DOI 10.1007/s00165-012-0230-y
The Author(s) © 2012. This article is published with open access at Springerlink.com
Formal Aspects of Computing (2012) 24: 497–518

Formal Aspects
of Computing

Reconciling real and stochastic time: the need for probabilistic refinement

J. Markovski[1], P. R. D'Argenio[2], J. C. M. Baeten[1,3] and E. P. de Vink[1,3]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands. E-mail: j.markovski@tue.nl
[2] FaMAF, Universidad Nacional de Córdoba, Córdoba, Argentina
[3] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Abstract. We conservatively extend an ACP-style discrete-time process theory with discrete stochastic delays. The semantics of the timed delays relies on time additivity and time determinism, which are properties that enable us to merge subsequent timed delays and to impose their synchronous expiration. Stochastic delays, however, interact with respect to a so-called race condition that determines the set of delays that expire first, which is guided by an (implicit) probabilistic choice. The race condition precludes the property of time additivity as the merger of stochastic delays alters this probabilistic behavior. To this end, we resolve the race condition using conditionally-distributed unit delays. We give a sound and ground-complete axiomatization of the process theory comprising the standard set of ACP-style operators. In this generalized setting, the alternative composition is no longer associative, so we have to resort to special normal forms that explicitly resolve the underlying race condition. Our treatment succeeds in the initial challenge to conservatively extend standard time with stochastic time. However, the 'dissection' of the stochastic delays to conditionally-distributed unit delays comes at a price, as we can no longer relate the resolved race condition to the original stochastic delays. We seek a solution in the field of probabilistic refinements that enable the interchange of probabilistic and nondeterministic choices.

Keywords: Timed and stochastic process algebras, Conservative extensions, race condition, Generally-distributed stochastic delays

1. Introduction

Like other approaches in concurrency theory, process algebra originally focused on qualitative aspects of systems, and gradually included quantitative aspects like time, probabilities, Markovian (exponential), and generally-distributed stochastic delays [Bae05]. In this paper, we consider timed and stochastic extensions of process algebras. It appears that these extensions have been introduced in separate, almost incompatible ways. Stochastic delays are a generalization of timed delays, as their duration is obtained by sampling from some probability distribution. Thus, one would expect that by taking Dirac point distributions, that assign probability one to a specific duration, one could either extend timed process theories with stochastic ones, or, alternatively, embed timed delays in stochastic process theories. However, both the former and the latter turned out to be nontrivial problems for bisimulation behavioral relations, sending us on a long journey [MV06, MV07, MV08, MV09]. It seems that the probabilistic nature of stochastic time imposes a semantics that is incompatible with the semantics of timed

Correspondence and offprint requests to: J. Markovski, E-mail: j.markovski@tue.nl

Supported by the research programme ProThOS, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO) under grant no. 600.065.120.11N124.

process theories. To reconcile these two notions a specific probabilistic refinement is needed. We elaborate these issues inspired by recent research on probabilistic refinement that may help us to bring our reconciliation effort home.

Timed process algebras Timed features were introduced in process algebras to model time-critical systems and to capture essential temporal behavior. The temporal aspects were added by *conservatively extending* some existing standard process theory. The extensions are conservative because they do not introduce any new equalities or behavior when restricted to the untimed part of the theories. For an overview and a generic approach to extensions with time, we refer to [NS92, Yi91]. We develop ACP-style process algebras, of which the most prominent timed versions are compiled in [BM02].

Time can be introduced in several ways. The time domain can be discrete or continuous, depending on the support set. Then, the timing itself can be relative, which is typically introduced by timed delay prefixes that give the duration of the timed delay, or timing can be absolute, realized as time-stamped actions. The behavioral equivalence usually requires that equivalent processes allow passage of time of equal duration. We choose for discrete relative timing in the form of timed delay prefix operators that induce timed transitions.

From a process-theoretical point of view, the identifying features of the timed process theories are *time determinism* and *time additivity*. Time determinism states that passage of time does not decide a choice by itself. As a consequence, timed prefixes and timed transitions are merged in the alternative and parallel composition. Time additivity allows subsequent timed delays to be merged together and to form an accumulative delay. This supports the intuition that passage of time by itself cannot be observed, so timed delays can be ‘dissected’ to suit our needs. Of interest is also the treatment of maximal progress, i.e., the priority of undelayable action transitions, that do not allow passage of time, over timed transitions.

In ACP-style process algebras, a weak nondeterministic choice in the alternative composition between undelayable actions and passage of time is assumed, similar to the choice between action transitions [BM02, BBR10]. The underlying intuition is that future alternatives should not be disabled by default, unless explicitly desired. In the latter case, this is accomplished by a maximal progress operator that disables passage of time in the presence of outgoing prioritized labeled transitions. It is also practice to derive composite notions, instead of introducing them as separate constructs. For example, the delayable action prefix that either delays indefinitely long or performs an undelayable action transition is derived by combining undelayable action and timed delay prefixes. Thus, we build higher-order constructs by relying on primitive operations, which also validates the choice of the primitives.

Stochastic extensions To support the combined modeling of functionality and performance of a system in a compositional manner, stochastic process algebras emerged. In stochastic process algebras, timed delays are replaced by stochastic delays that sample their duration from a probabilistic distribution. First came the Markovian process algebras, like TIPP, EMPA, PEPA, IMC [HMR94, BG98, Hil96, Her02] which exploited the memoryless property of the exponential distribution to give clear semantics and support compositional modeling. The memoryless property is a unique property of the geometrical and the exponential (or Markovian) distributions, which states that for Markovian delays, observed passage of time does not alter the distribution of the remainder of the delay up to its expiration. This enables interpolation of Markovian delays by employing interleaving [Her02]. In any case, the need for general distributions followed quickly as exponential delays are not efficient for the modeling of deterministic delays or high-variance heavy-tail distributions, e.g., the fixed timeouts of the Internet protocols or the distributions of the delays in media streaming services.

Prominent stochastic process algebras with generally-distributed delays include TIPP, GSMPA, SPADES, IGSMP, GSES, NMSPA, and MODEST [HMR94, BBG97, DK05b, Bra02, BKLL95, LN00, BDHK06]. More can be found in the review [BD04]. Despite the greater expressiveness, compositional modeling with general distributions proved to be challenging, as the memoryless property of the Markovian delays could not be relied upon [KD01, BD04]. This implies that upon expiration of some stochastic delays, all concurrent stochastic delays have to (re)adapt their probability distributions in order to ensure correct stochastic behavior. Usually, the underlying performance model is a generalized semi-Markov process that exploits clocks to memorize foregone passage of time in order to retain to some extent the Markov property of history independence [Gly89]. Similarly, the semantics of stochastic process algebras is given using clocks that represent the stochastic delays at a symbolic level. Such a symbolic representation allows for the manipulation of finite structures, e.g., stochastic automata [DK05a] that support SPADES or extensions of generalized semi-Markov processes [Bra02] for IGSMPs. The concrete execution model is subsequently obtained by sampling the clocks, frequently yielding infinite probabilistic timed transition systems.

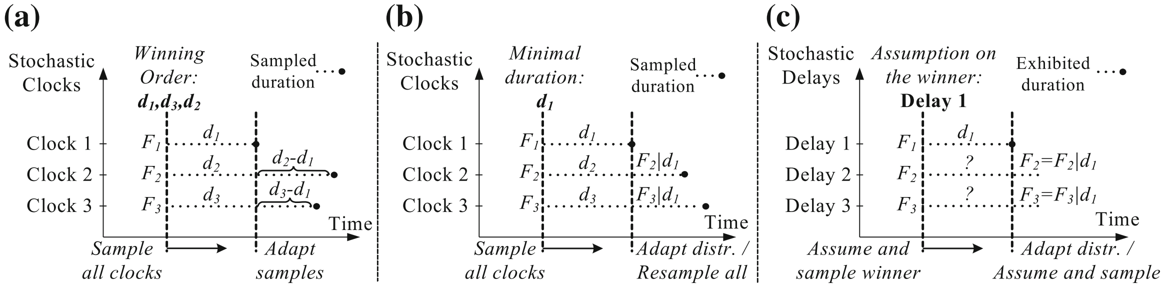

Fig. 1. a Residual lifetime semantics with clocks and **b** spent lifetime semantics with clocks. The notation $F_2|d_1$ and $F_3|d_1$ denotes that the distributions F_2 and F_3 of the clocks C_2 and C_3, respectively, have been shifted to the right by the duration d_1

Stochastic delays For the sampling of the clock two execution policies can be adopted: (1) a race condition [HMR94, DK05b, LN00, BDHK06], which enables the action transitions guarded by the clocks that expire first (the execution policy of Markov chains), and (2) pre-selection policy [BBG97, Bra02], which preselects the clocks by making a probabilistic choice (the execution policy of generalized semi-Markov processes). Notably, more execution policies have been developed for stochastic and generalized stochastic Petri nets, comprising exponential delays and immediate probabilistic choices [ABC$^+$95]. There, multiple transitions can be enabled and taken at the same time, leading to more complicated execution.

In absence of the memoryless property, the samples of the clocks must be updated after each stochastic delay transition. This is because the residual sample/distribution of the clock depends on the duration of time that the clock has been active. Again, the literature provides two techniques for doing this: (1) keeping track of the *residual lifetime* of clocks, i.e., the time that is left before the clock expires; or (2) keeping track of the *spent lifetime* of clocks, i.e., the time that the clock has been active.

The residual lifetime semantics [DK05b], depicted in Fig. 1a, supports performance analysis via discrete-event simulation, that is extensively exploited when analytical methods do not apply. However, it has been criticized for its being unfair as the outcome of the race condition is known upfront due to the early sampling of clocks. The spent lifetime semantics [HMR94, BBG97, Bra02, LN00], depicted in Fig. 1b, has been advocated for its correspondence to standard real-time, as the clocks increase as time passes. Additionally, the approach is considered fair with respect to the race condition as the clocks are first pre-sampled to statistically determine the minimal sample. Afterwards, the original samples are discarded, while the probability distributions of the remaining clocks are 'aged' with the minimal sample. However, the fairness comes at a price: re-sampling of the clocks is required after each resolution of the race condition.

An alternative, but equivalent approach taken by us is to make a probabilistic assumption on the outcomes of the race by conditioning the clocks that win the race, and, afterward, to sample from the (joint) probability distribution of the winning clocks [How71], cf. Fig. 1c. In this approach each clock is sampled only once. So, there is no need to keep track of the lifetimes of clocks. Instead, distributions have an 'age' which accounts for the time exhibited by previous samples. We refer to the samples as *stochastic delays*, resembling the notion of timed delays. In the present setting, we employ the race condition with spent-lifetime semantics. We rely on its interpretation in terms of conditional random variables, which makes a probabilistic assumption on the winning stochastic delays, i.e., the ones that expire. This is followed by conditioning the distributions of the losing delays, i.e., the ones that do not expire, on the time spent for the winning samples [How71].

In any case, one can observe that the property of time additivity is fairly difficult to preserve: extending a stochastic delay leads to it having a different distribution, imposing recalculation of the whole sampling process and, thus, ruining compositionality. Time determinism proves easier to handle as besides the initial probabilistic choice, one can easily enforce that passage of time does not disable future choices.

Motivation and contributions We focus on the interplay between real and stochastic delays that coexist in a single process theory. We investigate the possibility and means to (conservatively) extend timed process theories with stochastic delays. We also look into the possibility of extending timed delays with probabilistic features that might enable the derivation of stochastic delays, similar to delayable actions. We opt for discrete time as continuous distributions are more difficult to restrict to mimic timed delays. Moreover, extensions with discrete stochastic time are more complicated (if the probability distributions of the stochastic delays are measurable [CSKN05]) as there is non-zero probability that several delays expire simultaneously. We also touch on the problem of embedding real time in a stochastic-time setting. Finally, we consider the replacement of timed delays with stochastic ones and we examine closer the effect of such an experiment.

The relation between real and stochastic time has already been studied by others in various settings. Due to the nature of the stochastic process theories the results show an embedding or translations to a purely timed formalism. A structural translation from stochastic automata to timed automata with deadlines that preserve the timed traces is given in [D'A03]. This approach found its way into MODEST [BDHK06] as a means to introduce real and stochastic time as separate constructs in the same formalism. Also, a translation from IGSMP into pure real-time models called interactive timed automata is reported in [Bra02]. In [ABC$^+$94] a proposal of extending timed LOTOS is made by exploiting stochastic timers.

Our present work builds on earlier investigations. A preliminary effort to embed real-time in stochastic process algebras was given in [MV06]. A subsequent study led to the notion of dependent and independent race conditions that paved the way for the treatment of the expansion law and the maximal progress in the vein of timed process theories [MV07]. Equational theory that captures both real-time and stochastic-time aspects in the context of complete races was presented in [MV09]. There a decomposition of a race into disjoint events is exploited to explicitly state the outcomes of the race. Embedding of timed delays in stochastic process algebras was attempted in [MV08], where we introduced the notion of context-sensitive time interpolation (additivity), which is supposed to mimic the trivial race condition for timed delays. Here, we build mostly on [MV09, MV08] by extending the theory for incomplete races. Contrary to what we expected, this extension proved far from straightforward, taken upon in [Mar08], as then the alternative composition is no longer associative. Thus, we rely on normal forms that do preserve associativity to give an equational characterization.

Our main contribution is a stochastic process theory that conservatively extends (ACP-style) timed process theories that employ bisimulation-based semantics. Such process theories are of importance for (re)modeling and verification of (discretely-)timed systems for performance evaluation, or models stochastic systems that comprise timeouts or heavy-tail distributions, e.g., protocols for media-streaming services [BBD03]. To conservatively extend timed process algebras with stochastic delays, we introduced the concept of *racing timed delays*, which represent conditionally-distributed unit timed delays that are employed to explicitly resolve the race condition. Therefore, we enable symbolic manipulation of concurrent stochastic delays and symbolic resolution of the race condition. This is illustrated by modeling a discrete-time variant of the $G/G/1/\infty$ queue, for which we provide an explicit solution in which the race condition is resolved. The resolution of the race condition comes at a price as the supporting racing timed delays intertwine timed delays and probabilistic choices, unlike the stochastic delays, which make a probabilistic choice on the duration of the delay in the beginning, cf. Fig. 1. Unfortunately, existing research on combining nondeterministic and probabilistic choices points out that this issue cannot be satisfactorily resolved for bisimulation semantics without sacrificing compositionality [MMSS96, Low93, Seg95, Geo11]. Therefore, we outline a new challenge that states the need for a probabilistic refinement that allows reordering of probabilistic choices with respect to timed delays in order to relate racing timed and stochastic delays.

The rest of this paper is organized as follows. In Sect. 2, we elaborate on the notion of race condition, giving the necessary preliminaries. We follow with the development of the operational semantics of the process theory $\mathrm{TCP}^{\mathrm{drst}}$ in Sect. 3. In Sect. 4, we present the equational theory, relying on normal forms in which the race condition is explicitly resolved. In Sect. 5, we develop the process theory $\mathrm{DTCP}^{\mathrm{dst}}_{\mathrm{rec}}$ that comprises derived stochastic delays and delayable actions and we illustrate how to apply it to resolve the race condition by revisiting a discrete-time variant of the $G/G/1/\infty$ queue. We end with a discussion in Sect. 6 regarding the need of a probabilistic refinement that enables reordering of nondeterministic and probabilistic choices involving timed delays. Due to page limitations, we refer to [Mar08] for technical details.

