assumption $t_1 \neq t_2$, the two corresponding CIPA processes are not equivalent to each other, because the a-action of duration t_1 in the first CIPA process cannot be matched by the a-action of duration t_2 of the second CIPA process.

Summing up, on the one hand there are TCCS delays that cannot be associated with any visible action, and hence have to be translated into CIPA waitings. On the other hand, it is not always possible to assign a nonzero duration to every TCCS action. In particular, this is impossible in the case of untimed TCCS processes. In this respect, it is also worth reminding that the two untimed TCCS processes $a\,.\,\underline{\mathbf{0}} \mid b\,.\,\underline{\mathbf{0}}$ and $a\,.\,b\,.\,\underline{\mathbf{0}} + b\,.\,a\,.\,\underline{\mathbf{0}}$ are always equivalent under the interleaving view of concurrency, while the two ideally corresponding CIPA processes $a\,.\,\mathrm{nil} \mid b\,.\,\mathrm{nil}$ and $a\,.\,b\,.\,\mathrm{nil} + b\,.\,a\,.\,\mathrm{nil}$ are equivalent to each other only if actions a and b have duration zero, because in that case all visible transitions of both processes start at time 0.

As a consequence, for the sake of simplicity, uniformity, and full abstraction, when translating modified TCCS processes into modified CIPA processes we proceed as follows:

- Every TCCS instantaneous action a will be translated into a CIPA durational action a with $\Delta'(a) = 0$.
- The TCCS instantaneous action τ will be translated into a CIPA waiting of duration 0.
- Every TCCS delay t will be translated into a CIPA waiting of duration t, where $t \in \mathbb{N}_{>0}$.

4.3. Reverse Encoding and Weak Timed Bisimilarity

The encoding of modified TCCS processes into modified CIPA processes under action eagerness proceeds by induction on the syntactical structure of the former processes.

Definition 4.1. The reverse encoding $[\![\_]\!]^{\mathrm{r}} : \mathcal{P}'_{\mathrm{TCCS}} \to \mathcal{P}'_{\mathrm{CIPA}}$ under action eagerness is defined as follows:

$$\begin{array}{rcl}
[\![\underline{\mathbf{0}}]\!]^{\mathrm{r}} & = & \mathrm{nil} \\
[\![a\,.\,P]\!]^{\mathrm{r}} & = & a\,.\,[\![P]\!]^{\mathrm{r}} \\
[\![\tau\,.\,P]\!]^{\mathrm{r}} & = & \mathrm{wait}\,0\,.\,[\![P]\!]^{\mathrm{r}} \\
[\![(t)\,.\,P]\!]^{\mathrm{r}} & = & \mathrm{wait}\,t\,.\,[\![P]\!]^{\mathrm{r}} \\
[\![P_1 + P_2]\!]^{\mathrm{r}} & = & [\![P_1]\!]^{\mathrm{r}} + [\![P_2]\!]^{\mathrm{r}} \\
[\![P_1 \mid P_2]\!]^{\mathrm{r}} & = & [\![P_1]\!]^{\mathrm{r}} \mid [\![P_2]\!]^{\mathrm{r}} \\
[\![P \setminus L]\!]^{\mathrm{r}} & = & [\![P]\!]^{\mathrm{r}} \setminus L \\
[\![P\,[\varphi]]\!]^{\mathrm{r}} & = & [\![P]\!]^{\mathrm{r}}\,[\varphi] \\
[\![X]\!]^{\mathrm{r}} & = & X \\
[\![\mathrm{rec}\,X : P]\!]^{\mathrm{r}} & = & \mathrm{rec}\,X : [\![P]\!]^{\mathrm{r}}
\end{array}$$

with $\Delta'(a) = 0$ for all $a \in Act \setminus \{\tau\}$. We denote by $\mathcal{K}[\![P]\!]^{\mathrm{r}} \in \mathcal{KP}'$ the reverse encoding of $P \in \mathcal{P}'_{\mathrm{TCCS}}$ obtained by distributing clock "$0 \Rightarrow$" over all subprocesses of $[\![P]\!]^{\mathrm{r}} \in \mathcal{P}'_{\mathrm{CIPA}}$. ■

Process $\underline{\mathbf{0}}$ can be translated into process nil because neither can prevent time from advancing. The subsequent three clauses stem from the three considerations at the end of Sect. 4.2. The remaining clauses are obvious.

Unlike the direct encoding of Def. 3.1, the reverse encoding of Def. 4.1 does *not* preserve *strong* timed bisimilarity. Consider the two modified TCCS processes:

$$\begin{array}{rcl}
\bar{P}_1 & = & (1)\,.\,(1)\,.\,a\,.\,\underline{\mathbf{0}} \\
\bar{P}_2 & = & (2)\,.\,a\,.\,\underline{\mathbf{0}}
\end{array}$$

which satisfy $\bar{P}_1 \sim_{\mathrm{TCCS}} \bar{P}_2$. Their corresponding augmented modified CIPA processes are the following:

$$\begin{array}{rcl}
\mathcal{K}[\![\bar{P}_1]\!]^{\mathrm{r}} & = & 0 \Rightarrow (\mathrm{wait}\,1\,.\,\mathrm{wait}\,1\,.\,a\,.\,\mathrm{nil}) \\
\mathcal{K}[\![\bar{P}_2]\!]^{\mathrm{r}} & = & 0 \Rightarrow (\mathrm{wait}\,2\,.\,a\,.\,\mathrm{nil})
\end{array}$$

but $\mathcal{K}[\![\bar{P}_1]\!]^{\mathrm{r}} \not\sim_{\mathrm{CIPA}} \mathcal{K}[\![\bar{P}_2]\!]^{\mathrm{r}}$. The translation of delay prefixes in Def. 4.1 could be modified in such a way that every delay becomes a sequence of unitary waitings, so that also $\bar{P}_2$ would be encoded into the first CIPA process. However, this would loosen the correspondence between the transitions of the original processes and those of their encoded versions, and would be appropriate only in a discrete-time setting like ours.

This example emphasizes an important difference between TCCS and CIPA, which does not affect the direct encoding of Def. 3.1, because such an encoding cannot produce TCCS processes including consecutive delay prefixes (like $\bar{P}_1$ above) due to the presence of a τ-action in its third clause and hence the strict alternation of action prefixes and delay prefixes (apart from the encoding of nil). The reason why $\bar{P}_1 \sim_{\mathrm{TCCS}} \bar{P}_2$ holds is that in TCCS action execution is separated from time passing and the latter is subject to time

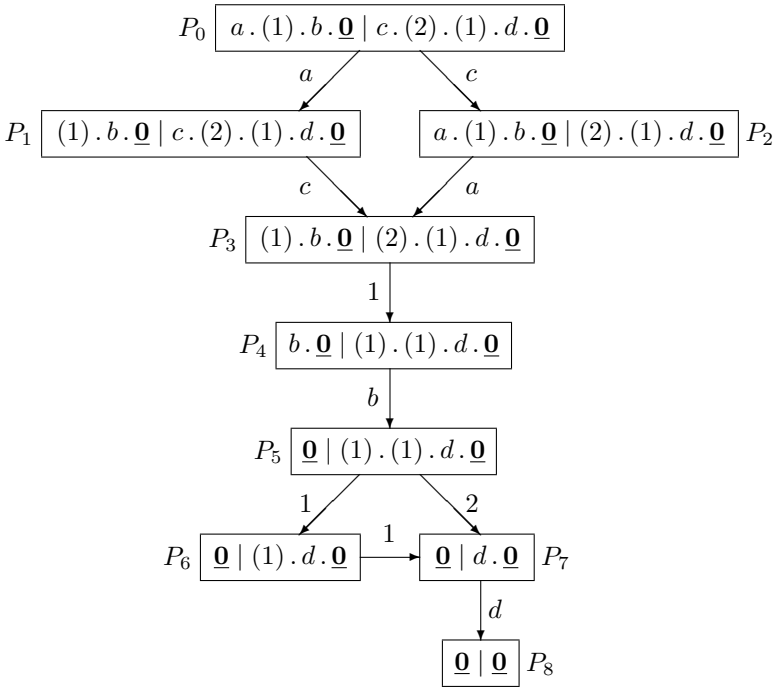

Figure 1: Labeled transition system for the modified TCCS process P_0

additivity, which is formalized through the operational semantic rules of TCCS. Thus, *both* $\bar{P}_1$ and $\bar{P}_2$ have a sequence of two delay transitions each labeled with 1 as well as a single delay transition labeled with 2, hence the strong bisimulation semantics for TCCS identifies those processes. On the other hand, $\mathcal{K}[\![\bar{P}_1]\!]^{\mathrm{r}}$ can only perform a sequence of two τ-transitions each having duration 1, while $\mathcal{K}[\![\bar{P}_2]\!]^{\mathrm{r}}$ can only perform a single τ-transition having duration 2, from which $\mathcal{K}[\![\bar{P}_1]\!]^{\mathrm{r}} \not\approx_{\mathrm{CIPA}} \mathcal{K}[\![\bar{P}_2]\!]^{\mathrm{r}}$ follows. It is however worth noting that $\mathcal{K}[\![\bar{P}_1]\!]^{\mathrm{r}} \approx_{\mathrm{CIPA}} \mathcal{K}[\![\bar{P}_2]\!]^{\mathrm{r}}$, because the weak bisimulation semantics for CIPA implements a form of *time additivity for waitings*. Therefore, we have to focus our attention on *weak* timed bisimilarities when investigating full abstraction for the reverse encoding.

4.4. Full Abstraction Result

The states of the labeled transition systems underlying a process $P \in \mathbb{P}'_{\mathrm{TCCS}}$ and the corresponding process $\mathcal{K}[\![P]\!]^{\mathrm{r}} \in \mathbb{KP}'$ are strictly related. Consider as an example the process $P_0 = a\,.\,(1)\,.\,b\,.\,\underline{\mathbf{0}} \mid c\,.\,(2)\,.\,(1)\,.\,d\,.\,\underline{\mathbf{0}}$. Its labeled transition system is depicted in Fig. 1, while the one for $\mathcal{K}[\![P_0]\!]^{\mathrm{r}}$ is shown in Fig. 2. It is easy to see that, as far as only visible actions and zero-valued local clocks are concerned, we have a one-to-one correspondence $\mathcal{K}[\![P_i]\!]^{\mathrm{r}} = K_i$ for $i = 0, 1, 2, 3$. However, the leading waitings in K_1 and K_2 produce τ-transitions that can be executed independently by one of the sequential subprocesses, thereby moving its own clock forward in time. In the meanwhile, the other subprocess still has actions to execute before that time. For instance, we have that $K_1 \xrightarrow[1]{\tau@0} K'_1$, but K'_1 can still perform the visible action c at time 0, and this must be performed before the visible action b at time 1 as ill-timed paths are not admitted.

On the other hand, the CIPA counterpart K_4 of the TCCS process P_4 cannot perform any τ-action, but $\mathcal{K}[\![P_4]\!]^{\mathrm{r}}$ has clock values not corresponding to those of K_4. Nevertheless, the b-action is performed at the same time, i.e., with timestamp 1, in both cases. Similarly, states P_5, P_6, and P_7, which are connected only by delay transitions, respectively result in $\mathcal{K}[\![P_5]\!]^{\mathrm{r}}$, $\mathcal{K}[\![P_6]\!]^{\mathrm{r}}$, and $\mathcal{K}[\![P_7]\!]^{\mathrm{r}}$, with the first one having local clock values different from those of K_5 and the other two having local clock values different from those of $K_{6,7}$. Focussing on P_5, we may observe that $\mathcal{K}[\![P_5]\!]^{\mathrm{r}} = (0 \Rightarrow \mathrm{nil}) \mid (0 \Rightarrow \mathrm{wait}\,1\,.\,\mathrm{wait}\,1\,.\,d\,.\,\mathrm{nil})$ is $\approx_{\mathrm{CIPA}}$-equivalent to $(0 \Rightarrow \mathrm{nil}) \mid (1 \Rightarrow \mathrm{wait}\,1\,.\,d\,.\,\mathrm{nil})$, which can be obtained from K_5 by decreasing by 1

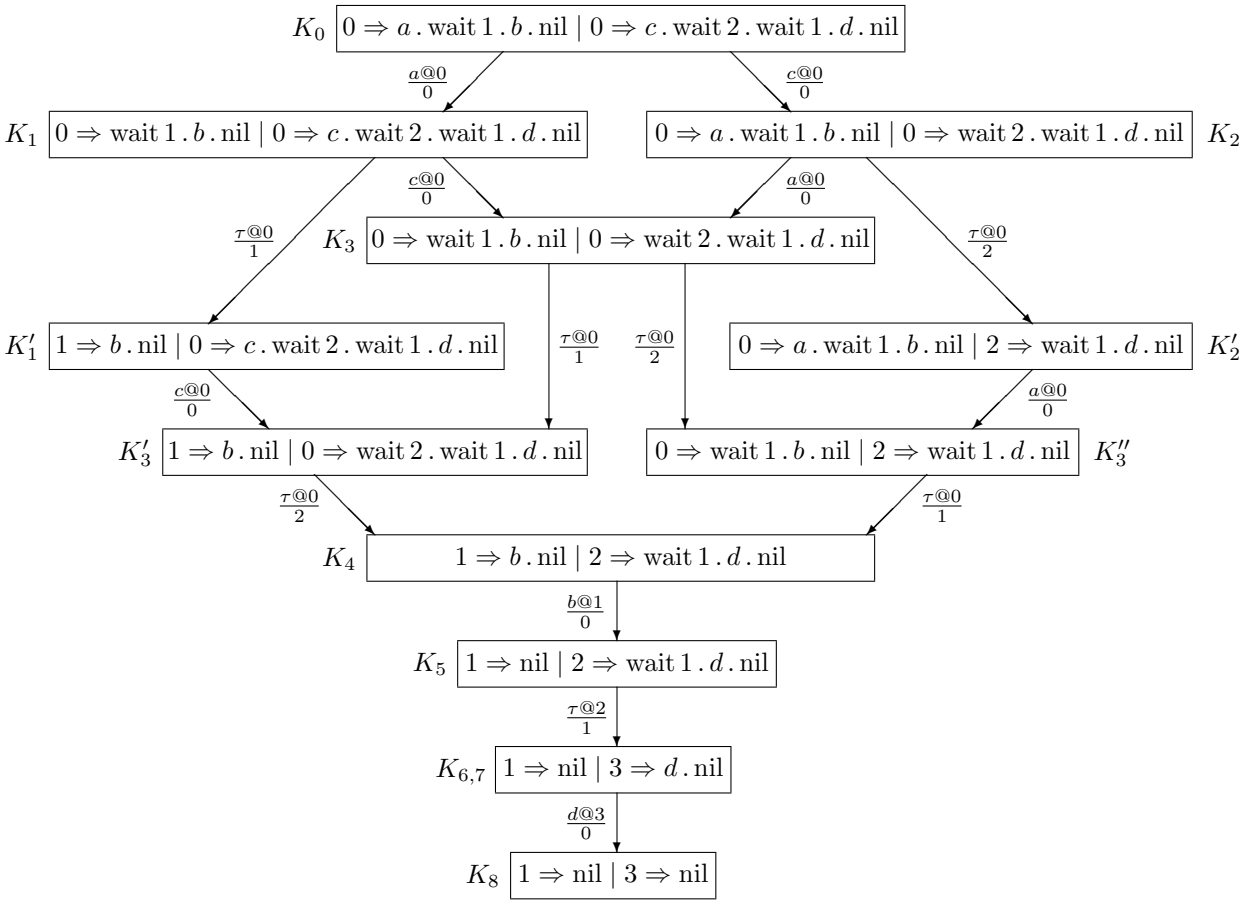

Figure 2: Labeled transition system for the augmented modified CIPA process $\mathcal{K}[\![P_0]\!]^{\mathrm{r}}$

all of its local clock values. Note that pairs of corresponding local clocks are different in absolute value, but agree on the relative differences: $1 - 0 = 2 - 1 = 1$. In general, this time shift operation can neglect the value of local clocks associated with nil subprocesses, because nil lets any amount of time pass.

To manage the clock discrepancies mentioned above, first of all we introduce a structural congruence $\equiv$ over $\mathbb{KP}'$ that permits to identify processes based on the following equalities for sequential subprocesses:

$$\begin{array}{rcl} k \Rightarrow \text{nil} & \equiv & k' \Rightarrow \text{nil} \\ k \Rightarrow (\text{wait}\,t\,.\,Q) & \equiv & (k+t) \Rightarrow Q \\ k \Rightarrow (\text{rec}\,X : Q) & \equiv & k \Rightarrow (Q\{\text{rec}\,X : Q \hookrightarrow X\}) \end{array}$$

where $k, k' \in \mathbb{N}$ and $t \in \mathbb{N}_{>0}$. Structural congruence is strictly related to weak bisimilarity.

Proposition 4.2. Let $K_1, K_2 \in \mathbb{KP}'$. If $K_1 \equiv K_2$, then $K_1 \approx_{\text{CIPA}} K_2$.

Proof It follows from the fact that the two processes in each of the equalities for $\equiv$ are $\approx_{\text{CIPA}}$-equivalent to each other and from the congruence property of $\approx_{\text{CIPA}}$, which can be proved as in [1]. ∎

Following [12], we then introduce a predicate $wf \subseteq \mathbb{KP}' \times \mathbb{N}$ and an update function $up\colon \mathbb{KP}' \times \mathbb{N} \to \mathbb{KP}'$. We let $wf(K, k)$ hold iff the local clocks of K not associated with nil subprocesses can be decreased by k without any of them becoming negative:

$$\frac{}{wf(k' \Rightarrow \mathrm{nil}, k)} \qquad \frac{k \leq k'}{wf(k' \Rightarrow (a\,.\,Q), k)} \qquad \frac{k \leq k'}{wf(k' \Rightarrow (\mathrm{wait}\,t\,.\,Q), k)} \qquad \frac{k \leq k'}{wf(k' \Rightarrow (\mathrm{rec}\,X : Q), k)}$$

$$\frac{wf(K_1, k) \quad wf(K_2, k)}{wf(K_1 + K_2, k)} \qquad \frac{wf(K_1, k) \quad wf(K_2, k)}{wf(K_1 \mid K_2, k)} \qquad \frac{wf(K, k)}{wf(K \setminus L, k)} \qquad \frac{wf(K, k)}{wf(K\,[\varphi], k)}$$

If $wf(K, k)$, then $up(K, k)$ is the process obtained from K by decreasing by k all of its local clocks:

$$\begin{array}{rcll}
up(k' \Rightarrow \mathrm{nil}, k) & = & (k' - k) \Rightarrow \mathrm{nil} & k \leq k' \\
up(k' \Rightarrow \mathrm{nil}, k) & = & 0 \Rightarrow \mathrm{nil} & k > k' \\
up(k' \Rightarrow (a\,.\,Q), k) & = & (k' - k) \Rightarrow (a\,.\,Q) & k \leq k' \\
up(k' \Rightarrow (\mathrm{wait}\,t\,.\,Q), k) & = & (k' - k) \Rightarrow (\mathrm{wait}\,t\,.\,Q) & k \leq k' \\
up(k' \Rightarrow (\mathrm{rec}\,X : Q), k) & = & (k' - k) \Rightarrow (\mathrm{rec}\,X : Q) & k \leq k' \\
up(K_1 + K_2, k) & = & up(K_1, k) + up(K_2, k) & \\
up(K_1 \mid K_2, k) & = & up(K_1, k) \mid up(K_2, k) & \\
up(K \setminus L, k) & = & up(K, k) \setminus L & \\
up(K\,[\varphi], k) & = & up(K, k)\,[\varphi] &
\end{array}$$

Using $\equiv$ and up, the non-initial states in Figs. 1 and 2 turn out to be related as follows, where increasing time shifts k in $up(\cdot, k)$ correspond to subsequent delay transitions.

- $\mathcal{K}[\![P_1]\!]^{\mathrm{r}} \;=\; K_1 \;\equiv\; K_1' \qquad \mathcal{K}[\![P_2]\!]^{\mathrm{r}} \;=\; K_2 \;\equiv\; K_2' \qquad \mathcal{K}[\![P_3]\!]^{\mathrm{r}} \;=\; K_3 \;\equiv\; K_3' \;\equiv\; K_3'' \;\equiv\; K_4$
- $\mathcal{K}[\![P_4]\!]^{\mathrm{r}} \;=\; (0 \Rightarrow b\,.\,\mathrm{nil}) \mid (0 \Rightarrow \mathrm{wait}\,1\,.\,\mathrm{wait}\,1\,.\,d\,.\,\mathrm{nil}) \;\equiv\; (0 \Rightarrow b\,.\,\mathrm{nil}) \mid (1 \Rightarrow \mathrm{wait}\,1\,.\,d\,.\,\mathrm{nil}) \;=\; up(K_4, 1)$
- $\mathcal{K}[\![P_5]\!]^{\mathrm{r}} \;=\; (0 \Rightarrow \mathrm{nil}) \mid (0 \Rightarrow \mathrm{wait}\,1\,.\,\mathrm{wait}\,1\,.\,d\,.\,\mathrm{nil}) \;\equiv\; (0 \Rightarrow \mathrm{nil}) \mid (1 \Rightarrow \mathrm{wait}\,1\,.\,d\,.\,\mathrm{nil}) \;=\; up(K_5, 1)$
- $\mathcal{K}[\![P_6]\!]^{\mathrm{r}} \;=\; (0 \Rightarrow \mathrm{nil}) \mid (0 \Rightarrow \mathrm{wait}\,1\,.\,d\,.\,\mathrm{nil}) \;\equiv\; (0 \Rightarrow \mathrm{nil}) \mid (1 \Rightarrow d\,.\,\mathrm{nil}) \;=\; up(K_{6,7}, 2)$
- $\mathcal{K}[\![P_7]\!]^{\mathrm{r}} \;=\; (0 \Rightarrow \mathrm{nil}) \mid (0 \Rightarrow d\,.\,\mathrm{nil}) \;=\; up(K_{6,7}, 3)$
- $\mathcal{K}[\![P_8]\!]^{\mathrm{r}} \;=\; (0 \Rightarrow \mathrm{nil}) \mid (0 \Rightarrow \mathrm{nil}) \;=\; up(K_8, 3)$

We can now present the full abstraction result with respect to weak timed bisimilarity for the reverse encoding of Def. 4.1 under action eagerness. Due to the same reason exemplified after Thm. 3.2, the validity of the result is limited to processes in which there are no occurrences of the restriction operator. Moreover, it is worth reminding that in TCCS time does not resolve choices, while in CIPA time passing is associated with action execution and explicit waitings, hence the operational semantic rules for alternative and parallel composition are quite different in the two calculi. In particular, if we consider a TCCS process whose topmost operator is an alternative composition in which at least one subprocess operand starts with a delay prefix, then no correspondence can be established between the transitions of the process and those of its encoded version, which means that full abstraction does not hold in this case.

For example, processes $\breve{P}_1 = a\,.\,\underline{\mathbf{0}} + (1)\,.\,\underline{\mathbf{0}}$ and $\breve{P}_2 = a\,.\,\underline{\mathbf{0}}$ satisfy $\breve{P}_1 \approx_{\mathrm{TCCS}} \breve{P}_2$, while their encodings $\mathcal{K}[\![\breve{P}_1]\!]^{\mathrm{r}} = (0 \Rightarrow a\,.\,\mathrm{nil}) + (0 \Rightarrow \mathrm{wait}\,1\,.\,\mathrm{nil})$ and $\mathcal{K}[\![\breve{P}_2]\!]^{\mathrm{r}} = 0 \Rightarrow a\,.\,\mathrm{nil}$ are such that $\mathcal{K}[\![\breve{P}_1]\!]^{\mathrm{r}} \not\approx_{\mathrm{CIPA}} \mathcal{K}[\![\breve{P}_2]\!]^{\mathrm{r}}$, because the τ-transition at time 0 of duration 1 corresponding to the waiting in $\mathcal{K}[\![\breve{P}_1]\!]^{\mathrm{r}}$ leads to a state that is not $\approx_{\mathrm{CIPA}}$-equivalent to $\mathcal{K}[\![\breve{P}_2]\!]^{\mathrm{r}}$. As another example, in which both processes start with precisely the same delay, $\breve{P}_3 = (1)\,.\,a\,.\,\underline{\mathbf{0}} + (1)\,.\,b\,.\,\underline{\mathbf{0}}$ and $\breve{P}_4 = (1)\,.\,(a\,.\,\underline{\mathbf{0}} + b\,.\,\underline{\mathbf{0}})$ satisfy $\breve{P}_3 \approx_{\mathrm{TCCS}} \breve{P}_4$, while their encodings $\mathcal{K}[\![\breve{P}_3]\!]^{\mathrm{r}} = (0 \Rightarrow \mathrm{wait}\,1\,.\,a\,.\,\mathrm{nil}) + (0 \Rightarrow \mathrm{wait}\,1\,.\,b\,.\,\mathrm{nil})$ and $\mathcal{K}[\![\breve{P}_4]\!]^{\mathrm{r}} = 0 \Rightarrow \mathrm{wait}\,1\,.\,(a\,.\,\mathrm{nil} + b\,.\,\mathrm{nil})$ are such that $\mathcal{K}[\![\breve{P}_3]\!]^{\mathrm{r}} \not\approx_{\mathrm{CIPA}} \mathcal{K}[\![\breve{P}_4]\!]^{\mathrm{r}}$, because each of the two τ-transitions at time 0 of duration 1 corresponding to the two waitings in $\mathcal{K}[\![\breve{P}_3]\!]^{\mathrm{r}}$ leads to a state that is $\approx_{\mathrm{CIPA}}$-equivalent to neither $\mathcal{K}[\![\breve{P}_4]\!]^{\mathrm{r}}$ nor its τ-derivative $1 \Rightarrow (a\,.\,\mathrm{nil} + b\,.\,\mathrm{nil})$.

