Expressing Priorities, External Probabilities and Time in Process Algebra via Mixed Open/Closed Systems

M. Bravetti

Technical Report UBLCS-2007-18

June 2007

Department of Computer Science
University of Bologna
Mura Anteo Zamboni 7
40127 Bologna (Italy)

The University of Bologna Department of Computer Science Research Technical Reports are available in PDF and gzipped PostScript formats via anonymous FTP from the area `ftp.cs.unibo.it:/pub/TR/UBLCS` or via WWW at URL `http://www.cs.unibo.it/`. Plain-text abstracts organized by year are available in the directory `ABSTRACTS`.

Recent Titles from the UBLCS Technical Report Series

2006-26 *FirmNet: The Scope of Firms and the Allocation of Task in a Knowledge-Based Economy*, Mollona, E., Marcozzi, A. November 2006.
2006-27 *Behavioral Coalition Structure Generation*, Rossi, G., November 2006.
2006-28 *On the Solution of Cooperative Games*, Rossi, G., December 2006.
2006-29 *Motifs in Evolving Cooperative Networks Look Like Protein Structure Networks*, Hales, D., Arteconi, S., December 2006.
2007-01 *Extending the Choquet Integral*, Rossi, G., January 2007.
2007-02 *Towards Cooperative, Self-Organised Replica Management*, Hales, D., Marcozzi, A., Cortese, G., February 2007.
2007-03 *A Model and an Algebra for Semi-Structured and Full-Text Queries (PhD Thesis)*, Buratti, G., March 2007.
2007-04 *Data and Behavioral Contracts for Web Services (PhD Thesis)*, Carpineti, S., March 2007.
2007-05 *Pattern-Based Segmentation of Digital Documents: Model and Implementation (PhD Thesis)*, Di Iorio, A., March 2007.
2007-06 *A Communication Infrastructure to Support Knowledge Level Agents on the Web (PhD Thesis)*, Guidi, D., March 2007.
2007-07 *Formalizing Languages for Service Oriented Computing (PhD Thesis)*, Guidi, C., March 2007.
2007-08 *Secure Gossiping Techniques and Components (PhD Thesis)*, Jesi, G., March 2007.
2007-09 *Rich Media Content Adaptation in E-Learning Systems (PhD Thesis)*, Mirri, S., March 2007.
2007-10 *User Interaction Widgets for Interactive Theorem Proving (PhD Thesis)*, Zacchiroli, S., March 2007.
2007-11 *An Ontology-based Approach to Define and Manage B2B Interoperability (PhD Thesis)*, Gessa, N., March 2007.
2007-12 *Decidable and Computational Properties of Cellular Automata (PhD Thesis)*, Di Lena, P., March 2007.
2007-13 *Patterns for Descriptive Documents: a Formal Analysis*, Dattolo, A., Di Iorio, A., Duca, S., Feliziani, A. A., Vitali, F., April 2007.
2007-14 *BPM + DM = BPDM*, Magnani, M., Montesi, D., May 2007.
2007-15 *A Study on Company Name Matching for Database Integration*, Magnani, M., Montesi, D., May 2007.
2007-16 *Fault Tolerance for Large Scale Protein 3D Reconstruction from Contact Maps*, Vassura, M., Margara, L., di Lena, P., Medri, F., Fariselli, P., Casadio, R., May 2007.
2007-17 *Computing the Cost of BPMN Diagrams*, Magnani, M., Montesi, D., June 2007.

Expressing Priorities, External Probabilities and Time in Process Algebra via Mixed Open/Closed Systems

M. Bravetti [1]

Technical Report UBLCS-2007-18

June 2007

Abstract

Defining operational semantics for a process algebra is often based either on labeled transition systems that account for interaction with a context or on the so-called reduction semantics: we assume to have a representation of the whole system and we compute unlabeled reduction transitions (leading to a distribution over states in the probabilistic case). In this paper we consider mixed models with states where the system is still open (towards interaction with a context) and states where the system is already closed. The idea is that (open) parts of a system "P" can be closed via an operator "$P \uparrow G$" that turns synchronized actions whose "handle" is specified inside "G" into prioritized reduction transitions (and, therefore, states performing them into closed states). We show that we can use the operator "$P \uparrow G$" to express multi-level priorities and external probabilistic choices (by assigning weights to handles inside G), and that, by considering reduction transitions as the only unobservable τ transitions, the proposed technique is compatible, for process algebra with general recursion, with both standard (probabilistic) observational congruence and a notion of equivalence which aggregates reduction transitions in a (much more aggregating) trace based manner. We also observe that the trace-based aggregated transition system can be obtained directly in operational semantics and we present the "aggregating" semantics. Finally, we discuss how the open/closed approach can be used to also express discrete and continuous (exponential probabilistic) time and we show that, in such timed contexts, the trace-based equivalence can aggregate more with respect to traditional lumping based equivalences over Markov Chains.

1. Università di Bologna, Dipartimento di Scienze dell'Informazione, Mura Anteo Zamboni 7, 40127 Bologna, Italy. E-mail: bravetti@cs.unibo.it

1 Introduction

In the literature, two main approaches are commonly used to define the semantics of a process algebra in an operational way. The first one, originally used to define the sematics of CCS [5], is based on *labeled* transition systems: the labels are used to represent both internal behaviours and possible behaviors obtained by interacting with a context. In the following we will refer to such labeled transition systems as *open* transition systems. The second one, used e.g. in [3], is based on the assumption to have a process algebraic representation of the whole system, and uses *unlabeled* reduction transitions to represent the system behavior, i.e. no behaviors possibly induced by a context are considered. In the following we will refer to such unlabeled reduction-based transition systems as *closed* transition systems. Note that, sometimes, reduction transitions can also be labeled: such labels however are not used to represent possible interactions with contexts, but are just informative labels describing what is happening internally in the system (they are useful to analyse the system behaviour by, e.g., model checking).

The distinction between open and closed transition systems is important, in particular, in the case we want to express quantitative behaviours as, e.g., prioritized and probabilistic choices. In the closed transition system of a whole system representation only prioritized behaviours (reductions) are represented and probabilistic choices are just internal: a possible representation is to make use of reduction transitions that lead to probability distribution over states (intead of just single states). On the contrary, in open transition systems, we have the problem of explicitely representing priorities and external probabilistic choices: *absolute* quantitative information, such as *priority levels* and *probabilistic weights*, associated to actions whose execution is just "potential", i.e. depends on the behavior of the context. Though very important from an expressive viewpoint, dealing with priorities and external probabilistic choices in open transition systems turned out to be problematic, especially when the issue of defining weak equivalences (that could be congruences) was considered (see, e.g., [2] for priorities): by directly attaching the quantitative information to actions the problem arises on (i) how to compute the quantitative value for synchronized actions and (ii) how to deal with distinguished τ actions carrying different quantitative information in the weak equivalence. A non-compositional way to deal, in a simple way, with the problem of expressing prioritized behaviours and external probabilistic choices in open transition systems is to use *schedulers*: we consider the open transition system of the whole system and we express weights and priority levels to be associated to actions in the scheduler definition. By applying such a scheduler to the (non-quantified) open transition system we obtain a (quantified) closed transition system as described above.

In this paper we propose a compositional solution to the problem above based on the idea of partially closing open transition systems via a process-algebraic operator. More precisely, we consider mixed models with states where the system is still open (towards interaction with a context) and states where the system is already closed. Moreover, we endow actions labeling open transitions with "handles" h: handles are used by the operator to identify the actions to which the quantitative information must be attached. The idea is that (open) parts of a system "P" can be closed via an operator "$P \uparrow G$" that turns τ_h actions whose handle h is specified inside "G" into reduction transitions that take *priority* with respect to labeled transitions (and, therefore, states performing such τ_h actions into closed states). Note that, as probably expected, only τ_h actions (e.g. in CCS synchronized actions), whose execution no longer depends on the context, can be closed by the operator.

In this way, we can use the operator "$P \uparrow G$" to express multi-level priorities by subsequent applications of the operator: actions closed by an inner application of the operator turn out to have higher priority with respect to actions closed by an outer application of the operator. For instance, by using a CCS-like parallel operator "$|$", $(a_{h_1}.P + b_{h_2}.Q|R)\uparrow\{h_2\}\uparrow\{h_1\}$, where output actions $\overline{a}$ and $\overline{b}$ occur in R with neutral handle $*$ (so that synchronization in "$|$", that involves also handles, gives rise to τ_{h_1} and τ_{h_2} actions), represents a prioritized choice between input actions a and b: if R offers synchronization (output) for both of them at the same time then the b action is executed otherwise the synchronization offered by R is executed. Moreover, we can extend the operator "$P \uparrow G$" to also express external probabilistic choices (at some priority level)

by assigning weights to handles inside G. For instance, $(a_{h_1}.P + b_{h_2}.Q|R) \uparrow \{\{(h_1, 1), (h_2, 3)\}\}$, where output actions $\overline{a}$ and $\overline{b}$ occur in R with neutral handle $*$, represents an external probabilistic choice between input actions a and b: if R offers synchronization (output) for both of them at the same time then they are executed with probabilities $.25$ (a action) and $.75$ (b action) otherwise the synchronization offered by R is executed with probability 1. Note that, since priority (and closure of external probability in a state) can be actually applied only when the synchronization context is considered and the involved actions turn from potential to internal, the proposed approach, which allows to put the "$P \uparrow G$" operator just outside this context (and not necessarily at the outermost syntactic level) does not "delay" the application of quantitative information with respect to the traditional approach based on attaching quantitative information directly to potential actions. Moreover, the usage of handles allows the closure operator to be applied selectively even to a single choice.

In the context of probabilistic process algebra literature, classifying states into standard states and quantified states is a natural language design choice that is commonly used to expess internal probabilistic choices (see e.g. [1]): this can be easily done by imposing probabilistic reduction transition to take priority with respect to standard action transitions. Moreover, in this respect, the approach that we adopt here gives us the following additional capabilities: (i) by giving the open/closed interpretation to states and by using an operator to both close the open system parts and, at the same time, assign a probabilistic quantification to them, we can additionally express external probability and also multi-level priority just as a consequence of the simple form of priority between the two kind of transitions; (ii) we can use the same technique in the reduced context of pure non-deterministic process algebra to give a solution to the long term open problem of expressing priorities in this context.

In the paper we consider full CCS with recursion: we use operator "$recX.P$" to express guarded and unguarded recursion. We use such a "core" process algebra (where we additionally attach handles to both internal and visible actions) to express open transition systems and we extend it in two subsequent steps: first we just consider non-deterministic prioritized reductions and a simple version of "$P \uparrow G$" where G is just a set of handles (giving us the ability to express multi-level prioirities only), then we also endow reductions with target probability distributions (thus expressing non-determinism among probabilistic reductions) and we extend the structure of set G in "$P \uparrow G$" to be composed by set of mappings from handles to weights. Note that, since the role of the core process algebra is just to compute τ_h transitions (possibly via process interaction) and "$P \uparrow G$" just acts on such transitions, i.e. we have a separation in two "layers" of the open transitions and of reduction (closed) transitions where the second ones are prioritized with respect to the first ones, our approach is not bound to the particular choice (CCS) of the core process algebra: we could have used any other process algebra.

Concerning equivalences, we are able to define weak equivalences that are compatible with the proposed technique by considering reduction transitions as the only unobservable τ transitions: the idea is that transitions of open states, even if τ_h, are still incomplete because they are not closed, i.e. we still have to apply quantification to them. More precisely, for both extensions of CCS we consider two kinds of weak equivalences that both deal with open transitions according to standard bisimulation and are distinguished for the treatment of reduction transitions. The first one aggregates reduction transitions in a trace-based manner: when a closed state is reached by an open transition, we just care about which open states are reachable by finite traces of reduction transitions and if non-escable divergence, e.g. a non-escapable loop of reduction transitions, can be reached. The intuition is that, being reduction transitions prioritized, it is natural to assume that a context cannot observe intermediate states in sequences of such transitions. The second one is simply standard observational congruence: Milner's one in the pure non-deterministic case and its probabilistic extension in [7] for transitions leading to probability distributions. Note that, even if obviously the trace-based equivalence aggregates much more than (probabilistic) observational congruence, we considered the latter to show that it is possible to make it compatible with multi-level priorities and external/internal probabilities.

As a main result we have that the trace-based equivalence is a congruence for the extension of CCS and that observational congruence is a congruence provided that "$\underline{0}$" is interpreted as

failure (so that it is allowed to be weakly equivalent to $recX.\tau.X$ without breaking congruence with respect to parallel) and successful termination "$\underline{1}$" is introduced in the process algebra.

We also observe that the aggregated transition system obtained by applying the trace-based equivalence to systems can be derived directly in operational semantics. By using an "aggregating" version of the operational semanitcs, we do not need to apply equivalence to reduce states, but the system state space is reduced directly by the operational semantics while we go from inner syntactic levels to outer ones and the system is progressively closed.