2. Race condition

We provide preliminaries of the central concepts of race condition, racing context, racing timed, and stochastic delays. We distinguish two types of race conditions to accommodate for compositional modeling and manipulation of stochastic delays. One type of race condition treats delays as having independent samples, whereas the other synchronizes on delays with the same name.

We use discrete random variables to represent durations of stochastic delays. The set of discrete distribution functions F such that $\mathrm{F}(n) = 0$ for $n \leqslant 0$ is denoted by $\mathcal{F}$; the set of the corresponding random variables by $\mathcal{V}$. We use X, Y, and Z to range over $\mathcal{V}$ and F_X, F_Y, and F_Z for their respective distribution functions. Also, W, L, V, and D range over $2^{\mathcal{V}}$. By assumption, the support set $\mathrm{supp}(X) = \{n > 0 \mid \mathrm{P}(X = n) > 0\}$ of a random variable X is finite or countably infinite. We write $\mathrm{f} \colon A \leftrightarrow B$ for f a bijection. The identity bijection on the set A is denoted by id_A. We write $p \subseteq A$ for a predicate $p\colon A \to \{\text{true}, \text{false}\}$. Composition of two relations $\mathrm{r}_1 \subseteq A \times B$ and $\mathrm{r}_2 \subseteq B \times C$ is given by $\mathrm{r}_2 \circ \mathrm{r}_1 \subseteq A \times C$ where $(x, z) \in \mathrm{r}_2 \circ \mathrm{r}_1$ if there exists a $y \in B$ such that $(x, y) \in r_1$ and $(y, z) \in r_2$. We restrict and rename functions on disjoint parts of the domain by $\mathrm{g}\{\mathrm{f}_1/D_1\} \ldots \{\mathrm{f}_n/D_n\}(x) = \mathrm{f}_i(x)$ if $x \in D_i$, and $\mathrm{g}(x)$ if $x \in D \backslash (\bigcup_{i=1}^{n} D_i)$, for functions $\mathrm{g}, \mathrm{f}_1, \ldots, \mathrm{f}_n\colon A \to B$ and disjoint subsets $D_1, \ldots, D_n \subseteq A$. By $\mathcal{P}(A)$ we denote the set of (standard) discrete probabilistic spaces (A, P) over the set A with probability measure P.

Stochastic delays A stochastic delay is a timed delay of a duration that is guided by a random variable. We use the random variable as the *name* of the stochastic delay. We observe simultaneous passage of time for a number of stochastic delays until one or some of them expire. This phenomenon is referred to as *race condition* and the underlying process as *race* [How71]. For multiple racing stochastic delays, different stochastic delays may be observed simultaneously as being the shortest. We refer to the delays with the shortest duration as *winners* and to the rest as *losers*. An outcome of a race is completely determined by the pair of sets of winners W and losers L, notation $\left[\begin{smallmatrix} W \\ L \end{smallmatrix}\right]$. We write $[W]$ for $\left[\begin{smallmatrix} W \\ \emptyset \end{smallmatrix}\right]$ and omit brackets wherever possible. Thus, $[X]$ represents a stochastic delay sampled from random variable X.

Outcomes of races may be involved in other races, so we refer to an outcome $\left[\begin{smallmatrix} W \\ L \end{smallmatrix}\right]$ as the (conditional) *stochastic delay* induced by the disjoint sets of winners W and losers L. By $W < L$ we denote the event $X_1 = X_2$ for all $X_1, X_2 \in W$ and $X < Y$ for all $X \in W$, $Y \in L$ and by $W < n$ for $n \in \mathbb{N}$ we denote the event $X_1 = X_2$ for all $X_1, X_2 \in W$ and $X < n$ for all $X \in W$. Similarly, we also use $W = n$. The probability of the outcome $\left[\begin{smallmatrix} W \\ L \end{smallmatrix}\right]$ is given by $\mathrm{P}(W < L)$ and the stochastic delay is guided by the conditional random variable $\langle X \mid W < L \rangle$ for every $X \in W$. Furthermore, two conditional stochastic delays $\left[\begin{smallmatrix} W_1 \\ L_1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} W_2 \\ L_2 \end{smallmatrix}\right]$ can race against each other and form a joint outcome if it is possible to consistently combine the sets of winners and losers, i.e., in the joint outcome no winner can come from the original sets of losers L_1 or L_2.

We take a closer look at the relation between the winners and the losers of the racing delays $\left[\begin{smallmatrix} W_1 \\ L_1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} W_2 \\ L_2 \end{smallmatrix}\right]$. There are three consistent combinations possible that refine the relation between the winners and the losers: (1) $L_1 \cap W_2 \neq \emptyset$, which means that the race must be won by W_1 and lost by $L_1 \cup W_2 \cup L_2$, (2) $W_1 \cap W_2 \neq \emptyset$, which means that the race must be won by $W_1 \cup W_2$ together and lost by $L_1 \cup L_2$, and (3) $W_1 \cap L_2 \neq \emptyset$, which means that the race must be won by W_2 and lost by $W_1 \cup L_1 \cup L_2$. Obviously, these 'restrictions' need to be disjoint and do not apply simultaneously; if more than one restriction holds, then the outcomes cannot be combined. For example, if both (1) and (2) hold, then L_1 and W_2 must exhibit the same sample and also W_1 and W_2 exhibit the same sample, wrongly implying that W_1 and L_1 exhibit the same sample, so the outcomes cannot be combined. If, in addition, we have that $W_1 \cup L_1 = W_2 \cup L_2$, we say that the race is *resolved*. The extra condition ensures that the outcomes stem from the same race, i.e., they have the same racing delays. For example, $\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} Y, Z \\ X \end{smallmatrix}\right]$ cannot form a joint outcome. These delays do not stem from the same race, which renders their combination inconsistent.

Resolved races Resolved races play an important role as they enumerate every possible outcome of the race. We define a predicate $\mathrm{rr}(\left[\begin{smallmatrix} W_1 \\ L_1 \end{smallmatrix}\right], \left[\begin{smallmatrix} W_2 \\ L_2 \end{smallmatrix}\right])$ that checks whether two delays $\left[\begin{smallmatrix} W_1 \\ L_1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} W_2 \\ L_2 \end{smallmatrix}\right]$ are in a resolved race:

$$\begin{aligned} \mathrm{rr}\left(\left[\begin{smallmatrix} W_1 \\ L_1 \end{smallmatrix}\right], \left[\begin{smallmatrix} W_2 \\ L_2 \end{smallmatrix}\right]\right) \triangleq{} & W_1 \cup L_1 = W_2 \cup L_2 \wedge ((L_1 \cap W_2 \neq \emptyset \wedge W_1 \cap W_2 \neq \emptyset) \\ & \vee (L_1 \cap W_2 \neq \emptyset \wedge W_1 \cap L_2 \neq \emptyset) \vee (W_1 \cap W_2 \neq \emptyset \wedge W_1 \cap L_2 \neq \emptyset)). \end{aligned} \tag{1}$$

By $\left[\begin{smallmatrix} W \\ L \end{smallmatrix}\right].p$ we denote a process term p prefixed by a stochastic delay $\left[\begin{smallmatrix} W \\ L \end{smallmatrix}\right]$. This prefixed term denotes a process that behaves as p after $\left[\begin{smallmatrix} W \\ L \end{smallmatrix}\right]$ expires. To express a race, we employ the alternative composition, e.g., $[X].p_1 + [Y].p_2$. As discussed above, there are three possible outcomes of this race in terms of the participating stochastic delays: (1) $\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$, (2) $\left[\begin{smallmatrix} X, Y \\ \emptyset \end{smallmatrix}\right]$, and (3) $\left[\begin{smallmatrix} Y \\ X \end{smallmatrix}\right]$. The passage of time of the stochastic delay $\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$ is guided by the conditional random variable $\langle X \mid X < Y \rangle$. In this case, the stochastic delay X expires, whereas Y becomes dependent on the amount of time that has passed for X. Intuitively, this is represented by the term $\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right].(p_1 + [Y].p_2)$, where both occurrences of Y refer to the same stochastic delay, i.e., the second occurrence of Y is bound by the first one. Similarly, we have $\left[\begin{smallmatrix} Y \\ X \end{smallmatrix}\right].([X].p_1 + p_2)$, when the winner is Y. In the case when both delays win, they expire together. Hence, by the notion of time determinism [NS92, BM02], which states that passage of time by itself cannot make a choice, the resulting term should be $\left[\begin{smallmatrix} X, Y \\ \emptyset \end{smallmatrix}\right].(p_1 + p_2)$. Thus, we can also write $\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right].(p_1 + [Y].p_2) + \left[\begin{smallmatrix} X, Y \\ \emptyset \end{smallmatrix}\right].(p_1 + p_2) + \left[\begin{smallmatrix} Y \\ X \end{smallmatrix}\right].([X].p_1 + p_2)$ instead of $[X].p_1 + [Y].p_2$ as both expressions have the same final outcomes. The advantage of the former is that it explicitly states all possible outcomes of the race and that these events are disjoint. Thus, we can clearly separate the disjoint stochastic behavior of the term depending on the resolved outcomes of the race of X and Y.

Dependent and independent races Consider the term $[X].p \parallel [X].p$, where $\parallel$ denotes parallel composition. The semantics of the race condition for the parallel composition is the same as for the alternative composition. We can interpret the race between the two processes above in two ways: (1) from the standard viewpoint of Markovian/race condition semantics, the process is a composition of two independent components that are competing for the same resource [How71], and (2) from the real-time perspective this composition synchronizes the two components that exhibit the same sample as they have the same name [BM02]. The former interpretation is according to the *independent* (stochastic) race condition and it enables compositional modeling. It states that stochastic delays with the same name have the same distribution, but do not necessarily exhibit the same sample. The latter interpretation is according to the *dependent* (timed) race condition that forces racing delays with the same name to always exhibit the same duration. It supports the existence of expansion laws and it enables resolution of races. We give an illustrative example.

Example 1 The term $\left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right].p_1 + \left[\begin{smallmatrix}X\\Z\end{smallmatrix}\right].p_2$ should be equivalent to the term $\left[\begin{smallmatrix}X\\Y,Z\end{smallmatrix}\right].(p_1 + p_2)$ if X is treated as a dependent stochastic delay. Both stochastic delays have a winner guided by X, which exhibits the same sample in both terms and, therefore, the winners of both delays must exhibit passage of time together. On the other hand, if X is treated as an independent stochastic delay, then the same term is equivalent to $\left[\begin{smallmatrix}X\\Y,Z,X'\end{smallmatrix}\right].(p_1 + \left[\begin{smallmatrix}X'\\Z\end{smallmatrix}\right].p_2) + \left[\begin{smallmatrix}X,X'\\Y,Z\end{smallmatrix}\right].(p_1 + p_2) + \left[\begin{smallmatrix}X'\\X,Y,Z\end{smallmatrix}\right].(\left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right].p_1 + p_2)$ for a random variable X' satisfying $\mathrm{F}_{X'} = \mathrm{F}_X$. In the standard independent race condition interpretation, the two occurrences of X can exhibit different samples that are guided by the same distribution. Therefore, they actually represent two distinct stochastic delays and the second occurrence of X is renamed to a new stochastic delay X' with the same distribution.

Dependence scope operator We introduce a dependence scope operator $|p|_D$ for $D \subseteq \mathcal{V}$ to specify dependent and independent delays. The racing delays in the races induced by the term p that are in D are treated as dependent. The names of dependent delays are important as they identify stochastic delays that exhibit the same sample. On the contrary, the names of the independent delays play no such role and they only identify stochastic delays with the same distribution. In the previous example, $|\left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right].p_2|_X$ would denote that X is a dependent stochastic delay, but Y is an independent one. Intuitively, this term is equivalent to $|\left[\begin{smallmatrix}X\\Z\end{smallmatrix}\right].p_2|_X$, for every Z such that $\mathrm{F}_Z = \mathrm{F}_Y$, but it is not equivalent to $|\left[\begin{smallmatrix}U\\Y\end{smallmatrix}\right].p_2|_U$ for any $U \neq X$, even if $\mathrm{F}_U = \mathrm{F}_X$. Multiple scopes intersect, i.e., $||p|_{D_2}|_{D_1}$ is equivalent to $|p|_{D_1 \cap D_2}$. For example, for $X \neq Y$, $||\left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right].p|_X|_Y$ denotes a process prefixed by the independent delay, i.e. $|\left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right].p|_\emptyset$, because $\{X\} \cap \{Y\} = \emptyset$.

The dependence scope plays an important role when giving operational semantics to the terms. Recall, the stochastic delay prefix $\left[\begin{smallmatrix}W\\L\end{smallmatrix}\right].p$ denotes an outcome of a race between the stochastic delays in $W \cup L$, where the winners are given by W and the losers are given by L. Moreover, it denotes that there was passage of time for the losing delays in L that may continue to persist in p. This means that the losers do not have their original distribution in the resulting process p and that their distributions must be 'aged' by the duration of the sample exhibited by the winners W. Therefore, the names of the losing delays must be protected in p, i.e., they become dependent. This is achieved by writing $|p|_L$ as the remaining term after the expiration of the delay given by $\left[\begin{smallmatrix}W\\L\end{smallmatrix}\right]$. Thus, $\left[\begin{smallmatrix}W\\L\end{smallmatrix}\right].p$ should be equivalent to $\left[\begin{smallmatrix}W\\L\end{smallmatrix}\right].\ |p|_L$ as the names in L must be preserved in p. This also means that the stochastic delays that are not in the set of losers L become independent. To support the interpretation of process terms as discussed above, the stochastic delays that are not encompassed by any dependence scope are considered as dependent. Thus, $\left[\begin{smallmatrix}W\\L\end{smallmatrix}\right].p$ is equivalent to $|\left[\begin{smallmatrix}W\\L\end{smallmatrix}\right].p|_{W \cup L}$.

Racing timed delays When a stochastic delay loses a race, its distribution is altered conditionally on the winning sample. However, its distribution does not have to be re-adapted each time it loses a race and, instead, it is sufficient to remember its age, i.e., the expired cumulative duration while losing races.

Definition 1 A distribution function F can be aged by $m \in \mathbb{N}$ if $\mathrm{F}(m) < 1$. The resulting distribution is denoted by $\mathrm{F}|m$, where $(\mathrm{F}|m)(n) \triangleq \frac{\mathrm{F}(n+m)-\mathrm{F}(m)}{1-\mathrm{F}(m)}$.

If the condition of Definition 1 is fulfilled, then $\mathrm{F}\,|\,m$ is again a probability distribution function. Since we work with distributions satisfying $\mathrm{F}(0) = 0$, we have that $\mathrm{F}\,|\,0 = \mathrm{F}$. Furthermore, iterative application of the aging function is the same as aging the function once with the accumulated duration [MV06].

Lemma 1 *If* $(\ldots(F\,|\,d_1)\ldots)\,|\,d_n$ *is defined for* $d_1, \ldots, d_n \in \mathbb{N}$, *then* $(\ldots(F\,|\,d_1)\ldots)\,|\,d_n = F\,|\,(\sum_{i=1}^{n} d_i)$.