The problem would not arise if we had parallel composition in place of alternative composition, because in that case no subprocess operand would be left out and CIPA well timedness would prevent some actions from being executed too soon. For instance, consider processes $\breve{P}_5 = (1)\,.\,a\,.\,\underline{\mathbf{0}} \mid b\,.\,\underline{\mathbf{0}}$ and $\breve{P}_6 = b\,.\,(1)\,.\,a\,.\,\underline{\mathbf{0}}$, which satisfy $\breve{P}_5 \approx_{\mathrm{TCCS}} \breve{P}_6$. Their encodings $\mathcal{K}[\![\breve{P}_5]\!]^{\mathrm{r}} = (0 \Rightarrow \mathrm{wait}\,1\,.\,a\,.\,\mathrm{nil}) \mid (0 \Rightarrow b\,.\,\mathrm{nil})$ and $\mathcal{K}[\![\breve{P}_6]\!]^{\mathrm{r}} = 0 \Rightarrow b\,.\,\mathrm{wait}\,1\,.\,a\,.\,\mathrm{nil}$ are $\approx_{\mathrm{CIPA}}$-equivalent because, after the τ-transition at time 0 of duration 1 corresponding

to its only waiting, the former process cannot perform a at time 1 as this would be followed by b at time 0 thereby resulting in an ill-timed path.

In conclusion, the reverse encoding from modified TCCS to modified CIPA can be fully abstract, under action eagerness and with respect to weak timed bisimilarity, only for restriction-free processes that are delay-choice free. A TCCS process is *delay-choice free* iff, for each occurrence of the alternative composition operator in the process, neither subprocess operand can perform a delay transition.

Following [12], we prove some preliminary results aimed at formally establishing a correspondence between the transitions of a modified TCCS process and the transitions of (augmented modified CIPA processes structurally congruent to) the encoded version of that process. The first preliminary result elicits a property of the reverse encoding that has to do with recursion, which will be exploited in the proof of the subsequent preliminary results.

Lemma 4.3. Let $P, \hat{P} \in \mathcal{P}'_{\mathrm{TCCS}}$ and $X, Y \in \mathit{Var}$. Then:

$$\mathcal{K}[\![P\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} \;=\; 0 \Rightarrow [\![P]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}$$

Proof We proceed by induction on the syntactical structure of $P \in \mathcal{P}'_{\mathrm{TCCS}}$:

- If $P = \underline{\mathbf{0}}$ or $P \in \mathit{Var} \setminus \{Y\}$, then:
$$\mathcal{K}[\![P\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} \;=\; 0 \Rightarrow \mathrm{nil} \;=\; 0 \Rightarrow [\![P]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}$$

- If $P = Y$, then:
$$\mathcal{K}[\![P\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} \;=\; \mathcal{K}[\![\mathrm{rec}\, X : \hat{P}]\!]^{\mathrm{r}} \;=\; 0 \Rightarrow \mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \;=\; 0 \Rightarrow [\![P]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}$$

- Let $P = \alpha\,.\,P'$ and assume that $\mathcal{K}[\![P'\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} = 0 \Rightarrow [\![P']\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}$. Then:
$$\begin{aligned}
\mathcal{K}[\![P\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} &= \mathcal{K}[\![\alpha\,.\,(P'\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\})]\!]^{\mathrm{r}} \\
&= 0 \Rightarrow \alpha\,.\,[\![P'\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} \\
&= 0 \Rightarrow \alpha\,.\,([\![P']\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}) \\
&= 0 \Rightarrow (\alpha\,.\,[\![P']\!]^{\mathrm{r}})\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\} \\
&= 0 \Rightarrow [\![P]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}
\end{aligned}$$

- Let $P = P_1 + P_2$ and for $i \in \{1, 2\}$ assume that $\mathcal{K}[\![P_i\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} = 0 \Rightarrow [\![P_i]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}$. Then:
$$\begin{aligned}
\mathcal{K}[\![P\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} &= \mathcal{K}[\![P_1\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\} + P_2\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} \\
&= 0 \Rightarrow [\![P_1\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} + 0 \Rightarrow [\![P_2\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} \\
&= 0 \Rightarrow [\![P_1]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\} + 0 \Rightarrow [\![P_2]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\} \\
&= 0 \Rightarrow ([\![P_1]\!]^{\mathrm{r}} + [\![P_2]\!]^{\mathrm{r}})\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\} \\
&= 0 \Rightarrow [\![P]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}
\end{aligned}$$

- The case $P = P_1 \mid P_2$ is similar to the previous one.

- Let $P = P' \setminus L$ and assume that $\mathcal{K}[\![P'\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} = 0 \Rightarrow [\![P']\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}$. Then:
$$\begin{aligned}
\mathcal{K}[\![P\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} &= \mathcal{K}[\![P'\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\} \setminus L]\!]^{\mathrm{r}} \\
&= 0 \Rightarrow [\![P'\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} \setminus L \\
&= 0 \Rightarrow [\![P']\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\} \setminus L \\
&= 0 \Rightarrow ([\![P']\!]^{\mathrm{r}} \setminus L)\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\} \\
&= 0 \Rightarrow [\![P]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}
\end{aligned}$$

- The case $P = P'\,[\varphi]$ is similar to the previous one.

- Let $P = \mathrm{rec}\, X' : P'$ and assume that $\mathcal{K}[\![P'\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} = 0 \Rightarrow [\![P']\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}$:

  - If $X' = Y$, then:
$$\mathcal{K}[\![P\{\mathrm{rec}\, X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} \;=\; \mathcal{K}[\![P]\!]^{\mathrm{r}} \;=\; 0 \Rightarrow [\![P]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}$$

– If $X' \neq Y$, then:

$$\begin{aligned}\mathcal{K}[\![P\{\mathrm{rec}\,X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} &= \mathcal{K}[\![\mathrm{rec}\,X' : (P'\{\mathrm{rec}\,X : \hat{P} \hookrightarrow Y\})]\!]^{\mathrm{r}} \\ &= 0 \Rightarrow \mathrm{rec}\,X' : [\![P'\{\mathrm{rec}\,X : \hat{P} \hookrightarrow Y\}]\!]^{\mathrm{r}} \\ &= 0 \Rightarrow \mathrm{rec}\,X' : ([\![P']\!]^{\mathrm{r}}\{\mathrm{rec}\,X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}) \\ &= 0 \Rightarrow (\mathrm{rec}\,X' : [\![P']\!]^{\mathrm{r}})\{\mathrm{rec}\,X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\} \\ &= 0 \Rightarrow [\![P]\!]^{\mathrm{r}}\{\mathrm{rec}\,X : [\![\hat{P}]\!]^{\mathrm{r}} \hookrightarrow Y\}\end{aligned}$$

■

The second preliminary result formalizes the correspondence between the action transitions of a modified TCCS process and the transitions, labeled with the same actions and duration zero, of the CIPA encoding of that process (see, e.g., states P_0, P_1, P_2 in Fig. 1 and states K_0, K_1, K_2 in Fig. 2).

Lemma 4.4. Let $\alpha \in Act$ and let $P, P' \in \mathbb{P}'_{\mathrm{TCCS}}$ be restriction and delay-choice free. Then $P \xrightarrow{\alpha} P'$ iff $\mathcal{K}[\![P]\!]^{\mathrm{r}} \xrightarrow[0]{\alpha@0} \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

Proof Given $\alpha \in Act$ and $P, P' \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free, the proof is divided into two parts:

$\Longrightarrow$) Assuming that $P \xrightarrow{\alpha} P'$, we prove that $\mathcal{K}[\![P]\!]^{\mathrm{r}} \xrightarrow[0]{\alpha@0} \mathcal{K}[\![P']\!]^{\mathrm{r}}$ by proceeding by induction on the length of the derivation of the action transition $P \xrightarrow{\alpha} P'$, intended as the number $n \in \mathbb{N}_{\geq 1}$ of operational semantic rules on the left-hand side of Table 1 that have been applied in order to derive the considered transition:

– If $n = 1$, then $P = \alpha\,.\,P'$. Therefore $\mathcal{K}[\![P]\!]^{\mathrm{r}} = 0 \Rightarrow (\alpha\,.\,[\![P']\!]^{\mathrm{r}}) \xrightarrow[0]{\alpha@0} 0 \Rightarrow [\![P']\!]^{\mathrm{r}} = \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

– Let $n > 1$ and suppose that the result holds for every transition derivable from a process in $\mathbb{P}'_{\mathrm{TCCS}}$, which is restriction and delay-choice free, by applying less than n operational semantic rules on the left-hand side of Table 1. There are several cases based on the syntactical structure of P (which is restriction and delay-choice free):

∗ If $P = P_1 + P_2$, then the transition derives from the fact that $P_i \xrightarrow{\alpha} P'$ for some $i \in \{1, 2\}$. From the induction hypothesis, it follows that $\mathcal{K}[\![P_i]\!]^{\mathrm{r}} \xrightarrow[0]{\alpha@0} \mathcal{K}[\![P']\!]^{\mathrm{r}}$, thus $\mathcal{K}[\![P]\!]^{\mathrm{r}} = \mathcal{K}[\![P_1]\!]^{\mathrm{r}} + \mathcal{K}[\![P_2]\!]^{\mathrm{r}} \xrightarrow[0]{\alpha@0} \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

∗ If $P = P_1 \mid P_2$, there are two subcases:

· If the transition is not originated from a synchronization, then it derives from the fact that $P_i \xrightarrow{\alpha} P'_i$ for some $i \in \{1, 2\}$. Without loss of generality, we assume $i = 1$, so that $P' = P'_1 \mid P_2$. From the induction hypothesis, it follows that $\mathcal{K}[\![P_1]\!]^{\mathrm{r}} \xrightarrow[0]{\alpha@0} [\![P'_1]\!]^{\mathrm{r}}$, thus $\mathcal{K}[\![P]\!]^{\mathrm{r}} = \mathcal{K}[\![P_1]\!]^{\mathrm{r}} \mid \mathcal{K}[\![P_2]\!]^{\mathrm{r}} \xrightarrow[0]{\alpha@0} \mathcal{K}[\![P'_1]\!]^{\mathrm{r}} \mid \mathcal{K}[\![P_2]\!]^{\mathrm{r}} = \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

· If the transition is originated from a synchronization, then it derives from the fact that $P_1 \xrightarrow{a} P'_1$ and $P_2 \xrightarrow{\bar{a}} P'_2$ for some $a \in Act \setminus \{\tau\}$, with $\alpha = \tau$ and $P' = P'_1 \mid P'_2$. From the induction hypothesis, it follows that $\mathcal{K}[\![P_1]\!]^{\mathrm{r}} \xrightarrow[0]{a@0} [\![P'_1]\!]^{\mathrm{r}}$ and $\mathcal{K}[\![P_2]\!]^{\mathrm{r}} \xrightarrow[0]{\bar{a}@0} [\![P'_2]\!]^{\mathrm{r}}$, thus $\mathcal{K}[\![P]\!]^{\mathrm{r}} = \mathcal{K}[\![P_1]\!]^{\mathrm{r}} \mid \mathcal{K}[\![P_2]\!]^{\mathrm{r}} \xrightarrow[0]{\tau@0} \mathcal{K}[\![P'_1]\!]^{\mathrm{r}} \mid \mathcal{K}[\![P'_2]\!]^{\mathrm{r}} = \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

∗ If $P = \bar{P}\,[\varphi]$, then the transition derives from the fact that $\bar{P} \xrightarrow{\beta} \bar{P}'$ for some $\beta \in Act$ such that $\varphi(\beta) = \alpha$, with $P' = \bar{P}'\,[\varphi]$. From the induction hypothesis, it follows that $\mathcal{K}[\![\bar{P}]\!]^{\mathrm{r}} \xrightarrow[0]{\beta@0} \mathcal{K}[\![\bar{P}']\!]^{\mathrm{r}}$, thus $\mathcal{K}[\![P]\!]^{\mathrm{r}} = [\![\bar{P}]\!]^{\mathrm{r}}\,[\varphi] \xrightarrow[0]{\alpha@0} [\![\bar{P}']\!]^{\mathrm{r}}\,[\varphi] = \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

* If $P = \mathrm{rec}\, X : \bar{P}$, then the transition derives from the fact that $\bar{P}\{\mathrm{rec}\, X : \bar{P} \hookrightarrow X\} \xrightarrow{\alpha} P'$. From the induction hypothesis, it follows that $\mathcal{K}[\![\bar{P}\{\mathrm{rec}\, X : \bar{P} \hookrightarrow X\}]\!]^{\mathrm{r}} \xrightarrow[0]{\alpha@0} \mathcal{K}[\![P']\!]^{\mathrm{r}}$, thus $\mathcal{K}[\![P]\!]^{\mathrm{r}} = 0 \Rightarrow (\mathrm{rec}\, X : [\![\bar{P}]\!]^{\mathrm{r}}) \xrightarrow[0]{\alpha@0} \mathcal{K}[\![P']\!]^{\mathrm{r}}$ because the unfolding of $0 \Rightarrow (\mathrm{rec}\, X : [\![\bar{P}]\!]^{\mathrm{r}})$, i.e., $0 \Rightarrow ([\![\bar{P}]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\bar{P}]\!]^{\mathrm{r}} \hookrightarrow X\})$, is equal to $\mathcal{K}[\![\bar{P}\{\mathrm{rec}\, X : \bar{P} \hookrightarrow X\}]\!]^{\mathrm{r}}$ by virtue of Lemma 4.3.

$\Longleftarrow$) Assuming that $\mathcal{K}[\![P]\!]^{\mathrm{r}} \xrightarrow[0]{\alpha@0} \mathcal{K}[\![P']\!]^{\mathrm{r}}$, the proof that $P \xrightarrow{\alpha} P'$ is similar to the previous one, in the sense that it proceeds by induction on the number of operational semantic rules of Table 2 that have been applied in order to derive the transition $\mathcal{K}[\![P]\!]^{\mathrm{r}} \xrightarrow[0]{\alpha@0} \mathcal{K}[\![P']\!]^{\mathrm{r}}$ and by performing a case analysis based on the syntactical structure of $\mathcal{K}[\![P]\!]^{\mathrm{r}}$. ■

The third preliminary result generalizes the second one by formalizing the correspondence between the action transitions of a modified TCCS process and the transitions, labeled with the same actions and duration zero, of all augmented modified CIPA processes that, after a suitable time shift, are structurally congruent to the CIPA encoding of the original process (see, e.g., state P_4 in Fig. 1 and state K_4 in Fig. 2, which satisfy $up(K_4, 1) \equiv \mathcal{K}[\![P_4]\!]^{\mathrm{r}}$).

Lemma 4.5. Let $\alpha \in Act$ and let $P \in \mathbb{P}'_{\mathrm{TCCS}}$ be restriction and delay-choice free:

1. If $P \xrightarrow{\alpha} P'$ for some $P' \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free, then for all $k \in \mathbb{N}$ and $K \in \mathbb{KP}'$ such that $wf(K, k)$ and $up(K, k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$ it holds that $K \xrightarrow[0]{\alpha@k} K'$ for some $K' \in \mathbb{KP}'$ such that $wf(K', k)$ and $up(K', k) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.
2. For all $k \in \mathbb{N}$ and $K \in \mathbb{KP}'$ such that $wf(K, k)$ and $up(K, k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$, if $K \xrightarrow[0]{\alpha@k} K'$ for some $K' \in \mathbb{KP}'$ such that $wf(K', k)$, then $P \xrightarrow{\alpha} P'$ for some $P' \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free such that $up(K', k) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

Proof Given $\alpha \in Act$ and $P \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free, the proof is divided into two parts:

1. Assuming that $P \xrightarrow{\alpha} P'$ for some $P' \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free, we prove the result by proceeding by induction on the length of the derivation of the action transition $P \xrightarrow{\alpha} P'$, intended as the number $n \in \mathbb{N}_{\geq 1}$ of operational semantic rules on the left-hand side of Table 1 that have been applied in order to derive the considered transition:
   - If $n = 1$, then $P = \alpha\,.\,P'$. Given $k \in \mathbb{N}$, the only process $K \in \mathbb{KP}'$ that satisfies the conditions $wf(K, k)$ and $up(K, k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$ is $K = k \Rightarrow (\alpha\,.\,[\![P']\!]^{\mathrm{r}})$ in the case that $\alpha \neq \tau$, or $K = k \Rightarrow (\mathrm{wait}\,0\,.\,[\![P']\!]^{\mathrm{r}})$ in the case that $\alpha = \tau$. In either case, $K \xrightarrow[0]{\alpha@k} k \Rightarrow [\![P']\!]^{\mathrm{r}}$. If we let $K' = k \Rightarrow [\![P']\!]^{\mathrm{r}}$, then $wf(K', k)$ and $up(K', k) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.
   - Let $n > 1$ and suppose that the result holds for every transition derivable from a process in $\mathbb{P}'_{\mathrm{TCCS}}$, which is restriction and delay-choice free, by applying less than n operational semantic rules on the left-hand side of Table 1. There are several cases based on the syntactical structure of P (which is restriction and delay-choice free):
     - If $P = P_1 + P_2$, then the transition derives from the fact that $P_i \xrightarrow{\alpha} P'$ for some $i \in \{1, 2\}$. Given $k \in \mathbb{N}$, any process $K \in \mathbb{KP}'$ that satisfies the conditions $wf(K, k)$ and $up(K, k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}} = \mathcal{K}[\![P_1]\!]^{\mathrm{r}} + \mathcal{K}[\![P_2]\!]^{\mathrm{r}}$ is of the form $K = K_1 + K_2$ where $K_j \in \mathbb{KP}', wf(K_j, k)$, and $up(K_j, k) \equiv \mathcal{K}[\![P_j]\!]^{\mathrm{r}}$ for all $j \in \{1, 2\}$. From the induction hypothesis, it follows that $K_i \xrightarrow[0]{\alpha@k} K'$, hence $K \xrightarrow[0]{\alpha@k} K'$, for some $K' \in \mathbb{KP}'$ such that $wf(K', k)$ and $up(K', k) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

Note that the negative premise in the CIPA rule for the + operator applied to K is satisfied because there cannot be local clocks in K that are less than k. This follows from the fact that either $up(K,k)$ is exactly equal to $\mathcal{K}[\![P]\!]^{\mathrm{r}}$, or it is the result of the application of structural congruence. In the former case, all local clocks in K are exactly equal to k. In the latter case, some waitings have been moved from a subprocess to the corresponding local clock, which then is greater than or equal to k. Thus, no other subprocess in K can perform an action at a time less than k.

– If $P = P_1 \mid P_2$, there are two subcases for the derivation of the transition. Before examining them, we observe that, given $k \in \mathbb{N}$, any process $K \in \mathbb{KP}'$ that satisfies the conditions $wf(K,k)$ and $up(K,k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}} = \mathcal{K}[\![P_1]\!]^{\mathrm{r}} \mid \mathcal{K}[\![P_2]\!]^{\mathrm{r}}$ is of the form $K = K_1 \mid K_2$ where $K_j \in \mathbb{KP}'$, $wf(K_j,k)$, and $up(K_j,k) \equiv \mathcal{K}[\![P_j]\!]^{\mathrm{r}}$ for all $j \in \{1,2\}$. Here are the two subcases:
  * If the transition is not originated from a synchronization, then it derives from the fact that $P_i \xrightarrow{\alpha} P_i'$ for some $i \in \{1,2\}$. Without loss of generality, we assume $i = 1$, so that $P' = P_1' \mid P_2$. From the induction hypothesis, it follows that $K_1 \xrightarrow[0]{\alpha@k} K_1'$ for some $K_1' \in \mathbb{KP}'$ such that $wf(K_1',k)$ and $up(K_1',k) \equiv \mathcal{K}[\![P_1']\!]^{\mathrm{r}}$, thus $K \xrightarrow[0]{\alpha@k} K_1' \mid K_2$ with $wf(K_1' \mid K_2, k)$ and $up(K_1' \mid K_2, k) = up(K_1',k) \mid up(K_2,k) \equiv \mathcal{K}[\![P_1']\!]^{\mathrm{r}} \mid \mathcal{K}[\![P_2]\!]^{\mathrm{r}} = \mathcal{K}[\![P']\!]^{\mathrm{r}}$. As far as the negative premise in the CIPA rule for the $\mid$ operator applied to K is concerned, the same observation made above for the + operator holds true.
  * If the transition is originated from a synchronization, then it derives from the fact that $P_1 \xrightarrow{a} P_1'$ and $P_2 \xrightarrow{\bar{a}} P_2'$ for some $a \in Act \setminus \{\tau\}$, with $\alpha = \tau$ and $P' = P_1' \mid P_2'$. From the induction hypothesis, it follows that $K_1 \xrightarrow[0]{a@k} K_1'$ for some $K_1' \in \mathbb{KP}'$ such that $wf(K_1',k)$ and $up(K_1',k) \equiv \mathcal{K}[\![P_1']\!]^{\mathrm{r}}$ as well as $K_2 \xrightarrow[0]{\bar{a}@k} K_2'$ for some $K_2' \in \mathbb{KP}'$ such that $wf(K_2',k)$ and $up(K_2',k) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$, thus $K \xrightarrow[0]{\tau@k} K_1' \mid K_2'$ with $wf(K_1' \mid K_2', k)$ and $up(K_1' \mid K_2', k) = up(K_1',k) \mid up(K_2',k) \equiv \mathcal{K}[\![P_1']\!]^{\mathrm{r}} \mid \mathcal{K}[\![P_2']\!]^{\mathrm{r}} = \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

– If $P = \bar{P}\,[\varphi]$, then the transition derives from the fact that $\bar{P} \xrightarrow{\beta} \bar{P}'$ for some $\beta \in Act$ such that $\varphi(\beta) = \alpha$, with $P' = \bar{P}'\,[\varphi]$. Given $k \in \mathbb{N}$, any process $K \in \mathbb{KP}'$ that satisfies the conditions $wf(K,k)$ and $up(K,k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}} = \mathcal{K}[\![\bar{P}]\!]^{\mathrm{r}}\,[\varphi]$ is of the form $K = \bar{K}\,[\varphi]$ where $\bar{K} \in \mathbb{KP}'$, $wf(\bar{K},k)$, and $up(\bar{K},k) \equiv \mathcal{K}[\![\bar{P}]\!]^{\mathrm{r}}$. From the induction hypothesis, it follows that $\bar{K} \xrightarrow[0]{\beta@k} \bar{K}'$ for some $\bar{K}' \in \mathbb{KP}'$ such that $wf(\bar{K}',k)$ and $up(\bar{K}',k) \equiv \mathcal{K}[\![\bar{P}']\!]^{\mathrm{r}}$, thus $K \xrightarrow[0]{\alpha@k} \bar{K}'\,[\varphi]$ with $wf(\bar{K}'\,[\varphi],k)$ and $up(\bar{K}'\,[\varphi],k) = up(\bar{K}',k)\,[\varphi] \equiv \mathcal{K}[\![\bar{P}']\!]^{\mathrm{r}}\,[\varphi] = \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

– If $P = \mathrm{rec}\,X : \bar{P}$, then the transition derives from the fact that $\bar{P}\{\mathrm{rec}\,X : \bar{P} \hookrightarrow X\} \xrightarrow{\alpha} P'$. Given $k \in \mathbb{N}$, the only processes in $\mathbb{KP}'$ that satisfy the conditions involving k and P are $K = k \Rightarrow (\mathrm{rec}\,X : [\![\bar{P}]\!]^{\mathrm{r}})$ and its unfolding $\bar{K} = k \Rightarrow ([\![\bar{P}]\!]^{\mathrm{r}}\{\mathrm{rec}\,X : [\![\bar{P}]\!]^{\mathrm{r}} \hookrightarrow X\})$, because $\bar{K} \equiv K$. Since by virtue of Lemma 4.3 it holds that $\mathcal{K}[\![\bar{P}\{\mathrm{rec}\,X : \bar{P} \hookrightarrow X\}]\!]^{\mathrm{r}} = 0 \Rightarrow ([\![\bar{P}]\!]^{\mathrm{r}}\{\mathrm{rec}\,X : [\![\bar{P}]\!]^{\mathrm{r}} \hookrightarrow X\}) = up(\bar{K},k)$, we proceed as follows in the two subcases:
  * In the subcase of K, from the induction hypothesis it follows that $\bar{K} \xrightarrow[0]{\alpha@k} K'$, hence $K \xrightarrow[0]{\alpha@k} K'$, for some $K' \in \mathbb{KP}'$ such that $wf(K',k)$ and $K' \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.
  * In the subcase of $\bar{K}$, it is as if we started from $\bar{P}\{\mathrm{rec}\,X : \bar{P} \hookrightarrow X\}$, a process whose syntactical structure has already been treated in this proof, hence $\bar{K} \xrightarrow[0]{\alpha@k} K'$ for some $K' \in \mathbb{KP}'$ such that $wf(K',k)$ and $K' \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

2. Let $k \in \mathbb{N}$ and $K \in \mathbb{KP}'$ be such that $wf(K,k)$ and $up(K,k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$. Assuming that $K \xrightarrow[0]{\alpha@k} K'$ for some $K' \in \mathbb{KP}'$ such that $wf(K',k)$, the proof that $P \xrightarrow{\alpha} P'$, for some $P' \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free such that $up(K',k) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$, is similar to the previous one, in the sense that it proceeds by induction on the number of operational semantic rules of Table 2 that have been applied in order to derive the transition $K \xrightarrow[0]{\alpha@k} K'$ and by performing a case analysis based on the syntactical structure of K. ■

The fourth preliminary result formalizes the correspondence between the delay transitions of a modified TCCS process and the *sequences* of τ-transitions labeled with durations greater than zero of *some* augmented modified CIPA processes, having no transitions of duration zero, that, after a suitable time shift, are structurally congruent to the CIPA encoding of the original process (see, e.g., state P_3 in Fig. 1 and state K_3 in Fig. 2 satisfying $K_3 = \mathcal{K}[\![P_3]\!]^{\mathrm{r}}$, as well as state P_5 in Fig. 1 and state K_5 in Fig. 2 satisfying $up(K_5,1) \equiv \mathcal{K}[\![P_5]\!]^{\mathrm{r}}$). Considering CIPA processes without transitions of duration zero is a consequence of the fact that, under action urgency, TCCS processes with delay transitions cannot have action transitions. In addition to that, as can be noted there are other three differences with respect to the correspondence established for action transitions with the previous two lemmas.

