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

Mario Bravetti[1],[2]

Dipartimento di Scienze dell'Informazione
Università di Bologna
Bologna, Italy

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 already 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.

Keywords: Process algebra, Priorities, Probabilities, Congruence property.

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 [10], 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 [5], 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 (instead of just single states). On the contrary, in open transition systems, we have the problem of explicitly 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., [8] 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 pri-

oirities 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 (where every mapping can generate a probabilistic reduction transition). 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 [13] for transitions leading to probability distributions. Note that, even if obviously the trace-based equivalence aggregates much more than standard (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 standard 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. Sect. 4 concerns the usage of the closure operator to express discrete and continuous (exponential) time. Finally, Sect. 5 is dedicated to comparison with related work. Proofs of theorems can be found in [3].

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 $\not\exists l \in Lab : s \xrightarrow{\ l\ }_o$.

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). 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 non-reducible states [3] 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 $s_1 \xrightarrow{l}_o \longrightarrow_c^* s_1' \, (\not\gg \vee \uparrow)$ 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 $(s_1', s_2') \in \beta$.

Two non-reducible 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.

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. $\tau.l.\underline{0} + \tau.recX.\tau.X \not\simeq \tau.l.\underline{0}$ [4] because $\tau.l.\underline{0} + \tau.recX.\tau.X \longrightarrow_c^+ l.\underline{0}$ and it can reach a divergent state, while $\tau.l.\underline{0} \longrightarrow_c^+ l.\underline{0}$ but it cannot reach a divergent state. On the contrary $\tau.l.\underline{0} + \tau.recX.(\tau.l.\underline{0} + \tau.X) \simeq \tau.l.\underline{0}$ 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.

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

---

[3] In general it is possible to consider also reducible states in the definition of weak equivalences, however this is not needed for defining weak congruences. See [3] for the extended definition.

[4] We assume syntactical precedence of prefix w.r.t. other operators when writing terms.

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 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

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

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: (*i*) For every $\alpha \in \mathcal{AN}, h \in \mathcal{H}$ there exists α' such that $\varphi(\alpha_h) = \alpha'_h$; (*ii*) $\varphi(\tau) = \tau$; and (*iii*) $\varphi(\overline{\alpha}) = \overline{\varphi(\alpha)}$. The meaning of the operators is the standard one of [10,11], 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 $\overline{b}$ occur 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 occurs in P with handle h_1 and in Q with handle h_2,

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

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

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

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

$$\frac{P \xrightarrow{\ \gamma\ } P' \quad \not\exists h \in G : P \xrightarrow{\ \tau_h\ }}{P \uparrow G \xrightarrow{\ \gamma\ } P' \uparrow G}$$

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

Table 1
Proposed variant of standard structural operational rules

represents a prioritized choice between the two input actions a: if both P and Q offer synchronization (input) on a 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 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) 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.*

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

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

Table 2
Additional rules for non-deterministic reduction transitions

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 directly denoting (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 $(S, Lab, Red, \text{--}\!\rightarrow_c, \longrightarrow_o)$, 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,
- $\text{--}\!\rightarrow_c \subseteq Red \times \{(S - Red) \cup \{\uparrow\}\}$ 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 $\text{--}\!\rightarrow_c$ 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 correspondence above) as follows.

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

- If $s_1 \xrightarrow{\;l\;}_o s'_1$ and $(s'_1 \text{--}\!\rightarrow_c \hat{s}''_1$ or $\hat{s}''_1 = s'_1 \not\gg)$ then, for some s'_2 and $\hat{s}''_2$, with

$s_2 \xrightarrow{\;l\;}_o s_2'$ and $(s_2' \dashrightarrow_c \hat{s}_2''$ or $\hat{s}_2'' = s_2' \gg)$, we have either $\hat{s}_1' = \hat{s}_2' = \uparrow$ or $(\hat{s}_1', \hat{s}_2') \in \beta$.

Two non-reducible 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.

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

- If $s_1 \xrightarrow{\;l\;}_o s_1'$ and $(s_1' \dashrightarrow_c \hat{s}_1''$ or $\hat{s}_1'' = s_1' \gg)$ then, for some s_2' and $\hat{s}_2''$, with $s_2 \xrightarrow{\;l\;}_o s_2'$ and $(s_2' \dashrightarrow_c \hat{s}_2''$ or $\hat{s}_2'' = s_2' \gg)$, we have either $\hat{s}_1' = \hat{s}_2' = \uparrow$ or $\hat{s}_1' \approx \hat{s}_2'$.
- If $s_1 \dashrightarrow_c \hat{s}_1'$ then, for some s_2', with $s_2 \dashrightarrow_c \hat{s}_2'$, 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 $(\mathcal{P}, OAct, Red, \dashrightarrow_c, \longrightarrow_o)$, 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

$$P \in Red \qquad\qquad \Longrightarrow P+Q, Q+P, P|Q, Q|P, P\backslash L, P[\varphi], P \uparrow G \in Red$$

$$P \xrightarrow{\;\tau_h\;}_o \wedge h \in G \quad \Longrightarrow P \uparrow G \in Red$$

$$P\{recX.P/X\} \in Red \Longrightarrow recX.P \in Red$$

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, $\dashrightarrow_c$ (here denoted simply by $\dashrightarrow$) 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 "$\dashrightarrow$" 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 $\dashrightarrow$ transitions.