As a direct consequence, to compute a total age of a distribution of a stochastic delay it suffices only to compute the sum of the durations of the samples of every race that it lost. Now, let us denote by $\sigma_{\emptyset}^{X}$ the event that the delay $[X]$ expires after one time unit has passed, i.e., in race condition terminology the stochastic delay $[X]$ wins a race with a sample of one unit timed delay and there are no losers. Let us assume that the age of X is m and let us denote by $X|m = \langle\, X - m \mid X > m \,\rangle$ the conditional random variable with distribution $\mathrm{F}_X|m$. Then, the probability of the event $\sigma_{\emptyset}^{X}$ is $\mathrm{P}((X|m) = 1)$, i.e., the probability that $[X]$ expired after $m + 1$ units of time. By $\sigma_{X}^{\emptyset}$, we denote the event that the delay $[X]$ does not expire in one time unit, i.e., the stochastic delay $[X]$ loses the race to a unit timed delay and there are no additional winners. Again, by assuming that X has age m, the probability of this event is $\mathrm{P}((X|m) > 1)$, and after the expiration of the timed delay, the age of X becomes $m + 1$. Thus, at each point in time we have two possibilities: either the delay expires, or the delay does not expire and it is aged by one time unit. Then, the process $[X].p$ can be specified as the solution of the recursive equation $A = \sigma_{\emptyset}^{X}.p + \sigma_{X}^{\emptyset}.A$, for a suitable recursion variable A.

In a generalized context, we specify a stochastic delay $\genfrac{[}{]}{0pt}{}{W}{L}.p$ as the solution of the recursive equation $B = \sigma_{L}^{W}.p + \sigma_{W\cup L}^{\emptyset}.B$, where either the winners expire and the losers are aged by one time unit or all racing delays are aged one time unit. We refer to $\sigma_{L}^{W}.\_$ as a unit timed delay prefix in a racing context induced by the sets of winner W and losers L, or a racing timed delay for short. The probability of the event is given by

$$\mathrm{RC}_1(W, L) = \mathrm{P}(W = 1, L > 1), \tag{2}$$

where the racing delays in $W \cup L$ can have their own ages as discussed above for a single delay $[X]$.

Racing timed delays in resolved races We emphasize that timed delays are not stochastic delays that impose a race condition and form joint outcomes to resolve it. Instead, they allow passage of one time unit in presupposed racing contexts that can be consistently merged as shown below. In the setting of this paper, we build a process theory for timed delays in a racing context and retrieve stochastic delays via guarded recursive specifications as indicated above. The standard unit timed delay prefix is embedded in the theory as $\sigma_{\emptyset}^{\emptyset}.\_$, i.e., a timed delay in an empty racing context. We omit the empty sets from the notation when clear from the context and we also write $\sigma^{n}.\_$ for $n \geqslant 1$ subsequent timed delays prefixes $\sigma.\_$. Timed delays can also be observed in a context of resolved races. If $\mathrm{rr}(\genfrac{[}{]}{0pt}{}{W_1}{L_1}, \genfrac{[}{]}{0pt}{}{W_2}{L_2})$ holds, then $\sigma_{L_1}^{W_1}$ and $\sigma_{L_2}^{W_2}$ are in the context of the resolved race between $\genfrac{[}{]}{0pt}{}{W_1}{L_1}$ and $\genfrac{[}{]}{0pt}{}{W_2}{L_2}$. However, this does not cover the case when there are no winners in the racing context, i.e., no stochastic delays expire after one unit time step. For that purpose we overload the resolved race predicate $\mathrm{rr}(\_)$ to $\mathrm{rr}(\sigma_{L_1}^{W_1}, \sigma_{L_2}^{W_2})$ as follows:

$$\begin{aligned}\mathrm{rr}(\sigma_{L_1}^{W_1}, \sigma_{L_2}^{W_2}) \triangleq {} & (W_1 \cup L_1 = W_2 \cup L_2) \wedge ((L_1 \cap W_2 \neq \emptyset \wedge W_1 \cap W_2 \neq \emptyset) \vee \\ & (L_1 \cap W_2 \neq \emptyset \wedge W_1 \cap L_2 \neq \emptyset) \vee (W_1 \cap W_2 \neq \emptyset \wedge W_1 \cap L_2 \neq \emptyset) \vee \\ & (W_1 = \emptyset \wedge W_2 \cap L_1 \neq \emptyset) \vee (W_2 = \emptyset \wedge W_1 \cap L_2 \neq \emptyset)).\end{aligned} \tag{3}$$

Recall that the predicate $\mathrm{rr}(\_)$ defines the context in which the race between the stochastic delays $\genfrac{[}{]}{0pt}{}{W_1}{L_1}$ and $\genfrac{[}{]}{0pt}{}{W_2}{L_2}$ is resolved. The extra conditions deal with the superfluous situation for the timed delays $\sigma_{W\cup L}^{\emptyset}$ and σ_{L}^{W} where in the context of one timed delay no racing delay has yet expired, whereas in the context of the other the winners have expired, creating a disjoint event. As stochastic delays can form inconsistent races, timed delays can also have inconsistent racing contexts. However, unlike the stochastic delays, the context of the timed delay is static, i.e., the racing condition is not resolved, but only endorsed. We illustrate the situation.

Example 2 The process $\sigma^{X}.p_1 + \sigma_{X}^{Y}.p_2$ can only deadlock. The process $\sigma^{X}.p_1$ performs a unit time step after which $[X]$ expires. The process $\sigma_{X}^{Y}.p_2$ performs a unit time step after which $[Y]$ expires in a context of a race in which $[Y]$ won over $[X]$. Thus, the process allows $[X]$ to expire in one timed unit, but it also requires $[Y]$ to expire in one time unit. However, $[Y]$ should delay less than $[X]$ as implied by the racing context of σ_{X}^{Y}, which leads to an inconsistency as there is no information about $[Y]$ in context of the first timed delay.

This illustrates the main difference between stochastic delays and racing timed delays as $[X].p_1 + \genfrac{[}{]}{0pt}{}{Y}{X}.p_2$ is equivalent to $\genfrac{[}{]}{0pt}{}{Y}{X}.([X].p_1 + p_2)$, after the resolution of the race between $[X]$ and $\genfrac{[}{]}{0pt}{}{Y}{X}$. This type of dynamics will be enabled by unfolding the guarded recursive specifications that model the stochastic delays.

Design choices We model processes using probabilistic discrete-time automata that have probabilistic discrete-time transition systems as the underlying model. We note that the automata used in this setting are not related to the probabilistic extensions of timed automata, e.g., the ones used in PRISM [KNP02]. Processes have outgoing timed delay transitions and undelayable action transitions that do not allow any passage of time. The choice between several action transitions is nondeterministic and, in general, depends on the environment as in standard process algebras. The choice between timed delays is probabilistic as it is induced by the racing context of the delays. We favor time-determinism, i.e., the principle that passage of time alone cannot make a choice [Yi91, NS92, BM02]. The probabilistic choices only resolve the race condition, but do not resolve the choice in the alternative composition. Also, we adopt the weak choice between undelayable actions and passage of time, i.e., we impose a nondeterministic choice on the undelayable action transitions and the passage of time in the vein of ACP-styled timed process algebras [NS92, BM02]. To support maximal progress, i.e., to prefer undelayable action to passage of time, we include a maximal progress operator in the theory together with encapsulation of actions, thereby disabling unwanted (interleaving) action transitions. We also opt for guarded recursion introduced by means of guarded recursive specifications. We derive delayable actions as solutions of guarded recursive equations that can perform an undelayable action at any point in time. Stochastic delays are introduced in the theory using guarded recursive specifications as briefly discussed above. We believe this approach to be systematic as it builds on well-established notions. Moreover, it helps us to identify the set of primitive operators that are sufficiently expressive so that when combined they provide other more complex features in the same theory. Finally, we note that recursion can be introduced in the theory using standard techniques outlined in [BBK87, BBR10].

3. Process theory $\mathrm{TCP}^{\mathrm{drst}}$

We introduce the theory $\mathrm{TCP}^{\mathrm{drst}}(\mathcal{A}, \mathcal{V}, \gamma)$ of communicating processes with discrete real and stochastic time, where $\mathcal{A}$ denotes the set of actions, $\mathcal{V}$ denotes the set of random variables, and γ is the ACP-style commutative and associative action synchronization function [BW90]. We give operational semantics to process terms using racing timed transition schemes, probabilistic discrete-time automata in which the probabilistic choice is implicitly and symbolically stated by the racing context. The states of the automaton determine the timed transitions, whereas we use an additional construct, called *environment*, to keep track of the ages of the racing delays. The environment is denoted by a function α that holds the age of the distribution function of each racing delay. We put $\alpha \colon \mathcal{V} \to \mathbb{N}$ and we write $\mathcal{E}$ for the set of all such environments. We recall that age 0 actually means that the stochastic delay has no age, i.e., it did not lose any race until that point. We do not dwell on details of resolving naming conflicts due to page limitations. This issue is solved using alpha conversion in the vein of [DK05a] and we refer the interested reader to [Mar08]. We also presuppose two functions $\mathrm{D}(\_)$ and $\mathrm{I}(\_)$ that identify the dependent and independent racing delays, respectively. Then the complete set $\mathrm{R}(p)$ of racing delays of a process term p is given by $\mathrm{R}(p) = \mathrm{D}(p) \cup \mathrm{I}(p)$, where $\mathrm{D}(p) \cap \mathrm{I}(p) = \emptyset$.

Definition 2 (Racing timed transition scheme) A racing timed transition scheme is a tuple $(S \times \mathcal{E}, A, V, \longrightarrow, \longmapsto, \downarrow, \mathrm{I})$, where the extended state $u = \langle s, \alpha \rangle \in S \times \mathcal{E}$ represents a state s in an environment α, A is a set of actions, V is a set of random variables giving the stochastic delay names, and

- $\longrightarrow\, \subseteq (S \times \mathcal{E}) \times A \times (S \times \mathcal{E})$ is the undelayable action transition relation.
- $\longmapsto\, \subseteq (S \times \mathcal{E}) \times 2^V \times 2^V \times (S \times \mathcal{E})$ is the racing timed delay transition relation. For every timed delay transition $u \overset{W}{\underset{L}{\longmapsto}} u'$ (in infix notation) it holds that the winners and the losers are disjoint, i.e., $W \cap L = \emptyset$. Moreover, for every $u \overset{W_1}{\underset{L_1}{\longmapsto}} u_1$ and $u \overset{W_2}{\underset{L_2}{\longmapsto}} u_2$ with $u \overset{W_1}{\underset{L_1}{\longmapsto}} u_1 \neq u \overset{W_2}{\underset{L_2}{\longmapsto}} u_2$, the predicate $\mathrm{rr}(\sigma^{W_1}_{L_1}, \sigma^{W_2}_{L_2})$ is satisfied.
- $\downarrow\, \subseteq S \times \mathcal{E}$ is the undelayable termination predicate.
- $\mathrm{I} \colon S \to 2^V$ is the independent racing delays function. It satisfies $\mathrm{I}(s) \subseteq \bigcup\{W \cup L \mid \langle s, \alpha \rangle \overset{W}{\underset{L}{\longmapsto}} \langle s', \alpha' \rangle\}$.

Definition 2 requires that the predicate $\mathrm{rr}(\sigma^{W_1}_{L_1}, \sigma^{W_2}_{L_2})$ holds for every two different timed delay transitions $u \overset{W_1}{\underset{L_1}{\longmapsto}} u_1 \neq u \overset{W_2}{\underset{L_2}{\longmapsto}} u_2$, implying that $W_1 \cup L_1 = W_2 \cup L_2$ and, thus, the existence of $\mathrm{R}(s) = W \cup L$ for every $\langle s, \alpha \rangle \overset{W}{\underset{L}{\longmapsto}} \langle s', \alpha' \rangle$. For notational convenience, we sometimes write $\mathrm{R}(u)$ instead of $\mathrm{R}(s)$ for $u = \langle s, \alpha \rangle$.

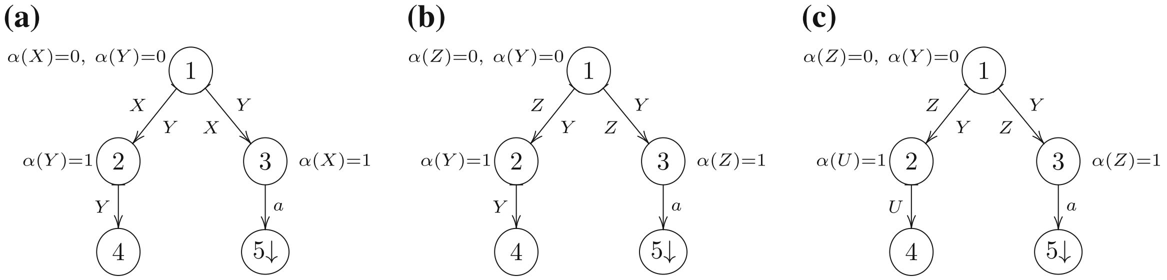

Fig. 2. Racing timed transition schemes: **a** and **b** are bisimilar, whereas **c** and **a**, and **c** and **b**, are not

We depict racing timed transition schemes as in Fig. 2. The states are numbered for ease of reference. State 1 of the racing timed transition scheme depicted in Fig. 2a has two outgoing transitions. Note that the race is incomplete as the outcomes where both X and Y are winners or losers are missing. The age of both delays in the beginning is 0. When the transition from state 1 to state 2 is taken the age of the loser Y is increased by 1 because it waited one time unit, whereas X has expired. The racing delays of state 1 are $\mathrm{R}(1) = \{X, Y\}$. If we assume that X is an independent delay, i.e., $\mathrm{I}(1) = \{X\}$, then Y is a dependent delay and $\mathrm{D}(1) = \{X, Y\}\setminus\{X\} = \{Y\}$. The termination predicate holds only in state 5 as indicated by ↓. In state 2 the only racing delay is Y, i.e., $\mathrm{R}(2) = \{Y\}$ and it is also a dependent delay as it has age 1, so $\mathrm{I}(2) = \emptyset$. Also the race in state 2 is not complete as the outcome when Y is a loser is missing.

Probabilistic timed transition system To represent the instantiation of a transition scheme with respect to a given assignment $\mathrm{d}\colon V \to \mathcal{F}$ of probability distributions we will employ *probabilistic timed transition systems*. The race condition is used to derive the underlying probability spaces that define the probabilistic behavior of each timed delay transition. In order to compute the correct distributions of the racing delays we will use the environment and the aging function. More precisely, the distribution of a racing delay $[X]$ in an environment α is given by $\mathrm{F}_X = \mathrm{d}(X) | \alpha(X)$.

Definition 3 A probabilistic timed transition system $(S, A, \to, \mapsto, \downarrow)$ is a tuple, where S is the set of states, A is a set of labels, $\to\ \subseteq S \times A \times S$ is the action transition relation, $\mapsto\colon S \to \mathcal{P}(\mathbb{N} \times S)$ is the probabilistic timed transition function, and $\downarrow\ \subseteq S$ is the undelayable termination predicate.

Each racing timed transition scheme coupled with probability distribution assignments induces a probabilistic timed transition system. The action transitions and termination predicate are adopted from the racing timed transition scheme. The probability measure of the (unit) timed delay is induced by its racing context.