Firstly, a tight correspondence like the one in Lemma 4.4 does not necessarily hold for delay transitions. Unlike [12], where the TCCS processes generated by the direct encoding of Def. 3.1 feature (apart from the translation of nil) a strict alternation between action prefixes and delay prefixes, the TCCS processes to which the reverse encoding is applied may start with a delay prefix. When mimicking a TCCS delay transition in CIPA, we have to pay attention to the different way in which time additivity is supported in TCCS and CIPA. For instance, process $\bar{P}_2 = (2) \,.\, a \,.\, \underline{\mathbf{0}}$ of Sect. 4.3 has a delay transition labeled with 1 that cannot be matched by $\mathcal{K}[\![\bar{P}_2]\!]^{\mathrm{r}}$, but can be matched by any process $K_k \in \mathbb{KP}'$ of the form $k \Rightarrow (\mathrm{wait}\,1 \,.\, \mathrm{wait}\,1 \,.\, a \,.\, \mathrm{nil})$ for $k \in \mathbb{N}$. Notice that $up(K_k,k)$, i.e., $K_0 = 0 \Rightarrow (\mathrm{wait}\,1 \,.\, \mathrm{wait}\,1 \,.\, a \,.\, \mathrm{nil})$, is syntactically different from, but structurally congruent to, process $\mathcal{K}[\![\bar{P}_2]\!]^{\mathrm{r}} = 0 \Rightarrow (\mathrm{wait}\,2 \,.\, a \,.\, \mathrm{nil})$ because we have that $0 \Rightarrow (\mathrm{wait}\,1 \,.\, \mathrm{wait}\,1 \,.\, a \,.\, \mathrm{nil}) \equiv 1 \Rightarrow (\mathrm{wait}\,1 \,.\, a \,.\, \mathrm{nil}) \equiv 2 \Rightarrow (a \,.\, \mathrm{nil}) \equiv 0 \Rightarrow (\mathrm{wait}\,2 \,.\, a \,.\, \mathrm{nil})$.

Secondly, aiming at establishing for delay transitions a looser correspondence like the one in Lemma 4.5, we can rely only on some – not all – augmented modified CIPA processes, having no transitions of duration zero, that, after a suitable time shift, are structurally congruent to the CIPA encoding of the original process. This is shown by the example above, where the delay transition of $\bar{P}_2$ labeled with 1 can be matched by K_k for all $k \in \mathbb{N}$, but not by $\mathcal{K}[\![\bar{P}_2]\!]^{\mathrm{r}}$, which is structurally congruent to itself after a zero time shift.

Thirdly, on the CIPA side it is necessary to work with sequences of τ-transitions, instead of individual τ-transitions. Consider for example the modified TCCS process $P = (1) \,.\, a \,.\, \underline{\mathbf{0}} \mid (1) \,.\, b \,.\, \underline{\mathbf{0}}$, which has a delay transition labeled with 1 to $P' = a \,.\, \underline{\mathbf{0}} \mid b \,.\, \underline{\mathbf{0}}$. Its CIPA encoding $\mathcal{K}[\![P]\!]^{\mathrm{r}} = 0 \Rightarrow (\mathrm{wait}\,1 \,.\, a \,.\, \mathrm{nil}) \mid 0 \Rightarrow (\mathrm{wait}\,1 \,.\, b \,.\, \mathrm{nil})$ has a τ-transition at time 0 of duration 1 to $K'_{\mathrm{l}} = 1 \Rightarrow (a \,.\, \mathrm{nil}) \mid 0 \Rightarrow (\mathrm{wait}\,1 \,.\, b \,.\, \mathrm{nil})$ and a τ-transition at time 0 of duration 1 to $K'_{\mathrm{r}} = 0 \Rightarrow (\mathrm{wait}\,1 \,.\, a \,.\, \mathrm{nil}) \mid 1 \Rightarrow (b \,.\, \mathrm{nil})$. Both K'_{l} and K'_{r} have a τ-transition at time 0 of duration 1 to $K' = 1 \Rightarrow (a \,.\, \mathrm{nil}) \mid 1 \Rightarrow (b \,.\, \mathrm{nil})$, but neither can execute its only enabled visible action at time 1 as this would generate an ill-timed path. Notice that $wf(K',1)$ and $up(K',1) = \mathcal{K}[\![P']\!]^{\mathrm{r}}$, while $wf(K'_{\mathrm{l}},1)$ and $wf(K'_{\mathrm{r}},1)$ do not hold because one of their local clocks is 0. Thus, only K' can be considered for establishing a correspondence with P', and K' can be reached from $\mathcal{K}[\![P]\!]^{\mathrm{r}}$ only after a sequence of two τ-transitions, each starting at time 0 and having duration 1.

In general, the τ-transitions in the sequence will not have the same duration. For instance, the modified TCCS process $\tilde{P} = (2) \,.\, \mathrm{rec}\, X : ((1) \,.\, a \,.\, \underline{\mathbf{0}} \mid (5) \,.\, b \,.\, X)$ has a delay transition of duration 3 that cannot be matched by a sequence of two τ-transitions with the same duration starting from $\mathcal{K}[\![\tilde{P}]\!]^{\mathrm{r}}$. However, $\mathcal{K}[\![\tilde{P}]\!]^{\mathrm{r}}$ can respond with a τ-transition at time 0 of duration 2 followed by a τ-transition at time 2 of duration 1. As another example, K_3 in Fig. 2 has a τ-transition of duration 2 that cannot be matched by the only delay transition of P_3 in Fig. 1. However, the sequence formed by the τ-transition of duration 2 from K_3 to K''_3 and the τ-transition of duration 1 from K''_3 to K_4 can be matched by the delay transition labeled with 1 from P_3 to P_4; notice that $wf(K''_3,1)$ does not hold, while $wf(K_4,1)$ and $up(K_4,1) \equiv \mathcal{K}[\![P_4]\!]^{\mathrm{r}}$.

Lemma 4.6. Let $t \in \mathbb{N}_{>0}$ and let $P \in \mathbb{P}'_{\mathrm{TCCS}}$ be restriction and delay-choice free:

1. If $P \stackrel{t}{\leadsto} P'$ for some $P' \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free, then for all $k \in \mathbb{N}$ there exists $K \in \mathbb{KP}'$, having no transitions of duration zero at time k and satisfying $wf(K,k)$ and $up(K,k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$, such that $K = K_0 \xrightarrow[t_0]{\tau@k_0} K_1 \xrightarrow[t_1]{\tau@k_1} \cdots \xrightarrow[t_{m-1}]{\tau@k_{m-1}} K_m = K'$ for some:
   - $m \in \mathbb{N}_{\geq 1}$,
   - $K_i \in \mathbb{KP}'$ for all $1 \leq i \leq m$,
   - $k_i \in \mathbb{N}_{\geq k}$ for all $0 \leq i \leq m-1$, with $k_0 = k$ and $k_i \leq k_j$ for $i \leq j$, and
   - $t_i \in \mathbb{N}_{>0}$ for all $0 \leq i \leq m-1$

   where $wf(K', k+t)$ and $up(K', k+t) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.
2. For all $k \in \mathbb{N}$ there exists $K \in \mathbb{KP}'$, having no transitions of duration zero at time k and satisfying $wf(K,k)$ and $up(K,k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$, such that, if $K = K_0 \xrightarrow[t_0]{\tau@k_0} K_1 \xrightarrow[t_1]{\tau@k_1} \cdots \xrightarrow[t_{m-1}]{\tau@k_{m-1}} K_m = K'$ for some:
   - $m \in \mathbb{N}_{\geq 1}$,
   - $K_i \in \mathbb{KP}'$ for all $1 \leq i \leq m$,
   - $k_i \in \mathbb{N}_{\geq k}$ for all $0 \leq i \leq m-1$, with $k_0 = k$ and $k_i \leq k_j$ for $i \leq j$, and
   - $t_i \in \mathbb{N}_{>0}$ for all $0 \leq i \leq m-1$

   where $wf(K', k+t)$, then $P \stackrel{t}{\leadsto} P'$ for some $P' \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free such that $up(K', k+t) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

Proof Given $t \in \mathbb{N}_{>0}$ and $P \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free, the proof is divided into two parts:

1. Assuming that $P \stackrel{t}{\leadsto} P'$ for some $P' \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free, we prove the result by proceeding by induction on the length of the derivation of the delay transition $P \stackrel{t}{\leadsto} P'$, intended as the number $n \in \mathbb{N}_{\geq 1}$ of operational semantic rules on the right-hand side of Table 1 that have been applied in order to derive the considered transition:
   - If $n = 1$ and $P = \underline{\mathbf{0}}$, then $P' = \underline{\mathbf{0}}$. Given $k \in \mathbb{N}$, we choose $K = k \Rightarrow \mathrm{wait}\, t\,.\,\mathrm{nil}$, which has no transitions of duration zero at time k and satisfies $wf(K,k)$ and $up(K,k) = 0 \Rightarrow \mathrm{wait}\, t.\mathrm{nil} \equiv t \Rightarrow \mathrm{nil} \equiv 0 \Rightarrow \mathrm{nil} = \mathcal{K}[\![P]\!]^{\mathrm{r}}$ by virtue of the structural congruence rule for nil. It holds that $K \xrightarrow[t]{\tau@k} (k+t) \Rightarrow \mathrm{nil}$. If we let $K' = (k+t) \Rightarrow \mathrm{nil}$, then $wf(K', k+t)$ and $up(K', k+t) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.
   - If $n = 1$ and $P = (t')\,.\,\bar{P}$ with $t' \in \mathbb{N}_{\geq t}$, there are two subcases:
     - If $t' = t$, then $P' = \bar{P}$. Given $k \in \mathbb{N}$, we choose $K = k \Rightarrow \mathrm{wait}\, t\,.\,[\![P']\!]^{\mathrm{r}}$, which has no transitions of duration zero at time k and satisfies $wf(K,k)$ and $up(K,k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$. It holds that $K \xrightarrow[t]{\tau@k} (k+t) \Rightarrow [\![P']\!]^{\mathrm{r}}$. If we let $K' = (k+t) \Rightarrow [\![P']\!]^{\mathrm{r}}$, then $wf(K', k+t)$ and $up(K', k+t) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.
     - If $t' > t$, then $P' = (t'-t)\,.\,\bar{P}$. Given $k \in \mathbb{N}$, we choose $K = k \Rightarrow \mathrm{wait}\, t\,.\,\mathrm{wait}\,(t'-t)\,.\,[\![\bar{P}]\!]^{\mathrm{r}}$, which has no transitions of duration zero at time k and satisfies $wf(K,k)$ and $up(K,k) = 0 \Rightarrow \mathrm{wait}\, t\,.\,\mathrm{wait}\,(t'-t)\,.\,[\![\bar{P}]\!]^{\mathrm{r}} \equiv t \Rightarrow \mathrm{wait}\,(t'-t)\,.\,[\![\bar{P}]\!]^{\mathrm{r}} \equiv (t + (t'-t)) \Rightarrow [\![\bar{P}]\!]^{\mathrm{r}} = t' \Rightarrow [\![\bar{P}]\!]^{\mathrm{r}} \equiv 0 \Rightarrow \mathrm{wait}\, t'\,.\,[\![\bar{P}]\!]^{\mathrm{r}} = \mathcal{K}[\![P]\!]^{\mathrm{r}}$. It holds that $K \xrightarrow[t]{\tau@k} (k+t) \Rightarrow \mathrm{wait}\,(t'-t)\,.\,[\![P']\!]^{\mathrm{r}}$. If we let $K' = (k+t) \Rightarrow \mathrm{wait}\,(t'-t)\,.\,[\![P']\!]^{\mathrm{r}}$, then $wf(K', k+t)$ and $up(K', k+t) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.
   - Let $n > 1$ and suppose that the result holds for every transition derivable from a process in $\mathbb{P}'_{\mathrm{TCCS}}$, which is restriction and delay-choice free, by applying less than n operational semantic rules on the right-hand side of Table 1. There are several cases based on the syntactical structure of P (which is restriction and delay-choice free):

– If $P = (t')\,.\,\bar{P}$ with $t' \in \mathbb{N}_{]0,t[}$, then the transition derives from the fact that $\bar{P} \overset{t-t'}{\rightsquigarrow} P'$. Given $\bar{k} \in \mathbb{N}$, from the induction hypothesis it follows that there exists $K_{\bar{k}} \in \mathbb{KP}'$, having no transitions of duration zero at time $\bar{k}$ and satisfying $wf(K_{\bar{k}}, \bar{k})$ and $up(K_{\bar{k}}, \bar{k}) \equiv \mathcal{K}[\![\bar{P}]\!]^{\mathrm{r}}$, such that $K_{\bar{k}} = K_{\bar{k},0} \overset{\tau@k_{\bar{k},0}}{\underset{t_{\bar{k},0}}{\longrightarrow}} K_{\bar{k},1} \overset{\tau@k_{\bar{k},1}}{\underset{t_{\bar{k},1}}{\longrightarrow}} \cdots \overset{\tau@k_{\bar{k},m_{\bar{k}}-1}}{\underset{t_{\bar{k},m_{\bar{k}}-1}}{\longrightarrow}} K_{\bar{k},m_{\bar{k}}} = K'_{\bar{k}}$ for some $m_{\bar{k}} \in \mathbb{N}_{\geq 1}$, $K_{\bar{k},i} \in \mathbb{KP}'$ for all $1 \leq i \leq m_{\bar{k}}$, $k_{\bar{k},i} \in \mathbb{N}_{\geq \bar{k}}$ for all $0 \leq i \leq m_{\bar{k}} - 1$ (with $k_{\bar{k},0} = \bar{k}$ and $k_{\bar{k},i} \leq k_{\bar{k},j}$ for $i \leq j$), and $t_{\bar{k},i} \in \mathbb{N}_{>0}$ for all $0 \leq i \leq m_{\bar{k}} - 1$, where $wf(K'_{\bar{k}}, \bar{k} + (t - t'))$ and $up(K'_{\bar{k}}, \bar{k} + (t - t')) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$. Given $k \in \mathbb{N}$, we choose $K = k \Rightarrow \mathrm{wait}\, t' \,.\, K_{k'}$ with $k' = k + t'$, which has no transitions of duration zero at time k and satisfies $wf(K, k)$ and $up(K, k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$. The result then follows by observing that $K \overset{\tau@k}{\underset{t'}{\longrightarrow}} K_{k'} \overset{\tau@k_{k',0}}{\underset{t_{k',0}}{\longrightarrow}} K_{k',1} \overset{\tau@k_{k',1}}{\underset{t_{k',1}}{\longrightarrow}} \cdots \overset{\tau@k_{k',m_{k'}-1}}{\underset{t_{k',m_{k'}-1}}{\longrightarrow}} K_{k',m_{k'}} = K'_{k'}$ where $wf(K'_{k'}, k' + (t - t'))$, i.e., $wf(K'_{k'}, k + t)$, and $up(K'_{k'}, k + t) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

– If $P = P_1 \mid P_2$, then the transition derives from the fact that $P_j \overset{t}{\rightsquigarrow} P'_j$ for all $j \in \{1, 2\}$, with $P' = P'_1 \mid P'_2$. Given $k \in \mathbb{N}$, from the induction hypothesis it follows that for all $j \in \{1, 2\}$ there exists $K_j \in \mathbb{KP}'$, having no transitions of duration zero at time k and satisfying $wf(K_j, k)$ and $up(K_j, k) \equiv \mathcal{K}[\![P_j]\!]^{\mathrm{r}}$, such that $K_j = K_0^j \overset{\tau@k_0^j}{\underset{t_0^j}{\longrightarrow}} K_1^j \overset{\tau@k_1^j}{\underset{t_1^j}{\longrightarrow}} \cdots \overset{\tau@k_{m_j-1}^j}{\underset{t_{m_j-1}^j}{\longrightarrow}} K_{m_j}^j = K'_j$ for some $m_j \in \mathbb{N}_{\geq 1}$, $K_i^j \in \mathbb{KP}'$ for all $1 \leq i \leq m_j$, $k_i^j \in \mathbb{N}_{\geq k}$ for all $0 \leq i \leq m_j - 1$ (with $k_0^j = k$ and $k_i^j \leq k_h^j$ for $i \leq h$), and $t_i^j \in \mathbb{N}_{>0}$ for all $0 \leq i \leq m_j - 1$, where $wf(K'_j, k + t)$ and $up(K'_j, k + t) \equiv \mathcal{K}[\![P'_j]\!]^{\mathrm{r}}$. We choose $K = K_1 \mid K_2$, which has no transitions of duration zero at time k and satisfies $wf(K, k)$ and $up(K, k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$. The result then follows by applying $m_1 + m_2$ times the first two operational semantic rules for parallel composition of Table 2 so to interleave the two sequences of transitions originating from K_1 and K_2 in a way that the resulting path is not ill timed, i.e., the merged sequence of k_i^j values is not decreasing (this is possible because no synchronization is involved). The last process of the resulting path is $K' = K'_1 \mid K'_2$, which satisfies $wf(K', k + t)$ and $up(K', k + t) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

– If $P = \bar{P}\,[\varphi]$, then the transition derives from the fact that $\bar{P} \overset{t}{\rightsquigarrow} \bar{P}'$, with $P' = \bar{P}'\,[\varphi]$. Given $k \in \mathbb{N}$, from the induction hypothesis it follows that there exists $\bar{K} \in \mathbb{KP}'$, having no transitions of duration zero at time k and satisfying $wf(\bar{K}, k)$ and $up(\bar{K}, k) \equiv \mathcal{K}[\![\bar{P}]\!]^{\mathrm{r}}$, such that $\bar{K} = K_0 \overset{\tau@k_0}{\underset{t_0}{\longrightarrow}} K_1 \overset{\tau@k_1}{\underset{t_1}{\longrightarrow}} \cdots \overset{\tau@k_{m-1}}{\underset{t_{m-1}}{\longrightarrow}} K_m = \bar{K}'$ for some $m \in \mathbb{N}_{\geq 1}$, $K_i \in \mathbb{KP}'$ for all $1 \leq i \leq m$, $k_i \in \mathbb{N}_{\geq k}$ for all $0 \leq i \leq m - 1$ (with $k_0 = k$ and $k_i \leq k_j$ for $i \leq j$), and $t_i \in \mathbb{N}_{>0}$ for all $0 \leq i \leq m - 1$, where $wf(\bar{K}', k + t)$ and $up(\bar{K}', k + t) \equiv \mathcal{K}[\![\bar{P}']\!]^{\mathrm{r}}$. We choose $K = \bar{K}\,[\varphi]$, which has no transitions of duration zero at time k and satisfies $wf(K, k)$ and $up(K, k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$. The result then follows by observing that $K = K_0\,[\varphi] \overset{\tau@k_0}{\underset{t_0}{\longrightarrow}} K_1\,[\varphi] \overset{\tau@k_1}{\underset{t_1}{\longrightarrow}} \cdots \overset{\tau@k_{m-1}}{\underset{t_{m-1}}{\longrightarrow}} K_m\,[\varphi] = \bar{K}'\,[\varphi] = K'$, where K' satisfies $wf(K', k + t)$ and $up(K', k + t) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$.

– If $P = \mathrm{rec}\, X : \bar{P}$, then the transition derives from the fact that $\bar{P}\{\mathrm{rec}\, X : \bar{P} \hookrightarrow X\} \overset{t}{\rightsquigarrow} P'$. Given $k \in \mathbb{N}$, from the induction hypothesis it follows that there exists $\bar{K} \in \mathbb{KP}'$, having no transitions of duration zero at time k and satisfying $wf(\bar{K}, k)$ and $up(\bar{K}, k) \equiv \mathcal{K}[\![\bar{P}\{\mathrm{rec}\, X : \bar{P} \hookrightarrow X\}]\!]^{\mathrm{r}}$, such that $\bar{K} = K_0 \overset{\tau@k_0}{\underset{t_0}{\longrightarrow}} K_1 \overset{\tau@k_1}{\underset{t_1}{\longrightarrow}} \cdots \overset{\tau@k_{m-1}}{\underset{t_{m-1}}{\longrightarrow}} K_m = K'$ for some $m \in \mathbb{N}_{\geq 1}$, $K_i \in \mathbb{KP}'$ for all $1 \leq i \leq m$, $k_i \in \mathbb{N}_{\geq k}$ for all $0 \leq i \leq m - 1$ (with $k_0 = k$ and $k_i \leq k_j$ for $i \leq j$), and $t_i \in \mathbb{N}_{>0}$ for all $0 \leq i \leq m - 1$, where $wf(K', k + t)$ and $up(K', k + t) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$. We choose $K = k \Rightarrow \mathrm{rec}\, X : [\![\bar{P}]\!]^{\mathrm{r}}$, which has no transitions of duration zero at time k and satisfies $wf(K, k)$ and $up(K, k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$ because its unfolding $k \Rightarrow [\![\bar{P}]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\bar{P}]\!]^{\mathrm{r}} \hookrightarrow X\}$ is equal to $\bar{K}$ by virtue of $up(\bar{K}, k) \equiv \mathcal{K}[\![\bar{P}\{\mathrm{rec}\, X : \bar{P} \hookrightarrow X\}]\!]^{\mathrm{r}}$ and Lemma 4.3. The result then

follows by observing that $K \xrightarrow[t_0]{\tau@k_0} K_1 \xrightarrow[t_1]{\tau@k_1} \cdots \xrightarrow[t_{m-1}]{\tau@k_{m-1}} K_m = K'$.