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

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

$$\frac{P --\to \hat{P}'}{P + Q --\to \hat{P}'}$$

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

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

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

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

Table 3
Additional rules for aggregated non-deterministic reduction 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 "artificial" 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) states 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 $\underline{1}$. Moreover we introduce a special action $\sqrt{}$, 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:

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

where now we have $\sqrt{} \in OAct$. 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 "$\sqrt{}$" moves, must be added (to Table 1):

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

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.*

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 $n \in \mathbf{N}^+$ (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 $tr_1, \ldots, tr_n$ 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 (history dependent) scheduler $sched_s$ from a state s is a partial function $sched_s : Tr_s \rightarrow PDist_S$ 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 $prob_{sched_s}(tr) = \prod_{1 \leq i \leq n-1} sched_s(tr_{\leq i})(tr_{i+i})$. [5] A scheduler $sched_s$ is *terminating* (by means of finite traces) if $\sum_{tr \in Tr_{sched_s}} prob_{sched_s}(tr) = 1$. [6] Terminating schedulers from s are ranged over by $tsched_s$.

We define $s \xrightarrow{}_c^* \sigma$, 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 $s \xrightarrow{}_c^+ \sigma$ 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 $\not\exists \sigma : s \xrightarrow{}_c^* \sigma \wedge \sigma(S_{\not\gg}) > 0$. 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 non-reducible states of a non-deterministic/probabilistic open/closed transition system $(S, Lab, \xrightarrow{}_c, \xrightarrow{}_o)$ is a weak equivalence if, whenever $(s_1, s_2) \in \beta$:

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

Two non-reducible 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.

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

- If $s_1 \xrightarrow{l}_o \xrightarrow{}_c^* \sigma(\not\gg \vee \uparrow)$ then, for some σ', $s_2 \xrightarrow{l}_o \xrightarrow{}_c^* \sigma'(\not\gg \vee \uparrow)$ and $\sigma_{\not\gg} \equiv_\approx \sigma'_{\not\gg}$.
- If $s_1 \xrightarrow{}_c^+ \sigma(\not\gg \vee \uparrow)$ then, for some σ', $s_2 \xrightarrow{}_c^+ \sigma'(\not\gg \vee \uparrow)$ 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 "$[p_1]P_1 + \ldots + [p_n]P_n$" ($\sum_{1 \leq i \leq n} p_i = 1$) [7] 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). $[.3]l.\underline{0} + [.7]recX.[1]X \not\simeq [1]l.\underline{0}$ 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 .7 to a divergent state, while the only distribution σ such that $\sigma(\not\gg \vee \uparrow)$ reachable by $[1]l.\underline{0}$ assigns probability 1 to $l.\underline{0}$ (i.e. 0 probability is assigned to divergent states). On the contrary, $[.3]l.\underline{0} + [.7]recX.([1]l.\underline{0} + [1]X) \simeq [1]l.\underline{0}$ (where the choice inside

---

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

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.

3.2 Probabilistic prioritized process algebra

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

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

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 $\mathbf{R}^+$ (representing weights) whose domains are disjoint and φ is a relabeling function over $OAct$ such that: (i) For every $\alpha \in \mathcal{AN}, h \in \mathcal{H}$ there exists α' such that $\varphi(\alpha_h) = \alpha'_h$; (ii) $\varphi(\overline{\alpha}) = \overline{\varphi(\alpha)}$. $\sum_{i \in I}[p_i].E_i$ 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 transitions, 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

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

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

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

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

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

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

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. 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 $\overline{c}$ occur in R with neutral handle $*$, represents a probabilistic/prioritized choice among input actions a, b and c: if R offers synchronization (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 corresponding action is executed with probability 1.

In general we can express (external) probabilistic choices at multiple priority levels by using $P \uparrow G$ 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 ——→* and ——→$^+$ slightly more complicate by using *probabilistic schedulers* like in [13]. 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 probabilistic 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). Moreover, the congruence for the parallel operator is crucially based on the adoption of schedulers with partial visibility. The definition of ——→* and ——→$^+$ 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 history 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. See [3] for details about congruence.