Finally, we build on the non-deterministic/probabilistic algebra by considering: discrete time, where reduction transitions take one time unit before reaching their probabilistic target, and continuous time, where reduction transitions take, instead, a probabilistic duration (denoted by the rate of an exponential distribution), to be executed. In both timed contexts we show that, by considering the trace-based equivalence, we can aggregate more with respect to the traditional lumping-based equivalences over Discrete Time or Continuous Time Markov Chains that correspond to a bisimulation-based matching of reductions. In particular, in the case of continuous time, if the semantics of parallel of reductions just gives rise to their non-deterministic interleaving (i.e. such a semantics it is not modified with respect to the untimed version in order to account for contemporaneous passage of time in reduction transitions) and just steady state probabilities are to be preserved by equivalence, then thanks to the *insensitivity property*, the trace-based equivalence just reduces to checking the mean overall duration of traces and, like in the untimed case, probabilities to reach non-reducible or divergent states.

The paper is structured as follows. Sect. 2, concerning management of multi-level Priorities in purely non-deterministic process algebra, presents the process algebra for non-deterministic open/closed systems and the related machinery: transition systems, the equivalences, syntax and semantics, congruence results and the aggregated semantics. Sect. 3 extends all the machinery of Sect. 2 to also deal with internal/external probabilistic choices. Finally, Sect. 4 concerns the usage of the closure operator to express discrete and continuous (exponential) time.

2 Multi-level Priorities

2.1 Partially open and partially closed non-deterministic transition systems

Definition 2.1 A non-deterministic open/closed transition system is a quadruple $(S, Lab, \longrightarrow_c, \longrightarrow_o)$, where

- S is a countable set of states,
- Lab is a countable set of labels of open transitions,
- $\longrightarrow_c \subseteq S \times S$ is a transition relation over states of S that represents *closed transitions*, i.e. reduction transitions,
- $\longrightarrow_o \subseteq S \times Lab \times S$ is a transition relation over states of S labeled over Lab that represents *open transitions*,

such that, for any $s \in S$, it holds that: $s \longrightarrow_c$ implies $\nexists l \in Lab : s \xrightarrow{l}_o$.

Note that, in the definition above and in the rest of the paper, we use: $s \xrightarrow{l}_o s'$ to stand for $(s, l, s') \in \longrightarrow_o$ and $s \xrightarrow{l}_o$ to stand for $\exists s' : s \xrightarrow{l}_o s'$ A similar notation is used for (unlabeled) reduction transition relation $\longrightarrow_c$. We assume predicate $\gg$ to single out reducible states, i.e. $s \gg$ if $s \longrightarrow_c$, $s \not\gg$ otherwise.

The constraint in Def. 2.1 guarantees that states of non-deterministic open/closed transition systems that have outgoing closed transitions (reducible states) cannot have outgoing open transitions and vice-versa. As a consequence system states can be classified into *closed system states* (states with outgoing closed transitions) and *open system states* (all other states). States with no outgoing transitions are assumed to be open (in Sect. 2.4 we will see an alternative approach where states with no outgoing transitions are assumed to be closed).

We use $\longrightarrow_c^+$ to denote the transitive closure of $\longrightarrow_c$ and $\longrightarrow_c^*$ to denote the transitive and reflexive closure of $\longrightarrow_c$. Predicate $\uparrow$ singles out (non-escapable) divergent states, i.e. $s \uparrow$ iff $\nexists s' : s \longrightarrow_c^* s' \wedge s' \not\gg$. Note that $s \uparrow$ implies $s \gg$. We assume predicate on states "$s\ (\not\gg \vee \uparrow)$" to be defined as "$(s \not\gg) \vee (s \uparrow)$".

Definition 2.2 A symmetric relation β over states of a non-deterministic open/closed transition system $(S, Lab, \longrightarrow_c, \longrightarrow_o)$ is a weak equivalence if, whenever $(s_1, s_2) \in \beta$:

- If $s_1 \xrightarrow{l}_o \longrightarrow_c^* s_1'\ (\not\gg \vee \uparrow)$ and $s_2 \not\gg$ then, for some s_2', with $s_2 \xrightarrow{l}_o \longrightarrow_c^* s_2'\ (\not\gg \vee \uparrow)$, we have: either $s_1' \uparrow$ and $s_2' \uparrow$, or $s_1' \not\gg$ and $s_2' \not\gg$ and $(s_1', s_2') \in \beta$.
- if $s_1 \longrightarrow_c^* s_1'\ (\not\gg \vee \uparrow)$ then, for some s_2', with $s_2 \longrightarrow_c^* s_2'\ (\not\gg \vee \uparrow)$, we have: either $s_1' \uparrow$ and $s_2' \uparrow$, or $s_1' \not\gg$ and $s_2' \not\gg$ and $(s_1', s_2') \in \beta$.

Two processes s_1, s_2 are weakly equivalent, written $s_1 \approx s_2$, iff (s_1, s_2) is included in some weak equivalence. ■

Note that in the case $s_1 \uparrow$, it is redundant to check, by a 0-length move of s_1, that s_2 can reach a (non escapable) divergent state (since $s_1 \uparrow$ implies that s_1 can also perform non 0-length moves to a divergent state); moreover in the case s_1 can reach a (non escapable) divergent state, it is redundant to allow s_2 to perform a 0-length move (since $s_2 \uparrow$ would imply that s_2 can also perform non 0-length moves to a divergent state). Finally, note that $s_1 \uparrow$ implies that $s_2 \uparrow$ (and viceversa).

Definition 2.3 Two states s_1, s_2 of a non-deterministic open/closed transition system $(S, Lab, \longrightarrow_c, \longrightarrow_o)$ are weakly congruent, written $s_1 \simeq s_2$, iff:

- If $s_1 \xrightarrow{l}_o \longrightarrow_c^* s_1'\ (\not\gg \vee \uparrow)$ then, for some s_2', with $s_2 \xrightarrow{l}_o \longrightarrow_c^* s_2'\ (\not\gg \vee \uparrow)$, we have: either $s_1' \uparrow$ and $s_2' \uparrow$, or $s_1' \approx s_2'$.
- if $s_1 \longrightarrow_c^+ s_1'\ (\not\gg \vee \uparrow)$ then, for some s_2', with $s_2 \longrightarrow_c^+ s_2'\ (\not\gg \vee \uparrow)$, we have: either $s_1' \uparrow$ and $s_2' \uparrow$, or $s_1' \approx s_2'$.

and a symmetrical constraint holds true for moves of s_2 as well. ■

Example 2.4 In the paper we will represent behaviors of states by means of process algebraic terms (as we will detail in the next Sect. 2.2), for the examples below the standard meaning of prefix (where τ represent a reduction transition), recursion and sum can be assumed. $\tau.l.\underline{0} + \tau.recX.\tau.X \not\approx l.\underline{0}$ [2] because $\tau.l.\underline{0} + \tau.recX.\tau.X \longrightarrow_c^* l.\underline{0}$ and it can reach a divergent state, while $l.\underline{0} \longrightarrow_c^* l.\underline{0}$ but it cannot reach a divergent state. On the contrary $\tau.l.\underline{0}+\tau.recX.(\tau.l.\underline{0}+\tau.X) \approx l.\underline{0}$ because $\tau.l.\underline{0}+\tau.recX.(\tau.l.\underline{0}+\tau.X) \longrightarrow_c^* l.\underline{0}$ and it cannot reach a divergent state (note that, since one of the two initial states cannot perform open transitions, they are not compared w.r.t. such kind of transitions).

2.2 Prioritized process algebra

The set of synchronization names $\mathcal{N}$ is ranged over by $a, b, c, \ldots$. The set of action names $\mathcal{N} \cup \{\overline{a} \mid a \in \mathcal{N}\} \cup \{\tau\}$, which includes input actions, output actions (identified by the overbar) and the special symbol τ to denote synchronized unnamed actions, is denoted by $\mathcal{AN}$, ranged over by $\alpha, \alpha', \ldots$. We extend complementation to the whole $\mathcal{AN}$ by assuming $\overline{\overline{a}} = a$ and $\overline{\tau} = \tau$. The finite set of handle names $\mathcal{H}$ is ranged over by $h, h', \ldots$. We assume synchronizing actions to yield unnamed actions and handlers of synchronizing actions to be composed by a given (arbitrarily defined) binary operator "$\propto$", i.e. when $\overline{a}_{h_1}$ synchronizes with a_{h_2} we get $\tau_{h_1 \propto h_2}$. From a modeling viewpoint it is convenient to adopt an operator "$\propto$" that has a neutral element, i.e. an handle $*$ (called neutral handle) such that $* \propto h = h \propto * = h$ for every handle h. The set of open actions (actions with handle) is denoted by $OAct = \{\alpha_h \mid \alpha \in \mathcal{AN} \wedge h \in \mathcal{H}\}$. The set of (all) actions is denoted by $Act = OAct \cup \{\tau\}$, that includes τ to express closed actions (actions

2. We assume syntactical precedence of prefix w.r.t. other operators when writing terms.

without handle). The set of term variables is Var, ranged over by $X, Y, \ldots$. The set $\mathcal{E}$ of behavior expressions, ranged over by E, F is defined by the following syntax.

$$E ::= \underline{0} \mid X \mid \alpha_h.E \mid \tau.E \mid E + E \mid E|E \mid E\backslash L \mid E[\varphi] \mid E\uparrow G \mid recX.E$$

where L is a finite subset of $\mathcal{N}$, G is a finite subset of $\mathcal{H}$ and φ is a relabeling function over Act such that:

- For every $\alpha \in \mathcal{AN}, h \in \mathcal{H}$ there exists α' such that $\varphi(\alpha_h) = \alpha'_h$.
- $\varphi(\tau) = \tau$
- $\varphi(\overline{\alpha}) = \overline{\varphi(\alpha)}$

The meaning of the operators is the standard one of [5, 6], where "$recX.E$" denotes recursion. The main differences and novelties are the following ones. Closed actions (actions τ) give rise to reduction (closed) transitions and are assumed to be prioritized with respect to open actions (actions α_h) that give rise to open transitions. The prioritization operator "$E\uparrow G$" turns unnamed open actions τ_h performable by E whose handle h is in G into closed actions τ thus turning them into prioritized actions and cutting possible unprioritized alternative behaviors. Closed terms are terms that do not include free variables (i.e. variables X not bound by a "$recX.E$" operator) and are called *processes*. The set $\mathcal{P}$ of processes is ranged over by P,Q,R. We omit trailing $\underline{0}$ when writing process terms.

The semantics of processes gives rise to the non-deterministic open/closed transition system $(\mathcal{P}, OAct, \longrightarrow_c, \longrightarrow_o)$, where $\longrightarrow_c$ (here denoted simply by $\longrightarrow$ and by explicit use of τ reduction labels) and $\longrightarrow_o$ (here denoted simply by $\longrightarrow$) are defined via structural operational semantics by the rules in Tables 1 and 2, plus symmetric rules. In Table 1 we take γ to range over the set of all actions Act: in the symmetric communication rule the handle of the τ transition is still $h_1 \propto h_2$, with h_1 handle of the output action and h_2 handle of the input action. $type(\gamma)$ yields the name in $\mathcal{N}$ of the action γ or τ if γ is an unnamed synchronized action (i.e. $\gamma = \tau$ or $\gamma = \tau_h$ for some handle h).

Example 2.5 The (non-deterministic open/closed) transition system of $\tau.P + \alpha_h.Q$ is the same as that of $\tau.P$. The transition system of $\tau_h.P + \alpha_{h'}.Q\uparrow\{h\}$, where $h' \neq h$, is the same as that of $\tau.P$.

The transition system of $(a_{h_1}.P + b_{h_2}.Q|R)\uparrow\{h_2\}\uparrow\{h_1\}$, where output actions $\overline{a}$ and $\overline{b}$ occurr in R with neutral handle $*$, represents a prioritized choice between input actions a and b: if R offers synchronization (output) for both of them at the same time then the b action is executed (since "$\uparrow\{h_2\}$" syntactically occurs before "$\uparrow\{h_1\}$") otherwise the synchronization offered by R is executed. For instance, if R is $\overline{a}_*.P' + \overline{b}_*.Q'$ then the transition system of the whole system is the same as that of $\tau.(Q|Q')$. If R is $\overline{a}_*.P'$ then the transition system of the whole system is the same as that of $\tau.(P|P')$. If R is $\overline{b}_*.P'$ then the transition system of the whole system is the same as that of $\tau.(Q|Q')$. The transition system of $(P|Q|\overline{a}_*)\uparrow\{h_2\}\uparrow\{h_1\}$, where input action a occurrs in P with handle h_1 and in Q with handle h_2, represents a prioritized choice between the two input actions a: if P and Q offer synchronization (input) for at the same time then the a action of Q is executed (since "$\uparrow\{h_2\}$" syntactically occurs before "$\uparrow\{h_1\}$") otherwise the synchronization offered by either P or Q is executed.

In general we can express multilevel priority by using operator $P\uparrow G$ to successively prioritize (and close) actions. We can use

$$P\uparrow G_n\uparrow G_{n-1}\ldots\uparrow G_1$$

to express that actions whose handle (after synchronization) belongs to G_n are at priority level n, actions whose handle belongs to G_{n-1} are at a lower priority level $n-1$, and so on...: actions whose handle belongs to G_1 are at the lowest (supposing that all actions used in P have been closed/prioritized) priority level 1.