2. Suppose that for all $k \in \mathbb{N}$ there exists $K \in \mathbb{KP}'$, having no transitions of duration zero at time k and satisfying $wf(K,k)$ and $up(K,k) \equiv \mathcal{K}[\![P]\!]^{\mathrm{r}}$, such that $K = K_0 \xrightarrow[t_0]{\tau@k_0} K_1 \xrightarrow[t_1]{\tau@k_1} \cdots \xrightarrow[t_{m-1}]{\tau@k_{m-1}} K_m = K'$ for some $m \in \mathbb{N}_{\geq 1}$, $K_i \in \mathbb{KP}'$ for all $1 \leq i \leq m$, $k_i \in \mathbb{N}_{\geq k}$ for all $0 \leq i \leq m-1$ (with $k_0 = k$ and $k_i \leq k_j$ for $i \leq j$), and $t_i \in \mathbb{N}_{>0}$ for all $0 \leq i \leq m-1$, where $wf(K', k+t)$. The proof that $P \overset{t}{\Longrightarrow} P'$, for some $P' \in \mathbb{P}'_{\mathrm{TCCS}}$ restriction and delay-choice free such that $up(K',k) \equiv \mathcal{K}[\![P']\!]^{\mathrm{r}}$, can be done by induction on the length m of the sequence of transitions originating from K. In particular, the proof of the base case $m = 1$ proceeds by induction on the number of operational semantic rules of Table 2 that have been applied in order to derive the transition $K_0 \xrightarrow[t_0]{\tau@k} K_1$ and by performing a case analysis based on the syntactical structure of K. ■

The fifth preliminary result states that it is possible to "reorder" a sequence of CIPA τ-transitions each possessing the same timestamp, so to have first those with duration zero (stemming from TCCS action transitions) and then those with positive duration (arising from TCCS delay transitions). In this way, the final subsequence of τ-transitions, each with positive duration in the reordered sequence, can be "undone" through the use of structural congruence, thus maintaining a relationship with the last state reached after performing the original sequence of τ-transitions. This will be useful in the proof of the full abstraction result to show the correspondence between the sequences of τ-transitions of the reverse encodings of two weakly bisimilar TCCS processes.

Lemma 4.7. Let $k \in \mathbb{N}$ and $K \in \mathbb{KP}'$ be such that $wf(K,k)$. For each computation:

$$K = K_0 \xrightarrow[t_0]{\tau@k} K_1 \xrightarrow[t_1]{\tau@k} \cdots \xrightarrow[t_{m-1}]{\tau@k} K_m = K'$$

where $m \in \mathbb{N}_{\geq 1}$, $K_i \in \mathbb{KP}'$ for all $1 \leq i \leq m$, and $t_i \in \mathbb{N}$ for all $0 \leq i \leq m-1$, there exists a computation:

$$K = K'_0 \, (\xrightarrow[0]{\tau@k})^h \, K'_h \xrightarrow[t'_h]{\tau@k} \cdots \xrightarrow[t'_{m-1}]{\tau@k} K'_m = K'$$

where $h \in \mathbb{N}_{\leq m}$, $K'_i \in \mathbb{KP}'$ for all $h \leq i \leq m$, and $\{t'_h, \ldots, t'_{m-1}\} = \{t_i \mid 0 \leq i \leq m-1 \wedge t_i \in \mathbb{N}_{>0}\}$, such that $wf(K'_i, k)$ and $K'_i \equiv K'$ for all $h \leq i \leq m$.

Proof Let $0 < h < m$ to avoid trivial cases. We start by showing that the reordering is possible. For each $0 \leq i \leq m-1$ such that $t_i = 0$, assume that $K_i \xrightarrow[0]{\tau@k} K_{i+1}$ is executed by the sequential component κ of K_i. We observe what follows:

- The component κ cannot have executed any preceding τ-transition in the original sequence with positive duration. To see this, suppose, by contradiction, that κ has previously executed a τ-transition of duration $t > 0$. Then its local clock would become equal to at least $k+t$, which implies that the i-th transition above should have a timestamp strictly greater than k. This contradicts the fact that all the τ-transitions in the sequence carry the same timestamp k. It thus follows that the i-th transition, which has duration zero, is independent of any other τ-transition with positive duration executed by a sequential component κ' different from κ, hence it can be scheduled before all of them.

- After executing the i-th transition above, the component κ can execute in the original sequence arbitrarily many other τ-transitions of duration zero and at most one τ-transition of positive duration. The latter, if any, occurs independently of any other sequential component κ' and thus can be scheduled at any position after all the τ-transitions of duration zero.

We now prove that $wf(K'_i, k)$ and $K'_i \equiv K'$ for all $h \leq i \leq m$. First, we notice that $wf(K'_i, k)$ simply follows from $wf(K,k)$ and the fact that K'_i is a derivative of K. Second, consider the last $m-h$ τ-transitions in the "reordered" sequence, which all have positive duration. Observe that each of them must be executed by

a different sequential component because, if a sequential component κ executed a τ-transition with positive duration, then its local clock would become greater than k and hence another τ-transition executed by the same κ would carry a timestamp different from k, which contradicts the fact that all the τ-transitions in the original sequence carry the same timestamp k. Thus, each of the last $m-h$ τ-transitions is executed by a different sequential component and corresponds to performing some wait t'_i for $h \leq i \leq m-1$, which increases the local clock of the sequential component by t'_i. The same effect can be produced by directly applying the structural congruence to each of those sequential components, thereby obtaining $K'_i \equiv K'$. ■

We finally exhibit the full abstraction result for the reverse encoding, whose proof exploits the correspondences between TCCS transitions and CIPA transitions established by Lemmas 4.4, 4.5, and 4.6, as well as the possibility of reordering certain sequences of CIPA τ-transitions established by Lemma 4.7.

Theorem 4.8. Let $P_1, P_2 \in \mathbb{P}'_{\rm TCCS}$ be restriction and delay-choice free. Then:
$$P_1 \approx_{\rm TCCS} P_2 \iff [\![P_1]\!]^{\rm r} \approx_{\rm CIPA} [\![P_2]\!]^{\rm r}$$

Proof As $[\![P_1]\!]^{\rm r} \approx_{\rm CIPA} [\![P_2]\!]^{\rm r}$ iff $\mathcal{K}[\![P_1]\!]^{\rm r} \approx_{\rm CIPA} \mathcal{K}[\![P_2]\!]^{\rm r}$, we show that $P_1 \approx_{\rm TCCS} P_2 \iff \mathcal{K}[\![P_1]\!]^{\rm r} \approx_{\rm CIPA} \mathcal{K}[\![P_2]\!]^{\rm r}$:

$\Longrightarrow$) Consider the following relation:
$$\mathcal{B}^{\rm CIPA}_{\rm TCCS} = \{(K_1, K_2) \in (\mathbb{K}\mathbb{P}')^2 \mid \exists P_1, P_2 \in \mathbb{P}'_{\rm TCCS} \text{ restriction and delay-choice free and } \exists k \in \mathbb{N} \text{ s.t.}$$
$$P_1 \approx_{\rm TCCS} P_2 \wedge wf(K_1, k) \wedge wf(K_2, k) \wedge up(K_1, k) \equiv \mathcal{K}[\![P_1]\!]^{\rm r} \wedge up(K_2, k) \equiv \mathcal{K}[\![P_2]\!]^{\rm r}\}$$
which is symmetric because so is $\approx_{\rm TCCS}$. We now prove that it is a CIPA weak timed bisimulation, from which the result will follow. Indeed, if we choose $P_1, P_2 \in \mathbb{P}'_{\rm TCCS}$ restriction and delay-choice free such that $P_1 \approx_{\rm TCCS} P_2$, then for $k = 0$ the processes $K_1 = 0 \Rightarrow [\![P_1]\!]^{\rm r}$ and $K_2 = 0 \Rightarrow [\![P_1]\!]^{\rm r}$ satisfy $wf(K_1, k)$, $wf(K_2, k)$, $up(K_1, k) \equiv \mathcal{K}[\![P_1]\!]^{\rm r}$, and $up(K_2, k) \equiv \mathcal{K}[\![P_2]\!]^{\rm r}$. Therefore $(K_1, K_2) \in \mathcal{B}^{\rm CIPA}_{\rm TCCS}$ and hence $K_1 \approx_{\rm CIPA} K_2$. Since $K_1 = \mathcal{K}[\![P_1]\!]^{\rm r}$ and $K_2 = \mathcal{K}[\![P_2]\!]^{\rm r}$, it holds that $\mathcal{K}[\![P_1]\!]^{\rm r} \approx_{\rm CIPA} \mathcal{K}[\![P_2]\!]^{\rm r}$.
Let $(K_1, K_2) \in \mathcal{B}^{\rm CIPA}_{\rm TCCS}$, so that there exist $P_1, P_2 \in \mathbb{P}'_{\rm TCCS}$ restriction and delay-choice free and $k \in \mathbb{N}$ such that $P_1 \approx_{\rm TCCS} P_2$, $wf(K_1, k)$, $wf(K_2, k)$, $up(K_1, k) \equiv \mathcal{K}[\![P_1]\!]^{\rm r}$, and $up(K_2, k) \equiv \mathcal{K}[\![P_2]\!]^{\rm r}$:

- If $K_1 \xrightarrow[0]{\alpha@k} K'_1$ for some $\alpha \in Act$ and $K'_1 \in \mathbb{K}\mathbb{P}'$, where $wf(K_1, k)$ implies $wf(K'_1, k)$, then from Lemma 4.5(2) it follows that $P_1 \xrightarrow{\alpha} P'_1$ for some $P'_1 \in \mathbb{P}'_{\rm TCCS}$ restriction and delay-choice free such that $up(K'_1, k) \equiv \mathcal{K}[\![P'_1]\!]^{\rm r}$. Since $P_1 \approx_{\rm TCCS} P_2$, we derive that $P_2 \stackrel{\hat{\alpha}}{\Longrightarrow} P'_2$ for some $P'_2 \in \mathbb{P}'_{\rm TCCS}$ restriction and delay-choice free such that $P'_1 \approx_{\rm TCCS} P'_2$. There are two cases:
  - If an α-transition is performed by P_2 or one of its τ-derivatives, then $P_2 \stackrel{\hat{\alpha}}{\Longrightarrow} P'_2$ coincides with $P_2 \, (\xrightarrow{\tau})^* \xrightarrow{\alpha} P'_2$. From repeated applications of Lemma 4.5(1), it follows that $K_2 \, (\xrightarrow[0]{\tau@k})^* \xrightarrow[0]{\alpha@k} K'_2$ for some $K'_2 \in \mathbb{K}\mathbb{P}'$ such that $wf(K'_2, k)$ and $up(K'_2, k) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$.
  - If no α-transition is performed by P_2 or one of its τ-derivatives (possible only when $\alpha = \tau$), then $P'_2 = P_2$ so, taking $K'_2 = K_2$, it holds that $wf(K'_2, k)$ and $up(K'_2, k) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$.

  In both cases, $K_2 \stackrel{\widehat{\alpha@k}}{\underset{0}{\Longrightarrow}} K'_2$. Since $P'_1 \approx_{\rm TCCS} P'_2$, $wf(K'_1, k)$, $wf(K'_2, k)$, $up(K'_1, k) \equiv \mathcal{K}[\![P'_1]\!]^{\rm r}$, and $up(K'_2, k) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$, it also holds that $(K'_1, K'_2) \in \mathcal{B}^{\rm CIPA}_{\rm TCCS}$.
- If $K_1 \xrightarrow[t]{\tau@k} K'_1$ for some $t \in \mathbb{N}_{>0}$ and $K'_1 \in \mathbb{K}\mathbb{P}'$, then it means that an initial wait t prefix of K_1 has been executed. By applying Lemma 4.7 to this single-transition sequence (i.e., in the special case in which $h = 0$), it follows that $wf(K'_1, k)$ and $up(K'_1, k) \equiv \mathcal{K}[\![P_1]\!]^{\rm r}$. Let $P'_1 = P_1$ and $P'_2 = P_2$, so that $P'_1 \approx_{\rm TCCS} P'_2$. Let $K'_2 = K_2$, so that $K_2 \stackrel{\widehat{\tau@k}}{\underset{t}{\Longrightarrow}} K'_2$ with $wf(K'_2, k)$ and $up(K'_2, k) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$. Since $P'_1 \approx_{\rm TCCS} P'_2$, $wf(K'_1, k)$, $wf(K'_2, k)$, $up(K'_1, k) \equiv \mathcal{K}[\![P'_1]\!]^{\rm r}$, and $up(K'_2, k) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$, it also holds that $(K'_1, K'_2) \in \mathcal{B}^{\rm CIPA}_{\rm TCCS}$.

$\Longleftarrow$) Consider the following relation:

$\mathcal{B}^{\rm TCCS}_{\rm CIPA} = \{(P_1,P_2) \in (\mathbb{P}'_{\rm TCCS})^2$ restriction and delay-choice free $\mid \exists K_1,K_2 \in \mathbb{KP}'$ and $\exists k \in \mathbb{N}$ s.t. $K_1 \approx_{\rm CIPA} K_2 \wedge wf(K_1,k) \wedge wf(K_2,k) \wedge up(K_1,k) \equiv \mathcal{K}[\![P_1]\!]^{\rm r} \wedge up(K_2,k) \equiv \mathcal{K}[\![P_2]\!]^{\rm r}\}$

which is symmetric because so is $\approx_{\rm CIPA}$. We now prove that it is a TCCS weak timed bisimulation, from which the result will follow. Indeed, if we choose $P_1,P_2 \in \mathbb{P}'_{\rm TCCS}$ restriction and delay-choice free such that $\mathcal{K}[\![P_1]\!]^{\rm r} \approx_{\rm CIPA} \mathcal{K}[\![P_2]\!]^{\rm r}$, then for $k=0$ the processes $K_1 = 0 \Rightarrow [\![P_1]\!]^{\rm r}$ and $K_2 = 0 \Rightarrow [\![P_2]\!]^{\rm r}$ satisfy $wf(K_1,k)$, $wf(K_2,k)$, $up(K_1,k) \equiv \mathcal{K}[\![P_1]\!]^{\rm r}$, and $up(K_2,k) \equiv \mathcal{K}[\![P_2]\!]^{\rm r}$. Since $K_1 = \mathcal{K}[\![P_1]\!]^{\rm r}$ and $K_2 = \mathcal{K}[\![P_2]\!]^{\rm r}$, it holds that $K_1 \approx_{\rm CIPA} K_2$. Therefore $(P_1,P_2) \in \mathcal{B}^{\rm TCCS}_{\rm CIPA}$ and hence $P_1 \approx_{\rm TCCS} P_2$.

Let $(P_1,P_2) \in \mathcal{B}^{\rm TCCS}_{\rm CIPA}$, so that P_1,P_2 are restriction and delay-choice free and there exist $K_1,K_2 \in \mathbb{KP}'$ and $k \in \mathbb{N}$ such that $K_1 \approx_{\rm CIPA} K_2$, $wf(K_1,k)$, $wf(K_2,k)$, $up(K_1,k) \equiv \mathcal{K}[\![P_1]\!]^{\rm r}$, and $up(K_2,k) \equiv \mathcal{K}[\![P_2]\!]^{\rm r}$:

- If $P_1 \xrightarrow{\alpha} P'_1$ for some $\alpha \in Act$ and $P'_1 \in \mathbb{P}'_{\rm TCCS}$ restriction and delay-choice free, then from Lemma 4.5(1) it follows that $K_1 \xrightarrow[0]{\alpha@k} K'_1$ for some $K'_1 \in \mathbb{KP}'$ such that $wf(K'_1,k)$ and $up(K'_1,k) \equiv \mathcal{K}[\![P'_1]\!]^{\rm r}$. Since $K_1 \approx_{\rm CIPA} K_2$, we derive that $K_2 \xRightarrow[0]{\widehat{\alpha@k}} K'_2$ for some $K'_2 \in \mathbb{KP}'$ such that $K'_1 \approx_{\rm CIPA} K'_2$, where $wf(K_2,k)$ implies $wf(K'_2,k)$. There are two cases:
  - If an α-transition at time k of duration 0 is performed by K_2 or one of its τ-derivatives, then $K_2 \xRightarrow[0]{\widehat{\alpha@k}} K'_2$ coincides with $K_2 \Longrightarrow \hat{K}'_2 \xrightarrow[0]{\alpha@k} K'_2$. Since $wf(K_2,k)$, by virtue of well timedness all the τ-transitions in $K_2 \Longrightarrow \hat{K}'_2$ carry the same timestamp k, thus Lemma 4.7 can be applied to this sequence. It follows that there exists a sequence of transitions $K_2 \,(\xrightarrow[0]{\tau@k})^* \bar{K}'_2$ for some $\bar{K}'_2 \in \mathbb{KP}'$ such that $\bar{K}'_2 \equiv \hat{K}'_2$ and $wf(\bar{K}'_2,k)$. From repeated applications of Lemma 4.5(2) starting directly from K_2, it follows that $P_2 \,(\xrightarrow{\tau})^* \bar{P}'_2$ for some $\bar{P}'_2 \in \mathbb{P}'_{\rm TCCS}$ restriction and delay-choice free such that $up(\bar{K}'_2,k) \equiv \mathcal{K}[\![\bar{P}'_2]\!]^{\rm r}$. From $\bar{K}'_2 \equiv \hat{K}'_2$, it follows that $wf(\hat{K}'_2,k)$ and $up(\hat{K}'_2,k) \equiv \mathcal{K}[\![\bar{P}'_2]\!]^{\rm r}$. Thus, from a final application of Lemma 4.5(2) to the transition $\hat{K}'_2 \xrightarrow[0]{\alpha@k} K'_2$, it follows that $\bar{P}'_2 \xrightarrow{\alpha} P'_2$ for some $P'_2 \in \mathbb{P}'_{\rm TCCS}$ restriction and delay-choice free such that $up(K'_2,k) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$. Summing up, $P_2 \,(\xrightarrow{\tau})^* \xrightarrow{\alpha} P'_2$ with $up(K'_2,k) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$.
  - If no α-transition at time k of duration 0 is performed by K_2 or one of its τ-derivatives (possible only when $\alpha = \tau$), then $K'_2 = K_2$ so, taking $P'_2 = P_2$, it holds that $wf(K'_2,k)$ and $up(K'_2,k) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$.

  In both cases, $P_2 \xRightarrow{\hat{\alpha}} P'_2$. Since $K'_1 \approx_{\rm CIPA} K'_2$, $wf(K'_1,k)$, $wf(K'_2,k)$, $up(K'_1,k) \equiv \mathcal{K}[\![P'_1]\!]^{\rm r}$, and $up(K'_2,k) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$, it also holds that $(P'_1,P'_2) \in \mathcal{B}^{\rm TCCS}_{\rm CIPA}$.
- If $P_1 \overset{t}{\rightsquigarrow} P'_1$ for some $t \in \mathbb{N}_{>0}$ and $P'_1 \in \mathbb{P}'_{\rm TCCS}$ restriction and delay-choice free, then from Lemma 4.6(1) it follows that there exists $\bar{K}_1 \in \mathbb{KP}'$, having no transitions of duration zero at time k and satisfying $wf(\bar{K}_1,k)$ and $up(\bar{K}_1,k) \equiv \mathcal{K}[\![P_1]\!]^{\rm r}$, such that $\bar{K}_1 = K^1_0 \xrightarrow[t_0]{\tau@k_0} K^1_1 \xrightarrow[t_1]{\tau@k_1} \cdots \xrightarrow[t_{m-1}]{\tau@k_{m-1}} K^1_m = K'_1$ for some $m \in \mathbb{N}_{\geq 1}$, $K^1_i \in \mathbb{KP}'$ for all $1 \leq i \leq m$, $k_i \in \mathbb{N}_{\geq k}$ for all $0 \leq i \leq m-1$ (with $k_0 = k$ and $k_i \leq k_j$ for $i \leq j$), and $t_i \in \mathbb{N}_{>0}$ for all $0 \leq i \leq m-1$, where $wf(K'_1,k+t)$ and $up(K'_1,k+t) \equiv \mathcal{K}[\![P'_1]\!]^{\rm r}$. From $up(\bar{K}_1,k) \equiv \mathcal{K}[\![P_1]\!]^{\rm r} \equiv up(K_1,k)$, it follows that $\bar{K}_1 \equiv K_1$ and, due to Prop. 4.2, we obtain that $\bar{K}_1 \approx_{\rm CIPA} K_1$. From $K_1 \approx_{\rm CIPA} K_2$, by transitivity of $\approx_{\rm CIPA}$ it follows that $\bar{K}_1 \approx_{\rm CIPA} K_2$. Thus, K_2 must be able to weak-simulate the sequence of m τ-transitions from $\bar{K}_1$ to K'_1. By definition of $\approx_{\rm CIPA}$, we have that $K_2 = K^2_0 \Longrightarrow K^2_1 \Longrightarrow \cdots \Longrightarrow K^2_m = \bar{K}'_2$, i.e., $K_2 \Longrightarrow \bar{K}'_2$, with $K^1_i \approx_{\rm CIPA} K^2_i$ for all $0 \leq i \leq m$. In particular, $K'_1 \approx_{\rm CIPA} \bar{K}'_2$.

  To conclude the proof, we have to demonstrate that $P_2 \overset{t}{\Longrightarrow} P'_2$ for some $P'_2 \in \mathbb{P}'$ restriction and

delay-choice free, together with the existence of $K_2' \in \mathbb{KP}'$ such that $K_1' \approx_{\mathrm{CIPA}} K_2'$, $wf(K_2', k+t)$, and $up(K_2', k+t) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$. There are three cases:

1. If $K_2 \Longrightarrow \bar{K}_2'$ is an empty sequence of τ-transitions, i.e., $\bar{K}_2' = K_2$, there are two subcases:
   (a) If $wf(\bar{K}_2', k+t)$, then to infer a t-delay transition for P_2 we can use the structural congruence to "undo" one passage of time of duration t in one of the sequential components of $\bar{K}_2'$. In other words, there exists $\bar{K}_2'' \in \mathbb{KP}'$ such that $\bar{K}_2'' \equiv \bar{K}_2' = K_2$, $wf(\bar{K}_2'', k)$, $up(\bar{K}_2'', k) \equiv \mathcal{K}[\![P_2]\!]^{\mathrm{r}}$, and $\bar{K}_2'' \xrightarrow[t]{\tau@k} \bar{K}_2'$. Since the only transition enabled at time k in $\bar{K}_2''$ is the one of duration t, $\bar{K}_2''$ has no transitions of duration zero at time k. Thus, Lemma 4.6(2) can be applied to derive that $P_2 \overset{t}{\rightsquigarrow} P_2'$ where $up(\bar{K}_2', k+t) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$. If we let $K_2' = \bar{K}_2'$, we obtain that $P_2 \overset{t}{\Longrightarrow} P_2'$, $K_1' \approx_{\mathrm{CIPA}} K_2'$, $wf(K_2', k+t)$, and $up(K_2', k+t) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$.
   (b) Suppose now that $\neg wf(\bar{K}_2', k+t)$. It holds, however, that $wf(\bar{K}_2', k)$, $K_1' \approx_{\mathrm{CIPA}} \bar{K}_2'$, and $wf(K_1', k+t)$. Let us first consider the case in which $K_1' \not\approx_{\mathrm{CIPA}} (k+t) \Rightarrow \mathrm{nil}$, i.e., K_1' or one of its τ-derivatives is able to perform a visible transition at a timestamp $\hat{k} \geq k+t$. Since $K_1' \approx_{\mathrm{CIPA}} \bar{K}_2'$, $\bar{K}_2'$ must be able to do further τ steps in order to increase its local clocks and be able to match the visible transition at time $\hat{k}$. Thus, $\bar{K}_2' \Longrightarrow K_2'' \xrightarrow[t']{\tau@\bar{k}} K_2'$ for some $\bar{k} \in \mathbb{N}_{\geq k}$, $t' \in \mathbb{N}_{>0}$, and $K_2', K_2'' \in \mathbb{KP}'$ such that $wf(K_2', k+t)$ and $\neg wf(K_2'', k+t)$. Moreover, $\bar{K}_2''$ is chosen in such a way that it has no transitions of duration zero at time k, which is always feasible because the possibly occurring transitions of duration zero can be all confined in the $\bar{K}_2' \Longrightarrow K_2''$ sequence. There are two further subcases:
      i. If all the τ-transitions in $\bar{K}_2' \Longrightarrow K_2''$ have positive duration, it follows that $\bar{K}_2' \equiv K_2'' \equiv K_2'$. Thus, $\bar{K}_2' \approx_{\mathrm{CIPA}} K_2'' \approx_{\mathrm{CIPA}} K_2'$, $wf(K_2'', k)$, and $up(K_2'', k) \equiv \mathcal{K}[\![P_2]\!]^{\mathrm{r}}$. Since K_2'' has no transitions of duration zero at time k, Lemma 4.6(2) can be applied to obtain that $P_2 \overset{t}{\rightsquigarrow} P_2'$ with $up(K_2', k+t) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$. It follows that $P_2 \overset{t}{\Longrightarrow} P_2'$, $K_1' \approx_{\mathrm{CIPA}} K_2'$, $wf(K_2', k+t)$, and $up(K_2', k+t) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$.
      ii. If some of the τ-transitions in $\bar{K}_2' \Longrightarrow K_2''$ have duration zero, they must preserve $\approx_{\mathrm{CIPA}}$. To see this, first notice that they cannot derive from a synchronization of two visible actions because, in this case, since $up(\bar{K}_2', k) \equiv \mathcal{K}[\![P_2]\!]^{\mathrm{r}}$ (due to the fact that $\bar{K}_2' = K_2$) and P_2 is restriction-free, then the same visible actions could be executed by $\bar{K}_2'$ or one of its τ-derivatives at a timestamp less than $k+t$. This would contradict the hypothesis that $\bar{K}_2' \approx_{\mathrm{CIPA}} K_1'$ because K_1' cannot match any visible action with timestamp less than $k+t$. Thus, any τ-transition of duration zero in the considered sequence must derive from the execution of a wait 0 prefix, which, due to the definition of the reverse encoding, can only derive from an explicit τ-action in P_2.
      The wait 0 prefix cannot resolve any choice. Suppose, by contradiction, that the wait 0 prefix is in alternative composition with another subprocess K_+ that is able to perform an action α. If α is a visible action, then it is executable at a timestamp less than $k+t$. This, as above, contradicts the fact that $K_1' \approx_{\mathrm{CIPA}} \bar{K}_2'$ and $wf(K_1', k+t)$. Thus, α must be a τ-action and must have positive duration or, if of duration zero, must lead to a process that will eventually perform, after a sequence of τ-actions of duration zero, at least a τ-action of positive duration because $\bar{K}_2' \Longrightarrow K_2'' \xrightarrow[t']{\tau@\bar{k}} K_2'$ and $wf(K_2', k+t)$. However, this contradicts the fact that P_2 is delay-choice free because there would be, in P_2, an alternative composition in which a subprocess operand is able to let time elapse. This subprocess is the one derived from K_+ by possible subsequent applications of Lemma 4.5(2) and then of Lemma 4.6(2).
      The wait 0 prefix cannot lead to a process in which a new choice is introduced. Suppose, again by contradiction, that the wait 0 prefix leads to a continuation process K_+' in

which there is an alternative composition. As in the previous case, the choice must be among τ-actions because no visible action can be performed at a time less than $k+t$. Moreover, K'_+ must eventually perform τ-actions of positive duration to increase its local clocks. This leads to the same contradiction as before, because from these transitions we can derive, through the application of Lemmas 4.5(2) and 4.6(2), that P_2 is not delay-choice free. To sum up, any transition of duration zero in $\bar{K}'_2 \Longrightarrow K''_2$, say $\widehat{K} \xrightarrow[0]{\tau@\bar{k}} \widehat{K}'$, derives from a wait 0 prefix that is not in alternative composition with any other action and that does not introduce any alternative composition. It follows that $\widehat{K} \approx_{\mathrm{CIPA}} \widehat{K}'$.