Definition 4 Let $R = (S \times \mathcal{E}, A, V, \longrightarrow, \longmapsto, \downarrow, \mathrm{I})$ be a racing timed transition scheme and $\mathrm{d}\colon V \to \mathcal{F}$ a distribution assignment function. Then, the pair (R, d) induces the probabilistic timed transition system $P = (S \times \mathcal{E}, A, \to, \mapsto, \downarrow)$, where the action transition and termination options $\to$ and $\downarrow$ of P are given by $\longrightarrow$ and $\downarrow$ of R, respectively, and $\mapsto(u) = ((1, S \times \mathcal{E}), \mathrm{P})$ is the probability space induced by the race, with

$$\mathrm{P}(1, u') = \begin{cases} \dfrac{\mathrm{RC}_1(W', L')}{\sum\{\mathrm{RC}_1(W, L) \mid u \xmapsto[L]{W} \overline{u}\}}, & \text{if } \mathrm{R}(u) = W' \cup L' \neq \emptyset \\ 1, & \text{otherwise} \end{cases}, \tag{4}$$

where $u \xmapsto[L']{W'} u'$ and $F_X = \mathrm{d}(X) \mid \alpha(X)$. The probability measure is normalized as the race need not be complete, i.e., $\sum_{u \xmapsto[L]{W} \overline{u}} \mathrm{RC}_1(W, L) \leqslant 1$. Only for complete races with all possible outcomes this sum equals one.

Bisimulation relation As a behavioral relation, we choose strong bisimulation. It requires timed delays to be in the same racing context modulo names of independent delays. This ensures that the related racing timed transition schemes have the same probabilistic behavior, i.e., they induce the same probabilistic timed transition systems when coupled with corresponding distribution assigning functions. As usual, bisimilar terms are required to have the same termination options and action transitions [BBR10, BR04].

Definition 5 Let $R \subseteq (S \times \mathcal{E})^2 \times (\mathcal{V} \leftrightarrow \mathcal{V})$ be a relation. Then R is a racing timed bisimulation if for all $(u_1, u_2, \mathrm{r}) \in R$ it holds that also $(u_2, u_1, \mathrm{r}^{-1}) \in R$, where $\mathrm{r}\colon \mathrm{R}(u_1) \leftrightarrow \mathrm{R}(u_2)$ is a bijection with $\mathrm{r}(\mathrm{I}(u_1)) = \mathrm{I}(u_2)$, $\mathrm{F}_X = \mathrm{F}_{\mathrm{r}(X)}$, and $\alpha_1(X) = \alpha_2(\mathrm{r}(X))$ for $X \in \mathrm{dom}\,(\mathrm{r})$ and $\alpha_1, \alpha_2 \in \mathcal{E}$, and

1. if $u_1\!\downarrow$ then $u_2\!\downarrow$;
2. if $u_1 \xrightarrow{a} u_1'$, then there exists $u_2' \in S \times \mathcal{E}$ with $u_2 \xrightarrow{a} u_2'$ and $(u_1', u_2', \mathrm{r}') \in R$ for some $\mathrm{r}' \in \mathcal{V} \leftrightarrow \mathcal{V}$; and
3. if $u_1 \xmapsto[L_1]{W_1} u_1'$, then there exists $u_2' \in S \times \mathcal{E}$ with $u_2 \xmapsto[L_2]{W_2} u_2'$ and $\mathrm{r}(W_1) = W_2$, $\mathrm{r}(L_1) = L_2$, and $(u_1', u_2', \mathrm{r}') \in R$ for some $\mathrm{r}' \in \mathcal{V} \leftrightarrow \mathcal{V}$ satisfying $\mathrm{r}'(X) = \mathrm{r}(X)$ for $X \in L_1 \cap \mathrm{D}(u_1')$.

We say that two states u_1 and u_2 are racing timed bisimilar, notation $u_1 \mathrel{\underline{\leftrightarrow}}_t u_2$, if there exists a bisimulation relation R such that $(u_1, u_2, \mathrm{r}) \in R$ for some $\mathrm{r} \in \mathcal{V} \leftrightarrow \mathcal{V}$.

The relationship between racing contexts of timed delays of bisimilar states is established using the bijection r. It is a bijection as the same number of racing delays must be present in both states. It also must respect the independent delays stated by $\mathrm{r}(\mathrm{I}(u_1)) = \mathrm{I}(u_2)$. The independent delays can have different names, but they must have the same distribution and age, meaning that they will exhibit the same probabilistic behavior. Conditions 1 and 2 state that bisimilar states have the same termination options and action transitions. The timed delay transitions have racing contexts induced by winners and losers related by r. Condition 3 requires that the losers, identified in the resulting state by $L_1 \cap \mathrm{D}(u_1')$, are backward compatible, i.e., they retain their names as they are bound in the first race that they lost. We illustrate the situation by an example.

Example 3 The racing timed transition scheme depicted in Fig. 2a is racing timed bisimilar to the one from Fig. 2b provided that $F_Z = F_X$. As X is an independent racing delay, it can be renamed to the delay Z with the same distribution. However, the racing timed transition scheme depicted in Fig. 2c is not racing timed bisimilar to the one from Fig. 2a (nor to the one from Fig. 2b) even if $F_U = F_Y$. This is because Y is a loser in a previous race and its name must be preserved. As we have strong bisimilarity the action transitions and termination options must be mutually simulated in bisimilar states.

Below we state bisimilarity to be a congruence for TCP$^{\mathrm{drst}}$. First we have the following result [Mar08].

Theorem 1 *Racing timed bisimilarity $\mathrel{\underline{\leftrightarrow}}_t$ is an equivalence relation.*

Signature of TCP$^{\mathrm{drst}}$ We introduce the signature of TCP$^{\mathrm{drst}}$. The deadlocked process 0 has no outgoing transitions, whereas successful termination is denoted by 1. Undelayable action prefix is a unary operator scheme $a.\_$, for every $a \in \mathcal{A}$. Similarly, timed delay prefixes are of the form $\sigma^W_L.\_$ for $W, L \subseteq \mathcal{V}$ disjoint. The dependence scope operator is given by $|\_|_D$, for a dependence binding set $D \subseteq \mathcal{V}$. The encapsulation operator $\partial_H(\_)$ for $H \subseteq \mathcal{A}$ blocks the actions in H. The maximal time progress operator $\theta_I(\_)$ for $I \subseteq \mathcal{A}$ gives priority to the undelayable actions in I over passage of time. The alternative composition is given by $\_ + \_$, at the same time representing a nondeterministic choice between action transitions and termination, a weak nondeterministic choice between action and timed delay transitions, and probabilistic choice of the (resolved) racing contexts of the timed delays. The parallel composition is given by $\_ \parallel \_$. It allows passage of time only if both components do so. The set of process terms $\mathcal{C}$ is formed by P given by:

$$P ::= 0 \mid 1 \mid a.P \mid \sigma^W_L.P \mid |P|_D \mid \partial_H(P) \mid \theta_I(P) \mid P + P \mid P \parallel P, \tag{5}$$

where $a \in \mathcal{A}$, $W, L, D \subseteq \mathcal{V}$ with $W \cap L = \emptyset$, and $H, I \subseteq \mathcal{A}$.

Structural operational semantics The semantics of a term $p \in \mathcal{C}$ in an environment $\alpha \in \mathcal{E}$ is given by the racing timed transition scheme $(\mathcal{C} \times \mathcal{E}, \mathcal{A}, \mathcal{V}, \longrightarrow, \longmapsto, \downarrow, \mathrm{I})$, where $\longrightarrow$, $\longmapsto$, and $\downarrow$ are defined by the operational rules given in Table 1. We omit symmetrical rules, which numbers are denoted in brackets. For notational convenience, we write α_0 for the environment such that $\alpha_0(X) = 0$, for $X \in \mathcal{V}$. Also, we write $\alpha + 1$ for the function satisfying $(\alpha + 1)(X) = \alpha(X) + 1$. We use three additional predicates in the operational rules: (1) $\langle p, \alpha \rangle \longmapsto$ denoting that the state has an outgoing timed delay transition, (2) $\langle p, \alpha \rangle \not\longmapsto$ denoting that the state has no outgoing timed delay transitions, and (3) $\langle p, \alpha \rangle \overset{a}{\not\longrightarrow}$ denoting that the state has no outgoing action transitions labeled by action a.

Table 1. Structural operational semantics

$$\mathbf{1}\ \frac{}{\langle 1, \alpha\rangle\!\downarrow} \qquad \mathbf{2}\ \frac{}{\langle a.p, \alpha\rangle \xrightarrow{a} \langle |p|_{\emptyset}, \alpha_0\rangle} \qquad \mathbf{3}\ \frac{}{\langle \sigma_L^W.p, \alpha\rangle \xmapsto[L]{W} \langle |p|_L, \alpha_0\{(\alpha+1)/L\}\rangle} \qquad \mathbf{4}\ \frac{\langle p, \alpha\rangle\!\downarrow}{\langle |p|_D, \alpha\rangle\!\downarrow} \qquad \mathbf{5}\ \frac{\langle p, \alpha\rangle \xrightarrow{a} \langle p', \alpha'\rangle}{\langle |p|_D, \alpha\rangle \xrightarrow{a} \langle p', \alpha'\rangle}$$

$$\mathbf{6}\ \frac{\langle p, \alpha\rangle \xmapsto[L]{W} \langle p', \alpha'\rangle}{\langle |p|_D, \alpha\rangle \xmapsto[L]{W} \langle p', \alpha'\rangle} \qquad \mathbf{7\,(8)}\ \frac{\langle p_1, \alpha\rangle\!\downarrow}{\langle p_1 + p_2, \alpha\rangle\!\downarrow} \qquad \mathbf{9\,(10)}\ \frac{\langle p_1, \alpha\rangle \xrightarrow{a_1} \langle p_1', \alpha_1\rangle}{\langle p_1 + p_2, \alpha\rangle \xrightarrow{a_1} \langle p_1', \alpha_1\rangle} \qquad \mathbf{11\,(12)}\ \frac{\langle p_1, \alpha\rangle \xmapsto[L_1]{W_1} \langle p_1', \alpha_1\rangle,\ \langle p_2, \alpha\rangle \not\longmapsto}{\langle p_1 + p_2, \alpha\rangle \xmapsto[L_1]{W_1} \langle p_1', \alpha_1\rangle}$$

$$\mathbf{13}\ \frac{\begin{gathered}\langle p_1, \alpha\rangle \xmapsto[L_1]{W_1} \langle p_1', \alpha_1\rangle,\ \langle p_2, \alpha\rangle \xmapsto[L_2]{W_2} \langle p_2', \alpha_2\rangle,\\ (W_1 \cup W_2) \cap (L_1 \cup L_2) = \emptyset\end{gathered}}{\langle p_1 + p_2, \alpha\rangle \xmapsto[L_1 \cup L_2]{W_1 \cup W_2} \langle p_1' + p_2', \alpha_1\{\alpha_2/L_2\}\rangle} \qquad \mathbf{14\,(15)}\ \frac{\begin{gathered}\langle p_1, \alpha\rangle \xmapsto[L_1]{W_1} \langle p_1', \alpha_1\rangle,\\ \mathrm{rr}(\sigma_{L_1}^{W_1}, \sigma_{L_2}^{W_2}) \text{ for all } W_2, L_2 \text{ such that } \langle p_2, \alpha\rangle \xmapsto[L_2]{W_2}\end{gathered}}{\langle p_1 + p_2, \alpha\rangle \xmapsto[L_1]{W_1} \langle p_1', \alpha_1\rangle}$$

$$\mathbf{16}\ \frac{\langle p_1, \alpha\rangle\!\downarrow,\ \langle p_2, \alpha\rangle\!\downarrow}{\langle p_1 \parallel p_2, \alpha\rangle\!\downarrow} \qquad \mathbf{17\,(18)}\ \frac{\langle p_1, \alpha\rangle \xrightarrow{a_1} \langle p_1', \alpha_1\rangle,\ \langle p_2, \alpha\rangle \not\longmapsto}{\langle p_1 \parallel p_2, \alpha\rangle \xrightarrow{a_1} \langle p_1' \parallel p_2, \alpha_1\rangle} \qquad \mathbf{19\,(20)}\ \frac{\langle p_1, \alpha\rangle \xrightarrow{a_1} \langle p_1', \alpha_1\rangle,\ \langle p_2, \alpha\rangle \longmapsto}{\langle p_1 \parallel p_2, \alpha\rangle \xrightarrow{a_1} \langle p_1' \parallel p_2, \alpha\{\alpha_0/(\mathcal{V} \setminus \mathrm{R}(p_2))\}\rangle}$$

$$\mathbf{21}\ \frac{\langle p_1, \alpha\rangle \xrightarrow{a_1} \langle p_1', \alpha_1\rangle,\ \langle p_2, \alpha\rangle \xrightarrow{a_2} \langle p_2', \alpha_2\rangle,\ \gamma(a_1, a_2) = a_3}{\langle p_1 \parallel p_2, \alpha\rangle \xrightarrow{a_3} \langle p_1' \parallel p_2', \alpha_0\rangle} \qquad \mathbf{22}\ \frac{\begin{gathered}\langle p_1, \alpha\rangle \xmapsto[L_1]{W_1} \langle p_1', \alpha_1\rangle,\ \langle p_2, \alpha\rangle \xmapsto[L_2]{W_2} \langle p_2', \alpha_2\rangle,\\ (W_1 \cup W_2) \cap (L_1 \cup L_2) = \emptyset\end{gathered}}{\langle p_1 \parallel p_2, \alpha\rangle \xmapsto[L_1 \cup L_2]{W_1 \cup W_2} \langle p_1' \parallel p_2', \alpha_1\{\alpha_2/L_2\}\rangle}$$

$$\mathbf{23}\ \frac{\langle p, \alpha\rangle\!\downarrow}{\langle \partial_H(p), \alpha\rangle\!\downarrow} \qquad \mathbf{24}\ \frac{\langle p, \alpha\rangle \xrightarrow{a} \langle p', \alpha'\rangle,\ a \notin H}{\langle \partial_H(p), \alpha\rangle \xrightarrow{a} \langle \partial_H(p'), \alpha'\rangle} \qquad \mathbf{25}\ \frac{\langle p, \alpha\rangle \xmapsto[L]{W} \langle p', \alpha'\rangle}{\langle \partial_H(p), \alpha\rangle \xmapsto[L]{W} \langle \partial_H(p'), \alpha'\rangle}$$

$$\mathbf{26}\ \frac{\langle p, \alpha\rangle\!\downarrow}{\langle \theta_I(p), \alpha\rangle\!\downarrow} \qquad \mathbf{27}\ \frac{\langle p, \alpha\rangle \xrightarrow{a} \langle p', \alpha'\rangle}{\langle \theta_I(p), \alpha\rangle \xrightarrow{a} \langle \theta_I(p'), \alpha'\rangle} \qquad \mathbf{28}\ \frac{\langle p, \alpha\rangle \xmapsto[L]{W} \langle p', \alpha'\rangle,\ \langle p, \alpha\rangle \overset{a}{\not\longrightarrow} \text{ for } a \in I}{\langle \theta_I(p), \alpha\rangle \xmapsto[L]{W} \langle \theta_I(p'), \alpha'\rangle}$$

Rule 1 states that the termination constant terminates regardless the current value of the environment. Rule 2 states that action prefixes enable action transitions and reset the ages of the racing delays to zero. Rule 3 states that timed delay prefixes enable timed transitions with racing contexts induced by the winners and the losers. The resulting environment contains the incremented ages of the losers. Rules 4–6 imply that the dependence scope does neither affect the termination nor the outgoing transitions of the term. Rules 7 and 8 state that the alternative composition has a termination option if at least one summand does. Rules 9 and 10 enable the nondeterministic choice between two action transitions. Rules 11 and 12 enable the weak choice between action transitions and timed delays. Rule 13 gives the synchronization of timed delays when the racing contexts can be merged. Rules 14 and 15 enable the resolution of races on disjoint events. A timed delay transition is in a context of a resolved race if it is in a resolved race with every timed delay transition of the other term. For example, the requirement that the timed delay $\sigma_{L_2}^{W_2}$ of the right summand is in a resolved race is ensured by the condition $\mathrm{rr}(\sigma_{L_1}^{W_1}, \sigma_{L_2}^{W_2})$ for all $\langle p_1, \alpha\rangle \xmapsto[L_1]{W_1}$. Rule 16 states that the parallel composition can terminate only when both components can. Rules 17–20 enable interleaving of action transitions in the parallel composition. Rules 17 and 18 state that the environment is reset when the other component cannot perform a timed delay transition. This is to preserve the desired property that only the ages of the losers persist in the environment. However, the relevant part of the environment must be preserved in case the other component can perform a timed delay as given by rules 19 and 20. Rule 21 allows for synchronization of action transitions if defined by the synchronization function. Similarly to the alternative composition, synchronization of timed delays is allowed when the racing contexts can be merged as given by rule 22. Rule 23 states that the termination option is not affected by the encapsulation operator. Rule 24 states that action transitions are allowed only if they are not labeled by actions that should be suppressed. Rule 25 states that the encapsulation does not affect the timed delays. The maximal progress operator does neither affect termination nor the action transitions as they are undelayable, given by rules 26 and 27. Timed delay transitions are exhibited only if the term cannot perform a prioritized action transition, given by rule 28.