Note that (i) closing/prioritizing actions makes it possible to abstract from them by means of a weak equivalence, so in a complete system we would expect all actions to be closed (ii) closing/prioritizing actions does not necessarily happen at the outermost syntactic level, like in the scenario above, where a similar effect could be obtained by just applying external (prioritized)

$$\gamma.P \xrightarrow{\gamma} P$$

$$\frac{P \xrightarrow{\alpha_h} P' \quad Q \not\gg}{P + Q \xrightarrow{\alpha_h} P'} \qquad \frac{P \xrightarrow{\alpha_h} P' \quad Q \not\gg}{P|Q \xrightarrow{\alpha_h} P'|Q}$$

$$\frac{P \xrightarrow{\overline{a}_{h_1}} P' \quad Q \xrightarrow{a_{h_2}} Q'}{P|Q \xrightarrow{\tau_{h_1 \propto h_2}} P'|Q'}$$

$$\frac{P \xrightarrow{\gamma} P'}{P\backslash L \xrightarrow{\gamma} P'\backslash L} \; type(\gamma) \notin L \qquad \frac{P \xrightarrow{\gamma} P'}{P[\varphi] \xrightarrow{\varphi(\gamma)} P'[\varphi]}$$

$$\frac{P \xrightarrow{\gamma} P' \quad \nexists h \in G : P \xrightarrow{\tau_h}}{P \uparrow G \xrightarrow{\gamma} P' \uparrow G}$$

$$\frac{P\{recX.P/X\} \xrightarrow{\gamma} P'}{recX.P \xrightarrow{\gamma} P'}$$

Table 1. Proposed variant of standard structural operational rules

$$\frac{P \xrightarrow{\tau} P'}{P + Q \xrightarrow{\tau} P'} \qquad \frac{P \xrightarrow{\tau} P'}{P|Q \xrightarrow{\tau} P'|Q}$$

$$\frac{P \xrightarrow{\tau_h} P'}{P \uparrow G \xrightarrow{\tau} P' \uparrow G} \; h \in G$$

Table 2. Additional rules for non-deterministic reduction transitions

schedulers to the transition system of P: synchronized actions should be closed at the innermost possible syntactic level so to make effective compositional reduction by means of the weak equivalence.

Theorem 2.6 Weak congruence "$\simeq$" is a congruence with respect to all the operators of the prioritized process algebra.

proof

We first show that weak equivalence $\approx$ is a congruence for non-reducible processes with respect to static operators.

We start with the parallel operator "$|$".

It is sufficient to show that:

$$\beta = \{(P_1|Q, P_2|Q) \mid P_1 \not\gg \wedge P_2 \not\gg \wedge Q \not\gg \wedge P_1 \approx P_2\}$$

is a weak bisimulation. Supposed that $P_1|Q \xrightarrow{\alpha_h} P_1'|Q' \longrightarrow^* P_1''|Q''(\not\gg \vee \uparrow)$ we have, due to the simple interleaving semantics of reduction transitions in "$|$": $P_1' \longrightarrow^* P_1''$ and $Q' \longrightarrow^* Q''$. We additionally consider $P_1'''(\not\gg \vee \uparrow)$ such that $P_1'' \longrightarrow^* P_1'''$.

We have three cases depending on how the $\xrightarrow{\alpha_h}$ is derived:

- $\alpha \neq \tau \wedge P_1 \xrightarrow{\alpha_h} P_1' \wedge Q' = Q$. Since $Q' = Q \not\gg$ we also have $Q'' = Q'$. Since $P_1 \approx P_2$, there exist P_2', P_2'' such that $P_2 \xrightarrow{\alpha_h} P_2' \longrightarrow^* P_2''(\not\gg \vee \uparrow)$ with either $P_1''' \uparrow \wedge P_2'' \uparrow$ or $P_1''' \not\gg \wedge P_2'' \not\gg$ and $P_1''' \approx P_2''$.
  Therefore we have: $P_2|Q \xrightarrow{\alpha_h} P_2'|Q \longrightarrow^* P_2''|Q$. If we now use $Q'' = Q \not\gg$ and $P_1''' \uparrow \iff P_1'' \uparrow$ and $P_1'' \not\gg \Rightarrow P_1''' = P_1''$, we have $P_2''|Q(\not\gg \vee \uparrow)$ and either $(P_1''|Q \uparrow \wedge P_2''|Q \uparrow)$ or $(P_1''|Q\beta P_2''|Q)$ and we are done.
- $\alpha \neq \tau \wedge Q \xrightarrow{\alpha_h} Q' \wedge P_1' = P_1$. Since $P_1' = P_1 \not\gg$ we also have $P_1'' = P_1'$. We have directly $P_2|Q \xrightarrow{\alpha_h} P_2|Q' \longrightarrow^* P_2|Q''$. If we now use $P_1'' = P_1 \not\gg$, we have $P_2|Q''(\not\gg \vee \uparrow)$ and either $(P_1|Q'' \uparrow \wedge P_2|Q'' \uparrow)$ or $(P_1|Q''\beta P_2|Q'')$ and we are done.
- $\alpha = \tau \wedge P_1 \xrightarrow{\alpha_h'} P_1' \wedge Q \xrightarrow{\overline{\alpha'}_h} Q'$ with $\alpha' \neq \tau$. Since $P_1 \approx P_2$, there exist P_2', P_2'' such that $P_2 \xrightarrow{\alpha_h'} P_2' \longrightarrow^* P_2''(\not\gg \vee \uparrow)$ with either $P_1''' \uparrow \wedge P_2'' \uparrow$ or $P_1''' \not\gg \wedge P_2'' \not\gg$ and $P_1''' \approx P_2''$.
  Therefore we have: $P_2|Q \xrightarrow{\tau_h} P_2'|Q' \longrightarrow^* P_2''|Q''$. If we now use $P_1''' \uparrow \iff P_1'' \uparrow$ and $P_1'' \not\gg \Rightarrow P_1''' = P_1''$, we have $P_2''|Q''(\not\gg \vee \uparrow)$ and either $(P_1''|Q'' \uparrow \wedge P_2''|Q'' \uparrow)$ or $(P_1''|Q''\beta P_2''|Q'')$ and we are done.

The proof for the other static operators, i.e. relabeling and restriction is just a much simplified version of the above proof as in the standard case. We instead report the proof of for the new operator "$P \uparrow G$" that is special because reduction transitions can be generated.

It is sufficient to show that:

$$\beta = \{(P_1 \uparrow G, P_2 \uparrow G) \mid P_1 \uparrow G \not\gg \wedge P_2 \uparrow G \not\gg \wedge P_1 \approx P_2\}$$

is a weak bisimulation. Let us suppose that $P_1 \uparrow G \xrightarrow{\alpha_h} P_1' \uparrow G \longrightarrow^* P_1'' \uparrow G(\not\gg \vee \uparrow)$. We preliminary consider $P_1'''(\not\gg \vee \uparrow)$ such that $P_1'' \longrightarrow^* P_1'''$. We have that, since reduction transitions executed by the "$\uparrow G$" operator are either copied or generated by closure, there exists $n \geq 0$ and non-reducible processes $P_1^1, \ldots, P_1^n$ such that $P_1' \longrightarrow^* P_1^1$ and, for $1 \leq i \leq n$, $P_1^i \xrightarrow{\tau_{h_i}} \longrightarrow^* P_1^{i+1}$ with $h_i \in G$, where $P_1^{n+1} = P_1'''$. We have that:

- $P_1 \xrightarrow{\alpha_h} P_1'$ and there is no $h' \in G$ such that $P_1 \xrightarrow{\tau_{h'}}$. Therefore there exist P_2', P_2^1 such that $P_2 \xrightarrow{\alpha_h} P_2' \longrightarrow^* P_2^1(\not\gg \vee \uparrow)$ with either $P_1^1 \uparrow \wedge P_2^1 \uparrow$ or $P_1^1 \not\gg \wedge P_2^1 \not\gg$ and $P_1^1 \approx P_2^1$.
  Note that, if $n \geq 1$ then $P_2^1 \not\gg$. Moreover there is no $h' \in G$ such that $P_2 \xrightarrow{\tau_{h'}}$ (easly proved by contradiction).
- There exist non-reducible processes $P_2^2, \ldots, P_2^n$ such that, for $1 \leq i \leq n-1$, we have $P_2^i \xrightarrow{\tau_{h_i}} \longrightarrow^* P_2^{i+1}$ with $P_1^{i+1} \approx P_2^{i+1}$. Finally, if $n \geq 1$, there exist P_2^{n+1} such that $P_2^n \xrightarrow{\tau_{h_i}} \longrightarrow^* P_2^{n+1}(\not\gg \vee \uparrow)$ with either $P_1^{n+1} \uparrow \wedge P_2^{n+1} \uparrow$ or $P_1^{n+1} \not\gg \wedge P_2^{n+1} \not\gg$ and $P_1^{n+1} \approx P_2^{n+1}$.
- In conclusion, we have $P_2 \uparrow G \xrightarrow{\alpha_h} P_2' \uparrow G \longrightarrow^* P_2^{n+1} \uparrow G$. We observe that $(P_1^{n+1} \uparrow G) \uparrow \Rightarrow (P_2^{n+1} \uparrow G) \uparrow$. This holds because, if P_1^{n+1} cannot reach, via reduction transitions or τ_h transitions with $h \in G$, a non-reducible state that performs no τ_h actions with $h \in G$; then also P_2^{n+1} cannot reach, via the same kind of transitions, such a state (easily proved, by contradiction, by subsequently matching $\xrightarrow{\tau_h} \longrightarrow^*$ transitions with $h \in G$, moving from equivalent states to equivalent states, similarly as done with the i-th indexed sequence above, and showing that P_1''' would reach a non-reducible state that performs no τ_h actions with $h \in G$). We also observe that $(P_1^{n+1} \uparrow G) \not\gg \Rightarrow (P_2^{n+1} \uparrow G) \not\gg$ and we recall $P_1''' = P_1^{n+1}$.
  If we now use such statements and $P_1'' \not\gg \Rightarrow P_1''' = P_1''$, we have $P_2^{n+1} \uparrow G(\not\gg \vee \uparrow)$ and either $(P_1'' \uparrow G) \uparrow \wedge (P_2^{n+1} \uparrow G) \uparrow$, or $P_1'' \uparrow G\beta P_2^{n+1} \uparrow G$ and we are done.

The congruence of weak congruence over all the operators can be showed by just applying the definition of weak congruence and, for static operators, by resorting to congruence of weak bisimulation. The only non-trivial case is parallel that we sketch in the following.

Let us suppose $P_1 \simeq P_2$.
Let us consider first $P_1|Q \xrightarrow{\alpha_h} P_1'|Q' \longrightarrow^* P_1''|Q''(\not\gg \vee \uparrow)$. We have three cases for the moves of P_1 and Q similar to the ones considered above for the congruence of weak bisimulation. In the case of a move of P_1, a corresponding move of P_2 must exist and either both divergent processes or non-reducible weak equivalent processes P_1'' and P_2'' are reached. In the case they are non-reducible and Q'' is non-reducible as well, due to congruence of weak bisimulation with respect to parallel we obtain $P_1''|Q''$ and $P_2''|Q''$.
Let us now consider $P_1|Q \longrightarrow^+ P_1'|Q'(\not\gg \vee \uparrow)$. In the case P_1 is reducible and originates some moves in the reduction sequence, a corresponding move of P_2 must exist and either both divergent processes or non-reducible weak equivalent processes P_1' and P_2' are reached. The proof then concludes as in the previous case.

Finally, concerning the recursion operator, from $E \simeq F$ (equivalence extended to terms with free variable occurences in the standard way) we derive $recX.E \simeq recX.F$ by showing that

$$\beta = \{(G\{recX.E/X\}, G\{recX.F/X\}) \mid G \text{ contains at most X free}\}$$

is a weak bisimulation up to $\approx$. This is done by induction on the derivation of transitions and by following a similar proof, for each operator as for its proof of congruence above.

2.3 Aggregating directly in operational semantics

The idea is that we can represent the behavior of a system in a minimal aggregated way by just saying which states s are reducible, i.e. such that $s \gg$, and by showing directly (i) which non-reducible states s' are reachable by reducible states s, i.e. $s \longrightarrow_c^+ s' \wedge s' \not\gg$, and (ii) weather a divergence state is reachable by reducible states s, i.e. $s \longrightarrow_c^+ s' \wedge s' \uparrow$ for some s'; instead of including all $\longrightarrow_c$ transitions in labeled transition systems. By doing this, we do not need to apply equivalence to reduce states, but the system state space is reduced directly by the operational semantics, while we go from inner syntactic levels to outer ones and the system is progressively closed.