Consider again the sequence of transitions $\bar{K}'_2 \Longrightarrow K''_2 \xrightarrow[t']{\tau@\bar{k}} K'_2$. To go on with the proof, recall that $\bar{K}'_2$ is such that $wf(\bar{K}'_2, k)$, $up(\bar{K}'_2, k) \equiv \mathcal{K}[\![P_2]\!]^{\mathrm{r}}$, and $\bar{K}'_2 \approx_{\mathrm{CIPA}} K'_1$. Let us examine the transitions from $\bar{K}'_2$ to K''_2 one after the other. If the transition at hand has positive duration, its source state is structurally congruent to its target state and hence it preserves the property of being $\approx_{\mathrm{CIPA}}$-equivalent to K'_1 and structurally congruent to $\mathcal{K}[\![P_2]\!]^{\mathrm{r}}$ (or to a τ-derivative of this) once all local clocks have been decreased by k. If the transition at hand has duration zero, say $\widehat{K} \xrightarrow[0]{\tau@\bar{k}} \widehat{K}'$, we apply Lemma 4.5(2) to derive that $P_2 \xrightarrow{\tau} \bar{P}_2$ for some $\bar{P}_2 \in \mathbb{P}'$ restriction and delay-choice free such that $wf(\widehat{K}', k)$, $up(\widehat{K}', k) \equiv \mathcal{K}[\![\bar{P}_2]\!]^{\mathrm{r}}$, and $\widehat{K}' \approx_{\mathrm{CIPA}} K'_1$. At the end, it holds that $K''_2 \approx_{\mathrm{CIPA}} K'_1$, $wf(K''_2, k)$, and $up(K''_2, k) \equiv \mathcal{K}[\![\hat{P}_2]\!]^{\mathrm{r}}$ for some $\hat{P}_2 \in \mathbb{P}'$ restriction and delay-choice free such that $P_2 \, (\xrightarrow{\tau})^* \, \hat{P}_2$.
Since K''_2 has no transitions of duration zero at time k and $wf(K'_2, k+t)$, Lemma 4.6(2) can be applied to the transition $K''_2 \xrightarrow[t']{\tau@\bar{k}} K'_2$. It follows that $\hat{P}_2 \overset{t}{\rightsquigarrow} P'_2$ for some $P'_2 \in \mathbb{P}'$ restriction and delay-choice free such that $up(K'_2, k+t) \equiv \mathcal{K}[\![P'_2]\!]^{\mathrm{r}}$. Moreover, from the fact that $K''_2 \xrightarrow[t']{\tau@\bar{k}} K'_2$ is a τ-transition of positive duration, it follows that $K''_2 \equiv K'_2$ and, thus, $K'_2 \approx_{\mathrm{CIPA}} K''_2 \approx_{\mathrm{CIPA}} K'_1$. Summing up, it holds that $P_2 \overset{t}{\Longrightarrow} P'_2$, $K'_1 \approx_{\mathrm{CIPA}} K'_2$, $wf(K'_2, k+t)$, and $up(K'_2, k+t) \equiv \mathcal{K}[\![P'_2]\!]^{\mathrm{r}}$.

It remains to consider the case in which $\bar{K}'_2 \approx_{\mathrm{CIPA}} K'_1 \approx_{\mathrm{CIPA}} (k+t) \Rightarrow \mathrm{nil}$, i.e., K'_1 and all of its τ-derivatives do not perform any visible action at any timestamp. We can reason along the same lines of cases 1(b)i and 1(b)ii above, apart from the possibility that $\bar{K}'_2$ is not able to perform enough τ-actions of positive duration to reach a process in which all the local clocks are greater than or equal to $k+t$. This could happen when some of its sequential components are of the form $(k+t') \Rightarrow \mathrm{nil}$ for some $t' \in \mathbb{N}_{<t}$. In this case, the structural congruence rule $k \Rightarrow \mathrm{nil} \equiv k' \Rightarrow \mathrm{nil}$ can be used to find a process $\hat{K}'_2 \equiv \bar{K}'_2$ in which each of these components is replaced with $(k + t' + (t - t')) \Rightarrow \mathrm{nil}$, with t' specific to the component, and then transformed into $(k+t') \Rightarrow \mathrm{wait}\,(t-t')\,.\,\mathrm{nil}$ through structural congruence. The process $\hat{K}'_2$ has the same properties as $\bar{K}'_2$, namely $\hat{K}'_2 \approx_{\mathrm{CIPA}} K'_1$, $wf(\hat{K}'_2, k)$, and $up(\hat{K}'_2, k) \equiv \mathcal{K}[\![P_2]\!]^{\mathrm{r}}$. Thus, it can replace $\bar{K}'_2$ to follow the same lines of the proof of cases 1(b)i and 1(b)ii above without the problem of the missing τ-transitions of positive duration.

2. If $K_2 \Longrightarrow \bar{K}'_2$ is a non-empty sequence of τ-transitions that have the same timestamp $\bar{k} \in \mathbb{N}_{\geq k}$, then we use Lemma 4.7 to reorder the τ-transitions so to expose those of duration zero first. We obtain that $K_2 \, (\xrightarrow[0]{\tau@\bar{k}})^* \, \bar{K}''_2 \Longrightarrow \bar{K}'_2$ where all the transitions in $\bar{K}''_2 \Longrightarrow \bar{K}'_2$ have positive duration and $\bar{K}''_2$ has no transitions of duration zero at time $\bar{k}$. From repeated applications of Lemma 4.5(2), it follows that $P_2 \, (\xrightarrow{\tau})^* \, \hat{P}_2$ for some $\hat{P}_2 \in \mathbb{P}'$ restriction and delay-choice free such that $wf(\bar{K}''_2, k)$ and $up(\bar{K}''_2, k) \equiv \mathcal{K}[\![\hat{P}_2]\!]^{\mathrm{r}}$. There are two subcases:

(a) If $wf(\bar{K}_2', k+t)$, then $\bar{K}_2'' \Longrightarrow \bar{K}_2'$ is not empty and Lemma 4.6(2) can be applied to derive that $\hat{P}_2 \stackrel{t}{\leadsto} P_2'$ for some $P_2' \in \mathbb{P}'$ restriction and delay-choice free such that $up(\bar{K}_2', k+t) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$. If we let $K_2' = \bar{K}_2'$, then it follows that $P_2 \stackrel{t}{\Longrightarrow} P_2'$, $K_1' \approx_{\mathrm{CIPA}} K_2'$, $wf(K_2', k+t)$, and $up(K_2', k+t) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$.

(b) If $\neg wf(\bar{K}_2', k+t)$, then we can reason along the same lines as case 1b and its subcases in order to extend the derivation $\bar{K}_2'' \Longrightarrow \bar{K}_2'$ by adding the necessary τ-transitions so to reach a process $K_2' \in \mathbb{K}\mathbb{P}'$ satisfying $wf(K_2', k+t)$ and $K_2' \approx_{\mathrm{CIPA}} K_1'$. It follows that $P_2 \, (\stackrel{\tau}{\longrightarrow})^* \, \hat{P}_2 \, (\stackrel{\tau}{\longrightarrow})^* \, \hat{P}_2' \stackrel{t}{\leadsto} P_2'$ for some $\hat{P}_2', P_2' \in \mathbb{P}'$ restriction and delay-choice free such that $up(K_2', k+t) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$. Hence, again, $P_2 \stackrel{t}{\Longrightarrow} P_2'$, $K_1' \approx_{\mathrm{CIPA}} K_2'$, $wf(K_2', k+t)$, and $up(K_2', k+t) \equiv \mathcal{K}[\![P_2']\!]^{\mathrm{r}}$.

3. If $K_2 \Longrightarrow \bar{K}_2'$ is a non-empty sequence of τ-transitions that do not have the same timestamp, we partition it into $\ell \in \mathbb{N}_{\geq 2}$ subsequences of τ-transitions respectively having the same timestamps $\bar{k}_0 < \bar{k}_1 < \cdots < \bar{k}_{\ell-1}$, with $\bar{k}_0 \geq k$. By repeatedly applying Lemma 4.7 to reorder the τ-transitions so to expose those of duration zero first, the sequence of τ-transitions becomes $K_2 = K_0^2 \, (\stackrel{\tau@k_0}{\underset{0}{\longrightarrow}})^* \, \bar{K}_0^2 \, (\stackrel{\tau@k_0}{\underset{t_0}{\longrightarrow}})^* \, K_1^2 \cdots K_{\ell-1}^2 \, (\stackrel{\tau@k_{\ell-1}}{\underset{0}{\longrightarrow}})^* \, \bar{K}_{\ell-1}^2 \, (\stackrel{\tau@k_{\ell-1}}{\underset{t_{\ell-1}}{\longrightarrow}})^* \, K_\ell^2 = \bar{K}_2'$.

We define a procedure such that at each step i a timed transition of duration $\hat{t}_i = \bar{k}_i - \bar{k}_{i-1}$ of process P_2 or one of its weakly timed bisimilar derivatives can be derived. A special case is when $i = 0$, in which $\hat{t}_0 = \bar{k}_0 - k \geq 0$. The procedure ends whenever a total amount of time t has elapsed as the sum of all the determined $\hat{t}_i$ (possibly with a final remainder) or when $i = \ell - 1$.

At step $i = 0$, we repeatedly apply Lemma 4.5(2) to derive that $P_2 \, (\stackrel{\tau}{\longrightarrow})^* \, \hat{P}_0^2$ for some $\hat{P}_0^2 \in \mathbb{P}'$ restriction and delay-choice free such that $wf(\bar{K}_0^2, k)$ and $up(\bar{K}_0^2, k) \equiv \mathcal{K}[\![\hat{P}_0^2]\!]^{\mathrm{r}}$. There are two subcases:

(a) If $\bar{k}_0 = k$, we use the fact that $K_1^2 \equiv \bar{K}_0^2$ – by virtue of the initial application of Lemma 4.7 – to substitute $\bar{K}_0^2$ for K_1^2 as starting process of the next step $i = 1$. Finally, we let $P_1^2 = \hat{P}_0^2$ and $\hat{t}_0 = 0$.

(b) If $\bar{k}_0 > k$, we let $\hat{t}_0 = \bar{k}_0 - k > 0$. It holds that $wf(K_1^2, k + \hat{t}_0)$ because the transition leading to K_1^2 has timestamp $\bar{k}_0 = k + \hat{t}_0$, thus all the local clocks in K_1^2 must be greater than or equal to $k + \hat{t}_0$ due to the absence of ill-timed paths. Lemma 4.6(2) can then be applied to the transitions $\bar{K}_0^2 \, (\stackrel{\tau@k_0}{\underset{t_0}{\longrightarrow}})^* \, K_1^2$ to obtain that $\hat{P}_0^2 \stackrel{\hat{t}_0}{\leadsto} P_1^2$ for some $P_1^2 \in \mathbb{P}'$ restriction and delay-choice free such that $up(K_1^2, k + \hat{t}_0) \equiv \mathcal{K}[\![P_1^2]\!]^{\mathrm{r}}$.

Thus, at the end of the first step, we have that $P_2 \stackrel{\hat{t}_0}{\Longrightarrow} P_1^2$ (meaning that $P_2 \Longrightarrow P_1^2$ if $\hat{t}_0 = 0$) for some $P_1^2 \in \mathbb{P}'$ restriction and delay-choice free such that $wf(K_1^2, k + \hat{t}_0)$ and $up(K_1^2, k + \hat{t}_0) \equiv \mathcal{K}[\![P_1^2]\!]^{\mathrm{r}}$.

At each step $i > 0$, we repeatedly apply Lemma 4.5(2) to derive that $P_i^2 \, (\stackrel{\tau}{\longrightarrow})^* \, \hat{P}_i^2$ for some $\hat{P}_i^2 \in \mathbb{P}'$ restriction and delay-choice free such that $wf(\bar{K}_i^2, k + \sum_{j=0}^{j=i-1} \hat{t}_j)$ and $up(\bar{K}_i^2, k + \sum_{j=0}^{j=i-1} \hat{t}_j) \equiv \mathcal{K}[\![\hat{P}_i^2]\!]^{\mathrm{r}}$. Let $\hat{t}_i = \bar{k}_i - \bar{k}_{i-1} > 0$. It holds that $wf(K_{i+1}^2, k + \sum_{j=0}^{j=i} \hat{t}_j)$ because the transition leading to K_{i+1}^2 has timestamp $\bar{k}_i = \bar{k}_{i-1} + \hat{t}_i = k + (\sum_{j=0}^{j=i-1} \hat{t}_j) + \hat{t}_i$, thus all the local clocks in K_{i+1}^2 must be greater than or equal to $\bar{k}_i$ due to the absence of ill-timed paths. Lemma 4.6(2) can then be applied to the transitions $\bar{K}_i^2 \, (\stackrel{\tau@k_i}{\underset{t_i}{\longrightarrow}})^* \, K_{i+1}^2$ to obtain that $\hat{P}_i^2 \stackrel{\hat{t}_i}{\leadsto} P_{i+1}^2$ for some $P_{i+1}^2 \in \mathbb{P}'$ restriction and delay-choice free such that $up(K_{i+1}^2, k + \sum_{j=0}^{j=i} \hat{t}_j) \equiv \mathcal{K}[\![P_{i+1}^2]\!]^{\mathrm{r}}$. There are three subcases:

(a) If $\bar{k}_{\ell-1} < k + t$ and $\neg wf(\bar{K}_2', k+t)$, we have shown that $P_2 \stackrel{\hat{t}}{\Longrightarrow} P_\ell^2$ for some $P_\ell^2 \in \mathbb{P}'$ restriction and delay-choice free and $\hat{t} = \sum_{j=0}^{j=\ell-1} \hat{t}_j < t$ such that $up(K_\ell^2, k + \hat{t}) \equiv$

$\mathcal{K}[\![P^2_\ell]\!]^{\rm r}$. Now, we can reason along the same lines as case 1b and its subcases in order to extend the derivation from process $\bar{K}'_2$ by adding the necessary τ-transitions so to reach a process $K'_2 \in \mathbb{KP}'$ satisfying $wf(K'_2, k+t)$ and $K'_2 \approx_{\rm CIPA} K'_1$. It then follows that $P^2_\ell \, (\stackrel{\tau}{\longrightarrow})^* \, \hat{P}'_2 \stackrel{t-\hat{t}}{\leadsto} P'_2$ for some $\hat{P}'_2, P'_2 \in \mathbb{P}'$ restriction and delay-choice free such that $up(K'_2, k+t) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$. Hence, $P_2 \stackrel{t}{\Longrightarrow} P'_2$, $K'_1 \approx_{\rm CIPA} K'_2$, $wf(K'_2, k+t)$, and $up(K'_2, k+t) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$.

(b) If $\bar{k}_{\ell-1} < k+t$ and $wf(\bar{K}'_2, k+t)$, we have shown that $P_2 \stackrel{\hat{t}}{\Longrightarrow} P^2_\ell$ for some $P^2_\ell \in \mathbb{P}'$ restriction and delay-choice free and $\hat{t} = \sum_{j=0}^{j=\ell-1} \hat{t}_j < t$ such that $up(K^2_\ell, k+\hat{t}\,) \equiv \mathcal{K}[\![P^2_\ell]\!]^{\rm r}$. Now, we can use the structural congruence to "undo" one passage of time of duration $t-\hat{t}$ in one of the sequential components of $\bar{K}'_2$. In other words, there exists $\bar{K}''_2 \in \mathbb{KP}'$ such that $\bar{K}''_2 \equiv \bar{K}'_2$, $wf(\bar{K}''_2, k+\hat{t})$, $up(\bar{K}''_2, k+\hat{t}) \equiv \mathcal{K}[\![P^2_\ell]\!]^{\rm r}$, and $\bar{K}''_2 \stackrel{\tau@k_{\ell-1}}{\underset{t-\hat{t}}{\longrightarrow}} \bar{K}'_2$. Since the only transition enabled at time $k+\hat{t}$ in $\bar{K}''_2$ is the one of duration $t-\hat{t}$, $\bar{K}''_2$ has no transitions of duration zero at time $k+\hat{t}$. Thus, Lemma 4.6(2) can be applied to obtain that $P^2_\ell \stackrel{t-\hat{t}}{\leadsto} P'_2$ for some $P'_2 \in \mathbb{P}'$ restriction and delay-choice free such that $up(\bar{K}'_2, k+t) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$. Now, if we let $K'_2 = \bar{K}'_2$, we obtain that $P_2 \stackrel{t}{\Longrightarrow} P'_2$, $K'_1 \approx_{\rm CIPA} K'_2$, $wf(K'_2, k+t)$, and $up(K'_2, k+t) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$.

(c) If $\bar{k}_{\ell-1} \geq k+t$, we have shown that $P_2 \stackrel{\hat{t}}{\Longrightarrow} P^2_\ell$ for some $P^2_\ell \in \mathbb{P}'$ restriction and delay-choice free and $\hat{t} = \sum_{j=0}^{j=\ell-1} \hat{t}_j \geq t$ such that $up(K^2_\ell, k+\hat{t}) \equiv \mathcal{K}[\![P^2_\ell]\!]^{\rm r}$. Now, if $\hat{t} = t$, then we let $K'_2 = \bar{K}'_2$ and $P'_2 = P^2_\ell$ to directly obtain that $P_2 \stackrel{t}{\Longrightarrow} P'_2$, $K'_1 \approx_{\rm CIPA} K'_2$, $wf(K'_2, k+t)$, and $up(K'_2, k+t) \equiv \mathcal{K}[\![P'_2]\!]^{\rm r}$. If, instead, $\hat{t} > t$, then we have to stop earlier, precisely at the latest step (less than $\ell - 1$) such that one of the previous subcases 3a or 3b applies. ■

5. Direct Encoding from MTIPP to IML

A direct encoding of MTIPP processes into IML processes under action eagerness can be easily established by following Def. 3.1, i.e., the direct encoding from CIPA to TCCS. The only important difference is that, when translating an exponentially timed action $<\alpha, \lambda>$, the exponential delay λ must precede – rather than follow – the instantaneous action α.

We now explain the reason behind this. For example, in the MTIPP process:

$$<\alpha_1, \lambda_1> . \underline{0} + <\alpha_2, \lambda_2> . \underline{0}$$

the choice between the two exponentially timed actions is probabilistic in that governed by the race policy. In contrast, in the IML process:

$$\alpha_1 \, . \, (\lambda_1) \, . \, \underline{0} + \alpha_2 \, . \, (\lambda_2) \, . \, \underline{0}$$

the choice is nondeterministic. When $\alpha_1 = \alpha_2 = \alpha$ and $\lambda_1 = \lambda_2 = \lambda$, we further observe that the MTIPP process is $\sim_{\rm MTIPP}$-equivalent to $<\alpha, 2 \cdot \lambda> . \underline{0}$ and the IML process is $\sim_{\rm IML}$-equivalent to $\alpha \, . \, (\lambda) \, . \, \underline{0}$, but the new MTIPP process would be translated into $\alpha \, . \, (2 \cdot \lambda) \, . \, \underline{0}$, which is not $\sim_{\rm IML}$-equivalent to the new IML process, and hence the encoding would not be fully abstract with respect to strong Markovian bisimilarity.

Therefore, the initial MTIPP process must instead be translated into:

$$(\lambda_1) \, . \, \alpha_1 \, . \, \underline{0} + (\lambda_2) \, . \, \alpha_2 \, . \, \underline{0}$$

which is $\sim_{\rm IML}$-equivalent to $(2 \cdot \lambda) \, . \, \alpha \, . \, \underline{0}$ – translation of $<\alpha, 2 \cdot \lambda> . \underline{0}$ – when $\alpha_1 = \alpha_2 = \alpha$ and $\lambda_1 = \lambda_2 = \lambda$. In other words, the instantaneous action in the translation has to witness the completion – rather than the beginning as in Def. 3.1 – of the execution of the original durational action.

Definition 5.1. The direct encoding $[\![\_]\!] : \mathcal{P}_{\mathrm{MTIPP}} \to \mathcal{P}_{\mathrm{IML}}$ under action eagerness is defined as follows:

$$\begin{array}{rcl} [\![\underline{0}]\!] & = & \underline{0} \\ [\![<\alpha, \lambda> . G]\!] & = & (\lambda) . \alpha . [\![G]\!] \\ [\![G_1 + G_2]\!] & = & [\![G_1]\!] + [\![G_2]\!] \\ [\![G_1 \parallel_S G_2]\!] & = & [\![G_1]\!] \parallel_S [\![G_2]\!] \\ [\![G \,/\, H]\!] & = & [\![G]\!] \,/\, H \\ [\![G\,[\varphi]]\!] & = & [\![G]\!]\,[\varphi] \\ [\![X]\!] & = & X \\ [\![\mathrm{rec}\, X : G]\!] & = & \mathrm{rec}\, X : [\![G]\!] \end{array}$$

■

The direct encoding above preserves strong Markovian bisimilarity only for processes that do not include synchronizations, i.e., in which all the occurrences of the parallel composition operator are of the form $\parallel_\emptyset$, thereby pinpointing the source of the different expressive power of MTIPP and IML under action eagerness. For example, given the two processes $G_1 = <a, \lambda> . \underline{0}$ and $G_2 = (<a, \lambda> . \underline{0} + <b, \mu> . \underline{0}) \parallel_{\{b\}} \underline{0}$, we have that $G_1 \sim_{\mathrm{MTIPP}} G_2$ because both of them can only perform the exponentially timed action $<a, \lambda>$, but $[\![G_1]\!] \not\sim_{\mathrm{IML}} [\![G_2]\!]$ because $[\![G_2]\!]$ has a spurious deadlock state deriving from the fact that the exponential delay μ may elapse before realizing that the instantaneous action b cannot be executed. The presence of spurious deadlock states is a consequence of the way in which exponentially timed actions have to be translated. It is worth noting that the synchronization freeness constraint needed for the direct encoding of MTIPP into IML under action eagerness is analogous to the restriction freeness constraint needed for the direct encoding of CIPA into TCCS under action eagerness (see Thm. 3.2).