An alternative way, with respect to adopting probabilistic schedulers with the above partial visibility condition, to obtain the congruence property of "$\simeq$" is to restrict the expressive power of the algebra, so that, as common in probabilistic process algebra, only pure probabilistic choices between reductions can be expressed in (sequential) processes. The usual way to do this (see e.g. [1]) 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 transitions is the same as that of "$+$". Moreover we have to consistently require that for every $P \upharpoonright G$ operator, the set G includes a single partial function g.

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 directly denoting 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. 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 3.6 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 correspondence above) as follows.

Definition 3.7 A symmetric relation β over non-reducible 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 $\hat{\sigma}_{\not\gg}(s_1')=1)$ 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 $\hat{\sigma}_{\not\gg} \equiv_\beta \hat{\sigma}'_{\not\gg}$.

Two non-reducible 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.

Definition 3.8 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 $\hat{\sigma}'_{\gg}(s_2')=1)$, we have $\hat{\sigma}_{\gg} \equiv_{\approx} \hat{\sigma}'_{\gg}$.

- If $s_1 --\rightarrow_c \hat{\sigma}$ then, for some $\hat{\sigma}'$, with $s_2 --\rightarrow_c \hat{\sigma}'$, we have $\hat{\sigma}_{\gg} \equiv_{\approx} \hat{\sigma}'_{\gg}$.

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

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

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

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

where [8] 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 over partial

---

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

$$\frac{P \gg \quad \nexists p, P' : P \xdashrightarrow{p} P'}{P \xdashrightarrow{1} \uparrow}$$

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

$$\frac{P \xdashrightarrow{p} \hat{P}'}{P + Q \xdashrightarrow{p} \hat{P}'}$$

$$\frac{P \xdashrightarrow{p} \hat{P}' \quad Q \not\gg}{P|Q \xdashrightarrow{p} \hat{P}'|Q} \qquad \frac{P \xdashrightarrow{p'} \hat{P}' \quad Q \xdashrightarrow{p''} \hat{Q}'}{P|Q \xdashrightarrow{p' \cdot p''} \hat{P}'|\hat{Q}'}$$

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

$$\frac{dom(g) \cap \{h' | P \xrightarrow{\tau_{h'}}\} = D \quad P \xrightarrow{\tau_h} P' \quad P' \uparrow G \not\gg}{P \uparrow G \xdashrightarrow{g(h)/g(D)} P' \uparrow G} \ h \in dom(g), g \in G$$

$$\frac{dom(g) \cap \{h' | P \xrightarrow{\tau_{h'}}\} = D \quad P \xrightarrow{\tau_h} P' \quad P' \uparrow G \xdashrightarrow{p} \hat{P}'' \uparrow G}{P \uparrow G \xdashrightarrow{(g(h)/g(D)) \cdot p} \hat{P}'' \uparrow G} \ h \in dom(g), g \in G$$

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

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

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 partial 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 the approach presented in Sect. 3.2 of replacing "+" with the "⊓" operator, can also be adopted, instead of the constraint (i) above, to reduce to the purely probabilistic case, i.e. to cause all reducible states to be purely probabilistic in the aggregated model. In particular, the aggregated reduction transitions for "⊓" are determined with the same rules used for parallel in Table 5.

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 [13] 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 $\sqrt{}$, 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 observational congruence is concerned, since, according to the definition given in [13], 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.

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 the trace-based equivalence, the probabilistic schedulers of [13] (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 and continuous time

A simple technique, previously 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) continuous 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

$$[\lambda_1][\mu]a.\underline{0} + [\lambda_2][\mu]b.\underline{0} = [\lambda_1 + \lambda_2]([\mu \cdot \frac{\lambda_1}{(\lambda_1 + \lambda_2)}]a.\underline{0} + [\mu \cdot \frac{\lambda_2}{(\lambda_1 + \lambda_2)}]b.\underline{0}). ^{[9]}$$

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

---

[9] 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.

example above, the states $[\mu]a.\underline{0}$ and $[\mu]b.\underline{0}$ that are not lumpable (cannot be put in the same equivalence class by Markovian bisimulation) can, instead, be merged by considering our equivalence: 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 sequentialization 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 preserves 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.