Definition 2.7 A non-deterministic aggregated open/closed transition system is a quintuple $(S, Lab, Red, --\rightarrow_c, \longrightarrow_o)$, where

- S is a countable set of states,
- Lab is a countable set of labels of open transitions,
- Red is the subset of S of reducible states,
- $--\rightarrow_c \subseteq Red \times \{(S - Red) \cup \{\uparrow\}\}$ is a transition relation, leading directly from reducible states to non-reducible states or to divergence "$\uparrow$", that represents multiple *closed transitions*
- $\longrightarrow_o \subseteq (S - Red) \times Lab \times S$ is a transition relation labeled over Lab that represents *open transitions*,

Similarly as before, given such a transition system, we use predicate $\gg$ to single out reducible states, i.e. $s \gg$ if $s \in Red$, $s \not\gg$ otherwise. We use $\hat{s}$ to range over $S \cup \{\uparrow\}$.

The aggregated semantics of processes can be obtained, by determining Red and $--\rightarrow_c$ from $\longrightarrow_c$ as explained above and by just leaving $\longrightarrow_o$ unchanged, from the semantics of Sect. 2.2.

Equivalence over non-deterministic aggregated open/closed transition system can be directly defined (by simply applying the correspondance above) as follows.

Definition 2.8 A symmetric relation β over states of a non-deterministic aggregated open/closed transition system $(S, Lab, Red, --\rightarrow_c, \longrightarrow_o)$ is a weak equivalence if, whenever $(s_1, s_2) \in \beta$:

- If $s_1 \xrightarrow{l}_o s_1'$ and ($s_1' --\rightarrow_c \hat{s}_1''$ or $\hat{s}_1'' = s_1' \not\gg$) and $s_2 \not\gg$ then, for some s_2' and $\hat{s}_2''$, with $s_2 \xrightarrow{l}_o s_2'$ and ($s_2' --\rightarrow_c \hat{s}_2''$ or $\hat{s}_2'' = s_2' \not\gg$), we have either $\hat{s}_1' = \hat{s}_2' = \uparrow$ or $(\hat{s}_1', \hat{s}_2') \in \beta$.
- If $s_1 --\rightarrow_c \hat{s}_1'$ or $\hat{s}_1' = s_1 \not\gg$ then, for some s_2', with $s_2 --\rightarrow_c \hat{s}_2'$ or $\hat{s}_2' = s_2 \not\gg$, we have either $\hat{s}_1' = \hat{s}_2' = \uparrow$ or $(\hat{s}_1', \hat{s}_2') \in \beta$.

Two processes s_1, s_2 are weakly equivalent, written $s_1 \approx s_2$, iff (s_1, s_2) is included in some weak equivalence. ■

Definition 2.9 Two states s_1, s_2 of a non-deterministic aggregated open/closed transition system $(S, Lab, Red, --\rightarrow_c, \longrightarrow_o)$ are weakly congruent, written $s_1 \simeq s_2$, iff:

- If $s_1 \stackrel{l}{\longrightarrow}_o s_1'$ and ($s_1' --\rightarrow_c \hat{s}_1''$ or $\hat{s}_1'' = s_1' \not>>$) then, for some s_2' and $\hat{s}_2''$, with $s_2 \stackrel{l}{\longrightarrow}_o s_2'$ and ($s_2' --\rightarrow_c \hat{s}_2''$ or $\hat{s}_2'' = s_2' \not>>$), we have either $\hat{s}_1' = \hat{s}_2' = \uparrow$ or $\hat{s}_1' \approx \hat{s}_2'$.
- If $s_1 --\rightarrow_c \hat{s}_1'$ then, for some s_2', with $s_2 --\rightarrow_c \hat{s}_2'$, we have either $\hat{s}_1' = \hat{s}_2' = \uparrow$ or $\hat{s}_1' \approx \hat{s}_2'$.

and a symmetrical constraint holds true for moves of s_2 as well. ■

The aggregated semantics can be also obtained directly from processes as follows. The non-deterministic aggregated open/closed transition system is $(\mathcal{P}, OAct, Red, --\rightarrow_c, \longrightarrow_o)$, where the set of reducible states Red is taken to be the smallest subset of $\mathcal{P}$ that includes terms $\tau.P$ for every $P \in \mathcal{P}$ and is such that

$$\begin{array}{lcl} P \in Red & \Longrightarrow & P+Q, Q+P, P|Q, Q|P, P\backslash L, P[\varphi], P\uparrow G \in Red \\ P \stackrel{\tau_h}{\longrightarrow}_o \ \wedge \ h \in G & \Longrightarrow & P\uparrow G \in Red \\ P\{recX.P/X\} \in Red & \Longrightarrow & recX.P \in Red \end{array}$$

and $\longrightarrow_o$ (here denoted simply by $\longrightarrow$) is still defined by the rules of Table 1 plus symmetric rules; however, differently from Sect. 2.2, here we take γ to just range over the set of open actions $OAct$ (thus now excluding τ) and we have that predicate $\gg$ (re-defined above) is directly determined from set Red. Finally, $--\rightarrow_c$ (here denoted simply by $--\rightarrow$) is defined by the rules of Table 3 plus symmetric rules, starting from Red and $\longrightarrow_o$. In Table 3, given a context for terms P "$con(P)$", we take "$con(\uparrow)$" to just stand for $\uparrow$. For instance, "$\uparrow |Q$" stands for $\uparrow$. Moreover, we take "$\uparrow | \uparrow$" to stand for $\uparrow$.

Note that, we need to preliminarily define set Red and to base the definition of "$--\rightarrow$" on Red because, in order to establish if a term P can be the target of an aggregated transition that does not lead to divergence, we cannot just require that P does not perform any such aggregated transition. This because, if P is, e.g., $recX.\tau.X$ that does not perform any such aggregated transition (just like $\underline{0}$), then the check above does not work. If unguarded recursion is somehow disallowed (in such a way that also cannot be "dinamically" generated by application of $P\uparrow G$), then the preliminary definition of set Red is not necessary and non-reducibility of states can be just determined by absence of $--\rightarrow$ transitions.

2.4 A variant compatible with standard observational congruence

The machinery for multilevel priorities can be modified to make it compatible with standard Milner's observational congruence. From the one hand we loose the distinction between reducible and unreducible states (i.e. $recX.\tau.X$ is now equated by weak bisimulation to $\underline{0}$), from the other hand we observe also intermediate (reducible) state in τ paths, so the equivalence becomes sensitive to the branching structure of τ behaviours and the state space reduction by aggregation of τ transitions (and elimination of intermediate states) less effective.

The crucial modification that we have to do in order to make the process algebra of Sect. 2.2 compatible with standard observational congruence concerns the parallel operator. Modifying the behaviour of parallel is necessary because with the definition of Sect. 2.2, e.g., while $a_h.\underline{0}|recX.\tau.X$ has the same transition system of $recX.\tau.X$, $a_h.\underline{0}|\tau.\underline{0}$ has the same transition system of $\tau.a_h.\underline{0}$, hence observational congruence cannot be a congruence. The problem is that, with observational congruence, $\underline{0}$ (that is weakly bisimilar to $recX.\tau.X$) must be considered by the parallel as a failure event that makes the whole system fail: i.e. the parallel must be such that the behaviour of $P|\underline{0}$ is that of $\underline{0}$ for any P.

The wanted behaviour for parallel is obtained as follows. We interpret $\underline{0}$ as failure and we introduce in the syntax of behaviour expressions $\mathcal{E}$ (and of processes $\mathcal{P}$) successful termination

$$\frac{P \gg \quad \nexists P' : P \dashrightarrow P'}{P \dashrightarrow \uparrow}$$

$$\frac{P \not\gg}{\tau.P \dashrightarrow P} \qquad \frac{P \dashrightarrow \hat{P}'}{\tau.P \dashrightarrow \hat{P}'}$$

$$\frac{P \dashrightarrow \hat{P}'}{P + Q \dashrightarrow \hat{P}'}$$

$$\frac{P \dashrightarrow \hat{P}' \quad Q \not\gg}{P|Q \dashrightarrow \hat{P}'|Q} \qquad \frac{P \dashrightarrow \hat{P}' \quad Q \dashrightarrow \hat{Q}'}{P|Q \dashrightarrow \hat{P}'|\hat{Q}'}$$

$$\frac{P \dashrightarrow \hat{P}'}{P\backslash L \dashrightarrow \hat{P}'\backslash L} \qquad \frac{P \dashrightarrow \hat{P}'}{P[\varphi] \dashrightarrow \hat{P}'[\varphi]}$$

$$\frac{P \xrightarrow{\tau_h} P' \quad P' \uparrow G \not\gg}{P \uparrow G \dashrightarrow P' \uparrow G} \; h \in G \qquad \frac{P \xrightarrow{\tau_h} P' \quad P' \uparrow G \dashrightarrow \hat{P}'' \uparrow G}{P \uparrow G \dashrightarrow \hat{P}'' \uparrow G} \; h \in G$$

$$\frac{P \dashrightarrow \hat{P}'}{P \uparrow G \dashrightarrow \hat{P}' \uparrow G} \qquad \frac{P\{recX.P/X\} \dashrightarrow \hat{P}'}{recX.P \dashrightarrow \hat{P}'}$$

Table 3. Additional rules for aggregated non-deterministic reduction transitions

$\underline{1}$. Moreover we introduce a special action $\surd$, denoting successful termination, that we add to the set $OAct$ of open actions. The new operational semantics is obtained by modifying the rule for unsynchronized parallel transitions of Table 1 as follows:

$$\frac{P \xrightarrow{\alpha_h} P' \quad Q \xrightarrow{\gamma}}{P|Q \xrightarrow{\alpha_h} P'|Q} \; \gamma \in OAct$$

where now we have $\surd \in OAct$. An analogous modification of the rule for $+$ (that would lead the behaviour of $P + \underline{0}$ to be that of $\underline{0}$) is optional.

Moreover the following two standard rules, concerning generation of "$\surd$" moves, must be added (to Table 1):

$$\underline{1} \xrightarrow{\surd} \underline{0} \qquad \frac{P \xrightarrow{\surd} P' \quad Q \xrightarrow{\surd} Q'}{P|Q \xrightarrow{\surd} P'|Q'}$$

From the modeling viewpoint the modifications above require successful termination of processes $\underline{1}$ to be explicitly used by modelers: in a parallel a process that internally fails (i.e. becomes $\underline{0}$) immediately makes the whole system fail. For instance in $a_h.\underline{0}|P$ the whole system fails as soon as the a_h action is executed; in $a_h.\underline{1}|P$, instead, the system waits for termination of P after execution of a_h. Finally note that in the scenario $(a_h.\underline{1}|P)\backslash a$ the system waits for P to execute an output on a as desirable from a modeling viewpoint, i.e. the system does not fail immediately because the lefthand process cannot execute actions. This happens because the cause that disallows action execution is external (the restriction) and not internal.

Theorem 2.10 Milner's observational congruence is a congruence with respect to all the operators of the prioritized process algebra with successful termination.
proof
In the proof we denote Milner's observational congruence (where $OAct$ actions, that include "$\surd$", are the observable actions) by "$\simeq$" and Milner's weak bisimulation by "$\approx$". $OAct$ is ranged over by γ, $Act = OAct \cup \{\tau\}$ is ranged over by θ. We first show that $\approx$ is a congruence with respect to static operators.

We start with the parallel operator "$|$".
It is sufficient to show that:

$$\beta = \{(P_1|Q, P_2|Q) \mid P_1 \approx P_2\}$$

is a weak bisimulation. Supposed that $P_1|Q \xrightarrow{\theta} P_1'|Q'$, we have six cases depending on how the $\xrightarrow{\theta}$ is derived:

- $\theta = \alpha_h, \alpha \neq \tau, P_1 \xrightarrow{\alpha_h} P_1' \wedge Q' = Q$. We must have $Q \xrightarrow{\gamma}$ for some γ. Since $P_1 \approx P_2$, there exists P_2' such that $P_2 \stackrel{\alpha_h}{\Longrightarrow} P_2'$ and $P_1' \approx P_2'$. Therefore $P_2|Q \stackrel{\alpha_h}{\Longrightarrow} P_2'|Q$ and $(P_1'|Q\beta P_2'|Q)$ and we are done.
- $\theta = \alpha_h, \alpha \neq \tau, Q \xrightarrow{\alpha_h} Q' \wedge P_1' = P_1$. We must have $P_1 \xrightarrow{\gamma}$ for some γ. Since $P_1 \approx P_2$, $P_2 \stackrel{\gamma}{\Longrightarrow}$ for some γ. Therefore $P_2|Q \stackrel{\alpha_h}{\Longrightarrow} P_2|Q'$ and $(P_1|Q'\beta P_2|Q')$ and we are done.
- $\theta = \tau_h, P_1 \xrightarrow{\alpha'_h} P_1' \wedge Q \xrightarrow{\overline{\alpha'}_h} Q'$ with $\alpha' \neq \tau$. Since $P_1 \approx P_2$, there exists P_2' such that $P_2 \stackrel{\alpha'_h}{\Longrightarrow} P_2'$ and $P_1' \approx P_2'$. Therefore $P_2|Q \stackrel{\tau_h}{\Longrightarrow} P_2'|Q'$ and $(P_1'|Q'\beta P_2'|Q')$ and we are done.
- $\theta = \surd, P_1 \xrightarrow{\surd} P_1' \wedge Q \xrightarrow{\surd} Q'$. This case is totally analogous to the previous one.
- $\theta = \tau, P_1 \xrightarrow{\tau} P_1' \wedge Q' = Q$. Since $P_1 \approx P_2$, there exists P_2' such that $P_2 \stackrel{\hat{\tau}}{\Longrightarrow} P_2'$ and $P_1' \approx P_2'$. Therefore $P_2|Q \stackrel{\hat{\tau}}{\Longrightarrow} P_2'|Q$ and $(P_1'|Q\beta P_2'|Q)$ and we are done.
- $\theta = \tau, Q \xrightarrow{\tau} Q' \wedge P_1' = P_1$. We have immediately $P_2|Q \xrightarrow{\tau} P_2|Q'$ and $(P_1|Q'\beta P_2|Q')$ and we are done.