To prove the full abstraction result, we first demonstrate some preliminary results that, in addition to deal with recursion, establish a correspondence between the transitions of an MTIPP process and sequences formed by a delay transition and an action transition of the encoded version of that process in IML.

Lemma 5.2. Let $G, \hat{G} \in \mathcal{P}_{\mathrm{MTIPP}}$ and $X, Y \in \mathit{Var}$. Then:

$$[\![G\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] \;=\; [\![G]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}$$

Proof We proceed by induction on the syntactical structure of $G \in \mathcal{P}_{\mathrm{MTIPP}}$:

- If $G = \underline{0}$ or $G \in \mathit{Var} \setminus \{Y\}$, then:
$$[\![G\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] \;=\; G \;=\; [\![G]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}$$
- If $G = Y$, then:
$$[\![G\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] \;=\; [\![\mathrm{rec}\, X : \hat{G}]\!] \;=\; \mathrm{rec}\, X : [\![\hat{G}]\!] \;=\; [\![G]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}$$
- Let $G = <\alpha, \lambda>.G'$ and assume that $[\![G'\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] = [\![G']\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}$. Then:
$$\begin{array}{rcl} [\![G\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] & = & [\![<\alpha, \lambda> . (G'\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\})]\!] \\ & = & (\lambda) . \alpha . [\![G'\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] \\ & = & (\lambda) . \alpha . ([\![G']\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}) \\ & = & ((\lambda) . \alpha . [\![G']\!])\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\} \\ & = & [\![G]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\} \end{array}$$
- Let $G = G_1 + G_2$ and for $i \in \{1, 2\}$ assume that $[\![G_i\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] = [\![G_i]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}$. Then:
$$\begin{array}{rcl} [\![G\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] & = & [\![G_1\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\} + G_2\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] \\ & = & [\![G_1\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] + [\![G_2\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] \\ & = & [\![G_1]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\} + [\![G_2]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\} \\ & = & ([\![G_1]\!] + [\![G_2]\!])\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\} \\ & = & [\![G]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\} \end{array}$$
- The case $G = G_1 \parallel_S G_2$ is similar to the previous one.

- Let $G = G' \,/\, H$ and assume that $[\![G'\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] = [\![G']\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}$. Then:
$$\begin{aligned}
[\![G\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] &= [\![G'\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\} \,/\, H]\!] \\
&= [\![G'\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] \,/\, H \\
&= [\![G']\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\} \,/\, H \\
&= ([\![G']\!] \,/\, H)\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\} \\
&= [\![G]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}
\end{aligned}$$

- The case $G = G'\,[\varphi]$ is similar to the previous one.

- Let $G = \mathrm{rec}\, X' : G'$ and assume that $[\![G'\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] = [\![G']\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}$:

  - If $X' = Y$, then:
$$[\![G\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] = [\![G]\!] = [\![G]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}$$
  - If $X' \neq Y$, then:
$$\begin{aligned}
[\![G\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] &= [\![\mathrm{rec}\, X' : (G'\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\})]\!] \\
&= \mathrm{rec}\, X' : [\![G'\{\mathrm{rec}\, X : \hat{G} \hookrightarrow Y\}]\!] \\
&= \mathrm{rec}\, X' : ([\![G']\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}) \\
&= (\mathrm{rec}\, X' : [\![G']\!])\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\} \\
&= [\![G]\!]\{\mathrm{rec}\, X : [\![\hat{G}]\!] \hookrightarrow Y\}
\end{aligned}$$

■

Lemma 5.3. Let $G \in \mathbb{P}_{\mathrm{MTIPP}}$. Then $[\![G]\!]$ has no action transitions and:
$$\sum_{\alpha \in Act} rate_{\mathrm{a}}(G, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) = rate([\![G]\!], \mathbb{P}_{\mathrm{IML}})$$
when G is synchronization free:

Proof We proceed by induction on the syntactical structure of $G \in \mathbb{P}_{\mathrm{MTIPP}}$:

- If $G = \underline{0}$, then $[\![G]\!] = \underline{0}$ and hence has no action transitions. Moreover:
$$\sum_{\alpha \in Act} rate_{\mathrm{a}}(G, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) = 0 = rate([\![G]\!], \mathbb{P}_{\mathrm{IML}})$$
with G synchronization free.

- If $G = <\alpha, \lambda> . G'$, then $[\![G]\!] = (\lambda) . \alpha . [\![P']\!]$ and hence has no action transitions. Moreover:
$$\sum_{\beta \in Act} rate_{\mathrm{a}}(G, \beta, \mathbb{P}_{\mathrm{MTIPP}}) = \lambda = rate([\![G]\!], \mathbb{P}_{\mathrm{IML}})$$
regardless of G being synchronization free or not.

- Let $G = G_1 + G_2$ and for $i \in \{1, 2\}$ assume that $[\![G_i]\!]$ has no action transitions. Then $[\![G]\!] = [\![G_1]\!] + [\![G_2]\!]$ has no action transitions either. Moreover, from the assumption that $\sum_{\alpha \in Act} rate_{\mathrm{a}}(G_i, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) = rate([\![G_i]\!], \mathbb{P}_{\mathrm{IML}})$ when G_i is synchronization free for $i \in \{1, 2\}$, it follows that:
$$\begin{aligned}
\sum_{\alpha \in Act} rate_{\mathrm{a}}(G, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) &= \sum_{\alpha \in Act} rate_{\mathrm{a}}(G_1, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) + \sum_{\alpha \in Act} rate_{\mathrm{a}}(G_2, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) \\
&= rate([\![G_1]\!], \mathbb{P}_{\mathrm{IML}}) + rate([\![G_2]\!], \mathbb{P}_{\mathrm{IML}}) \\
&= rate([\![G]\!], \mathbb{P}_{\mathrm{IML}})
\end{aligned}$$
with G synchronization free.

- The case $G = G_1 \,\|_{\emptyset}\, G_2$ is similar to the previous one because, due to the emptiness of the synchronization set, the total exit rate of G is again the sum of the total exit rates of G_1 and G_2. On the other hand, the total exit rate of $[\![G]\!]$ can only be the sum of the total exit rates of $[\![G_1]\!]$ and $[\![G_2]\!]$, because exponential delays cannot synchronize.

- Let $G = G' \,/\, H$ and assume that $[\![G']\!]$ has no action transitions. Then $[\![G]\!] = [\![G']\!] \,/\, H$ has no action transitions either. Moreover, from the assumption that $\sum_{\alpha \in Act} rate_{\mathrm{a}}(G', \alpha, \mathbb{P}_{\mathrm{MTIPP}}) = rate([\![G']\!], \mathbb{P}_{\mathrm{IML}})$ when G' is synchronization free, it follows that:
$$\sum_{\alpha \in Act} rate_{\mathrm{a}}(G, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) = \sum_{\alpha \in Act} rate_{\mathrm{a}}(G', \alpha, \mathbb{P}_{\mathrm{MTIPP}}) = rate([\![G']\!], \mathbb{P}_{\mathrm{IML}}) = rate([\![G]\!], \mathbb{P}_{\mathrm{IML}})$$
with G synchronization free.

- The case $G = G'\,[\varphi]$ is similar to the previous one.
- Let $G = \mathrm{rec}\,X : G'$ and assume that $[\![G'\{\mathrm{rec}\,X : G' \hookrightarrow X\}]\!]$ has no action transitions. Then $[\![G]\!] = \mathrm{rec}\,X : [\![G']\!]$ has no action transitions either, as $[\![G']\!]\{\mathrm{rec}\,X : [\![G']\!] \hookrightarrow X\} = [\![G'\{\mathrm{rec}\,X : G' \hookrightarrow X\}]\!]$ due to Lemma 5.2. Moreover, from the assumption that $\sum_{\alpha \in Act} rate_{\mathrm{a}}(G'\{\mathrm{rec}\,X : G' \hookrightarrow X\}, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) = rate([\![G'\{\mathrm{rec}\,X : G' \hookrightarrow X\}]\!], \mathbb{P}_{\mathrm{IML}})$ when $G'\{\mathrm{rec}\,X : G' \hookrightarrow X\}$ is synchronization free, it follows that:
$$\begin{aligned}\sum_{\alpha \in Act} rate_{\mathrm{a}}(G, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) &= \sum_{\alpha \in Act} rate_{\mathrm{a}}(G'\{\mathrm{rec}\,X : G' \hookrightarrow X\}, \alpha, \mathbb{P}_{\mathrm{MTIPP}}) \\ &= rate([\![G'\{\mathrm{rec}\,X : G' \hookrightarrow X\}]\!], \mathbb{P}_{\mathrm{IML}}) \\ &= rate([\![G']\!]\{\mathrm{rec}\,X : [\![G']\!] \hookrightarrow X\}, \mathbb{P}_{\mathrm{IML}}) \\ &= rate([\![G]\!], \mathbb{P}_{\mathrm{IML}})\end{aligned}$$
with G synchronization free. ■

Lemma 5.4. Let $G, G' \in \mathbb{P}_{\mathrm{MTIPP}}$ be synchronization free, $\alpha \in Act$, and $\lambda \in \mathbb{R}_{>0}$. Then $G \stackrel{\alpha,\lambda}{\longrightarrow} G'$ iff $[\![G]\!] \stackrel{\lambda}{\leadsto} F \stackrel{\alpha}{\longrightarrow} [\![G']\!]$ for some synchronization-free $F \in \mathbb{P}_{\mathrm{IML}}$ having only that action transition.

Proof Given $G, G' \in \mathbb{P}_{\mathrm{MTIPP}}$ synchronization free, $\alpha \in Act$, and $\lambda \in \mathbb{R}_{>0}$, the proof is divided into two parts:

$\Longrightarrow$) Assuming that $G \stackrel{\alpha,\lambda}{\longrightarrow} G'$, we prove that $[\![G]\!] \stackrel{\lambda}{\leadsto} F \stackrel{\alpha}{\longrightarrow} [\![G']\!]$ for some synchronization-free $F \in \mathbb{P}_{\mathrm{IML}}$ having only that action transition by proceeding by induction on the length of the derivation of $G \stackrel{\alpha,\lambda}{\longrightarrow} G'$, intended as the number $n \in \mathbb{N}_{\geq 1}$ of operational semantic rules of Table 3 that have been applied in order to derive the considered transition:

- If $n = 1$, then $G = <\alpha, \lambda>.\,G'$. Therefore $[\![G]\!] = (\lambda).\,\alpha.\,[\![G']\!] \stackrel{\lambda}{\leadsto} \alpha.\,[\![G']\!] \stackrel{\alpha}{\longrightarrow} [\![G']\!]$ with $\alpha.\,[\![G']\!]$ being synchronization free and having only that action transition.
- Let $n > 1$ and suppose that the result holds for every transition derivable from a synchronization-free MTIPP process by applying less than n operational semantic rules of Table 3. There are several cases based on the syntactical structure of G:
  - If $G = G_1 + G_2$, then the transition derives from the fact that $G_i \stackrel{\alpha,\lambda}{\longrightarrow} G'$ for some $i \in \{1, 2\}$. From the induction hypothesis, it follows that $[\![G_i]\!] \stackrel{\lambda}{\leadsto} F \stackrel{\alpha}{\longrightarrow} [\![G']\!]$ for some synchronization-free $F \in \mathbb{P}_{\mathrm{IML}}$ having only that action transition. Thus $[\![G]\!] = [\![G_1]\!] + [\![G_2]\!] \stackrel{\lambda}{\leadsto} F \stackrel{\alpha}{\longrightarrow} [\![G']\!]$ with F being synchronization free and having only that action transition.
  - If $G = G_1 \parallel_\emptyset G_2$, then the transition derives from the fact that $G_i \stackrel{\alpha,\lambda}{\longrightarrow} G_i'$ for some $i \in \{1, 2\}$. Without loss of generality, we assume $i = 1$, so that $G' = G_1' \parallel_\emptyset G_2$. From the induction hypothesis, it follows that $[\![G_1]\!] \stackrel{\lambda}{\leadsto} F_1 \stackrel{\alpha}{\longrightarrow} [\![G_1']\!]$ for some synchronization-free $F_1 \in \mathbb{P}_{\mathrm{IML}}$ having only that action transition. Thus $[\![G]\!] = [\![G_1]\!] \parallel_\emptyset [\![G_2]\!] \stackrel{\lambda}{\leadsto} F_1 \parallel_\emptyset [\![G_2]\!] \stackrel{\alpha}{\longrightarrow} [\![G_1']\!] \parallel_\emptyset [\![G_2]\!] = [\![G']\!]$ with $F_1 \parallel_\emptyset [\![G_2]\!]$ being synchronization free and having only that action transition because $[\![G_2]\!]$ has no action transitions by virtue of Lemma 5.3.
  - If $G = \bar{G} / H$, then the transition derives from the fact that $\bar{G} \stackrel{\beta,\lambda}{\longrightarrow} \bar{G}'$ with $G' = \bar{G}' / H$ for some $\beta \in Act$ such that $\beta \in H \cup \{\tau\}$ if $\alpha = \tau$, $\beta = \alpha$ otherwise. From the induction hypothesis, it follows that $[\![\bar{G}]\!] \stackrel{\lambda}{\leadsto} \bar{F} \stackrel{\beta}{\longrightarrow} [\![\bar{G}']\!]$ for some synchronization-free $\bar{F} \in \mathbb{P}_{\mathrm{IML}}$ having only that action transition. Thus $[\![G]\!] = [\![\bar{G}]\!] / H \stackrel{\lambda}{\leadsto} \bar{F} / H \stackrel{\alpha}{\longrightarrow} [\![\bar{G}']\!] / H = [\![G']\!]$ with $\bar{F} / H$ being synchronization free and having only that action transition.
  - The case $G = \bar{G}\,[\varphi]$ is similar to the previous one with β satisfying $\varphi(\beta) = \alpha$.

* If $G = \mathrm{rec}\, X : \bar{G}$, then the transition derives from the fact that $\bar{G}\{\mathrm{rec}\, X : \bar{G} \hookrightarrow X\} \xrightarrow{\alpha,\lambda} G'$. From the induction hypothesis, it follows that $[\![\bar{G}\{\mathrm{rec}\, X : \bar{G} \hookrightarrow X\}]\!] \xrightarrow{\lambda}\!\!\rightsquigarrow F \xrightarrow{\alpha} [\![G']\!]$ for some synchronization-free $F \in \mathbb{P}_{\mathrm{IML}}$ having only that action transition. Thus $[\![G]\!] = \mathrm{rec}\, X : [\![\bar{G}]\!] \xrightarrow{\lambda}\!\!\rightsquigarrow F \xrightarrow{\alpha} [\![G']\!]$ because the unfolding of $\mathrm{rec}\, X : [\![\bar{G}]\!]$, i.e., $[\![\bar{G}]\!]\{\mathrm{rec}\, X : [\![\bar{G}]\!] \hookrightarrow X\}$, is equal to $[\![\bar{G}\{\mathrm{rec}\, X : \bar{G} \hookrightarrow X\}]\!]$ by virtue of Lemma 5.2.

$\Longleftarrow$) Assuming that $[\![G]\!] \xrightarrow{\lambda}\!\!\rightsquigarrow F \xrightarrow{\alpha} [\![G']\!]$ for some synchronization-free $F \in \mathbb{P}_{\mathrm{IML}}$ having only that action transition, the proof that $G \xrightarrow{\alpha,\lambda} G'$ is similar to the previous one, in the sense that it proceeds by induction on the number of operational semantic rules on the right-hand side of Table 4 that have been applied in order to derive the delay transition $[\![G]\!] \xrightarrow{\lambda}\!\!\rightsquigarrow F$ and by performing a case analysis based on the syntactical structure of $[\![G]\!]$. ■

Theorem 5.5. Let $G_1, G_2 \in \mathbb{P}_{\mathrm{MTIPP}}$ be synchronization free. Then:

$$G_1 \sim_{\mathrm{MTIPP}} G_2 \iff [\![G_1]\!] \sim_{\mathrm{IML}} [\![G_2]\!]$$

Proof The proof is divided into two parts:

$\Longrightarrow$) Consider the relation $\mathcal{B}^{\mathrm{IML}}_{\mathrm{MTIPP}} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3$ over $\mathbb{P}_{\mathrm{IML}}$ where:

$$\begin{aligned}
\mathcal{B}_1 &= \{([\![G_1]\!], [\![G_2]\!]) \mid G_1, G_2 \in \mathbb{P}_{\mathrm{MTIPP}} \text{ synchronization free} \wedge G_1 \sim_{\mathrm{MTIPP}} G_2\} \\
\mathcal{B}_2 &= \{(F_1, F_2) \mid F_1, F_2 \in \mathbb{P}_{\mathrm{IML}} \text{ synchronization free} \wedge \\
&\qquad \exists G_1', G_2' \in \mathbb{P}_{\mathrm{MTIPP}} \text{ synchronization free}.\, G_1' \sim_{\mathrm{MTIPP}} G_2' \wedge \\
&\qquad \exists \alpha \in \mathit{Act}.\, F_1 \xrightarrow{\alpha} [\![G_1']\!] \wedge F_2 \xrightarrow{\alpha} [\![G_2']\!] \text{ are the only action transitions of } F_1, F_2\} \\
\mathcal{B}_3 &= \{(F, F) \mid F \in \mathbb{P}_{\mathrm{IML}} \text{ not occurring in any pair of } \mathcal{B}_1 \cup \mathcal{B}_2\}
\end{aligned}$$

Relation $\mathcal{B}^{\mathrm{IML}}_{\mathrm{MTIPP}}$ is an equivalence relation because so is $\sim_{\mathrm{MTIPP}}$. Moreover, it turns out to be an IML strong Markovian bisimulation, as we show below:

- If we take $([\![G_1]\!], [\![G_2]\!]) \in \mathcal{B}_1$, then by virtue of Lemma 5.3 neither $[\![G_1]\!]$ nor $[\![G_2]\!]$ has action transitions, hence there are no action transitions to be matched. Given an equivalence class $\mathcal{C} \in \mathbb{P}_{\mathrm{IML}}/\mathcal{B}^{\mathrm{IML}}_{\mathrm{MTIPP}}$, there are two cases related to delay transition matching:
  * If $\mathcal{C}$ is originated from $\mathcal{B}_1 \cup \mathcal{B}_3$, then by virtue of Lemma 5.4 we have that:
    $$\mathit{rate}([\![G_1]\!], \mathcal{C}) \;=\; 0 \;=\; \mathit{rate}([\![G_2]\!], \mathcal{C})$$
  * If $\mathcal{C}$ is originated from $\mathcal{B}_2$ when considering $\alpha \in \mathit{Act}$ and the equivalence class $[G']_{\sim_{\mathrm{MTIPP}}}$, then by virtue of Lemma 5.4 and $G_1 \sim_{\mathrm{MTIPP}} G_2$ we have that:
    $$\mathit{rate}([\![G_1]\!], \mathcal{C}) \;=\; \mathit{rate}_{\mathrm{a}}(G_1, \alpha, [G']_{\sim_{\mathrm{MTIPP}}}) \;=\; \mathit{rate}_{\mathrm{a}}(G_2, \alpha, [G']_{\sim_{\mathrm{MTIPP}}}) \;=\; \mathit{rate}([\![G_2]\!], \mathcal{C})$$
- If we take $(F_1, F_2) \in \mathcal{B}_2$, then any possible delay transition of F_1 and F_2 is preempted, under action eagerness, by the only action transition of F_1 and F_2, respectively, hence there are no delay transitions to be matched. As far as action transition matching is concerned, if $F_1 \xrightarrow{\alpha} [\![G_1']\!]$, then $F_2 \xrightarrow{\alpha} [\![G_2']\!]$ with $([\![G_1']\!], [\![G_2']\!]) \in \mathcal{B}_1 \subseteq \mathcal{B}^{\mathrm{IML}}_{\mathrm{MTIPP}}$ because G_1' and G_2' are synchronization free and $G_1' \sim_{\mathrm{MTIPP}} G_2'$, and vice versa.

$\Longleftarrow$) Consider the relation $\mathcal{B}^{\mathrm{MTIPP}}_{\mathrm{IML}} = \mathcal{B}_1 \cup \mathcal{B}_2$ over $\mathbb{P}_{\mathrm{MTIPP}}$ where:

$$\begin{aligned}
\mathcal{B}_1 &= \{(G_1, G_2) \mid G_1, G_2 \in \mathbb{P}_{\mathrm{MTIPP}} \text{ synchronization free} \wedge [\![G_1]\!] \sim_{\mathrm{IML}} [\![G_2]\!]\} \\
\mathcal{B}_2 &= \{(G, G) \mid G \in \mathbb{P}_{\mathrm{MTIPP}} \text{ not synchronization free}\}
\end{aligned}$$

Relation $\mathcal{B}^{\mathrm{MTIPP}}_{\mathrm{IML}}$ is an equivalence relation because so is $\sim_{\mathrm{IML}}$. Moreover, it turns out to be an MTIPP strong Markovian bisimulation, as we now show. If we take $(G_1, G_2) \in \mathcal{B}_1$ and we consider $\alpha \in \mathit{Act}$ and $\mathcal{C} \in \mathbb{P}_{\mathrm{MTIPP}}/\mathcal{B}^{\mathrm{MTIPP}}_{\mathrm{IML}}$, there are two cases:

- If $\mathcal{C}$ is originated from $\mathcal{B}_2$, then:
  $$\mathit{rate}_{\mathrm{a}}(G_1, \alpha, \mathcal{C}) \;=\; 0 \;=\; \mathit{rate}_{\mathrm{a}}(G_2, \alpha, \mathcal{C})$$

– If $\mathcal{C}$ is originated from $\mathcal{B}_1$ when considering the equivalence class $[[\![G]\!]]_{\sim_{\rm IML}}$, then by virtue of Lemma 5.4 and $[\![G_1]\!] \sim_{\rm IML} [\![G_2]\!]$ we have that:

$$rate_{\rm a}(G_1, \alpha, \mathcal{C}) \;=\; rate([\![G_1]\!], \mathcal{C}_{\alpha,G}) \;=\; rate([\![G_2]\!], \mathcal{C}_{\alpha,G}) \;=\; rate_{\rm a}(G_2, \alpha, \mathcal{C})$$

where $\mathcal{C}_{\alpha,G}$ is the equivalence class of $\sim_{\rm IML}$ containing all the synchronization-free processes F whose only action transition is $F \stackrel{\alpha}{\longrightarrow} [\![G']\!] \in [[\![G]\!]]_{\sim_{\rm IML}}$. ■

6. Reverse Encoding from IML to MTIPP

The definition of a reverse encoding from IML to MTIPP under action eagerness raises issues similar to those discussed in Sects. 4.1 and 4.2 for the mapping of TCCS to CIPA, especially with respect to the association of durations with actions. Following Def. 4.1, every exponential delay λ is translated into an exponentially timed τ-action with rate λ, while every instantaneous action α is transformed into a so-called *immediate action* $<\alpha, \infty>$, which takes place at an infinite speed and hence has duration zero [23, 8, 5].

Such immediate actions are added to the syntax of MTIPP and the operational semantic rules of Table 3 are extended accordingly, by further imposing that no synchronization can take place between an exponentially timed action and an immediate action and that the synchronization of two identically named immediate actions is still immediate. Moreover, the strong Markovian bisimilarity of Def. 2.8 is modified by introducing a double action exit rate check based on the following two values:

$$rate_{\rm a}(G, \alpha, \mathcal{C}, {\rm exp}) \;=\; \sum\{\!|\, \lambda \in \mathbb{R}_{>0} \mid \exists G' \in \mathcal{C}.\, G \stackrel{\alpha,\lambda}{\longrightarrow} G' \,|\!\}$$

$$rate_{\rm a}(G, \alpha, \mathcal{C}, {\rm imm}) \;=\; \begin{cases} 1 & \text{if } \exists G' \in \mathcal{C}.\, G \stackrel{\alpha,\infty}{\longrightarrow} G' \\ 0 & \text{otherwise} \end{cases}$$

We denote by $\mathcal{P}'_{\rm MTIPP}$ the resulting set of process terms, with $\mathbb{P}'_{\rm MTIPP}$ being the subset of closed and guarded process terms, and by $\sim'_{\rm MTIPP}$ the resulting strong Markovian bisimilarity.