Term model Based on the structural operational semantics, we can relate TCP$^{\mathrm{drst}}$ terms as well. Two terms $p_1, p_2 \in \mathcal{C}$ are racing timed bisimilar, notation $p_1 \leftrightarroweq_t p_2$ if there exists a racing timed bisimulation relation R such that $(\langle p_1, \alpha_0\rangle, \langle p_2, \alpha_0\rangle, \mathrm{r}) \in R$ for some $\mathrm{r} \in \mathcal{V} \leftrightarrow \mathcal{V}$ satisfying $\mathrm{r}(X) = X$ for $X \in \mathrm{D}(p_1)$. The condition that $\mathrm{r}(X) = X$ for $X \in \mathrm{D}(p_1)$ states that bisimilar terms must have the same dependent delays. This preserves the congruence property as dependent delays are explicitly aged by the timed delay prefix σ_L^W, whereas independent delays cannot have an explicit age dependence.

Table 2. Race-consistent renaming operation

$$0[{}^{Y}/_{X}] = 0 \qquad 1[{}^{Y}/_{X}] = 1 \qquad (a.p)[{}^{Y}/_{X}] = a.p \qquad \partial_H(p)[{}^{Y}/_{X}] = \partial_H(p[{}^{Y}/_{X}]) \qquad \theta_I(p)[{}^{Y}/_{X}] = \theta_I(p[{}^{Y}/_{X}])$$

$$(p_1 + p_2)[{}^{Y}/_{X}] = p_1[{}^{Y}/_{X}] + p_2[{}^{Y}/_{X}] \qquad (p_1 \parallel p_2)[{}^{Y}/_{X}] = p_1[{}^{Y}/_{X}] \parallel p_2[{}^{Y}/_{X}] \qquad (\sigma_L^W.p)[{}^{Y}/_{X}] = \sigma_L^W.p \quad \text{if } X \notin W \cup L$$

$$(\sigma_L^W.p)[{}^{Y}/_{X}] = \sigma_L^{(W\setminus\{X\})\cup\{Y\}}.p \quad \text{if } X \in W \qquad (\sigma_L^W.p)[{}^{Y}/_{X}] = \sigma_{(L\setminus\{X\})\cup\{Y\}}^{W}.p[{}^{Y}/_{X}] \quad \text{if } X \in L$$

$$(|p|_D)[{}^{Y}/_{X}] = |p|_D \quad \text{if } X \notin D \qquad (|p|_D)[{}^{Y}/_{X}] = |p[{}^{Y}/_{X}]|_{(D\setminus\{X\})\cup\{Y\}} \quad \text{if } X \in D$$

Table 3. Axioms for the dependence scope operator

$$|0|_{\emptyset} = 0 \ \mathbf{A1} \qquad |1|_{\emptyset} = 1 \ \mathbf{A2} \qquad |a.p|_{\emptyset} = a.p \ \mathbf{A3} \qquad a.p = a.\,|p|_{\emptyset} \ \mathbf{A4}$$

$$\sigma_L^W.p = |\sigma_L^W.p|_{W\cup L} \ \mathbf{A5} \qquad \sigma_L^W.p = \sigma_L^W.\,|p|_L \ \mathbf{A6} \qquad ||p|_{D_1}|_{D_2} = |p|_{D_1\cap D_2} \ \mathbf{A7}$$

The congruence property of racing timed bisimilarity is stated in the following theorem [Mar08].

Theorem 2 *Racing timed bisimilarity relation $\mathrel{\underline{\leftrightarrow}}_t$ is a congruence on $\mathcal{C}$.*

Now, we have all the prerequisites to define the term model of TCP$^{\mathrm{drst}}$ modulo racing timed bisimulation.

Definition 6 The term model of TCP$^{\mathrm{drst}}$ is the quotient algebra $\mathbb{P}(\mathrm{TCP}^{\mathrm{drst}})/\mathrel{\underline{\leftrightarrow}}_t$, where

$$\begin{aligned}\mathbb{P}(\mathrm{TCP}^{\mathrm{drst}}) = (&\mathcal{C},\ 0,\ 1,\ a.\_ \text{ for } a \in \mathcal{A},\ \sigma_L^W.\_ \text{ for } W, L \subseteq \mathcal{V}, \text{ with } W \cap L = \emptyset,\ |\_|_D \text{ for } D \subseteq \mathcal{V},\\ &\partial_H(\_) \text{ for } H \subseteq \mathcal{A},\ \theta_I(\_) \text{ for } I \subseteq \mathcal{A},\ \_ + \_,\ \_ \parallel \_).\end{aligned} \tag{6}$$

That is, $\mathbb{P}(\mathrm{TCP}^{\mathrm{drst}})$ is the algebra with carrier set $\mathcal{C}$, comprising all process terms and operations of TCP$^{\mathrm{drst}}$.

4. Equational theory

The main complication in dealing with process terms that induce incomplete races is that the alternative composition is no longer associative, unless all races are resolved. Furthermore, the expansion of the parallel composition and the resolution of the maximal progress operator require resolved races. This impels us to present the equational theory of TCP$^{\mathrm{drst}}$ in terms of process normal forms with resolved races. First, we give axioms for manipulation of the dependence scope operator. We employ them to derive a normal form that enumerates all possible outcomes of a race, making the alternative composition associative. It is unique for the timed delays modulo commutativity, associativity, and naming of independent delays. We use this normal form to give expansion laws for the other operators, such that the expansions are again in the same normal form. Thereafter, we prove the theory to be sound and ground-complete. At the end, we introduce guarded recursive specifications to obtain the process theory TCP$^{\mathrm{drst}}_{\mathrm{rec}}$.

Renaming of independent delays As already discussed, the main idea behind having two types of race condition is that systems are modeled by independent delays whereas, the race condition is resolved by assigning unique names to racing delays and afterwards treating them as dependent. Thus, we need a mechanism for renaming independent delays (and making them dependent).

Example 4 Given the simple component $|\sigma_Y^X.\sigma^Y.a.0|_{\emptyset}$, we can use it as a building block of the system $|\sigma_Y^X.\sigma^Y.a.0|_{\emptyset} \parallel |\sigma_Y^X.\sigma^Y.a.0|_{\emptyset}$. However, for composition purposes, we revert to the system $|\sigma_Y^X.\sigma^Y.a.0 \parallel \sigma_V^U.\sigma^V.a.0|_{\emptyset}$, where $\mathrm{F}_X = \mathrm{F}_U$ and $\mathrm{F}_Y = \mathrm{F}_V$. The advantage of encompassing the whole term within a single dependence scope is that all independent delays are given unique names. Moreover, the dependent delays are 'declared' in the parameter of the dependence scope.

It is clear that naming conflicts may arise in situations as in Example 4 above. Care has to be taken to rename losing delays consistently as their names are made dependent and bound by the winners in the first race that they lost. To this end, we define a race-consistent renaming operation $p[{}^{Y}/_{X}]$ in Table 2, which renames X to Y in the process term p.

Dependence scope Axioms to manipulate the dependence scope operators are given in Table 3. Axioms A1–A3 deal with terms that have no timed delays, so they impose an empty dependence scope. Axiom A4 states that there is no dependence of timed delays that are enabled by an action transition. Axiom A5 states that all delays are treated as dependent by default. Axiom A6 states that the losers of a timed delay retain their names and that they are treated as dependent in the remaining process. Axiom A7 states that multiple scope operators intersect. It enables the replacement of iterative application of scope operators by a single one.

Theorem 3 *The axioms in Table 3 are sound with respect to the operational rules of Table 1.*

The axioms in Table 3 enable manipulation of iterated applications of the dependence scope operator and scopes encompassing action or timed delay prefixed processes. Next, we deal with the alternative composition.

Alternative composition In general, associativity does not hold for the alternative composition. Intuitively, the condition for resolved racing contexts is problematic as it may depend on the order we merge racing contexts of timed delays in incomplete races. The following example illustrates the situation.

Example 5 Consider the terms $(\sigma^{X}_{Y}.0 + \sigma^{Z}.0) + \sigma^{Y,Z}_{X}.0$ and $\sigma^{X}_{Y}.0 + (\sigma^{Z}.0 + \sigma^{Y,Z}_{X}.0)$. The transition scheme of the first term has two outgoing transitions, viz. $(\sigma^{X}_{Y}.0 + \sigma^{Z}.0) + \sigma^{Y,Z}_{X}.0 \xmapsto[Y]{X,Z} 0 + 0$ and $(\sigma^{X}_{Y}.0 + \sigma^{Z}.0) + \sigma^{Y,Z}_{X}.0 \xmapsto[X]{Y,Z} 0$ because $(\{X\} \cup \{Z\}) \cap (\{Y\} \cup \emptyset) = \emptyset$ and $\mathrm{rr}(\sigma^{X,Z}_{Y}, \sigma^{Y,Z}_{X})$ holds. However, the second process only deadlocks as the timed delay transitions $\xmapsto[Y]{X}$ of $\sigma^{X}_{Y}.0$ and $\xmapsto[X]{Y,Z}$ of $\sigma^{Z}.0 + \sigma^{Y,Z}_{X}.0$ are in inconsistent racing contexts.

Nevertheless, associativity holds for terms that comprise alternative composition of action prefixed terms and timed delay prefixed terms in a context of resolved races as in such cases there is no merging of the timed delays. Thus, the timed delay transitions are distinctly modeled by the prefixes. Such a term p can be represented in a 'normal' form that is unique for the timed delays modulo commutativity, associativity, and naming of independent delays (see Remark 1 below). Specifically, if p defines a context of a resolved race, it can be equivalently written as $p = \big| \sum_{i=1}^{m} a_i.p_i + \sum_{j=1}^{n} \sigma^{W_j}_{L_j}.q_j \ (+1) \big|_D$, where $W_j \cup L_j = \mathrm{R}(p)$ for every $1 \leqslant j \leqslant n$ is the set of racing delay names, $D \subseteq \mathrm{R}(p)$ determines the dependent delay names, whereas $\mathrm{I}(p) = \mathrm{R}(p) \setminus D$, the summand 1 may or may not exist, and $\mathrm{rr}(\sigma^{W_j}_{L_j}, \sigma^{W_{j'}}_{L_{j'}})$ holds for $1 \leqslant j < j' \leqslant n$. The notation $\sum_{i=1}^{m} p_i$ is shorthand for $p_1 + \cdots + p_m$, if $m > 0$, and equals 0, otherwise.

Remark 1 Unlike standard head normal forms, e.g. [BM02, BW90], we do not have $a_i.p_i \neq a_{i'}.p_{i'}$ for $1 \leqslant i < i' \leqslant m$, at this point. This is a prerequisite for the uniqueness of the normal form and we discuss it later on. Similarly, associativity still holds if we relax the condition of the timed delay prefixes to require that $\mathrm{rr}(\sigma^{W_j}_{L_j}, \sigma^{W_{j'}}_{L_{j'}})$ holds or $(W_j = W_{j'}$ and $L_j = L_{j'})$ for $1 \leqslant j, j' \leqslant n$, relying on the fact that $\sigma^{W}_{L}.p_1 + \sigma^{W}_{L}.p_2 \mathrel{\underline{\leftrightarrow}}_t \sigma^{W}_{L}.(p_1 + p_2)$. We also note that when restricting to race-complete process specifications, the associativity of the alternative composition holds [MV09]. In this special case, the inconsistency of Example 5 cannot occur as all timed delays of the remaining resolved racing contexts would also be available.

Based on the normal forms we give the expansion law A8 for the alternative composition of two terms.

Theorem 4 *Let* $p = \big| \sum_{i=1}^{m} a_i.p_i + \sum_{j=1}^{n} \sigma^{W_j}_{L_j}.q_j \ (+1) \big|_D$ *and* $p' = \big| \sum_{k=1}^{m'} a'_k.p'_k + \sum_{\ell=1}^{n'} \sigma^{W'_\ell}_{L'_\ell}.q'_\ell \ (+1) \big|_{D'}$*, with* $D \subseteq W_j \cup L_j$, $D' \subseteq W'_\ell \cup L'_\ell$, $\mathrm{rr}(\sigma^{W_j}_{L_j}, \sigma^{W_{j'}}_{L_{j'}})$ *valid for* $1 \leqslant j < j' \leqslant n$*, and* $\mathrm{rr}(\sigma^{W'_\ell}_{L'_\ell}, \sigma^{W'_{\ell'}}_{L'_{\ell'}})$ *valid for* $1 \leqslant \ell < \ell' \leqslant n'$*. If* $\mathrm{I}(p) \cap \mathrm{R}(p') = \mathrm{R}(p) \cap \mathrm{I}(p') = \emptyset$*, then the normal form of their alternative composition is given by*

$$p + p' = \Bigg| \sum_{i=1}^{m} a_i.p_i + \sum_{k=1}^{m'} a'_k.p'_k + \sum_{j,\ell:\,(W_j \cup W'_\ell) \cap (L_j \cup L'_\ell) = \emptyset} \sigma^{W_j \cup W'_\ell}_{L_j \cup L'_\ell}.(|q_j|_{L_j} + |q'_\ell|_{L'_\ell}) +$$

$$\sum_{j:\,\mathrm{rr}(\sigma^{W_j}_{L_j}, \sigma^{W'_\ell}_{L'_\ell}) \textit{ for all } 1 \leqslant \ell \leqslant n'} \sigma^{W_j}_{L_j}.q_j + \sum_{\ell:\,\mathrm{rr}(\sigma^{W_j}_{L_j}, \sigma^{W'_\ell}_{L'_\ell}) \textit{ for all } 1 \leqslant j \leqslant n} \sigma^{W'_\ell}_{L'_\ell}.q'_\ell \ (+\,1) \Bigg|_{D \cup D'} \qquad \mathbf{A8}$$

where the summand 1 *exists if* p *or* p' *contains it.*

Using Theorem 4 we can represent every term comprising an alternative composition of deadlock, termination, and action and timed prefixed terms in a normal form provided there are no naming conflicts of the independent delays. In case there are such conflicts, we resolve them by renaming the independent delays.

Encapsulation The encapsulation operator suppresses undesired action transitions and, unlike the alternative composition, it does not require resolved races as it freely propagates through the timed delay prefixes [Mar08]. However, for the sake of uniformity, we give it in terms of normal forms as well.