The proof for the other static operators, i.e. relabeling and restriction is just a much simplified version of the above proof as in the standard case. We instead report the proof of for the new operator "$P \uparrow G$" that is special because reduction transitions can be generated.

It is sufficient to show that:

$$\beta = \{(P_1 \uparrow G, P_2 \uparrow G) \mid P_1 \approx P_2\}$$

is a weak bisimulation. Supposed that $P_1 \uparrow G \xrightarrow{\theta} P_1' \uparrow G$, we have three cases depending on how the $\xrightarrow{\theta}$ is derived:

- $\theta = \gamma, P_1 \xrightarrow{\gamma} P_1'$. We must have that $\nexists h \in G : P_1 \xrightarrow{\tau_h}$. Since $P_1 \approx P_2$, there exists P_2' such that $P_2 \stackrel{\gamma}{\Longrightarrow} P_2'$ and $P_1' \approx P_2'$. Moreover, called P_2'' the intermediate state of the weak transition above such that $P_2'' \xrightarrow{\gamma}$ it must be that $\nexists h \in G : P_2'' \xrightarrow{\tau_h}$ (by contradiction, if such a transition existed then $P_2 \stackrel{\tau_h}{\Longrightarrow}$, hence $P_1 \stackrel{\tau_h}{\Longrightarrow}$, but P_1 is not allowed to perform τ actions and it does not perform the τ_h action). Therefore $P_2 \uparrow G \stackrel{\gamma}{\Longrightarrow} P_2' \uparrow G$ and $(P_1' \uparrow G\beta P_2' \uparrow G)$ and we are done.
- $\theta = \tau, P_1 \xrightarrow{\tau_h} P_1'$ with $h \in G$. Since $P_1 \approx P_2$, there exists P_2' such that $P_2 \stackrel{\tau_h}{\Longrightarrow} P_2'$ and $P_1' \approx P_2'$. Therefore $P_2 \uparrow G \stackrel{\tau}{\Longrightarrow} P_2' \uparrow G$ and $(P_1' \uparrow G\beta P_2' \uparrow G)$ and we are done.
- $\theta = \tau, P_1 \xrightarrow{\tau} P_1'$ with $h \in G$. Since $P_1 \approx P_2$, there exists P_2' such that $P_2 \stackrel{\hat{\tau}}{\Longrightarrow} P_2'$ and $P_1' \approx P_2'$. Therefore $P_2 \uparrow G \stackrel{\hat{\tau}}{\Longrightarrow} P_2' \uparrow G$ and $(P_1' \uparrow G\beta P_2' \uparrow G)$ and we are done.

The congruence of observational congruence over all the operators can be showed by just applying the definition of observational congruence and, for static operators, by resorting to congruence of weak bisimulation, as in the standard way. In the case of parallel, supposed $P_1 \simeq$

P_2, we consider $P_1|Q \xrightarrow{\theta} P_1'|Q'$. We have six cases for the moves of P_1 and Q similar to the ones considered above for the congruence of weak bisimulation.

Finally, concerning the recursion operator, from $E \simeq F$ (equivalence extended to terms with free variable occurences in the standard way) we derive $recX.E \simeq recX.F$ by showing that

$$\beta = \{(G\{recX.E/X\}, G\{recX.F/X\}) \mid G \text{ contains at most X free}\}$$

is a weak bisimulation up to $\approx$. This is done, as in the standard way, by induction on the derivation of transitions and by following a similar proof, for each operator as for its proof of congruence above.

3 Adding Probabilities

3.1 Partially open and partially closed non-deterministic and probabilistic transition systems

First of all we introduce the following notation that will be used in the rest of the paper. Let f be a partial function from an arbitrary domain $\mathcal{D}$ to real numbers $\mathbb{R}$. Given a subset D of $dom(f)$ and supposed that $\sum_{s \in D} f(s) \in \mathbb{R}$, we use $f(D)$ to denote such a sum.

A partial discrete probability distribution over a countable set of states S is a function $\sigma : S \longrightarrow [0,1]$ such that $\sigma(S) \leq 1$. A discrete probability distribution σ is a partial discrete probability distribution such that $\sigma(S) = 1$. We denote by $PDist_S$ the set of discrete probability distributions over states S.

In the case S is infinite, it is convinient to introduce the following notatation to denote discrete probability distributions in a finite way. Given a partial function f from S to $[0,1]$ such that $\sigma(dom(f)) = 1$, we use it to denote a probability distribution by writing σ_f defined as: $\sigma_f(s) = f(s)$ if $s \in dom(f)$, $\sigma_f(s) = 0$ otherwise.

Definition 3.1 A non-deterministic/probabilistic open/closed transition system is a quadruple $(S, Lab, \longrightarrow_c, \longrightarrow_o)$, where

- S is a countable set of states,
- Lab is a countable set of labels of open transitions,
- $\longrightarrow_c \subseteq S \times PDist_S$ is a transition relation from states of S to discrete probability distributions over S that represents *closed transitions*, i.e. reduction transitions,
- $\longrightarrow_o \subseteq S \times Lab \times S$ is a transition relation over states of S labeled over Lab that represents *open transitions*,

such that, for any $s \in S$, it holds that: $s \longrightarrow_c$ implies $\nexists l \in Lab : s \xrightarrow{l}_o$.

Note that, in the definition above and in the rest of the paper, we use: $s \longrightarrow_c \sigma$ to stand for $(s, \sigma) \in \longrightarrow_c$ and $s \longrightarrow_c$ to stand for $\exists \sigma : s \longrightarrow_c \sigma$. We assume predicate $\gg$ to single out reducible states, i.e. $s \gg$ if $s \longrightarrow_c$, $s \not\gg$ otherwise.

We extend predicates $P(s)$ defined on states to hold on discrete probability distributions over states as follows: $P(\sigma)$ iff $\forall\, s \in S.\, \sigma(s) > 0 \Rightarrow P(s)$. For instance, $\sigma \not\gg$ stands for $\forall\, s \in S.\, \sigma(s) > 0 \Rightarrow s \not\gg$. Moreover, given a predicate $P(s)$ defined on states, we take: S_P to denote the subset of S of states s that satisfy $P(s)$, i.e. $S_P = \{s \in S \mid P(s)\}$; σ_P to denote the partial discrete probability distribution obtained from σ by considering only probability associated to states s that satisfy $P(s)$, i.e. $\forall\, s \in S$ we have $\sigma_P(s) = \sigma(s)$ if $P(s)$, $\sigma_P(s) = 0$ otherwise. For instance, $S_{\not\gg}$ denotes the set of non-reducible states and $\sigma_{\not\gg}$ is the partial discrete probability distribution obtained from σ by considering only probability associated to non reducible states.

A finite trace tr of reduction transitions is a function $tr : \{1, \ldots, n\} \longrightarrow S$, for some $n \in \mathbb{N}^+$ (the length of the trace), such that for every $i \in \{1, \ldots, n-1\}$ there exists σ such that $tr(i) \longrightarrow_c \sigma$ and $\sigma(tr(i+i)) > 0$. We denote by Tr the set of such traces and by Tr_s the subset of all traces tr in Tr such that $tr(1) = s$. In the following we will denote the states of a trace tr just as tr_1, ..., tr_n standing for $tr(1), \ldots, tr(n)$. Moreover, given a trace tr of length n, we use tr_{fin} to denote its

final state tr_n (the only state of the trace that can be a non-reducible state) and $tr_{\leq i}$, with $i \leq n$, to denote the trace of length i that is a prefix of tr.

A (hystory dependent) scheduler $sched_s$ from a state s is a partial function $sched_s : Tr_s \rightharpoonup PDist_S$ such that $sched(tr) = \sigma$ implies $tr_{fin} \longrightarrow_c \sigma$ and satisfies: $tr \in dom(sched_s)$ implies $tr_{\leq n-1} \in dom(sched_s)$ and $sched_s(tr_{\leq n-1})(tr_n) > 0$, where n is the length of tr. Tr_{sched_s}, representing finite traces that can be scheduled going from s all the way until one of scheduler's halt states, is the subset of all traces tr in Tr_s such that $tr \notin dom(sched_s)$, $tr_{\leq n-1} \in dom(sched_s)$ and $sched_s(tr_{\leq n-1})(tr_n) > 0$, where n is the length of tr. The probability of a trace $tr \in Tr_{sched_s}$ of length n under a scheduler $sched_s$ is defined by $prob_{sched_s}(tr) = \prod_{1 \leq i \leq n-1} sched_s(tr_{\leq i})(tr_{i+i})$. [3] A scheduler $sched_s$ is *terminating* (by means of finite traces) if $\sum_{tr \in Tr_{sched_s}} prob_{sched_s}(tr) = 1$. [4] Terminating schedulers from s are ranged over by $tsched_s$.

We define $s \longrightarrow_c^* \sigma$, with $\sigma \in PDist_S$, to hold if and only if there exists a scheduler $tsched_s$ such that for every $s' \in S$ it holds $\sigma(s') = \sum_{tr \in Tr_{tsched_s} \wedge tr_{fin} = s'} prob_{tsched_s}(tr)$. The definition of $s \longrightarrow_c^+ \sigma$ is the same with the additional constraint of $tsched_s \neq \emptyset$. Predicate $\uparrow$ singles out (non-escapable) divergent states, i.e. $s \uparrow$ iff $\nexists \sigma : s \longrightarrow_c^* \sigma \wedge \sigma(S_{\not\gg}) > 0$. Note that $s \uparrow$ implies $s \gg$.

Given an equivalence relation β over states S, we say that two partial discrete probability distributions σ' and σ'' are equivalent, written $\sigma' \equiv_\beta \sigma''$ if, for every equivalence class $C \in S/\beta$, it holds that $\sum_{s \in C} \sigma'(s) = \sum_{s \in C} \sigma''(s)$.

Definition 3.2 An equivalence relation β over states of a non-deterministic/probabilistic open/closed transition system $(S, Lab, \longrightarrow_c, \longrightarrow_o)$ is a weak equivalence if, whenever $(s_1, s_2) \in \beta$:

- If $s_1 \xrightarrow{l}_o \longrightarrow_c^* \sigma(\not\gg \vee \uparrow)$ and $s_2 \not\gg$ then, for some σ', $s_2 \xrightarrow{l}_o \longrightarrow_c^* \sigma'(\not\gg \vee \uparrow)$ and $\sigma_{\not\gg} \equiv_\beta \sigma'_{\not\gg}$.
- If $s_1 \longrightarrow_c^* \sigma(\not\gg \vee \uparrow)$ then, for some σ', $s_2 \longrightarrow_c^* \sigma'(\not\gg \vee \uparrow)$ and $\sigma_{\not\gg} \equiv_\beta \sigma'_{\not\gg}$.

Two states s_1, s_2 are weakly equivalent, written $s_1 \approx s_2$, iff (s_1, s_2) is included in some weak equivalence. ■

Note that in the case $s_1 \uparrow$, it is redundant to check, by a 0-length move of s_1, that s_2 can reach a distribution where only divergent states have non-zero probability (since $s_1 \uparrow$ implies that s_1 can also perform non 0-length moves to a distribution where only divergent states have non-zero probability); moreover in the case s_1 can reach a distribution where only divergent states have non-zero probability, it is redundant to allow s_2 to perform a 0-length move (since $s_2 \uparrow$ would imply that s_2 can also perform non 0-length moves to a distribution where only divergent states have non-zero probability). Finally, note that $s_1 \uparrow$ implies that $s_2 \uparrow$ (and viceversa).