Definition 6.1. The reverse encoding $[\![\_]\!]^{\rm r} : \mathcal{P}_{\rm IML} \to \mathcal{P}'_{\rm MTIPP}$ under action eagerness is defined as follows:

$$\begin{array}{rcl}
[\![\underline{0}]\!]^{\rm r} &=& \underline{0} \\
[\![\alpha\,.\,F]\!]^{\rm r} &=& <\alpha, \infty>\,.\,[\![F]\!]^{\rm r} \\
[\![(\lambda)\,.\,F]\!]^{\rm r} &=& <\tau, \lambda>\,.\,[\![F]\!]^{\rm r} \\
[\![F_1 + F_2]\!]^{\rm r} &=& [\![F_1]\!]^{\rm r} + [\![F_2]\!]^{\rm r} \\
[\![F_1 \parallel_S F_2]\!]^{\rm r} &=& [\![F_1]\!]^{\rm r} \parallel_S [\![F_2]\!]^{\rm r} \\
[\![F \,/\, H]\!]^{\rm r} &=& [\![F]\!]^{\rm r} \,/\, H \\
[\![F\,[\varphi]]\!]^{\rm r} &=& [\![F]\!]^{\rm r}\,[\varphi] \\
[\![X]\!]^{\rm r} &=& X \\
[\![{\rm rec}\, X : F]\!]^{\rm r} &=& {\rm rec}\, X : [\![F]\!]^{\rm r}
\end{array}$$
■

We now show that the reverse encoding above preserves strong Markovian bisimilarity. Unlike Thm. 5.5, the limitation to synchronization-free processes is not needed, because the order in which instantaneous action prefixes and exponential delay prefixes alternate in an IML process is preserved in the MTIPP process produced by the translation, hence so are possible spurious deadlock states. This is different from the deterministic-time setting, where the reverse encoding from TCCS to CIPA still needs the restriction-freeness constraint of the direct encoding due to the fact that timelocks are possible only in TCCS.

Lemma 6.2. Let $F, \hat{F} \in \mathcal{P}_{\rm IML}$ and $X, Y \in \mathit{Var}$. Then:

$$[\![F\{{\rm rec}\, X : \hat{F} \hookrightarrow Y\}]\!] \;=\; [\![F]\!]\{{\rm rec}\, X : [\![\hat{F}]\!] \hookrightarrow Y\}$$

Proof Similar to the proof of Lemma 5.2. ■

Lemma 6.3. Let $F, F' \in \mathbb{P}_{\rm IML}$, $\alpha \in \mathit{Act}$, and $\lambda \in \mathbb{R}_{>0}$. Then:

1. $F \stackrel{\alpha}{\longrightarrow} F'$ iff $[\![F]\!]^{\rm r} \stackrel{\alpha,\infty}{\longrightarrow} [\![F']\!]^{\rm r}$.
2. $F \stackrel{\lambda}{\rightsquigarrow} F'$ iff $[\![F]\!]^{\rm r} \stackrel{\tau,\lambda}{\longrightarrow} [\![F']\!]^{\rm r}$.

Proof Given $F, F' \in \mathbb{P}_{\mathrm{IML}}$, we proceed as follows:

1. Let $\alpha \in Act$. The proof is divided into two parts:

   $\Longrightarrow$) Assuming that $F \stackrel{\alpha}{\longrightarrow} F'$, we prove that $[\![F]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$ by proceeding by induction on the length of the derivation of the action transition $F \stackrel{\alpha}{\longrightarrow} F'$, intended as the number $n \in \mathbb{N}_{\geq 1}$ of operational semantic rules on the left-hand side of Table 4 that have been applied in order to derive the considered transition:

   - If $n = 1$, then $F = \alpha \,.\, F'$. Therefore $[\![F]\!]^{\mathrm{r}} = <\alpha, \infty> \,.\, [\![F']\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$.
   - Let $n > 1$ and suppose that the result holds for every transition derivable from an IML process by applying less than n operational semantic rules on the left-hand side of Table 4. There are several cases based on the syntactical structure of F:
     * If $F = F_1 + F_2$, then the transition derives from the fact that $F_i \stackrel{\alpha}{\longrightarrow} F'$ for some $i \in \{1, 2\}$. From the induction hypothesis, it follows that $[\![F_i]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$. Thus $[\![F]\!]^{\mathrm{r}} = [\![F_1]\!]^{\mathrm{r}} + [\![F_2]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$.
     * If $F = F_1 \parallel_S F_2$, there are two subcases:
       · If $\alpha \notin S$, then the transition derives from the fact that $F_i \stackrel{\alpha}{\longrightarrow} F_i'$ for some $i \in \{1, 2\}$. Without loss of generality, we assume $i = 1$, so that $F' = F_1' \parallel_S F_2$. From the induction hypothesis, it follows that $[\![F_1]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F_1']\!]^{\mathrm{r}}$. Thus $[\![F]\!]^{\mathrm{r}} = [\![F_1]\!]^{\mathrm{r}} \parallel_S [\![F_2]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F_1']\!]^{\mathrm{r}} \parallel_S [\![F_2]\!]^{\mathrm{r}} = [\![F']\!]^{\mathrm{r}}$.
       · If $\alpha \in S$, then the transition derives from the fact that $F_i \stackrel{\alpha}{\longrightarrow} F_i'$ for all $i \in \{1, 2\}$, so that $F' = F_1' \parallel_S F_2'$. From the induction hypothesis, it follows that $[\![F_i]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F_i']\!]^{\mathrm{r}}$. Thus $[\![F]\!]^{\mathrm{r}} = [\![F_1]\!]^{\mathrm{r}} \parallel_S [\![F_2]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F_1']\!]^{\mathrm{r}} \parallel_S [\![F_2']\!]^{\mathrm{r}} = [\![F']\!]^{\mathrm{r}}$.
     * If $F = \bar{F} \,/\, H$, then the transition derives from the fact that $\bar{F} \stackrel{\beta}{\longrightarrow} \bar{F}'$ with $F' = \bar{F}' \,/\, H$ for some $\beta \in Act$ such that $\beta \in H \cup \{\tau\}$ if $\alpha = \tau$, $\beta = \alpha$ otherwise. From the induction hypothesis, it follows that $[\![\bar{F}]\!]^{\mathrm{r}} \stackrel{\beta,\infty}{\longrightarrow} [\![\bar{F}']\!]^{\mathrm{r}}$. Thus $[\![F]\!]^{\mathrm{r}} = [\![\bar{F}]\!]^{\mathrm{r}} \,/\, H \stackrel{\alpha,\infty}{\longrightarrow} [\![\bar{F}']\!]^{\mathrm{r}} \,/\, H = [\![F']\!]^{\mathrm{r}}$.
     * The case $F = \bar{F}\,[\varphi]$ is similar to the previous one with β satisfying $\varphi(\beta) = \alpha$.
     * If $F = \mathrm{rec}\, X : \bar{F}$, then the transition derives from the fact that $\bar{F}\{\mathrm{rec}\, X : \bar{F} \hookrightarrow X\} \stackrel{\alpha}{\longrightarrow} F'$. From the induction hypothesis, it follows that $[\![\bar{F}\{\mathrm{rec}\, X : \bar{F} \hookrightarrow X\}]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$. Thus $[\![F]\!]^{\mathrm{r}} = \mathrm{rec}\, X : [\![\bar{F}]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$ because the unfolding of $\mathrm{rec}\, X : [\![\bar{F}]\!]^{\mathrm{r}}$, i.e., $[\![\bar{F}]\!]^{\mathrm{r}}\{\mathrm{rec}\, X : [\![\bar{F}]\!]^{\mathrm{r}} \hookrightarrow X\}$, is equal to $[\![\bar{F}\{\mathrm{rec}\, X : \bar{F} \hookrightarrow X\}]\!]^{\mathrm{r}}$ by virtue of Lemma 6.2.

   $\Longleftarrow$) Assuming that $[\![F]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$, the proof that $F \stackrel{\alpha}{\longrightarrow} F'$ is similar to the previous one, in the sense that it proceeds by induction on the number of operational semantic rules of Table 3 that have been applied in order to derive the transition $[\![F]\!]^{\mathrm{r}} \stackrel{\alpha,\infty}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$ and by performing a case analysis based on the syntactical structure of $[\![F]\!]^{\mathrm{r}}$.

2. Let $\lambda \in \mathbb{R}_{>0}$. The proof is divided into two parts:

   $\Longrightarrow$) Assuming that $F \stackrel{\lambda}{\leadsto} F'$, the proof that $[\![F]\!]^{\mathrm{r}} \stackrel{\tau,\lambda}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$ is similar to the one of the first part of the previous result, in the sense that it proceeds by induction on the number of operational semantic rules on the right-hand side of Table 4 that have been applied in order to derive the delay transition $F \stackrel{\lambda}{\leadsto} F'$ and by performing a case analysis based on the syntactical structure of F.

   $\Longleftarrow$) Assuming that $[\![F]\!]^{\mathrm{r}} \stackrel{\tau,\lambda}{\longrightarrow} [\![F']\!]^{\mathrm{r}}$, the proof that $F \stackrel{\lambda}{\leadsto} F'$ is similar to the one of the second part of the previous result, in the sense that it proceeds by induction on the number of operational

semantic rules of Table 3 that have been applied in order to derive the transition $[\![F]\!]^{\rm r} \stackrel{\tau,\lambda}{\longrightarrow} [\![F']\!]^{\rm r}$ and by performing a case analysis based on the syntactical structure of $[\![F]\!]^{\rm r}$. ■

Theorem 6.4. Let $F_1, F_2 \in \mathbb{P}_{\rm IML}$. Then:

$$F_1 \sim_{\rm IML} F_2 \iff [\![F_1]\!]^{\rm r} \sim'_{\rm MTIPP} [\![F_2]\!]^{\rm r}$$

Proof The proof is divided into two parts:

$\Longrightarrow$) Consider the relation $\mathcal{B}^{\rm MTIPP}_{\rm IML} = \mathcal{B}_1 \cup \mathcal{B}_2$ over $\mathbb{P}'_{\rm MTIPP}$ where:

$$\begin{array}{rcl} \mathcal{B}_1 & = & \{([\![F_1]\!]^{\rm r}, [\![F_2]\!]^{\rm r}) \mid F_1, F_2 \in \mathbb{P}_{\rm IML} \wedge F_1 \sim_{\rm IML} F_2\} \\ \mathcal{B}_2 & = & \{(G, G) \mid G \in \mathbb{P}'_{\rm MTIPP} \text{ containing } <\alpha, \lambda> \text{ with } \alpha \neq \tau\} \end{array}$$

Relation $\mathcal{B}^{\rm MTIPP}_{\rm IML}$ is an equivalence relation because so is $\sim_{\rm IML}$. Moreover, it turns out to be an MTIPP strong Markovian bisimulation, as we now show. If we take $([\![F_1]\!]^{\rm r}, [\![F_2]\!]^{\rm r}) \in \mathcal{B}_1$ and we consider $\mathcal{C} \in \mathbb{P}'_{\rm MTIPP}/\mathcal{B}^{\rm MTIPP}_{\rm IML}$, there are two cases:

- If $\mathcal{C}$ is originated from $\mathcal{B}_2$, then for all $\alpha \in Act$ we have that:
$$\begin{array}{rcl} rate_{\rm a}([\![F_1]\!]^{\rm r}, \alpha, \mathcal{C}, {\rm exp}) & = & 0 \;=\; rate_{\rm a}([\![F_2]\!]^{\rm r}, \alpha, \mathcal{C}, {\rm exp}) \\ rate_{\rm a}([\![F_1]\!]^{\rm r}, \alpha, \mathcal{C}, {\rm imm}) & = & 0 \;=\; rate_{\rm a}([\![F_2]\!]^{\rm r}, \alpha, \mathcal{C}, {\rm imm}) \end{array}$$
- If $\mathcal{C}$ is originated from $\mathcal{B}_1$ when considering the equivalence class $[F']_{\sim_{\rm IML}}$, then by virtue of Lemma 6.3 and $F_1 \sim_{\rm IML} F_2$ we have that:
$$rate_{\rm a}([\![F_1]\!]^{\rm r}, \alpha, \mathcal{C}, {\rm exp}) \;=\; 0 \;=\; rate_{\rm a}([\![F_2]\!]^{\rm r}, \alpha, \mathcal{C}, {\rm exp})$$
for $\alpha \neq \tau$ and:
$$rate_{\rm a}([\![F_1]\!]^{\rm r}, \tau, \mathcal{C}, {\rm exp}) \;=\; rate(F_1, [F']_{\sim_{\rm IML}}) \;=\; rate(F_2, [F']_{\sim_{\rm IML}}) \;=\; rate_{\rm a}([\![F_2]\!]^{\rm r}, \tau, \mathcal{C}, {\rm exp})$$
Moreover, given $\alpha \in Act$, from Lemma 6.3 it follows that:
$$rate_{\rm a}([\![F_1]\!]^{\rm r}, \alpha, \mathcal{C}, {\rm imm}) \;=\; rate_{\rm a}([\![F_2]\!]^{\rm r}, \alpha, \mathcal{C}, {\rm imm})$$
iff either both F_1 and F_2 have an action transition labeled with α to a process in $[F']_{\sim_{\rm IML}}$, or neither of them has, which is indeed the case because $F_1 \sim_{\rm IML} F_2$.

$\Longleftarrow$) Consider the following relation over $\mathbb{P}_{\rm IML}$:

$$\mathcal{B}^{\rm IML}_{\rm MTIPP} \;=\; \{(F_1, F_2) \mid F_1, F_2 \in \mathbb{P}_{\rm IML} \wedge [\![F_1]\!]^{\rm r} \sim'_{\rm MTIPP} [\![F_2]\!]^{\rm r}\}$$

Relation $\mathcal{B}^{\rm IML}_{\rm MTIPP}$ is an equivalence relation because so is $\sim'_{\rm MTIPP}$. Moreover, it turns out to be an IML strong Markovian bisimulation, as we now show. If we take $(F_1, F_2) \in \mathcal{B}^{\rm IML}_{\rm MTIPP}$ and we consider $\mathcal{C} \in \mathbb{P}_{\rm IML}/\mathcal{B}^{\rm IML}_{\rm MTIPP}$, we have that:

- If $F_1 \stackrel{\alpha}{\longrightarrow} F'_1$ for some $\alpha \in Act$ and $F'_1 \in \mathbb{P}_{\rm IML}$, then $[\![F_1]\!]^{\rm r} \stackrel{\alpha,\infty}{\longrightarrow} [\![F'_1]\!]^{\rm r}$ by virtue of Lemma 6.3. From $[\![F_1]\!]^{\rm r} \sim'_{\rm MTIPP} [\![F_2]\!]^{\rm r}$, it follows that $[\![F_2]\!]^{\rm r} \stackrel{\alpha,\infty}{\longrightarrow} [\![F'_2]\!]^{\rm r}$ for some $F'_2 \in \mathbb{P}_{\rm IML}$ such that $[\![F'_1]\!]^{\rm r} \sim'_{\rm MTIPP} [\![F'_2]\!]^{\rm r}$. As a consequence $F_2 \stackrel{\alpha}{\longrightarrow} F'_2$ by virtue of Lemma 6.3, with $(F'_1, F'_2) \in \mathcal{B}^{\rm IML}_{\rm MTIPP}$.
- If F_1 and F_2 have no action transitions, then by virtue of Lemma 6.3 and $[\![F_1]\!]^{\rm r} \sim'_{\rm MTIPP} [\![F_2]\!]^{\rm r}$ we have that:
$$rate(F_1, \mathcal{C}) \;=\; rate_{\rm a}([\![F_1]\!]^{\rm r}, \tau, [G']_{\sim'_{\rm MTIPP}}, {\rm exp}) \;=\; rate_{\rm a}([\![F_2]\!]^{\rm r}, \tau, [G']_{\sim'_{\rm MTIPP}}, {\rm exp}) \;=\; rate(F_2, \mathcal{C})$$
where $[G']_{\sim'_{\rm MTIPP}}$ is the equivalence class originating $\mathcal{C}$. ■

7. Variants for Lazy Actions and Maximal Progress

In the last four sections, we have dealt with encodings based on the assumption that action execution is urgent. In this section, we consider the case in which actions are *lazy*, in the sense that the beginning of their execution can be arbitrarily delayed, and the case in which *maximal progress* is enforced, meaning that only internal actions must be performed as soon as they get enabled. In both cases, we discuss whether and how each of the four encodings changes together with the corresponding full abstraction result.

7.1. Deterministic Time

A lazy variant of TCCS was introduced in [30] by replacing the stopped process $\mathbf{0}$ with the inactive process $\underline{\mathbf{0}}$ that lets any amount of time pass according to the following additional rule:

$$\underline{\mathbf{0}} \stackrel{t}{\longrightarrow\!\!\!\!\!\sim} \underline{\mathbf{0}}$$

and, most importantly, by defining a further rule for delaying action prefix as follows:

$$\alpha \,.\, P \stackrel{t}{\longrightarrow\!\!\!\!\!\sim} \alpha \,.\, P$$

with $t \in \mathbb{N}_{>0}$ in both selfloop rules. We denote by $\mathcal{P}_{\mathrm{TCCS,L}}$ the resulting language.

A lazy variant of CIPA was introduced in [13]. This variant allows durational actions to start their execution at any time instant after the one indicated by the local clocks of the subprocesses participating in the action execution. However, this is not the only modification. Since in CIPA actions are eager, the global clock implicitly associated with a process is guaranteed to be equal to the minimum value among those of the local clocks explicitly associated with the various subprocesses. In lazy CIPA, instead, this is not necessarily the case, so we have to make sure that no durational action starts its execution at a time instant before the minimum of the global clock and the local clocks of the subprocesses participating in the action execution. As a consequence, $\mathcal{KP}$ is replaced with the set $\mathcal{KP}_{\mathrm{gc}}$ of augmented process terms in which the global clock $g \in \mathbb{N}$ is made explicit through the following syntax:

$$K \rhd g$$

and the operational and bisimulation semantics are extended accordingly, with the rules of Table 2 for durational action prefix and waiting prefix becoming:

$$\frac{k' \geq \max(k, g)}{(k \Rightarrow a \,.\, Q) \rhd g \stackrel{a@k'}{\underset{\Delta(a)}{\longrightarrow}} ((k' + \Delta(a)) \Rightarrow Q) \rhd k'} \qquad \frac{k' \geq \max(k, g)}{(k \Rightarrow \mathrm{wait}\, t \,.\, Q) \rhd g \stackrel{\tau@k'}{\underset{t}{\longrightarrow}} ((k' + t) \Rightarrow Q) \rhd k'}$$

We denote by $\mathcal{P}_{\mathrm{CIPA,L}}$ the resulting language and by $\sim_{\mathrm{CIPA,L}}$ and $\approx_{\mathrm{CIPA,L}}$ the resulting bisimilarities.

The direct encoding $[\![\_]\!]_{\mathrm{L}} : \mathcal{P}_{\mathrm{CIPA,L}} \to \mathcal{P}_{\mathrm{TCCS,L}}$ from lazy CIPA to lazy TCCS is the same as the one of Def. 3.1 from eager CIPA to eager TCCS, with the difference that nil is directly translated into $\underline{\mathbf{0}}$. It preserves strong timed bisimilarity without the limitation to restriction-free processes of Thm. 3.2, because in TCCS lazy actions let time advance also when occurring in a restriction set.

Theorem 7.1. Let $Q_1, Q_2 \in \mathbb{P}_{\mathrm{CIPA,L}}$. Then:

$$Q_1 \sim_{\mathrm{CIPA,L}} Q_2 \iff [\![Q_1]\!]_{\mathrm{L}} \sim_{\mathrm{TCCS}} [\![Q_2]\!]_{\mathrm{L}}$$

Proof See [12]. ∎

The reverse encoding $[\![\_]\!]^{\mathrm{r}}_{\mathrm{L}} : \mathcal{P}_{\mathrm{TCCS,L}} \to \mathcal{P}'_{\mathrm{CIPA,L}}$ from lazy TCCS to modified lazy CIPA is the same as the one of Def. 4.1 from modified eager TCCS to modified eager CIPA. It preserves weak timed bisimilarity without the limitation to restriction-free processes of Thm. 4.8 for the same reason explained before Thm. 7.1. Moreover, we observe that the other constraint of Thm. 4.8, i.e., delay-choice freeness, boils down to the absence of occurrences of the alternative composition operator, as in TCCS any lazy action can let time advance and hence cannot prevent delay transitions from being executed.

Theorem 7.2. Let $P_1, P_2 \in \mathbb{P}_{\mathrm{TCCS,L}}$ be delay-choice free. Then:

$$P_1 \approx_{\mathrm{TCCS}} P_2 \iff [\![P_1]\!]^{\mathrm{r}}_{\mathrm{L}} \approx_{\mathrm{CIPA,L}} [\![P_2]\!]^{\mathrm{r}}_{\mathrm{L}}$$

Proof Similar to the proof of Thm. 4.8. ∎

To obtain a maximal-progress variant of TCCS, we have to add the same two selfloop rules introduced for lazy TCCS, with the premise $\alpha \neq \tau$ in the second one. Moreover, we have to enforce the eagerness of τ-actions arising from the synchronization of a visible action with its coaction. This is accomplished by modifying the delay transition rule for parallel composition of Table 1 as follows:

$$\frac{\begin{array}{c} P_1 \stackrel{t}{\longrightarrow\!\!\!\!\!\sim} P'_1 \quad P_2 \stackrel{t}{\longrightarrow\!\!\!\!\!\sim} P'_2 \quad \neg(P_1 \stackrel{a}{\longrightarrow} \hat{P}_1 \wedge P_2 \stackrel{\bar{a}}{\longrightarrow} \hat{P}_2) \\ \neg(P_1 \stackrel{t'}{\longrightarrow\!\!\!\!\!\sim} \tilde{P}_1 \wedge P_2 \stackrel{t'}{\longrightarrow\!\!\!\!\!\sim} \tilde{P}_2 \wedge \tilde{P}_1 \stackrel{b}{\longrightarrow} \tilde{P}'_1 \wedge \tilde{P}_2 \stackrel{\bar{b}}{\longrightarrow} \tilde{P}'_2 \wedge t' < t) \end{array}}{P_1 \mid P_2 \stackrel{t}{\longrightarrow\!\!\!\!\!\sim} P'_1 \mid P'_2}$$

We denote by $\mathcal{P}_{\mathrm{TCCS,MP}}$ the resulting language.

Likewise, in a maximal-progress variant of CIPA, an explicit global clock is necessary as in lazy CIPA, with the premise of the rule for waiting prefix becoming $k' = \max(k, g)$. Moreover, we have to enforce the eagerness of τ-actions arising from the synchronization of a visible action with its coaction. This is accomplished by modifying the rule for synchronization of lazy CIPA (an extension of the one of Table 2) as follows:

$$\frac{K_1 \rhd g \xrightarrow[\Delta(a)]{a@k} K_1' \rhd g' \quad K_2 \rhd g \xrightarrow[\Delta(a)]{\bar{a}@k} K_2' \rhd g' \quad \neg(K_1 \rhd g \xrightarrow[\Delta(b)]{b@k'} \tilde{K}_1 \rhd \tilde{g} \ \wedge\ K_2 \rhd g \xrightarrow[\Delta(b)]{\bar{b}@k'} \tilde{K}_2 \rhd \tilde{g} \ \wedge\ k' < k)}{(K_1 \mid K_2) \rhd g \xrightarrow[\Delta(a)]{\tau@k} (K_1' \mid K_2') \rhd g'}$$

We denote by $\mathcal{P}_{\mathrm{CIPA,MP}}$ the resulting language and by $\sim_{\mathrm{CIPA,MP}}$ and $\approx_{\mathrm{CIPA,MP}}$ the resulting bisimilarities.

The direct encoding $[\![\_]\!]_{\mathrm{MP}} : \mathcal{P}_{\mathrm{CIPA,MP}} \to \mathcal{P}_{\mathrm{TCCS,MP}}$ from maximal-progress CIPA to maximal-progress TCCS is the same as the one for the lazy case. It preserves strong timed bisimilarity without the limitation to restriction-free processes of Thm. 3.2, because the only eager action, τ, cannot occur in a restriction set.

Theorem 7.3. Let $Q_1, Q_2 \in \mathbb{P}_{\mathrm{CIPA,MP}}$. Then:

$$Q_1 \sim_{\mathrm{CIPA,MP}} Q_2 \iff [\![Q_1]\!]_{\mathrm{MP}} \sim_{\mathrm{TCCS}} [\![Q_2]\!]_{\mathrm{MP}}$$

Proof Similar to the proof of Thm. 7.1. ∎

The reverse encoding $[\![\_]\!]^{\mathrm{r}}_{\mathrm{MP}} : \mathcal{P}_{\mathrm{TCCS,MP}} \to \mathcal{P}'_{\mathrm{CIPA,MP}}$ from maximal-progress TCCS to modified maximal-progress CIPA is the same as the one of Def. 4.1 from modified eager TCCS to modified eager CIPA. It preserves weak timed bisimilarity without the limitation to restriction-free processes of Thm. 4.8 for the same reason explained before Thm. 7.3. Moreover, we observe that the other constraint of Thm. 4.8, i.e., delay-choice freeness, implies that the only occurrences of the alternative composition operator admitted in a TCCS process are those in which each subprocess operand can only perform τ-transitions.