Theorem 5 *Let* $p = |\sum_{i=1}^{m} a_i.p_i + \sum_{j=1}^{n} \sigma_{L_j}^{W_j}.q_j\ (+1)|_D$, *where* $D \subseteq W_j \cup L_j$ *and* $\mathrm{rr}(\sigma_{L_j}^{W_j}, \sigma_{L_{j'}}^{W_{j'}})$ *holds for* $1 \leqslant j < j' \leqslant n$. *Then,*

$$\partial_H(p) = \left| \sum_{a_i \notin H} a_i.\partial_H(p_i) + \sum_{j=1}^{n} \sigma_{L_j}^{W_j}.\partial_H(q_j)\,(+1) \right|_D \qquad \textbf{A9}$$

where $(+1)$ *depends on* p.

Parallel composition The expansion law of the parallel composition requires resolved racing contexts as illustrated by the following example.

Example 6 Let $p = (\sigma_X.0 + \sigma^Y.0) \parallel \sigma^X.0$. By first resolving the race in the left operand and afterwards eliminating the parallel composition according to the operational rules one readily obtains that $p = \sigma_X^Y.0 \parallel \sigma^X.0 = 0$. However, if we attempt to naively expand the parallel composition as it is done in timed process theories, we would wrongly obtain that $p = \sigma_X.0 \parallel \sigma^X.0 + \sigma^Y.0 \parallel \sigma^X.0 = 0 + \sigma^{X,Y}.0 = \sigma^{X,Y}.0$.

Theorem 6 *Let* $p = |\sum_{i=1}^{m} a_i.p_i + \sum_{j=1}^{n} \sigma_{L_j}^{W_j}.q_j\ (+1)|_D$ *and* $p' = |\sum_{k=1}^{m'} a'_k.p'_k + \sum_{\ell=1}^{n'} \sigma_{L'_\ell}^{W'_\ell}.q'_\ell\ (+1)|_{D'}$, *with* $D \subseteq W_j \cup L_j$, $D' \subseteq Will \cup L'_\ell$, $\mathrm{rr}(\sigma_{L_j}^{W_j}, \sigma_{L_{j'}}^{W_{j'}})$ *for* $1 \leqslant j < j' \leqslant n$, *and* $\mathrm{rr}(\sigma_{L'_\ell}^{W'_\ell}, \sigma_{L'_{\ell'}}^{W'_{\ell'}})$ *for* $1 \leqslant \ell < \ell' \leqslant n'$. *If* $\mathrm{I}(p) \cap \mathrm{R}(p') = \mathrm{R}(p) \cap \mathrm{I}(p') = \emptyset$, *then the normal form of their parallel composition is given by*

$$\begin{aligned} p \parallel p' = \big|\, & \textstyle\sum_{i=1}^{m} a_i.(|p_i|_\emptyset \parallel p') + \sum_{k=1}^{m'} a'_k.(p \parallel |p'_k|_\emptyset) + \sum_{i,k:\, \gamma(a_i, a'_k) = b_{ik}} b_{ik}.(|p_i|_\emptyset \parallel |p'_k|_\emptyset) + \\ & \textstyle\sum_{j,\ell:\, (W_j \cup W'_\ell) \cap (L_j \cup L'_\ell) = \emptyset} \sigma_{L_j \cup L'_\ell}^{W_j \cup W'_\ell}.(|q_j|_{L_j} \parallel |q'_\ell|_{L'_\ell})\,(+1)\, |_{D \cup D'} \end{aligned} \qquad \textbf{A10}$$

where the summand 1 *exists only if it exists in both* p *and* p'.

Different from the alternative composition, the parallel composition is associative for closed $\mathrm{TCP^{drst}}$ terms and, thus, supports compositional modeling. Intuitively, the parallel composition is associative as it eliminates resolved races that obstructed the associativity of the alternative composition, cf. Table 1 [Mar08].

Theorem 7 *Parallel composition is associative, i.e.,* $(p \parallel p') \parallel p'' = p \parallel (p' \parallel p'')$ *for all* $p, p', p'' \in \mathcal{C}$.

Maximal progress Typically, resolution of the maximal progress operator requires an additional operator that ascertains that a process has no timed delay transitions [BM02]. Alternatively, as presented, one can employ normal forms that make the undelayable action transitions and the timed delay transitions explicit. As only passage of time is important, the resolution of maximal progress does not demand resolved races. Nevertheless, for the sake of compactness we give an expansion law for the existing normal forms.

Theorem 8 *Let* $p = |\sum_{i=1}^{m} a_i.p_i + \sum_{j=1}^{n} \sigma_{L_j}^{W_j}.q_j\ (+1)|_D$, *where* $D \subseteq W_j \cup L_j$ *and* $\mathrm{rr}(\sigma_{L_j}^{W_j}, \sigma_{L_{j'}}^{W_{j'}})$ *holds for* $1 \leqslant j < j' \leqslant n$. *Then,*

$$\theta_I(p) = |\textstyle\sum_{i=1}^{m} a_i.\theta_I(p_i)(+1)|_D \quad \text{if } a_i \in I \text{ for some } i \in \{1, \ldots, m\} \qquad \textbf{A11}$$

$$\theta_I(p) = |\textstyle\sum_{i=1}^{m} a_i.\theta_I(p_i) + \sum_{j=1}^{n} \sigma_{L_j}^{W_j}.\theta_I(q_j)(+1)|_D \quad \text{if } \bigcup_{i=1}^{m} \{a_i\} \cap I = \emptyset \qquad \textbf{A12}$$

where the existence of $(+1)$ *depends on* p.

Now that we provided expansion laws for all operators, we proceed by giving head normal forms for closed $\mathrm{TCP^{drst}}$ terms that support the further development of the theory.

Ground completeness Using the axioms/expansion laws for every operator, it is not difficult to see that every closed $\mathrm{TCP^{drst}}$ term can be represented again in normal form. To eliminate multiple instances of bisimilar action prefixed terms in alternative composition we employ the idempotence axiom

$$a.p + a.p = a.p \qquad \textbf{A13}$$

which enables unique normal forms as discussed above in Remark 1.

Table 4. Operational rules for guarded recursion

$$\mathbf{29}\ \frac{\langle \mu\, p.S, \alpha\rangle{\downarrow},\ A = p \in S}{\langle \mu\, A.S, \alpha\rangle{\downarrow}} \qquad \mathbf{30}\ \frac{\langle \mu\, p.S, \alpha\rangle \xrightarrow{a} \langle p', \alpha'\rangle,\ A = p \in S}{\langle \mu\, A.S, \alpha\rangle \xrightarrow{a} \langle p', \alpha'\rangle} \qquad \mathbf{31}\ \frac{\langle \mu\, p.S, \alpha\rangle \xmapsto[L]{W} \langle p', \alpha'\rangle,\ A = p \in S}{\langle \mu\, A.S, \alpha\rangle \xmapsto[L]{W} \langle p', \alpha'\rangle}$$

Table 5. Definition of $\mu p.S$

$$\mu\, 0.S = 0 \qquad \mu\, 1.S = 1 \qquad \mu(a.p).S = a.(\mu\, p.S) \qquad \mu(\sigma_L^W.p).S = \sigma_L^W.(\mu\, p.S) \qquad \mu(\mu\, A.S).S = \mu\, A.S$$

$$\mu(\partial_H(p)).S = \partial_H(\mu\, p.S) \qquad \mu(\theta_I(p)).S = \theta_I(\mu\, p.S) \qquad \mu(p_1 + p_2).S = \mu\, p_1.S + \mu\, p_2.S \qquad \mu(p_1 \parallel p_2).S = \mu\, p_1.S \parallel \mu\, p_2.S$$

Corollary 1 Every $p \in \mathcal{C}$ can be represented in a unique head normal form modulo commutativity, associativity, and naming of independent delays $p = \lvert \sum_{i=1}^{m} a_i.p_i + \sum_{j=1}^{n} \sigma_{L_j}^{W_j}.q_j\ (+1) \rvert_D$, where $a_i.p_i \neq a_{i'}.p_{i'}$ for $1 \leqslant i < i' \leqslant m$, $D \subseteq \mathrm{R}(p) = W_j \cup L_j$, $\mathrm{rr}(\sigma_{L_j}^{W_j}, \sigma_{L_{j'}}^{W_{j'}})$ holds for $1 \leqslant j < j' \leqslant n$, and the summand 1 is optional.

The availability of a head normal form is technically important. It is instrumental for proving ground-completeness and showing uniqueness of solutions of guarded recursive specification [BBK87].

Theorem 9 *Axioms A1–A13 are ground-complete for the term model* $\mathbb{P}(\mathrm{TCP}^{\mathrm{drst}})/{\leftrightarrows_t}$.

Guarded recursive specifications We introduce guarded recursive specifications and obtain the process theory $\mathrm{TCP}^{\mathrm{drst}}_{\mathrm{rec}}$. Guardedness is a well-known concept that typically guarantees unique solution of the recursive specifications [BBK87]. The prerequisite is that every recursion variable must be prefixed, which ensures well-defined (predictable) behavior of the process.

A guarded recursive equation is an equation of the form $A = p$, where $A \in \mathcal{R}$ is a recursion variable, and p is a term over the signature of $\mathrm{TCP}^{\mathrm{drst}}$ that additionally contains variables from $\mathcal{R}$. Moreover, the term can be rewritten in such a way that the variables only appear in subterms prefixed by $a.\_$ or $\sigma_L^W.\_$ for $a \in \mathcal{A}$ and $W, L \in \mathcal{V}$ provided that $W \cap L = \emptyset$. A guarded recursive specification $S \in \mathcal{G}$, $\mathcal{G}$ denoting the set of specifications of interest, is a set of guarded recursive equations with one equation for every variable. The set of recursion variables of a specification S is denoted by $\mathcal{R}(S)$.

The definitions of racing, dependent, and independent delay names are straightforwardly extended to guarded recursive specifications. For renaming of delays we have that $A[{}^{X}/_{Y}] = A'$, where $A' = p[{}^{X}/_{Y}]$ provided that $A = p$. Solutions of recursive specifications in the term model are process terms that when replaced for the recursion variables give valid equations in the term model. By the constant $\mu\, A.S$ we denote a process term that is a solution for the recursion variable $A \in \mathcal{R}(S)$ defined by the guarded recursive specification S. Typically, a solution of a single variable is of interest, which we refer to as the solution of the guarded recursive specification. The structural operational semantics is given in Table 4. We generalize the notation $\mu\, A.S$ to $\mu\, p.S$, for an arbitrary term $p \in \mathcal{C}$ that contains variables from $\mathcal{R}(S)$ [BBK87] in Table 5.

It is straightforward from the structural operational semantics that every equation of a guarded recursive specification has a solution. Thus, the restrictive recursive definition principle, abbreviated as RDP$^-$, that every guarded recursive specification has a solution is sound in $\mathrm{TCP}^{\mathrm{drst}}_{\mathrm{rec}}$. Also, it should come as no surprise that the bisimulation relation is a congruence for recursion and that the axioms and expansion laws are sound for $\mathbb{P}(\mathrm{TCP}^{\mathrm{drst}}_{\mathrm{rec}})$, forming the standard term model $\mathbb{P}(\mathrm{TCP}^{\mathrm{drst}}_{\mathrm{rec}})/{\leftrightarrows_t}$. It is readily observed that $\mathrm{TCP}^{\mathrm{drst}}_{\mathrm{rec}}$ is a conservative extension of $\mathrm{TCP}^{\mathrm{drst}}$ [BM02, BBR10]. Additionally, it is not difficult to show that the head normal form of Corollary 1 is preserved. Now, by an adaptation of the proofs of [BBK87] along the lines of [BBR10] it can be shown that the recursive specification principle holds, relying on the existence of the head normal norm. This principle, abbreviated as RSP, states that every guarded recursive specification has at most one solution in the model. As a consequence of the validity of the principles RDP$^-$ and RSP in the model, all guarded recursive specifications have a unique solution in $\mathbb{P}(\mathrm{TCP}^{\mathrm{drst}}_{\mathrm{rec}})/{\leftrightarrows_t}$.

5. Derived delayable action and stochastic delay prefixes

In this section we express delayable action and stochastic delay prefixes by means of guarded recursive specifications comprising undelayable actions and timed delays as given in the process theory $\mathrm{TCP}^{\mathrm{drst}}$. We show that when dealing with such process specifications, we need not to resort to the specifications that comprise timed delay prefixes, but we can manipulate with the higher-order constructions directly. This gives rise to a ground-complete derived theory of communicating process with discrete stochastic time called $\mathrm{DTCP}^{\mathrm{dst}}_{\mathrm{rec}}(\mathcal{A}, \mathcal{V}, \mathcal{R}, \gamma)$. We illustrate the approach by modeling a discrete-time variant of the $G/G/1/\infty$ queue as a classical example.

Delayable deadlock, termination, and action prefix We define the delayable action prefix scheme $\overline{a}.\_$ for $a \in \mathcal{A}$ by taking the approach of [BM02] and putting $\overline{a}.p = \mu\, A.\{A = a.p + \sigma.A\}$. This process allows for the undelayable action a at every time slice. If the action is taken, then the process continues to behave as p and, otherwise, the process is delayed one unit of time. As the semantics of the processes is given per time unit, the process captures the intuition of a delayable action.

Of interest is the application of the encapsulation and the maximal progress operator on the delayable action prefix. For encapsulation one obtains $\partial_a(A) = \partial_a(a.p + \sigma.A) = 0 + \sigma.\partial_a(A) = \sigma.\partial_a(A)$. So, the resulting process can only delay arbitrary long. From the discussion on stochastic delays above, it should be clear that this process is not a stochastic delay as there are no winners. However, it plays a role in the theory as it occurs as an encapsulation of a delayable action. We represent this process in the theory as the constant process $\overline{0}$, called delayable deadlock, where $\overline{0} = \mu\, B.\{B = \sigma.B\}$. It is a neutral element in the alternative composition for a delayable action. To see this, assume that the definitions of $\overline{a}.p$ and $\overline{0}$ are as above. Then for $\overline{a}.p + \overline{0}$ we have that $A + B = (a.p + \sigma.A) + \sigma.B = a.p + \sigma.(A + B)$, i.e., $\overline{a}.p + \overline{0} = \overline{a}.p$.

For the application of the maximal progress operator we have $\theta_a(A) = \theta_a(a.p + \sigma.A) = a.\theta_a(p)$, i.e., it turns a delayable action prefix into an undelayable one. We can similarly define a delayable termination process constant as $\overline{1} = \mu\, C.\{C = 1 + \sigma.C\}$. It is a neutral element for the parallel composition [Mar08].

Stochastic delay prefix We specify stochastic delays as discussed above, i.e., as an expiration observed per unit of time in the same racing context. The stochastic delay prefix $\left[\begin{smallmatrix}W\\L\end{smallmatrix}\right].p$ is defined as the solution of the following guarded recursive equation $\left[\begin{smallmatrix}W\\L\end{smallmatrix}\right].p = \mu\, A.\{A = \sigma_L^W.p + \sigma_{W \cup L}.A\}$. The solution of this guarded recursive specification is an infinite racing timed transition scheme. The 'paths' in the probabilistic timed transition system induced by this scheme that end in p represent the duration of the stochastic delay. The underlying process is well-defined as the probability that a path of infinite length is taken in the probabilistic timed transition system that is induced by some assignment of distributions is equal to zero. This is because the probability distributions of the racing delays are aged by 1 in every state by the expiration of the timed delay $\sigma_{W \cup L}$ from above and $\lim_{n\to\infty} \mathrm{F}(n) = 1$ for every $\mathrm{F} \in \mathcal{F}$.