Definition 3.3 Two states s_1, s_2 of a non-deterministic/probabilistic open/closed transition system $(S, Lab, \longrightarrow_c, \longrightarrow_o)$ are weakly congruent, written $s_1 \simeq s_2$, iff:

- If $s_1 \xrightarrow{l}_o \longrightarrow_c^* \sigma(\not\gg \vee \uparrow)$ then, for some σ', $s_2 \xrightarrow{l}_o \longrightarrow_c^* \sigma'(\not\gg \vee \uparrow)$ and $\sigma_{\not\gg} \equiv_\approx \sigma'_{\not\gg}$.
- If $s_1 \longrightarrow_c^+ \sigma(\not\gg \vee \uparrow)$ then, for some σ', $s_2 \longrightarrow_c^+ \sigma'(\not\gg \vee \uparrow)$ and $\sigma_{\not\gg} \equiv_\approx \sigma'_{\not\gg}$.

and a symmetrical constraint holds true for moves of s_2 as well. ■

Example 3.4 Below we represent a reduction transition that leads to a probability distribution over states by means of a sum "$[p_1]P_1 + \ldots + [p_n]P_n$" ($\sum_{1 \leq i \leq n} p_i = 1$) [5] where each target state is prefixed by a probability. On the contrary non-deterministic choices between (open or reduction) transitions are still represented via standard "$P + Q$" sums (the formal definitions will be given in next Sect. 3.2). $[.3]l.\underline{0} + [.7]recX.[1]X \not\approx l.\underline{0}$ because the only distributions σ such that $\sigma(\not\gg \vee \uparrow)$ reachable by $[.3]l.\underline{0} + [.7]recX.[1]X$ assign probability .3 to $l.\underline{0}$ and probability

3. We assume an empty product to yield 1.
4. We assume an empty summation to yield 0.
5. In the case of a distribution where all probability is given to a single target term P the sum reduces to $[1]P$.

.7 to a divergent state, while the only distribution σ such that $\sigma(\not\gg \vee \uparrow)$ reachable by $l.\underline{0}$ assigns probability 1 to $l.\underline{0}$ (i.e. 0 probability is assigned to divergent states). On the contrary, $[.3]l.\underline{0} + [.7]recX.([1]l.\underline{0} + [1]X) \approx l.\underline{0}$ (where the choice inside recursion is non-deterministic) because the only distributions σ such that $\sigma(\not\gg \vee \uparrow)$ reachable by $[.3]l.\underline{0} + [.7]recX.([1]l.\underline{0} + [1]X)$ assign probability 1 to $l.\underline{0}$: no divergent states can be reached by the initial state (note that, since one of the two initial states cannot perform open transitions, they are not compared w.r.t. such kind of transitions).

3.2 Probabilistic prioritized process algebra

The set $\mathcal{E}$ of behavior expressions, ranged over by E, F is defined by the following syntax.

$$E ::= \underline{0} \mid X \mid \alpha_h.E \mid \sum_{i \in I}[p_i].E_i \mid E + E \mid E|E \mid E\backslash L \mid E[\varphi] \mid E \uparrow G \mid recX.E$$

where $\sum_{i \in I} p_i = 1$, L is a finite subset of $\mathcal{N}$, G is a finite set of partial functions from $\mathcal{H}$ to $\mathbb{R}^+$ (representing weights) whose domains are disjoint and φ is a relabeling function over $OAct$ such that:

- For every $\alpha \in \mathcal{AN}, h \in \mathcal{H}$ there exists α' such that $\varphi(\alpha_h) = \alpha'_h$.
- $\varphi(\overline{\alpha}) = \overline{\varphi(\alpha)}$

$\sum_{i \in I}[p_i].E_i$ represents a (discrete) probabilistic choice among terms E_i, where E_i is chosen with probability p_i. The prioritization operator "$E \uparrow G$", for every partial function $g \in G$, turns all open transitions τ_h performable by E whose handlers h are (distinguished and) in the domain of g, into a single closed reduction transition leading to a probability distribution over the target states of the open tranitions, where probabilities are proportional to the weights associated to the handlers by g. Moreover, as in the pure nondeterministic case, it cuts possible unprioritized alternative open behaviors. Again we assume the set $\mathcal{P}$ of processes (i.e. closed terms) to be ranged over by P,Q.

The semantics of processes gives rise to the non-deterministic/probabilistic open/closed transition system $(\mathcal{P}, OAct, \longrightarrow_c, \longrightarrow_o)$, where $\longrightarrow_c$ (here denoted simply by $\longrightarrow$ with no label) and $\longrightarrow_o$ (here denoted simply by $\longrightarrow$) are defined via structural operational semantics by the rules in Tables 1 and 4, plus symmetric rules. In Table 1, differently from Sect. 2.2, here we take γ to just range over the set of open actions $OAct$ (thus now excluding τ that is not considered in this section), and we consider $h \in G$ to be an abuse of notation for $h \in dom(g)$ for some $g \in G$, i.e. $h \in \bigcup_{g \in G} dom(g)$. In Table 4, given a context for terms P "$con(P)$" and a probability distribution σ, we take "$con(\sigma)$" to stand for the probability distribution such that: $con(\sigma)(con(P)) = \sigma(P)$, for every $P \in \mathcal{P}$; $con(\sigma)(P') = 0$, for every $P' \in \mathcal{P}$ that is not in the form $con(P)$ for some P. For instance, $\sigma|Q(P|Q) = \sigma(P)$, for every $P \in \mathcal{P}$; $\sigma|Q(P') = 0$ if P' is not in the form $P|Q$ for some P.

Example 3.5 The (non-deterministic/probabilistic open/closed) transition system of $\sum_{i \in I}[p_i].P + \alpha_h.Q$ is the same as that of $\sum_{i \in I}[p_i].P$. The transition system of $\tau_{h_1}.P_1 + \tau_{h_2}.P_2 + \alpha_{h'}.Q \uparrow \{\{(h_1,1),(h_2,3),(h_3,2)\}\}$, where h_1, h_2, h_3, h' are distinguished handlers, is the same as that of $[.25]P_1 + [.75]P_2$. The transition system of $\tau_{h_1}.P_1 + \tau_{h_2}.P_2 + \tau_{h_3}.P_3 + \tau_{h_4}.P_4 \uparrow \{\{(h_1,1),(h_2,3)\}\{(h_3,1),(h_4,1)\}\{(h',1)\}\}$, where h_1, h_2, h_3, h_4, h' are distinguished handlers, is the same as that of $([.25]P_1 + [.75]P_2) + ([.5]P_3 + [.5]P_4)$. The transition system of $\tau_{h_1}.P_1 + \tau_{h_2}.P_2 + \tau_{h_2}.P_3 + \alpha_{h'}.Q \uparrow \{\{(h_1,1),(h_2,3)\}\}$, where h_1, h_2, h' are distinguished handlers, is the same as that of $([.25]P_1 + [.75]P_2) + ([.25]P_1 + [.75]P_3)$.

The transition system of $(a_{h_1}.P + b_{h_2}.Q|R) \uparrow \{\{(h_1,1),(h_2,3)\}\}$, where output actions $\overline{a}$ and $\overline{b}$ occurr in R with neutral handle $*$, represents an external probabilistic choice between input actions a and b: if R offers synchronization (output) for both of them at the same time then they are executed with probabilities .25 (a action) and .75 (b action) otherwise the synchronization offered by R is executed. The transition system of $(a_{h_1}.P_1 + b_{h_2}.P_2 + c_{h_3}.P_3|R) \uparrow \{\{(h_3,1)\}\} \uparrow \{\{(h_1,1),(h_2,3)\}\}$, where output actions $\overline{a}$, $\overline{b}$ and $\overline{c}$ occurr in R with neutral handle $*$, represents a probabilistic/prioritized choice among input actions a, b and c: if R offers synchronization

$$\sum_{i\in I}[p_i].P_i \longrightarrow \sigma_{\{(P_i,\sum_{j\in I:P_j=P_i} p_j)|i\in I\}}$$

$$\frac{P \longrightarrow \sigma}{P+Q \longrightarrow \sigma} \qquad \frac{P \longrightarrow \sigma}{P|Q \longrightarrow \sigma|Q}$$

$$\frac{P \longrightarrow \sigma}{P\backslash L \longrightarrow \sigma\backslash L} \qquad \frac{P \longrightarrow \sigma}{P[\varphi] \longrightarrow \sigma[\varphi]}$$

$$\frac{dom(g)\cap\{h|P \xrightarrow{\tau_h}\} = D \neq \emptyset \quad \forall h\in D.\, P \xrightarrow{\tau_h} P_h}{P\uparrow G \longrightarrow \sigma_{\{(P_h,(\sum_{h'\in D:P_{h'}=P_h} g(h'))/g(D))|h\in D\}}\uparrow G} \; g\in G$$

$$\frac{P \longrightarrow \sigma}{P\uparrow G \longrightarrow \sigma\uparrow G} \qquad \frac{P\{recX.P/X\} \longrightarrow \sigma}{recX.P \longrightarrow \sigma}$$

Table 4. Additional rules for non-deterministic/probabilistic reduction transitions

(output) for all of them at the same time (in general if the synchronization set offered by R includes output c) then the c action is executed (since "$\uparrow\{\{(h_3,1)\}\}$" syntactically occurs before "$\uparrow\{\{(h_1,1),(h_2,3)\}\}$"); otherwise if output on c is not offered and both output on actions a and b are offered then a is executed with probability $.25$ and b with probability $.75$; finally if just output on action a or on action b is offered that the correspondig action is executed with probability 1.

In general we can express (external) probabilistic choices at multiple priority levels by using operator $P\uparrow G$ to successively prioritize (and close) actions. We can use

$$P\uparrow G_n\uparrow G_{n-1}\ldots\uparrow G_1$$

to express that actions whose handle (after synchronization) belongs to G_n are at priority level n and a non-deterministic/probabilistic choice among them occurs based on the weight functions in G_n, actions whose handle belongs to G_{n-1} are at a lower priority level $n-1$ and a non-deterministic/probabilistic choice among them occurs based on the weight functions in G_{n-1}, and so on...: actions whose handle belongs to G_1 are at the lowest (supposing that all actions used in P have been closed/prioritized) priority level 1 and a non-deterministic/probabilistic choice among them occurs based on the weight functions in G_1.

As far as the congruence property of "$\simeq$" is concerned, first of all we have to make the definition of $\longrightarrow^*$ and $\longrightarrow^+$ slightly more complicate by using *probabilistic* schedulers like in [7]. Such schedulers lead to an increased capability of equating states (without modifying the definition of equivalence): e.g. single system transitions can be matched even if the distribution of one of them is just obtained as a probabilistic combination of the distributions of the others (instead of matching transitions by requiring them to have the same distribution). The adoption of probabilstic schedulers is essential for the aggregation of multiple occurrences of the same states in a probabilistic choice, as e.g. in $[.2]P+[.8]P$ that has the same semantics as $[1]P$, (and ultimately for the aggregation of states belonging to the same equivalence class) to be compatible with equivalence (weak congruence). This because, if we, e.g., replace one occurrence of P with $P'\simeq P$, we get $[.2]P+[.8]P'$ that should be weakly congruent to $[.2]P+[.8]P$. However, while a scheduler for $[.2]P+[.8]P'$ (supposing P and P' to be reducible processes) can perform two different scheduling sequences of the behaviours of P and P' because such processes (that are the initial states of the two sequences) are different, a scheduler for $[.2]P+[.8]P$ cannot perform a corresponding pair of scheduling sequences because, after the initial reduction tran-

sition, it reaches with probability 1 a single state P, hence it cannot perform two distingushed scheduling sequences. Differently from the schedulers considered in Sect. 3.1, the adoption of probabilistic schedulers yields weak transitions that enjoy the following property (see [4] for a proof): given $p_i, i \in I$ with $\sum_{i\in I} p_i = 1$ and distributions $\sigma_i, i \in I$ such that $P \longrightarrow^+ \sigma_i$, there exists $P \longrightarrow^+ \sigma$ such that $\forall P'.\, \sigma(P') = \sum p_i \cdot \sigma_i(P')$, i.e. it is always possible to build a single "combined" probabilistic scheduler. The congruence property of the $\sum_{i\in I}[p_i]P_i$ and (also of the non-deterministic $+$ operator, since now non-deterministic choices are scheduled probabilistically) is a simple consequence, as described above, of the aggregation property of probabilistic schedulers. The congruence property of weak bisimulation with respect to the $P \uparrow G$ operator is proven similarly as in the pure non-deterministic case (see Sect. 2.2). The only significant difference concerns the derivation of the reduction move $P_2' \uparrow G \longrightarrow^* P_2'' \uparrow G$ from the initial move $P_1' \uparrow G \longrightarrow^* P_1'' \uparrow G$. Such a move determines sequences of schedulers for P_1, where every scheduler in the sequence reach open states performing τ_h transitions with $h \in dom(g)$ for some $g \in G$. Since equivalent schedulers exist for P_2', they must be used to form a unique scheduler for $P_2' \uparrow G$, i.e. when the τ_h transitions are turned into probabilistic reductions. This can be done by exploiting the aggregation property of probabilstic schedulers: states belonging to the same (weak) equivalence class in the target distribution of a scheduler of P_1' are starting states for (in general) different schedulings for the next scheduler in the sequence (as determined by the global weak transition performed by $P_1' \uparrow G$); since the states are equivalent such schedulings can be aggregated into a single one according to the distribution of the states in the class and the obtained schuduling used for the states in the same equivalence class in the target distribution of the scheduler of P_2' corresponding to that initially considered for P_1'.