Theorem 7.4. Let $P_1, P_2 \in \mathbb{P}_{\mathrm{TCCS,MP}}$ be delay-choice free. Then:

$$P_1 \approx_{\mathrm{TCCS}} P_2 \iff [\![P_1]\!]^{\mathrm{r}}_{\mathrm{MP}} \approx_{\mathrm{CIPA,MP}} [\![P_2]\!]^{\mathrm{r}}_{\mathrm{MP}}$$

Proof Similar to the proof of Thm. 7.2. ∎

7.2. Stochastic Time

Unlike TCCS, introducing a selfloop rule for delaying action prefix is not appropriate in IML. The reason is that stochastic time resolves choices due to the adoption of the race policy, hence additional delay transitions may interfere with the resolution of nondeterministic choices among actions. To achieve action laziness in IML, we have to change the bisimulation semantics, not the operational semantics. This is accomplished by modifying the second clause of Def. 2.10 in such a way that the exit rate equality check is always performed, so that no delay transition is neglected in the bisimulation game. We denote by $\sim_{\mathrm{IML,L}}$ the resulting bisimilarity.

The operational semantic rules of Table 3 also give rise to a lazy variant of MTIPP. This stems from the possibility of delaying the beginning of action execution inherent to the memoryless property of exponentially distributed durations. Indeed, if an exponentially timed action does not finish its execution within t time units, the residual execution time has the same distribution as the whole action duration, hence the beginning of the execution of the action can be thought of as being delayed by t time units with respect to the instant in which the action has become enabled.

The direct encoding $[\![\_]\!]_{\mathrm{L}} : \mathcal{P}_{\mathrm{MTIPP}} \to \mathcal{P}_{\mathrm{IML}}$ from lazy MTIPP to lazy IML is the same as the one of Def. 5.1 from eager MTIPP to eager IML. It preserves strong Markovian bisimilarity only for sequential processes, i.e., processes with no occurrences of the parallel composition operator, which is a limitation stronger than the synchronization freeness of Thm. 5.5.

As an example, given $G_1 = <\alpha, \lambda> . \underline{0} \parallel_{\emptyset} <\beta, \mu> . \underline{0}$ and $G_2 = <\alpha, \lambda> . <\beta, \mu> . \underline{0} + <\beta, \mu> . <\alpha, \lambda> . \underline{0}$, we have that $G_1 \sim_{\mathrm{MTIPP}} G_2$ thanks to the memoryless property of exponential distributions, but $[\![G_1]\!]_{\mathrm{L}} \not\sim_{\mathrm{IML,L}} [\![G_2]\!]_{\mathrm{L}}$ because the former contains states having both action transitions and delay transitions

– with both kinds of transition being considered in the bisimulation game due to the lazy interpretation of action execution – whereas this is not the case with the latter. In other words, non-synchronizing concurrent MTIPP subprocesses give rise to an IML process in which some states may have both kinds of transition, while this is not possible with the sequential MTIPP process obtained from the interleaving of the original MTIPP subprocesses.

Theorem 7.5. Let $G_1, G_2 \in \mathbb{P}_{\rm MTIPP}$ be sequential. Then:
$$G_1 \sim_{\rm MTIPP} G_2 \iff [\![G_1]\!]_{\rm L} \sim_{\rm IML,L} [\![G_2]\!]_{\rm L}$$

Proof Similar to the proof of Thm. 5.5. In particular, we observe that, under the sequentiality assumption, which implies the absence of $\|_\emptyset$ operators, the α-transition of Lemma 5.4, mentioned as the only action transition of F, is indeed the only transition of F. ■

The reverse encoding $[\![\text{-}]\!]^{\rm r} : \mathcal{P}_{\rm IML} \to \mathcal{P}'_{\rm MTIPP}$ from lazy IML to modified lazy MTIPP is the same as the one of Def. 6.1 from eager IML to modified eager MTIPP. It preserves strong Markovian bisimilarity without the limitation to sequential processes of Thm. 7.5, because the possible presence of states having both action transitions and delay transitions in an IML process is preserved in the MTIPP process produced by the reverse encoding. Not even the limitation to synchronization-free processes of Thm. 5.5 is necessary, because the order in which instantaneous actions and exponential delays alternate in an IML process is preserved in the MTIPP process produced by the reverse encoding, hence so are possible spurious deadlocks.

Theorem 7.6. Let $F_1, F_2 \in \mathbb{P}_{\rm IML}$. Then:
$$F_1 \sim_{\rm IML,L} F_2 \iff [\![F_1]\!]^{\rm r}_{\rm L} \sim'_{\rm MTIPP} [\![F_2]\!]^{\rm r}_{\rm L}$$

Proof Similar to the proof of Thm. 6.4. ■

To obtain a maximal-progress variant of IML – thus retrieving the original version of [21] – we have to change the bisimulation semantics again. This is accomplished by modifying the second clause of Def. 2.10 in such a way that the exit rate equality check is performed only when neither the challenger F_1 nor the defender F_2 has τ-transitions. We denote by $\sim_{\rm IML,MP}$ the resulting bisimilarity.

The operational semantic rules of Table 3 produce a maximal-progress variant of MTIPP too. The reason is that maximal progress is in some sense between eagerness and laziness, and both the latter two are expressed by the considered rules.

The direct encoding $[\![\text{-}]\!]_{\rm MP} : \mathcal{P}_{\rm MTIPP} \to \mathcal{P}_{\rm IML}$ from maximal-progress MTIPP to maximal-progress IML is the same as the one of Def. 5.1 from eager MTIPP to eager IML. It preserves strong Markovian bisimilarity only for sequential processes, which is a limitation stronger than the synchronization freeness of Thm. 5.5, for the same reason exemplified before Thm. 7.5 (where $[\![G_1]\!]_{\rm MP} \sim_{\rm IML,MP} [\![G_2]\!]_{\rm MP}$ only for $\alpha = \tau = \beta$).

Theorem 7.7. Let $G_1, G_2 \in \mathbb{P}_{\rm MTIPP}$ be sequential. Then:
$$G_1 \sim_{\rm MTIPP} G_2 \iff [\![G_1]\!]_{\rm MP} \sim_{\rm IML,MP} [\![G_2]\!]_{\rm MP}$$

Proof Similar to the proof of Thm. 7.5. ■

The reverse encoding $[\![\text{-}]\!]^{\rm r} : \mathcal{P}_{\rm IML} \to \mathcal{P}'_{\rm MTIPP}$ from maximal-progress IML to modified maximal-progress MTIPP is the same as the one of Def. 6.1 from eager IML to modified eager MTIPP. It preserves strong Markovian bisimilarity over all processes for the same reasons explained before Thm. 7.6.

Theorem 7.8. Let $F_1, F_2 \in \mathbb{P}_{\rm IML}$. Then:
$$F_1 \sim_{\rm IML,MP} F_2 \iff [\![F_1]\!]^{\rm r}_{\rm MP} \sim'_{\rm MTIPP} [\![F_2]\!]^{\rm r}_{\rm MP}$$

Proof Similar to the proof of Thm. 7.6. ■

8. Conclusions

In this paper, we have performed a study of the comparative expressiveness of integrated-time process calculi, in which actions are durational, and orthogonal-time process calculi, in which actions are instantaneous and delays are expressed separately. Starting from [12], where a fully abstract encoding from the former calculi to the latter ones was provided in the deterministic-time setting, here we have addressed the opposite direction, as well as both directions in the stochastic-time setting.

In general, it is well known that orthogonal-time calculi are more expressive than integrated-time ones. The reason is that the former permit to describe untimed processes too; moreover, in the stochastic-time setting, they also provide modeling support for nondeterminism, intended as implementation/scheduling freedom or lack of information. Nevertheless, it is worth investigating their comparative expressiveness in more detail, and under different action interpretations such as eagerness, laziness, and maximal progress, to pinpoint the specific circumstances in which durational and durationless actions are interchangeable.

8.1. Summary of Results for the Deterministic-Time Setting

In this setting, it is the reverse encoding from TCCS to CIPA developed in this paper that has emphasized the higher expressive power of the former calculus with respect to the latter. While it is natural to translate a CIPA action into a TCCS action followed by a TCCS delay (in this order to comply with CIPA operational semantics), the translation of TCCS actions and delays has not been obvious. Likewise, it has been technically involved to establish a correspondence between TCCS delay transitions and CIPA transitions. It was not necessary to create this correspondence in the study of the direct encoding of [12] due to the way CIPA actions were translated.

Additionally, the reverse encoding has elicited not only the fact that timelocks are possible only in TCCS – as the direct encoding did – but also the different support to time additivity in TCCS and CIPA – witnessed by the possibility of preserving only weak timed bisimilarity – and the different influence that time has with respect to choice resolution in TCCS and CIPA – witnessed by the necessity of an additional constraint to achieve full abstraction. A fact common to both encodings is that the number of constraints needed to achieve full abstraction decreases when moving from eagerness to laziness and maximal progress:

- *Direct encoding from CIPA to TCCS.* This was provided in [12] (see Def. 3.1) together with the following full abstraction results with respect to strong timed bisimilarity:
  - *Eagerness.* Bisimilarity is preserved only over restriction-free processes (Thm. 3.2), because in TCCS the restriction operator may cause timelocks that are not possible in CIPA.
  - *Laziness.* Bisimilarity is preserved over all processes (Thm. 7.1), because in TCCS lazy actions let time advance also when occurring in a restriction set.
  - *Maximal progress.* Bisimilarity is preserved over all processes (Thm. 7.3), because the only eager action, τ, cannot occur in a restriction set.
- *Reverse encoding from TCCS to CIPA.* This is given in Def. 4.1. It has required some slight modifications to both calculi in order to address the issues raised at the end of [12]. We have shown that it cannot preserve strong timed bisimilarity due to further issues related to the fact that, while TCCS supports time additivity through its operational semantic rules, CIPA supports time additivity only for waitings through its weak bisimulation semantics. This difference did not emerge in the direct encoding of [12] because of the strict alternation of action prefixes and delay prefixes in the TCCS processes produced by the encoding itself. For the reverse encoding, we have proved the following full abstraction results with respect to *weak* timed bisimilarity:
  - *Eagerness.* Bisimilarity is preserved only over processes that are both restriction-free and delay-choice-free (Thm. 4.8). Restriction freeness is necessary because in TCCS the restriction operator may cause timelocks that are not possible in CIPA, precisely as in the direct encoding. Delay-choice freeness is also necessary because in TCCS time does not resolve choices, while in CIPA

time passing is associated with action execution and explicit waitings, hence the operational semantic rules for alternative composition of the two calculi produce transitions among which a correspondence cannot always be established. Not even this difference between TCCS and CIPA emerged in the direct encoding of [12] because of the absence of initial delay prefixes in the TCCS processes produced by the encoding itself.

– *Laziness.* Bisimilarity is preserved only over delay-choice-free processes (Thm. 7.2). As in the direct encoding, restriction freeness is no longer necessary because in TCCS lazy actions let time advance also when occurring in a restriction set. Delay-choice freeness boils down to the absence of occurrences of the alternative composition operator, as in TCCS any lazy action can let time advance and hence cannot prevent delay transitions from being executed.

– *Maximal progress.* Bisimilarity is preserved only over delay-choice-free processes (Thm. 7.4). As in the direct encoding, restriction freeness is no longer necessary because the only eager action, τ, cannot occur in a restriction set. Delay-choice freeness implies that the only occurrences of the alternative composition operator admitted in a TCCS process are those in which each subprocess operand can only perform τ-transitions.

8.2. Summary of Results for the Stochastic-Time Setting

In this setting, it is again the reverse encoding to reveal the different expressive power between IML and MTIPP. While it is natural to translate an MTIPP action into an IML delay followed by an IML action (in this order to comply with the race policy), the translation of IML actions has required the introduction of immediate actions in MTIPP. Surprisingly, unlike the deterministic-time setting, constraints are not necessary to achieve full abstraction in the reverse encoding, while they are essential in the direct encoding and become tighter when moving from eagerness to laziness and maximal progress:

- *Direct encoding from MTIPP to IML.* This is given in Def. 5.1. While the direct encoding from CIPA to TCCS translates any durational action into an instantaneous action followed by a fixed delay – as CIPA operational semantics keeps track of the time at which an action begins its execution – we have shown that the race policy imposes a different order: any durational action has to be translated into an exponential delay followed by an instantaneous action. For the direct encoding from MTIPP to IML, we have proved the following full abstraction results with respect to strong Markovian bisimilarity:

  – *Eagerness.* Bisimilarity is preserved only over synchronization-free processes (Thm. 5.5), because the IML processes produced by the encoding may have spurious deadlock states – due to the way in which exponentially timed actions must be translated – that are not possible in the original MTIPP processes. This synchronization-freeness constraint can be viewed as the dual of the restriction-freeness constraint of the direct encoding from CIPA to TCCS because, on the orthogonal-time side, the former avoids deadlocks while the latter avoids timelocks.

  – *Laziness.* Bisimilarity is preserved only over sequential processes (Thm. 7.5), which constitute a proper subset of synchronization-free processes. The reason is that non-synchronizing concurrent MTIPP subprocesses give rise to an IML process in which some states may have both action transitions and delay transitions – with the former not pre-empting the latter as actions are lazy – while this is not possible with the sequential MTIPP process obtained from the interleaving of the original MTIPP subprocesses.

  – *Maximal progress.* Bisimilarity is preserved only over sequential processes (Thm. 7.7) for the same reason explained in the previous case, with the only difference that in IML action transitions take precedence over delay transitions when the action is τ.

- *Reverse encoding from IML to MTIPP.* This is given in Def. 6.1. Unlike the reverse encoding from TCCS to CIPA, it has required some significant language modifications, but only on the integrated-time side, i.e., only to MTIPP. For the reverse encoding from IML to MTIPP, we have proved the following full abstraction results with respect to strong Markovian bisimilarity:

– *Eagerness.* Bisimilarity is preserved over all processes (Thm. 6.4). Synchronization-freeness required by the direct encoding is no longer necessary, because the order in which instantaneous actions and exponential delays alternate in an IML process is preserved in the MTIPP process produced by the reverse encoding, hence so are possible spurious deadlock states. This is different from the deterministic-time setting, where the reverse encoding still needs the restriction-freeness constraint of the direct encoding.

– *Laziness.* Bisimilarity is preserved over all processes (Thm. 7.6). Sequentiality required by the direct encoding is no longer necessary, because the possible presence of states having both action transitions and delay transitions in an IML process is preserved in the MTIPP process produced by the reverse encoding. Even synchronization-freeness is no longer necessary for the same reason explained in the previous case.

– *Maximal progress.* Bisimilarity is preserved over all processes (Thm. 7.8) for the same reasons explained in the previous case.

8.3. Language Modifications and Full Abstraction Constraints

Unlike the formulation of the direct encoding from CIPA to TCCS introduced in [12] (see Def. 3.1), the reverse encoding of Def. 4.1 can be designed only after allowing the stopped process of TCCS to let time pass as in [30] and after introducing zero durations and zero waitings in CIPA. We observe that these language modifications alter only slightly the expressiveness of the considered process languages. The full abstraction of both encodings is inevitably limited to restriction-free processes under action eagerness, meaning that synchronizations can take place but cannot be enforced. Moreover, the additional delay-choice-freeness constraint is necessary for the reverse encoding. Since time does not resolve choices in TCCS, this is not a severe limitation.

The formulation of the direct encoding from MTIPP to IML of Def. 5.1 does not require any language modification, while the reverse encoding of Def. 6.1 calls for the introduction of immediate actions in MTIPP, which significantly enhance the expressive power with a performance abstraction capability. In contrast, the full abstraction result is valid for all processes in the case of the reverse encoding, but for the direct encoding it is valid only over synchronization-free processes or sequential processes, depending on whether the action execution interpretation is eagerness or not, which is quite restrictive.

A precise formalization of the impact of the above-mentioned language modifications and full abstraction constraints on the expressiveness of the considered calculi is outside the scope of this paper.

8.4. Possible Uses of the Proposed Encodings

Defining encodings between different languages is useful not only to formalize the circumstances under which the considered languages have identical or different expressiveness. Indeed, the presence of encodings permits the interchange of concepts, modeling methodologies, and analysis techniques.

For instance, the reverse encoding from TCCS to CIPA can be exploited for a more efficient equivalence checking of TCCS processes based on the compact representation of CIPA processes developed in [12], which gives rise to finite state spaces in spite of the presence of local clocks within states. As another example, the reverse encoding from IML to MTIPP can be employed to obtain descriptions that are closer to the underlying continuous-time Markov chains, so to enhance the confidence of performance modelers familiar with those stochastic processes.

8.5. Future Work

We would like to continue the investigation started in [9] to compare in the uniform framework of ULTraS [7] the various deterministically timed and stochastically timed models and languages that have been proposed in the literature. This is not an easy task, as the differences between deterministic time and stochastic time are far more radical than those between integrated time and orthogonal time.

As mentioned in this paper, in the deterministic-time setting the resolution of choices is nondeterministic and not affected by time, while in the stochastic-time setting it can be influenced by time, in which case it is

probabilistic. Moreover, we have seen that in the deterministic-time setting laws such as time determinism and time additivity are usually enforced, and that the different action interpretations are implemented through the operational semantic rules. In the stochastic-time setting, instead, there are no analogous laws. Moreover, the different action interpretations are implemented through the bisimulation semantics in the orthogonal-time case, while the distinction among them is blurred by the memoryless property of exponential distributions in the integrated-time case.

Acknowledgment: We are very grateful to the anonymous reviewers for their comments and suggestions. This work has been funded by MIUR-PRIN project CINA.

References

[1] L. Aceto and D. Murphy. Timing and causality in process algebra. *Acta Informatica*, 33:317–350, 1996.
[2] J.C.M. Baeten and J.A. Bergstra. Real time process algebra. *Formal Aspects of Computing*, 3:142–188, 1991.
[3] M. Bernardo. On the expressiveness of Markovian process calculi with durational and durationless actions. In *Proc. of the 1st Int. Symp. on Games, Automata, Logics and Formal Verification (GANDALF 2010)*, volume 25 of *EPTCS*, pages 199–213, 2010.
[4] M. Bernardo. On the tradeoff between compositionality and exactness in weak bisimilarity for integrated-time Markovian process calculi. *Theoretical Computer Science*, 563:99–143, 2015.
[5] M. Bernardo and M. Bravetti. Performance measure sensitive congruences for Markovian process algebras. *Theoretical Computer Science*, 290:117–160, 2003.
[6] M. Bernardo, F. Corradini, and L. Tesei. Timed process calculi: From durationless actions to durational ones. In *Proc. of the 15th Italian Conf. on Theoretical Computer Science (ICTCS 2014)*, volume 1231, pages 21–32. CEUR-WS, 2014.
[7] M. Bernardo, R. De Nicola, and M. Loreti. A uniform framework for modeling nondeterministic, probabilistic, stochastic, or mixed processes and their behavioral equivalences. *Information and Computation*, 225:29–82, 2013.
[8] M. Bernardo and R. Gorrieri. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theoretical Computer Science*, 202:1–54, 1998.
[9] M. Bernardo and L. Tesei. Encoding timed models as uniform labeled transition systems. In *Proc. of the 10th European Performance Engineering Workshop (EPEW 2013)*, volume 8168 of *LNCS*, pages 104–118. Springer, 2013.
[10] T. Bolognesi and F. Lucidi. LOTOS-like process algebras with urgent or timed interactions. In *Proc. of the 4th Int. Conf. on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE 1991)*, volume C-2 of *IFIP Transactions*, pages 249–264, 1991.
[11] P. Buchholz. Markovian process algebra: Composition and equivalence. In *Proc. of the 2nd Int. Workshop on Process Algebra and Performance Modelling (PAPM 1994)*, pages 11–30. University of Erlangen, Technical Report 27-4, 1994.
[12] F. Corradini. Absolute versus relative time in process algebras. *Information and Computation*, 156:122–172, 2000.
[13] F. Corradini and M. Pistore. Specification and verification of timed lazy systems. In *Proc. of the 21st Int. Symp. on Mathematical Foundations of Computer Science (MFCS 1996)*, volume 1113 of *LNCS*, pages 279–290. Springer, 1996.
[14] F. Corradini, W. Vogler, and L. Jenner. Comparing the worst-case efficiency of asynchronous systems with PAFAS. *Acta Informatica*, 38:735–792, 2002.
[15] R. De Nicola, D. Latella, M. Loreti, and M. Massink. A uniform definition of stochastic process calculi. *ACM Computing Surveys*, 46(1:5):1–35, 2013.
[16] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the ACM*, 43:555–600, 1996.
[17] R. Gorrieri, M. Roccetti, and E. Stancampiano. A theory of processes with durational actions. *Theoretical Computer Science*, 140:73–94, 1995.
[18] N. Götz, U. Herzog, and M. Rettelbach. Multiprocessor and distributed systems design: The integration of functional specification and performance analysis using stochastic process algebras. In *Proc. of the 16th Int. Symp. on Computer Performance Modelling, Measurement and Evaluation (PERFORMANCE 1993)*, volume 729 of *LNCS*, pages 121–146. Springer, 1993.
[19] H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *Proc. of the 11th IEEE Real-Time Systems Symp. (RTSS 1990)*, pages 278–287. IEEE-CS Press, 1990.
[20] M. Hennessy and T. Regan. A process algebra for timed systems. *Information and Computation*, 117:221–239, 1995.
[21] H. Hermanns. *Interactive Markov Chains*. Springer, 2002. Volume 2428 of LNCS.
[22] H. Hermanns and M. Rettelbach. Syntax, semantics, equivalences, and axioms for MTIPP. In *Proc. of the 2nd Int. Workshop on Process Algebra and Performance Modelling (PAPM 1994)*, pages 71–87. University of Erlangen, Technical Report 27-4, 1994.
[23] H. Hermanns, M. Rettelbach, and T. Weiss. Formal characterisation of immediate actions in SPA with nondeterministic branching. *Computer Journal*, 38:530–541, 1995.
[24] J. Hillston. The nature of synchronisation. In *Proc. of the 2nd Int. Workshop on Process Algebra and Performance Modelling (PAPM 1994)*, pages 51–70. University of Erlangen, Technical Report 27-4, 1994.
[25] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
[26] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[27] K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.
[28] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
[29] F. Moller and C. Tofts. A temporal calculus of communicating systems. In *Proc. of the 1st Int. Conf. on Concurrency Theory (CONCUR 1990)*, volume 458 of *LNCS*, pages 401–415. Springer, 1990.
[30] F. Moller and C. Tofts. Behavioural abstraction in TCCS. In *Proc. of the 19th Int. Coll. on Automata, Languages and Programming (ICALP 1992)*, volume 623 of *LNCS*, pages 559–570. Springer, 1992.
[31] X. Nicollin and J. Sifakis. An overview and synthesis on timed process algebras. In *Proc. of the REX Workshop on Real Time: Theory in Practice*, volume 600 of *LNCS*, pages 526–548. Springer, 1991.
[32] X. Nicollin and J. Sifakis. The algebra of timed processes ATP: Theory and application. *Information and Computation*, 114:131–178, 1994.
[33] C. Priami. Stochastic π-calculus. *Computer Journal*, 38:578–589, 1995.
[34] J. Quemada, D. de Frutos, and A. Azcorra. TIC: A timed calculus. *Formal Aspects of Computing*, 5:224–252, 1993.
[35] G.M. Reed and A.W. Roscoe. A timed model for communicating sequential processes. *Theoretical Computer Science*, 58:249–261, 1988.
[36] E.W. Stark, R. Cleaveland, and S.A. Smolka. A process-algebraic language for probabilistic I/O automata. In *Proc. of the 14th Int. Conf. on Concurrency Theory (CONCUR 2003)*, volume 2761 of *LNCS*, pages 189–203. Springer, 2003.
[37] W.J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.
[38] I. Ulidowski and S. Yuen. Extending process languages with time. In *Proc. of the 6th Int. Conf. on Algebraic Methodology and Software Technology (AMAST 1997)*, volume 1349 of *LNCS*, pages 524–538. Springer, 1997.
[39] Wang Yi. CCS + time = an interleaving model for real time systems. In *Proc. of the 18th Int. Coll. on Automata, Languages and Programming (ICALP 1991)*, volume 510 of *LNCS*, pages 217–228. Springer, 1991.