Example 7 We illustrate the manipulation with stochastic delays specified in this manner by demonstrating that $p_1 = [X].p + [Y].q$ and $p_2 = \left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right].(|p|_{\emptyset} + [Y].q) + [X, Y].(p + q) + \left[\begin{smallmatrix}Y\\X\end{smallmatrix}\right].([X].p + |q|_{\emptyset})$ are equivalent. We put $[X].p = \mu\, A_1.S$ for $A_1 = \sigma^X.p + \sigma_X.A_1 \in S$ and $[Y].q = \mu\, A_2.S$ for $A_2 = \sigma^Y.q + \sigma_Y.A_2 \in S$ for p_1 and $\left[\begin{smallmatrix}X\\Y\end{smallmatrix}\right].(|p|_{\emptyset} + [Y].q) = \mu\, A_3.S$, $[X, Y].(p + q) = \mu\, A_4.S$, and $\left[\begin{smallmatrix}Y\\X\end{smallmatrix}\right].([X].p + |q|_{\emptyset}) = \mu\, A_5.S$ for p_2. Then,

$$\begin{aligned} S = \{&A_1 = \sigma^X.p + \sigma_X.A_1,\ A_2 = \sigma^Y.q + \sigma_Y.A_2,\\ &A_3 = \sigma^X_Y.(|p|_{\emptyset} + A_2) + \sigma_{X,Y}.A_3,\ A_4 = \sigma^{X,Y}.(p + q) + \sigma_{X,Y}.A_4,\ A_5 = \sigma^Y_X.(A_1 + |q|_{\emptyset}) + \sigma_{X,Y}.A_5\} \end{aligned} \tag{7}$$

and we write $p_1 = \mu(A_1 + A_2).S$ and $p_2 = \mu(A_3 + A_4 + A_5).S$. We calculate

$$\begin{aligned} A_1 + A_2 &= (\sigma^X.p + \sigma_X.A_1) + (\sigma^Y.q + \sigma_Y.A_2)\\ &= \sigma^{X,Y}.(p + q) + \sigma^X_Y.(|p|_{\emptyset} + A_2) + \sigma^Y_X.(A_1 + |q|_{\emptyset}) + \sigma_{X,Y}.(A_1 + A_2) \end{aligned} \tag{8}$$

$$\begin{aligned} A_3 + A_4 + A_5 &= (\sigma^X_Y.(|p|_{\emptyset} + A_2) + \sigma_{X,Y}.A_3) + (\sigma^{X,Y}.(p + q) + \sigma_{X,Y}.A_4) + (\sigma^Y_X.(A_1 + |q|_{\emptyset}) + \sigma_{X,Y}.A_5)\\ &= \sigma^{X,Y}.(p + q) + \sigma^X_Y.(|p|_{\emptyset} + A_2) + \sigma^Y_X.(A_1 + |q|_{\emptyset}) + \sigma_{X,Y}.(A_3 + A_4 + A_5) \end{aligned} \tag{9}$$

Now, in view of the principles RDP^- and RSP for guarded recursion, p_1 and p_2 have the same solution.

Note that p_1 and p_2 do not specify explicitly any recursive equation and use only a stochastic delay prefix of the form $\left[\begin{smallmatrix}W\\L\end{smallmatrix}\right].\_$ for $W, L \subseteq \mathcal{V}$ with $W \neq \emptyset$ and $W \cap L = \emptyset$. Actually, we can manipulate stochastic delay prefixed terms directly in any context without having to resort to the recursive specifications at all (as originally proposed in [MV08, MV07]). However, unrestricted interaction between timed and stochastic delays requires the representation of stochastic delays in terms of guarded recursive specifications.

Example 8 Consider $\theta_I(\sigma^3.a.p + \genfrac{[}{]}{0pt}{}{W}{L}.b.q)$ for $I = \{a, b\}$. Let $\genfrac{[}{]}{0pt}{}{W}{L}.b.q = \mu\, B.\{B = \sigma_L^W.b.q + \sigma_{W\cup L}.B\}$. Then

$$\begin{aligned}\theta_I(\sigma^3.a.p + B) &= \theta_I(\sigma.\sigma^2.a.p + (\sigma_L^W.b.q + \sigma_{W\cup L}.B))\\ &= \theta_I(\sigma_L^W.(b.q + \sigma^2.a.p) + \sigma_{W\cup L}.(\sigma^2.a.p + B))\\ &= \sigma_L^W.\theta_I(b.q + \sigma^2.a.p) + \sigma_{W\cup L}.\theta_I(\sigma.\sigma.a.p + \sigma_L^W.b.q + \sigma_{W\cup L}.B)\\ &= \sigma_L^W.b.\theta_I(q) + \sigma_{W\cup L}.\theta_I(\sigma_L^W.(\sigma.a.p + b.q) + \sigma_{W\cup L}.(\sigma.a.p + B))\\ &= \sigma_L^W.b.\theta_I(q) + \sigma_{W\cup L}.(\sigma_L^W.b.\theta_I(q) + \sigma_{W\cup L}.\theta_I(\sigma.a.p + B))\\ &= \sigma_L^W.b.\theta_I(q) + \sigma_{W\cup L}.(\sigma_L^W.b.\theta_I(q) + \sigma_{W\cup L}.(\sigma_L^W.(a.\theta_I(p) + b.\theta_I(q) + \sigma_{W\cup L}.b.\theta_I(q)))).\end{aligned} \tag{10}$$

In the final step of the derivation we unfold B and apply the maximal progress operator. Even though no winner has expired, the maximal progress operator prohibits the expiration of the stochastic delay after time slice 3 as given by $\sigma_{W\cup L}.b.\theta_I(q)$.

Such an interaction between the timed and stochastic delays can also be used to specify a probabilistic behavior after a passage of time [MV09]. However, the theory cannot express a standard probabilistic choice between processes that do not allow passage of time.

Interaction between the prefix operators Next, we investigate a common type of synchronization between delayable action and stochastic delay prefixes.

Example 9 Consider $\partial_a(\overline{a}.p \parallel \genfrac{[}{]}{0pt}{}{W}{L}.q)$, where we suppress the synchronizing action as in standard compositional modeling. Let $\overline{a}.p = \mu A.\{A = a.p + \sigma.A\}$ and $\genfrac{[}{]}{0pt}{}{W}{L}.q = \mu B.\{B = \sigma_L^W.q + \sigma_{W\cup L}.B\}$. Then,

$$\begin{aligned}\partial_a(A \parallel B) &= \partial_a((a.p + \sigma.A) \parallel (\sigma_L^W.q + \sigma_{W\cup L}.B))\\ &= \partial_a(a.(p \parallel B) + \sigma_L^W.(A \parallel |q|_L) + \sigma_{W\cup L}.(A \parallel B))\\ &= \sigma_L^W.\partial_a(A \parallel |q|_L) + \sigma_{W\cup L}.\partial_a(A \parallel B),\end{aligned} \tag{11}$$

i.e., $\partial_a(\overline{a}.p \parallel \genfrac{[}{]}{0pt}{}{W}{L}.q) = \genfrac{[}{]}{0pt}{}{W}{L}.\partial_a(\overline{a}.p \parallel |q|_L)$.

If $q = \overline{b}.q'$ and the synchronization of a and b is defined, i.e., $\gamma(a, b) = c$ for some $c \in \mathcal{A}$, then it is also common to prioritize this communication. For example, this can be a communication via a channel, so naturally one wants this communication to happen as soon as it is enabled. In that case, one typically has a specification of the form $\theta_I(\partial_H(\overline{a}.p \parallel \genfrac{[}{]}{0pt}{}{W}{L}.\overline{b}.q'))$ for $H = \{a, b\}$ and $I = \{c\}$. Then, by extending the previous derivation with $\overline{b}.q' = \mu\, C.\{C = b.q' + \sigma.C\}$ one obtains:

$$\begin{aligned}\theta_I(\partial_H(A \parallel B)) &= \theta_I(\sigma_L^W.\partial_H(A \parallel |q|_L) + \sigma_{W\cup L}.\partial_H(A \parallel B))\\ &= \sigma_L^W.\theta_I(\partial_H((a.p + \sigma.A) \parallel (b.q' + \sigma.C))) + \sigma_{W\cup L}.\theta_I(\partial_H(A \parallel B))\\ &= \sigma_L^W.\theta_I(\partial_H(a.(p \parallel \overline{b}.q') + b.(a.p \parallel q') + c.(p \parallel q') + \sigma.(A \parallel C))) + \sigma_{W\cup L}.\theta_I(\partial_H(A \parallel B))\\ &= \sigma_L^W.\theta_I(c.\partial_H(p \parallel q') + \sigma.\partial_H(A \parallel C)) + \sigma_{W\cup L}.\theta_I(\partial_H(A \parallel B))\\ &= \sigma_L^W.c.\theta_I(\partial_H(p \parallel q')) + \sigma_{W\cup L}.\theta_I(\partial_H(A \parallel B)),\end{aligned} \tag{12}$$

i.e., $\theta_I(\partial_H(\overline{a}.p \parallel \genfrac{[}{]}{0pt}{}{W}{L}.\overline{b}.q')) = \genfrac{[}{]}{0pt}{}{W}{L}.c.\theta_I(\partial_H(p \parallel q'))$.

The composition of a stochastic delay prefixed process and the delayable deadlock constant can also be resolved in terms of stochastic delay processes. Unlike the compositions with delayable actions, the delayable deadlock propagates through the stochastic delay. We show the case of the alternative composition where the stochastic delay prefixed term $\genfrac{[}{]}{0pt}{}{W}{L}.q$ is defined as above in (11), and $\overline{0} = \mu\, C.\{C = \sigma.C\}$:

$$B + C = (\sigma_L^W.q + \sigma_{W\cup L}.B) + \sigma.C = \sigma_L^W.(q + C) + \sigma_{W\cup L}.(B + C), \tag{13}$$

i.e., $\genfrac{[}{]}{0pt}{}{W}{L}.q + \overline{0} = \genfrac{[}{]}{0pt}{}{W}{L}.(q + \overline{0})$.

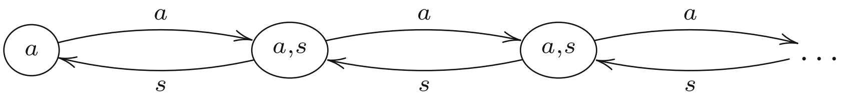

Fig. 3. Generalized semi-Markov model of the $G/G/1/\infty$ queue

Example 9 and the discussion above illustrate that the synchronization of passage of time of stochastic delay and delayable action prefixed terms can be handled without resorting to guarded recursive specifications comprising timed delay prefixes. Together with Example 7 and the above discussions involving delayable actions motivated us to develop a theory in the framework of $\mathrm{TCP}^{\mathrm{drst}}_{\mathrm{rec}}$ that directly manipulates with delayable action and stochastic delay prefixes [MV08, Mar08]. Unfortunately, such a theory is not closed for timed prefixes as it does not provide for time interpolation, as already observed in Example 8. Therefore, it cannot conservatively extend timed process theories employing the existing notions of behavioral equivalence.

The $G/G/1/\infty$ queue We proceed by specifying and solving the recursive specification of the discrete-time variant of the $G/G/1/\infty$ queue [Gly89, KD01, BD04, DK05b, MV07, MV09] to demonstrate the derived process theory. The queue can be compactly modeled by a generalized semi-Markov process [Gly89] given in Fig. 3. Here, a denotes the event of an arrival job and s is an event of a processed job. The states are labeled by the clocks that correspond to events. In every state the clocks with which the state is labeled are reset, whereas the others are updated typically using spent-lifetime semantics. After the expiration of a clock, the transition labeled by the name of the event is taken. The model shows that jobs arrive constantly in the queue and the server processes one job at a time.

We specify the discrete-time $G/G/1/\infty$ queue in our setting by using the components A, Q_0, and S:

$$A = \| [X].\overline{s}_1.A|_{\emptyset} \qquad Q_0 = \overline{r}_1.Q_1; \quad Q_{k+1} = \overline{r}_1.Q_{k+2} + \overline{s}_2.Q_k, \text{ if } k \geqslant 0 \qquad S = \overline{r}_2.[Y].\overline{s}_3.S \tag{14}$$

The equation for A models the arrival process distributed according to X. This delay corresponds to the clock a in the generalized semi-Markov representation of the process in Fig. 3. The process modeled by Q_0 is the standard representation of a queue. It comprises delayable actions and it is always able to receive a new job or to offer a job that has already been queued. Finally, the process given by S models the server that has processing time distributed according to Y. Its counterpart in Fig. 3 is given by the clock s. It is always ready to accept a job when it is idle. The specification of the $G/G/1/\infty$ queue itself is given by

$$G = \theta_I(\partial_H(A \parallel Q_0 \parallel S)), \tag{15}$$

where $\gamma(r_1, s_1) = c_1$, $\gamma(r_2, s_2) = c_2$, $H = \{s_1, r_1, s_2, r_2\}$, and $I = \{c_1, c_2, s_3\}$. Following Examples 7 and 9:

$$\begin{aligned} G = G_0 &= \| [X].c_1.c_2.G_1|_{\emptyset} \\ G_1 &= \left[{X \atop Y}\right].c_1.G_2 + \left[{Y \atop X}\right].s_3.G_0 + [X, Y].(c_1.s_3.c_2.G_1 + s_3.c_1.c_2.G_1) \\ G_k &= \left[{X \atop Y}\right].c_1.G_{k+1} + \left[{X, Y \atop \emptyset}\right].(c_1.s_3.c_2.G_k + s_3.c_1.c_2.G_k) + \left[{Y \atop X}\right].s_3.c_2.G_{k-1} \text{ for } k > 1, \end{aligned} \tag{16}$$

which gives the solution of the recursive relation in terms of processes comprising resolved races. Note that G_i denotes i jobs waiting to be processed. One can readily observe in the solution the behavior of the queue in terms of the racing condition between the delays modeling the arrival and processing times.

6. Towards reconciling real and stochastic time

It can be readily observed that the process theory $\mathrm{TCP}^{\mathrm{drst}}_{\mathrm{rec}}$ conservatively extends timed process algebras [BM02, BBR10] that comprise the same operations. Namely, in our setting both time additivity and time determinism are preserved as it holds that $\sigma^m.\sigma^n.p \leftrightarrow_t \sigma^{m+n}.p$ and $\sigma.p + \sigma.q \leftrightarrow_t \sigma.(p+q)$. Moreover, by employing guarded recursive equations we can introduce generally-distributed stochastic delays and express the race condition. All seems to be in place until we attempt to relate the derived stochastic delays of section 5 with standard stochastic delays, represented either by stochastic clocks or stochastic delays [HMR94, BBG97, DK05b, Bra02, LN00, BDHK06, MV08].