Moreover, the congruence for the parallel operator is crucially based on the adoption of schedulers with partial visibility. The definition of $\longrightarrow^*$ and $\longrightarrow^+$ must be further complicated by additionally requiring that the corresponding scheduler satisfies the following *partial visibility condition*: the decision about the probabilistic reduction of a given (sequential) process to be performed in a state must depend only on the state of such a process and on the hystory of the states of such a process. In general, when such a scheduler reaches a state: first decides which (sequential) process must perform a probabilistic reduction (this decision can depend on the whole state and on the history of whole states like for schedulers defined in Sect. 3.1), then decides which probabilistic reduction of the chosen process is to be performed by using partial visibility as explained above. Such a property is natural, since, like for probabilities, the decisions about the choice of the reductions to be performed on a process should not depend on the decisions about the choice of the reductions to be performed in the other processes.

If such a partial visibility is assumed, then the congruence property of "$\simeq$" with respect to all the operators of the probabilistic prioritized process algebra can be proven similarly as for the pure non-deterministic case (see Sect. 2.2).

In the case of the parallel operator "$|$" we still have to show that:

$$\beta = \{(P_1|Q, P_2|Q) \mid P_1 \not\gg \wedge P_2 \not\gg \wedge Q \not\gg \wedge P_1 \approx P_2\}$$

is a weak bisimulation. Supposed that $P_1|Q \xrightarrow{\alpha_h} P_1'|Q' \longrightarrow^* \sigma(\not\gg \vee \uparrow)$ we first consider $\hat{\sigma}$ such that $P_1'|Q' \longrightarrow^* \hat{\sigma}$, $\hat{\sigma}_{\not\gg} = \sigma_{\not\gg}$ for all $(P_1''|Q'')$ with $\hat{\sigma}(P_1''|Q'') > 0$ we have $P_1''(\not\gg \vee \uparrow)$ and $Q''(\not\gg \vee \uparrow)$. Such $\hat{\sigma}$ is easily obtained by lengthening the probabilistic scheduler yielding σ. Due to the interleaving semantics of reduction transitions in "$|$" and to the partial visibility of schedulers there exist $\sigma_1(\not\gg \vee \uparrow)$ and $\sigma'(\not\gg \vee \uparrow)$ such that $P_1' \longrightarrow^* \sigma_1$ and $Q' \longrightarrow^* \sigma'$ and $\hat{\sigma}_{\not\gg} = (\sigma_1|\sigma')_{\not\gg}$, where $\sigma_1|\sigma'$ is defined as: $\forall P_1'', Q'' \in \mathcal{P}.\, \sigma_1|\sigma'(P_1''|Q'') = \sigma_1(P_1'') \cdot \sigma'(Q'')$. Such σ_1 and σ' are easily obtained by considering the probabilistic schedulers obtained by projecting the scheduler yielding $\hat{\sigma}$ on the sequential processes executed by P_1' (and then then scheduling them in an arbitrary way) and Q', respectively. The projection of a scheduler on a sequential process just yields a scheduler for the sequential process that, given a sequence of states traversed in the sequential process, takes the same probabilistic decision as the initial scheduler takes (assuming that such a sequential process will perform the next move) for any sequence of states of the whole system such that the sequential process traverses the given sequence of states (the decision is

the same for any choice of such a global sequence, due to the partial visibility property). We have then the same three cases, as for the the pure non-deterministic case, depending on how the $\xrightarrow{\alpha_h}$ is derived. In the most complex situation, i.e. $\alpha = \tau$, we have that both σ_1 and σ' can be yielded by non-empty schedulers and that there exists σ_2 such that $P_2' \longrightarrow^* \sigma_2(\not\gg \vee \uparrow)$ and $\sigma_{1\not\gg} \equiv_{\approx} \sigma_{2\not\gg}$. We derive, by scheduling the two schedulers for parallel processes in an arbitrary way, $P_1'|Q' \longrightarrow^* \sigma_2|\sigma'(\not\gg \vee \uparrow)$. Moreover $\sigma_{\not\gg} = \hat{\sigma}_{\not\gg} = (\sigma_1|\sigma')_{\not\gg} \equiv_\beta (\sigma_2|\sigma')_{\not\gg}$ (because equivalence classes of β are made of terms $P''|Q''$ where P'' are processes of the same equivalence class of $\approx$ and Q'' is fixed).

A counterexample showing that partial visibility of schedulers is needed is shown below.

Example 3.6 The system $[.2]([1]a + [1]b) + [.8]c|([.1]a + [.9]b) + [1]c \longrightarrow^* \sigma$ such that $\sigma(a|a) = .2 \cdot .1, \sigma(b|b) = .2 \cdot .9, \sigma(c|c) = .8$ (and σ is 0 for all other terms), i.e. it can perform, by means of an appropriate terminating scheduler, the probabilistic execution $[.2]([.1][1](a|a) + [.9][1](b|b)) + [.8][1](c|c)$. Note that such a scheduler uses the knowledge about the state chosen in the lefthand process to decide about the probabilistic reduction transition to perform in the righthand process and then it does the vice-versa. If we consider the process $([.2]a + [.8]c) + ([.2]b + [.8]c)$, that is equivalent to the lefthand process, no scheduler would allow to reach such a distribution σ. This because no matter which process is scheduled first, if it reaches a or b, then no scheduling exists for the other one that forces it to choose the same action.

3.3 Aggregating directly in operational semantics

The idea is that, similarly as in the purely non-deterministic case, we can represent the behavior of a system in a minimal aggregated way by just saying which states s are reducible, i.e. such that $s \gg$, and by showing directly which distributions $\hat{\sigma}$ over non-reducible states and divergence (denoted by $\uparrow$) are reachable by reducible states s, i.e. $\hat{\sigma} \in PDist_{S_{\not\gg} \cup \{\uparrow\}}$ such that $s \longrightarrow_c^+ \sigma \wedge \sigma(\not\gg \vee \uparrow)$ and $\sigma_{\not\gg} = \hat{\sigma}_{\not\gg}$ instead of including all $\longrightarrow_c$ transitions in labeled transition systems. By doing this, we do not need to apply equivalence to reduce states, but the system state space is reduced directly by the operational semantics, while we go from inner syntactic levels to outer ones and the system is progressively closed. Note however, that, in the case probabilistic schedulers are adopted, since infinite schedulings are possible, in the general case, we have (continuously) infinite $\hat{\sigma}$ distributions reachable by states. Adopting the non-probabilistic schedulers of Sect. 3.1 does not solve completely the problem, since, e.g., $RecX.[.2]X + [.8]([1]a + [1]b)$ would reach a (enumerable) infinite number of $\hat{\sigma}$ distributions too. Only by restricting to the case where all choices are purely probabilistic (see below), we can be sure of branching finiteness.

Definition 3.7 A non-deterministic/probabilistic aggregated open/closed transition system is a quintuple $(S, Lab, Red, --\rightarrow_c, \longrightarrow_o)$, where

- S is a countable set of states,
- Lab is a countable set of labels of open transitions,
- Red is the subset of S of reducible states,
- $--\rightarrow_c \subseteq Red \times PDist_{(S-Red)\cup\{\uparrow\}}$ is a transition relation, leading directly from reducible states to discrete probability distributions over non-reducible states and divergence "$\uparrow$", that represents multiple *closed transitions*
- $\longrightarrow_o \subseteq (S - Red) \times Lab \times S$ is a transition relation labeled over Lab that represents *open transitions*,

As usual, we use predicate $\gg$ to single out reducible states, i.e. $s \gg$ if $s \in Red$, $s \not\gg$ otherwise. We use $\hat{\sigma}$ to range over $PDist_{(S-Red)\cup\{\uparrow\}}$.

The aggregated semantics of processes can be obtained, by determining Red and $--\rightarrow_c$ from $\longrightarrow_c$ as explained above and by just leaving $\longrightarrow_o$ unchanged, from the semantics of Sect. 2.2.

Equivalence over non-deterministic/probabilistic aggregated open/closed transition system can be directly defined (by simply applying the correspondance above) as follows.

Definition 3.8 A symmetric relation β over states of a non-deterministic/probabilistic aggregated open/closed transition system $(S, Lab, Red, --\rightarrow_c, \longrightarrow_o)$ is a weak equivalence if, whenever $(s_1, s_2) \in \beta$:

- If $s_1 \xrightarrow{l}_o s_1'$ and $(s_1' --\rightarrow_c \hat{\sigma}$ or $\hat{\sigma}_{\not\gg}(s_1') = 1)$ and $s_2 \not\gg$ then, for some s_2' and $\hat{\sigma}'$, with $s_2 \xrightarrow{l}_o s_2'$ and $(s_2' --\rightarrow_c \hat{\sigma}'$ or $\hat{\sigma}'_{\not\gg}(s_2')$=1), we have $\hat{\sigma}_{\not\gg} \equiv_\beta \hat{\sigma}'_{\not\gg}$.
- If $s_1 --\rightarrow_c \hat{\sigma}$ or $\hat{\sigma}_{\not\gg}(s_1') = 1$ then, for some $\hat{\sigma}'$, with $s_2 --\rightarrow_c \hat{\sigma}'$ or $\hat{\sigma}'_{\not\gg}(s_2') = 1$, we have $\hat{\sigma}_{\not\gg} \equiv_\beta \hat{\sigma}'_{\not\gg}$.

Two processes s_1, s_2 are weakly equivalent, written $s_1 \approx s_2$, iff (s_1, s_2) is included in some weak equivalence. ■

Definition 3.9 Two states s_1, s_2 of a non-deterministic/probabilistic aggregated open/closed transition system $(S, Lab, Red, --\rightarrow_c, \longrightarrow_o)$ are weakly congruent, written $s_1 \simeq s_2$, iff:

- If $s_1 \xrightarrow{l}_o s_1'$ and $(s_1' --\rightarrow_c \hat{\sigma}$ or $\hat{\sigma}_{\not\gg}(s_1')$=1) then, for some s_2' and $\hat{\sigma}'$, with $s_2 \xrightarrow{l}_o s_2'$ and $(s_2' --\rightarrow_c \hat{\sigma}'$ or $\hat{\sigma}'_{\not\gg}(s_2')$=1), we have $\hat{\sigma}_{\not\gg} \equiv\approx \hat{\sigma}'_{\not\gg}$.
- If $s_1 --\rightarrow_c \hat{\sigma}$ then, for some $\hat{\sigma}'$, with $s_2 --\rightarrow_c \hat{\sigma}'$, we have $\hat{\sigma}_{\not\gg} \equiv\approx \hat{\sigma}'_{\not\gg}$.

and a symmetrical constraint holds true for moves of s_2 as well. ■

The aggregated semantics can be also obtained directly from processes similarly as in the non-deterministic case. In the following we show how this can be done in the pure probabilistic case, i.e. for processes such that: (i) we have at most one probabilistic choice occurring (unguarded) in the scope of non deterministic choices, (ii) for every $P \uparrow G$ operator, the set G includes a single partial function g. We will then discuss how the presented semantics can be extended to the general non-deterministic/probabilistic case.

The non-deterministic/probabilistic aggregated open/closed transition system is $(\mathcal{P}, OAct, Red, --\rightarrow_c, \longrightarrow_o)$, where the set of reducible states Red is taken to be the smallest subset of $\mathcal{P}$ that includes terms $\sum_{i\in I}[p_i].P_i$, where P_i are arbitrary processes of $\mathcal{P}$, and is such that

$$\begin{array}{lcl} P \in Red & \Longrightarrow & P+Q, Q+P, P|Q, Q|P, P\backslash L, P[\varphi], P\uparrow G \in Red \\ P \xrightarrow{\tau_h}_o \ \wedge \ \exists g \in G\colon h \in dom(g) & \Longrightarrow & P\uparrow G \in Red \\ P\{recX.P/X\} \in Red & \Longrightarrow & recX.P \in Red \end{array}$$

and $\longrightarrow_o$ (denoted simply by $\longrightarrow$) is still defined by the rules of Table 1 plus symmetric rules; however, differently from Sect. 2.2, here we take γ to just range over the set of open actions $OAct$ (thus now excluding τ) and we have that predicate $\gg$ (re-defined above) is directly determined from set Red. Finally, $--\rightarrow_c$ (here denoted simply by $--\rightarrow$) is defined by

$$P --\rightarrow \hat{\sigma} \quad \Leftrightarrow \quad \forall \hat{P'} \in \mathcal{P} \cup \{\uparrow\}.\, \hat{\sigma}(\hat{P'}) = \sum_{P \xrightarrow{p} \hat{P'}} p$$

where [6] the probability labeled multi-transition relation $--\rightarrow$, a multi-set over $\mathcal{P} \times [0,1] \times \mathcal{P}$, is defined by the rules of Table 5 plus symmetric rules, starting from Red and $\longrightarrow$: in Table 5 a transition is taken with multiplicity n if it can be derived in n different ways.

As in the pure non-deterministic case, if unguarded recursion is somehow disallowed, then the preliminary definition of set Red is not necessary and non-reducibility of states can be just determined by absence of $--\rightarrow$ transitions.

The semantics above can be extended to deal with the general non-deterministic/probabilistic case by just adding information, representing scheduling choices, to reduction transitions. This must be done so to distinguish, in a given reducible state, outgoing probabilistic transitions belonging to different schedulers. The information can be produced as an additional label that records application of operators by their derivation rules. Another possibility is to define the semantics directly on reduction transitions $P --\rightarrow \hat{\sigma}$. It is possible to do this by defining a preorder

6. In the summation, a distinguished istance of p is considered for each multiple instance of that same transition $P \xrightarrow{p} \hat{P'}$.

over partial probability distributions that coincides with point to point $\leq$ on the probability associated to states and by defining the semantics of a term to be the one with the minimal paritial probability distributions satisfying the operational semantics. The use of such a pre-order can be seen, for instance, in term $recX.([.4]l.\underline{0} + [.6]X)$, whose semantics is evaluated by starting from a partial probability distribution that assigns zero to all states and incrementing such a partial probability distribution by applying the operational rules.

Note that another way, common in the literature (see, e.g., [1]), to force the system to be purely probabilistic is to adopt a different "$+$" operator, where probabilistic (reduction) transitions do not resolve the choice. More precisely, by using the notation for (non-aggregated) probabilistic transitions used in this paper and by denoting such an operator with "$\sqcap$", the semantics is:

$$\frac{P \longrightarrow \sigma}{P \sqcap Q \longrightarrow \sigma \sqcap Q}$$

and a symmetric rule, i.e. the same rules for reduction transitions that we have for parallel, while the semantics for open transition is the same as that of "$+$". Aggregated reduction transitions for "$\sqcap$" are determined with the same rules used for parallel in Table 5. The use of "$\sqcap$" instead of "$+$" and of restricted $P \uparrow G$ operators, where the set G includes a single partial function g, guarantees that all reducible states are purely probabilistic in the aggregated model.

3.4 A variant compatible with probabilistic standard observational congruence

The machinery for internal/external probability and multilevel priorities can be modified to make it compatible with (probabilistic) standard Milner's observational congruence. From the one hand we loose the distinction between reducible and unreducible states (i.e. $recX.\tau.X$ is now equated by weak bisimulation to $\underline{0}$), from the other hand we observe also intermediate (reducible) state in τ paths, so the equivalence becomes sensitive to the branching structure of τ behaviours and the state space reduction by aggregation of τ transitions (and elimination of intermediate states) less effective.

More precisely, we consider probabilistic observational congruence and probabilistic weak bisimulation equivalence as defined in [7] for the so-called "simple model": non-deterministic/ probabilistic open/closed transition systems can be seen as a restriction of such a model where: (i) closed reduction transitions correspond to probabilistic τ transitions and (ii) open labeled transitions correspond to probabilistic labeled (non-τ) transitions that lead to a distribution giving probability 1 to a single target state.

As in the pure non-deterministic case, the crucial modification that we have to do in order to make the process algebra of Sect. 3.2 compatible with probabilistic observational congruence concerns the parallel operator. This because, in terms of the probabilistic algebra we have, e.g., that while $a_h.\underline{0}|recX.[1]X$ has the same transition system of $recX.[1]X$, $a_h.\underline{0}|[1]\underline{0}$ has the same transition system of $[1]a_h.\underline{0}$, hence observational congruence cannot be a congruence.

We must therefore consider $\underline{0}$ (that is weakly bisimilar to $recX.[1]X$) as a failure event. As a consequence: we introduce in the syntax of behaviour expressions $\mathcal{E}$ (and of processes $\mathcal{P}$) successful termination $\underline{1}$, we add to the set $OAct$ of open actions a special action $\surd$, denoting successful termination, and we modify the operational semantics of Table 1 exactly as in the pure non-deterministic case.

As far as the congruence property of probabilistic observatonal congruence is concerned, since, according to the definition given in [7], probabilistic weak equivalence matches single probabilistic reductions to weak transitions (instead of "maximal" weak transitions into weak transitions like in the trace-based equivalence), here the adoption of probabilistic schedulers and the requirement about partial visibility of schedulers are not needed. The congruence property of probabilistic observatonal congruence with respect to all the operators of the probabilistic prioritized process algebra can be proven similarly as for the pure non-deterministic case (see Sect. 2.4).

In the case we consider a generalized definition of probabilistic weak bisimulation where arbitrary weak transitions must be matched by weak transitions then we have to adopt, as for

$$\frac{P \gg \quad \nexists p, P' : P \stackrel{p}{--\!\!\rightarrow} P'}{P \stackrel{1}{--\!\!\rightarrow} \uparrow}$$

$$\frac{P_j \not\gg}{\sum_{i \in I}[p_i].P_i \stackrel{p_j}{--\!\!\rightarrow} P_j}\ j \in I \qquad \frac{P_j \stackrel{p}{--\!\!\rightarrow} \hat{P}'}{\sum_{i \in I}[p_i].P_i \stackrel{p_j \cdot p}{--\!\!\rightarrow} \hat{P}'}\ j \in I$$

$$\frac{P \stackrel{p}{--\!\!\rightarrow} \hat{P}'}{P + Q \stackrel{p}{--\!\!\rightarrow} \hat{P}'}$$

$$\frac{P \stackrel{p}{--\!\!\rightarrow} \hat{P}' \quad Q \not\gg}{P|Q \stackrel{p}{--\!\!\rightarrow} \hat{P}'|Q} \qquad \frac{P \stackrel{p'}{--\!\!\rightarrow} \hat{P}' \quad Q \stackrel{p''}{--\!\!\rightarrow} \hat{Q}'}{P|Q \stackrel{p' \cdot p''}{--\!\!\rightarrow} \hat{P}'|\hat{Q}'}$$

$$\frac{P \stackrel{p}{--\!\!\rightarrow} \hat{P}'}{P \backslash L \stackrel{p}{--\!\!\rightarrow} \hat{P}' \backslash L} \qquad \frac{P \stackrel{p}{--\!\!\rightarrow} \hat{P}'}{P[\varphi] \stackrel{p}{--\!\!\rightarrow} \hat{P}'[\varphi]}$$

$$\frac{dom(g) \cap \{h' | P \stackrel{\tau_{h'}}{\longrightarrow}\} = D \quad P \stackrel{\tau_h}{\longrightarrow} P' \quad P' \uparrow G \not\gg}{P \uparrow G \stackrel{g(h)/g(D)}{--\!\!\rightarrow} P' \uparrow G}\ h \in dom(g), g \in G$$

$$\frac{dom(g) \cap \{h' | P \stackrel{\tau_{h'}}{\longrightarrow}\} = D \quad P \stackrel{\tau_h}{\longrightarrow} P' \quad P' \uparrow G \stackrel{p}{--\!\!\rightarrow} \hat{P}'' \uparrow G}{P \uparrow G \stackrel{(g(h)/g(D)) \cdot p}{--\!\!\rightarrow} \hat{P}'' \uparrow G}\ h \in dom(g), g \in G$$

$$\frac{P \stackrel{p}{--\!\!\rightarrow} \hat{P}'}{P \uparrow G \stackrel{p}{--\!\!\rightarrow} \hat{P}' \uparrow G} \qquad \frac{P\{recX.P/X\} \stackrel{p}{--\!\!\rightarrow} \hat{P}'}{recX.P \stackrel{p}{--\!\!\rightarrow} \hat{P}'}$$

Table 5. Additional rules for aggregated non-deterministic/probabilistic reduction transitions

the trace-based equivalence, the probabilistic schedulers of [7] (a phenomenon similar to the sequence of schedulers in the proof of congruence for the trace-based equivalence with respect to the "$P \uparrow G$" operator arises, due to the decomposition of the weak transitions into single transitions and re-composition in the other term).

4 Possible extensions: discrete time and continuous time

A simple techinque, previoulsy used in the literature (e.g. in the context of continuous time, with exponential distributions), to add capability to express time to a process algebra is to attach the timing information to actions when a model is considered to be complete.

By exploiting our approach, it is possible to do this compositionally: when a part of a system is closed via the "$P \uparrow G$" operator, we can put inside set G the timing information to be attached to actions. We can express, e.g.: (exponentially distributed) continuos time by putting rates of exponential distributions instead of weights inside G and by letting the semantics of "$P \uparrow G$" to additionally label (with respect to that considered in Sect. 3.3) reduction transitions with the assigned (overall) rate; discrete time by assuming that the resulting reduction transition take one time unit to be executed (and by preserving the possibility to include weights inside G to express probabilistic choices).

When timing is considered, trace-based equivalence is established by additionally requiring, w.r.t. that considered in the probabilistic case (see Def. 3.2 and Def. 3.3), that the (continuous or discrete) probability distribution of time associated to matching aggregated reduction transitions ($\longrightarrow_c^*$ or $\longrightarrow_c^+$) must be the same. Moreover in the general case (if we do not want equivalent systems to just preserve particular properties, as we will discuss below) it is necessary to require that, not only the mean probability distribution over states reached by aggregated reduction transitions ($\longrightarrow_c^*$ or $\longrightarrow_c^+$) are compared, but also, probability distributions conditioned on the amount of time taken by aggregated reduction transitions (i.e. a probability distribution is matched for every possible, discrete or continuous, time value).

With respect to bisimulation-based (ordinary lumping-based) markovian aggregation, which requires (as for the equivalence considered Sect. 3.4 for probabilistic systems) to preserve the branching structure of reduction transitions, the obtained equivalence is more coarse. For example, with discrete time

$$[p_1][1]a.\underline{0} + [p_2][1]b.\underline{0} = [1]([p_1]a.\underline{0} + [p_2]b.\underline{0})$$

and with continuous exponentially distributed time

$$[\lambda_1][\mu]a.\underline{0} + [\lambda_2][\mu]b.\underline{0} = [\lambda_1 + \lambda_2]([\mu \cdot \frac{\lambda_1}{(\lambda_1 + \lambda_2)}]a.\underline{0} + [\mu \cdot \frac{\lambda_2}{(\lambda_1 + \lambda_2)}]b.\underline{0}).$$ [7]

Such examples show how, by considering coarser equivalences with respect to bisimulation (as trace-based or even testing-based equivalences), we can reduce the number of system states by merging states (that otherwise would not be mergeable, due to necessity of preserving the branching structure) and still obtain systems with the same transient state (and consequently steady state) behaviors. For instance, in the example above, the states $[\mu]a.\underline{0}$ and $[\mu]b.\underline{0}$ that are not lumpable (cannot be put in the same quivalence class by markovian bisimulation) can, instead, be merged by considering our equivalance: even if the states are not lumpable such aggregation is correct from a stochastic viewpoint. Similarly, in the discrete time case, for the states $[1]a.\underline{0}$ and $[1]b.\underline{0}$.

Finally, we would like to note that, in the continuous time case, if a parallel operator like that of Sect. 3.2 is considered, where, in the case of parallel of closed states, the reduction transition to be executed is just non-deterministically chosen (i.e. time reduction transitions are non-deterministically interleaved by parallel, thus obtaining a sequentilization of their execution time), then it is possible to adopt a very coarse version of the equivalence which just matches the mean time for performing aggregated transitions (instead of matching the time distribution) and the mean probability distribution over states reached by aggregated reduction transitions (instead of probability distributions conditioned on time). Due to the *insensitivity property* of the considered systems (time distributions are never really contemporaneously executed because of the priority of reduction transitions over open transitions and of the way parallel of closed states is defined) such an equivalence can be a congruence and preseves the steady state behavior of systems. The obtained aggregating power is much greater with respect to the general equivalence above. More precisely every system can be turned into an equivalent aggregated one where reducible states directly reach, via exponential rate-labeled reduction transitions, distributions over non-reducible states or non-escapable divergent states: rates are obtained as the inverse of the mean time for performing aggregated transitions and reached distributions are just given by the mean probability distribution reached by aggregated reduction transitions.

References

[1] S. Andova, *"Process Algebra with Probabilistic Choice"*, in Proc. of *Formal Methods for Real-Time and Probabilistic Systems, 5th International AMAST Workshop*, LNCS 1601:111-129, 1999.

7. In order for the aggregation to take place it is essential that the states reachable after the first exponential phase have all the same total rate, i.e. sum of rates performable exponential delays (μ in the example) , otherwise the second phase, when aggregated, would become hyperexponentially distributed, instead of just exponentially distributed.

[2] R. Cleaveland, G. Luttgen, V. Natarajan, *"Priority in Process Algebras"*, in Handbook of Process Algebra, Chapter 12, pp. 711-765, Elsevier, 2001

[3] M. Bravetti, R. Gorrieri, R. Lucchi, G. Zavattaro. *"Quantitative Information in the Tuple Space Coordination Model"*, *Theoretical Computer Science*, 346:1, pages 28-57, Elsevier, 2005.

[4] N. Lynch, R. Segala, F. Vaandrager, *"Observing Branching Structure through Probabilistic Contexts"*, To appear in *Siam Journal on Computing*.
Available at `http://theory.lcs.mit.edu/tds/lynch-pubs.html`

[5] R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989.

[6] R. Milner, *"A complete axiomatization for observational congruence of finite-state behaviours"*, in Information and Computation 81:227-247, 1989

[7] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1995.