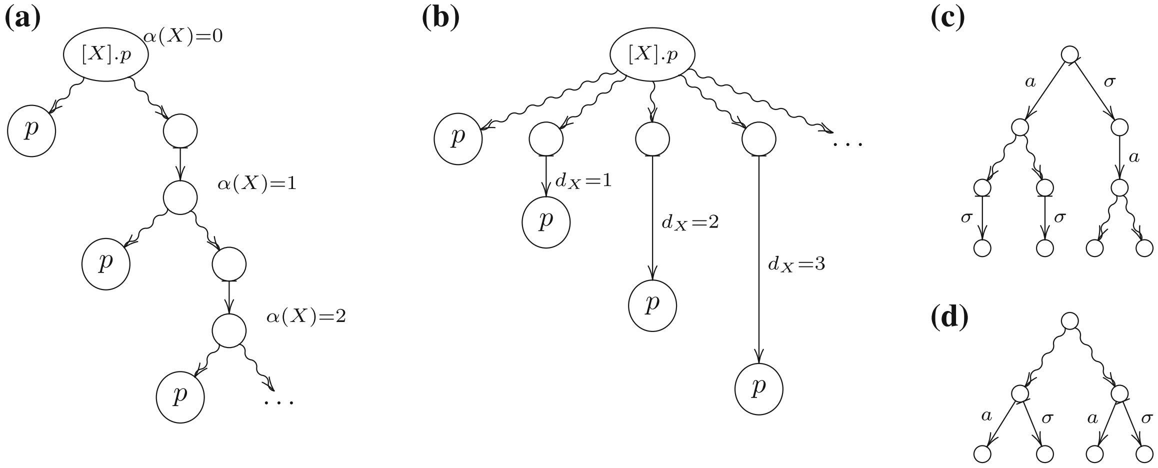

Fig. 4. **a** Derived versus **b** standard stochastic delays: Iconic representation of the probabilistic transition system that captures the resolution of the race condition for $[X].p$; **c** Iconic representation of $a.0 \parallel \sigma.(0 \oplus 0)$; and **d** Iconic representation of $a.0 \parallel (\sigma.0 \oplus \sigma.0)$. ($\oplus$ denotes a probabilistic choice)

The problem lies in the instantiation of the underlying probabilistic transition system and, more particularly, in the positioning of the internal probabilistic choices that result from the resolution of the race condition. We depict the situation in an iconic fashion in Fig. 4, where $\rightsquigarrow$ denotes internal probabilistic choices, whereas $\mapsto$ denotes passage of time.

In both cases, Fig. 4a and b, the expected time and probability to reach the process p are the same. Under the assumption that the exact time of making the probabilistic choice is not observable, both processes should be considered equivalent. Unfortunately, our notion of bisimulation does not relate the processes of Fig. 4a and Fig. 4b [Mar08]. Moreover, the problem to consistently interchange probabilistic and nondeterministic choices is far from trivial and it has been already studied in (the untimed setting) for more than a decade. The general consensus seems to be that mixing probabilistic and nondeterministic choices in concurrent processes creates compositionality issues, no matter whether the nondeterministic choices are resolved first [CSV07, Low93, Seg95], or the probabilistic ones [ABN11, MMSS96].

In [Low93], trace equivalences for processes comprising action and probabilistic choice are proposed, where the nondeterministic choice is resolved by employing so-called almighty randomized schedulers. Almighty schedulers can resolve nondeterministic choices in any possible way, without taking into consideration the process from which this choice stemmed from. In case a nondeterministic choice is duplicated by lifting of a probabilistic choice over it, it would be fair that the same decision is made in both instances of the same nondeterministic choice. For that reason, almighty schedulers often overestimate the probability that some event has happened. Moreover, the equivalences proposed in [CSV07, Low93, Seg95] are not congruences, mainly because of the use of the almighty schedulers. The first process-algebraic treatment is given in the seminal work [MMSS96], where processes are given as probability distributions over standard CSP processes [Hoa85]. The probabilistic choices were ‘lifted’ and resolved first, following a nondeterministic process with duplicated nondeterministic choices that were ‘pushed down’. Unfortunately, the idempotence law for the internal nondeterministic choice [Hoa85] was no longer in place, as the action prefix does not distribute over the probabilistic choice. This work is among the first to note the overestimation problem of almighty schedulers. Similar compositionality problems employing may-testing semantics arise in [ABN11].

The observations made in the reviewed research and the overestimation problem of the almighty schedulers, already noted in [MMSS96, Low93, Seg95], suggested that the power of the schedulers must be restricted. The compositionality problem in trace semantics [Seg95] was addressed first in [dAHJ01], which employs a scheduler that relies on the complete history of the system. A restriction is added in [GD09], such that the scheduler can only use local component information to resolve local nondeterministic choices. Other works consider specialized forms of parallel composition or impose structural model restrictions, for a detailed overview we refer to [Geo11]. A satisfactory conservative extension of CSP that employs ready-trace semantics and respects compositionality, is given in [GA10], building upon [MMSS96]. Recently, a full process algebraic treatment of this approach is given in [GA12].

The approach of [GA12] employs ready-trace semantics, a weaker behavioral relation than the standard ACP-style bisimulation. An initial suggestion would be to treat the timed delays and probabilistic choices using ready-trace semantics, as they can be considered unobservable, while preserving bisimilar behavior of the action transitions. However, if action transitions and timed delays can stem from the same state, which is one of our signature design choices, then the action transitions must also be allowed to distribute over probabilistic choices. The situation is depicted in Fig. 4c and d, in which the transition systems of $a.0 \parallel \sigma.(0 \oplus 0)$; and $a.0 \parallel (\sigma.0 \oplus \sigma.0)$ are given for a probabilistic choice operator $\oplus$ that (typically) has priority in the parallel composition. We observe that in order to relate the transitions systems in Fig. 4c and d, we have to be able to relate processes in which the action transition distributes over the probabilistic choice, leading to the same compositionality problems as discussed in the overview above. This suggests that in order to reconcile real and stochastic time, we may have to resort to a weak behavioral relation, like ready-trace semantics, and proceed along the lines of [GA10]. However, in that case, we no longer conservatively extend timed process theories (with bisimulation semantics), which was our original goal.

The opposite approach to embed timed delays in stochastic process theories, has been taken upon in [MV08]. It turns out, from the same reasons as above, that time additivity cannot be completely preserved. Instead the concept of context-sensitive interpolation is introduced as a compromise, which treats timed delays as being in a trivial race with each other. The signature axiom $\sigma^m.p + \sigma^n.q = \sigma^m.(p + \sigma^{n-m}.q)$ for $m < n$, states that we can interpolate if the context requires us to do so, but we cannot interpolate timed delays as desired, i.e., $\sigma^m.\sigma^n.p$ is no longer equivalent to $\sigma^{m+n}.p$.

7. Concluding remarks

We conservatively extended timed (ACP-style) process theories with stochastic time. To this end, we introduced the notion of racing timed delays, which present conditionally-distributed timed delays that are used to resolve the race condition. We defined stochastic delays as guarded recursive specifications employing racing timed prefixes. This enabled us to symbolically manipulate and resolve the race condition. We illustrated the main features of the theory by revisiting a discrete-time variant of the $G/G/1/\infty$ queue. The conservative extension of the theory came at a price that we could no longer relate the derived stochastic delays with the standard notion of stochastic delays. Namely, the employment of racing timed delays resulted in intertwining of the conditional probabilistic choices and the unit timed delays, whereas the standard stochastic delays make a probabilistic choice on the duration, given as an intact timed delay. To reconcile these two notions, a specific probabilistic refinement is needed that allows the interchange of nondeterministic and probabilistic choices in a setting with timed delays. We outlined the main characteristics of this refinement, taking into account recent work in this subject area. Future work should characterize this probabilistic refinement, which will enable a derivation of a performance model. Then, the process theory can be applied to model stochastic systems comprising general distributions and/or timeouts.

Acknowledgements

We thank Sonja Georgievska for insightful comments and discussions on early drafts of this paper.

References

[ABC+94] Ajmone Marsan M, Bianco A, Ciminiera L, Sisto R, Valenzano A (1994) A LOTOS extension for the performance analysis of distributed systems. IEEE/ACM Trans Netw 2(2):151–165

[ABC+95] Ajmone Marsan M, Balbo G, Conte G, Donatelli S, Franceschinis G (1995) Modelling with generalized stochastic Petri nets. Wiley, New York

[ABN11] Acciai L, Boreale M, De Nicola R (2011) Linear-time and may-testing in a probabilistic reactive setting. In: Proceedings of FMOODS 2011. Lecture Notes in Computer Science, vol 6722. Springer, Berlin, pp 29–43

[Bae05] Baeten JCM (2005) A brief history of process algebra. Theor Comput Sci 335:131–146

[BBD03] Bryans J, Bowman H, Derrick J (2003) Model checking stochastic automata. ACM Trans Comput Logic 4(4):452–492

[BBG97] Bravetti M, Bernardo M, Gorrieri R (1997) From EMPA to GSMPA: allowing for general distributions. In: Proceedings of PAPM'97, Enschede, pp 17–33

[BBK87] Baeten JCM, Bergstra JA, Klop JW (1987) On the consistency of Koomen's fair abstraction rule. Theor Comput Sci 51(1):129–176

[BBR10] Baeten JCM, Basten T, Reniers MA (2010) Process algebra: equational theories of communicating processes. In: Cambridge tracts in theoretical computer science, vol 50. Cambridge University Press, Cambridge

[BD04] Bravetti M, D'Argenio PR (2004) Tutte le algebre insieme: concepts, discussions and relations of stochastic process algebras with general distributions. In: Validation of stochastic systems—a guide to current research. Lecture Notes in Computer Science, vol 2925. Springer, Berlin, pp 44–88

[BDHK06] Bohnenkamp HC, D'Argenio PR, Hermanns H, Katoen J-P (2006) MODEST: a compositional modeling formalism for hard and softly timed systems. IEEE Trans Softw Eng 32:812–830

[BG98] Bernardo M, Gorrieri R (1998) A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time. Theor Comput Sci 202(1–2):1–54

[BKLL95] Brinksma E, Katoen J-P, Langerak R, Latella D (1995) A stochastic causality-based process algebra. Comput J 38(7):552–565

[BM02] Baeten JCM, Middelburg CA (2002) Process algebra with timing. In: Monographs in theoretical computer science. Springer, Berlin

[BR04] Baeten JCM, Reniers MA (2004) Timed process algebra (with a focus on explicit termination and relative timing). In: Proceedings of SFM 2004. Lecture Notes of Computer Science, vol 3185. Springer, Berlin, pp 59–97

[Bra02] Bravetti M (2002) Specification and analysis of stochastic real-time systems. PhD thesis, Università di Bologna

[BW90] Baeten JCM, Weijland WP (1990) Process algebra. Cambridge tracts in theoretical computer science, vol 18. Cambridge University Press, Cambridge

[CSKN05] Cattani S, Segala R, Kwiatkowska M, Norman G (2005) Stochastic transition systems for continuous state spaces and non-determinism. In: Proceedings of FoSSaCS'05. Lecture Notes of Computer Science, vol 3441. Springer, Berlin, pp 125–139

[CSV07] Cheung L, Stoelinga M, Vaandrager F (2007) A testing scenario for probabilistic processes. J ACM 54

[D'A03] D'Argenio PR (2003) From stochastic automata to timed automata: abstracting probability in a compositional manner. In: Proceedings of WAIT 2003, Buenos Aires

[dAHJ01] de Alfaro L, Henzinger T, Jhala R (2001) Compositional methods for probabilistic systems. In: Proceedings of CONCUR 2001. Lecture Notes in Computer Science, vol 2154. Springer, Berlin, pp 351–365

[DK05a] D'Argenio PR, Katoen J-P (2005) A theory of stochastic systems, part I: stochastic automata. Inform Comput 203(1):1–38

[DK05b] D'Argenio PR, Katoen J-P (2005) A theory of stochastic systems, part II: process algebra. Inform Comput 203(1):39–74

[GA10] Georgievska S, Andova S (2010) Retaining the probabilities in probabilistic testing theory. In: Proceedings of FOSSACS 2010. Lecture Notes in Computer Science, vol 6014. Springer, Berlin, pp 79–93

[GA12] Georgievska S, Andova S (2012) Probabilistic CSP: preserving the laws via restricted schedulers. In: Proceedings of MMB 2012. VDE Verlag (to appear). Available from: http://www.win.tue.nl/sgeorgie/.

[GD09] Giro S, D'Argenio PR (2009) On the expressive power of schedulers in distributed probabilistic systems. EPTCS 253:45–71

[Geo11] Georgievska S (2011) Probability and hiding in concurrent processes. PhD thesis, Eindhoven University of Technology

[Gly89] Glynn PW (1989) A GSMP formalism for discrete event systems. Proc IEEE 77(1):14–23

[Her02] Hermanns H (2002) Interactive Markov chains: the quest for quantified quality. Lecture Notes in Computer Science, vol 2428. Springer, Berlin

[Hil96] Hillston J (1996) A Compositional approach to performance modelling. Cambridge University Press, Cambridge

[HMR94] Hermanns H, Mertsiotakis V, Rettelbach M (1994) Performance analysis of distributed systems using TIPP. In: Proceedings of UKPEW'94. University of Edinburgh, pp 131–144

[Hoa85] Hoare CAR (1985) Communicating sequential processes. Prentice Hall, Englewood Cliffs

[How71] Howard RA (1971) Dynamic probabilistic systems. Wiley, New York

[KD01] Katoen JP, D'Argenio PR (2001) General distributions in process algebra. In: Lectures on formal methods and performance analysis. Lecture Notes in Computer Science, vol 2090. Springer, Berlin, pp 375–429

[KNP02] Kwiatkowska M, Norman G, Parker D (2002) PRISM: probabilistic symbolic model checker. In: Proceedings of TOOLS 2002. Lecture Notes in Computer Science, vol 2324. Springer, Berlin, pp 200–204

[LN00] López N, Núñez M (2000) NMSPA: a non-Markovian model for stochastic processes. In: Proceedings of ICDS 2000. IEEE, pp 33–40

[Low93] Lowe G (1993) Representing nondeterministic and probabilistic behaviour in reactive processes. Technical report PRG-TR-11-93, Oxford University Computing Labs

[Mar08] Markovski J (2008) Real and stochastic time in processs algebras for performance evaluation. PhD thesis, Eindhoven University of Technology

[MMSS96] Morgan C, McIver A, Seidel K, Sanders JW (1996) Refinement-oriented probability for CSP. Formal Aspects Comput 8:617–647

[MV06] Markovski J, de Vink EP (2006) Embedding real-time in stochastic process algebras. In: Proceedings of EPEW 2006. Lecture Notes of Computer Science, vol 4054. Springer, Berlin, pp 47–62

[MV07] Markovski J, de Vink EP (2007) Real-time process algebra with stochastic delays. In: Proceedings of ACSD 2007. IEEE, pp 177–186

[MV08] Markovski J, de Vink EP (2008) Extending timed process algebra with discrete stochastic time. In: Proceedings of AMAST 2008. Lecture Notes of Computer Science, vol 5140. Springer, Berlin, pp 268–283

[MV09] Markovski J, de Vink EP (2009) Performance evaluation of distributed systems based on a discrete real- and stochastic-time process algebra. Fundam Inform 95(1):157–186

[NS92] Nicollin X, Sifakis J (1992) An overview and synthesis of timed process algebras. In: Real-time: theory in practice. Lecture Notes of Computer Science, vol 600. Springer, Berlin, pp 526–548

[Seg95] Segala R (1995) Modeling and Verification of randomized distributed real-time systems. Phd thesis, MIT

[Yi91] Yi W (1991) CCS + time = an interleaving model for real-time systems. In: Proceedings of ICALP'91. Lecture Notes of Computer Science, vol 510. Springer, Berlin, pp 217–228

Received 8 December 2011

Revised 25 March 2012

Accepted 7 April 2012 by Peter Höfner, Robert van Glabbeek and Ian Hayes

Published online 2 July 2012