5   Related Work

In this section we compare the approach introduced in this paper with other work on extending standard process algebra with priorities. In order to do this we initially use the notions of stratified and unstratified approach, like in [12]. In stratified approaches to priority actions are assigned fixed priority levels (e.g actions are endowed with numbers to denote their priority level). On the contrary, in unstratified approaches the same action can have different priority depending where it appears in a term. For example, the approaches of [7,4] are stratified: assuming for simplicity that just two priority levels exist, a choice where a takes priority over b is denoted in [7] by $\underline{a}.P + b.Q$ (high priority actions are underlined), similarly in [4]. The approach of [6], where the same choice would be represented by $a.P +\rangle b.Q$, is, instead, not stratified. This because the priority of a over b is specific to this particular choice; elsewhere in the term we could have another choice where b has priority over a instead. Another example of unstratified approach is that of [12]: the same choice is represented in [12] by $a.P + \{a\}{:}b.Q$, where $\{a\}$ is a priority guard denoting the set of actions for which the context must not offer a synchronization in order for b to be executable. Again, elsewhere in the term we could use the same

actions but with exchanged pre-emption power.

In our approach the above choice is represented by $a_h.P + b_{h'}.Q$ occurring inside a term $R \uparrow \{h\}$ (which also includes the synchronization context for the choice), in turn occurring inside a term $R' \uparrow \{h'\}$. Our approach could be classified as un-stratified because the same choice $a_h.P + b_{h'}.Q$ could also occur elsewhere in the term specifying the whole system (not inside $R' \uparrow \{h'\}$) and exchanged priorities be assigned to a and b. However, note that the possibility to express multi-level priorities in an unstratified manner in our approach is actually based on a stratification of actions in two levels, where reductions have priority over standard process algebra actions (endowed with handles).

The most significant difference that characterizes our approach with respect to the other ones cited before is the idea to further delay the moment in which the priority of actions is expressed: it is not encoded directly in actions (like in stratified approaches) but it is also not even expressed at the syntactical point where the actions (like in [12]) or the choice in which they are directly involved (like in [6]) occur in a term. The priority is expressed inside the operator $R \uparrow G$ which is to be put just outside the synchronization context which decides which actions will actually occur. This is the reason why such an operator can prioritize only τ_h actions, i.e. actions which are no more potential (i.e. for which all needed synchronization has been performed and that can no longer be blocked by a context, e.g. a restriction operator in CCS). Technically, such an operator is similar to the operator $\theta(P)$ introduced in [2] which prioritizes actions of P according to a fixed partial order among actions expressing a priority relation. The main difference are: the priority relation applied with $R \uparrow G$ depends on the set G, while in $\theta(P)$ is fixed; and $R \uparrow G$ can prioritize just τ_h actions that are not potential (cannot be blocked by a context) and turns them into τ (the only prioritized action), while $\theta(P)$ can also prioritize (just potential) a actions and does not change the actions when prioritizing them. The latter difference is related to a different idea behind the use of the operator: in this paper the operator is just aimed at prioritizing actions after they have been turned from potential to actual by the synchronization context. In [2], instead, the operator could be applied to potential actions too, thus giving rise to an approach where priority of actions is not mandatorily delayed until the synchronization context is considered as in our approach. Such a difference between $R \uparrow G$ and $\theta(P)$ and in their use is also related to the issue of compatibility with weak bisimulation: in order for $\theta(P)$ to be compatible with weak bisimulation only priority relations such that, whenever an action a has priority over an action b, then also τ must have priority over b, must be considered. A simple counterexample is given by the equivalent terms $c.(\tau.(a.P + b.Q) + b.Q)$ and $c.(a.P + b.Q)$: if we apply the operator $\theta(P)$ to both of them and we assume a priority relation where just a (and not τ) has priority over b, then the resulting terms are not equivalent. Since in our approach actions τ are always prioritized, the operator $R \uparrow G$ is a congruence for weak bisimulation.

The main benefit of forcing the modeler to delay the moment in which the priority of actions is expressed until we are outside of the synchronization context

is to avoid the two main sources of complications of the approaches cited above: managing synchronization of prioritized/unprioritized actions and defining transition relation and weak equivalences which are based on moves conditioned on the environment. Concerning the former, typical questions, especially in the context of stratified approaches, are: Are actions with different priority levels allowed to synchronize? In the affirmative case which is the resulting priority level? Some approaches deal with this issues by disallowing actions with a different priority to synchronize (as [7]), other by enforcing some asymmetry in the synchronization (in [6] and [8], where CCS synchronization is adopted, prioritized choices are allowed only for input actions; in [4] the generative-reactive synchronization model is adopted where the priority assumed by the synchronization is the priority of the generative action and the reactive priorities are just used to choose among the possible synchronizations with the prioritized actions). Only [12] fully exploits the advantages of being unstratified: it is able to deal with a symmetrical form of communication by taking the priority of the synchronization to be determined by the union of the guards of the single actions. Concerning the latter, the presence of transitions (in particular τ transitions) at several levels of priority (for stratified approaches) or conditioned on the actions offered by the environment (for unstratified approaches) causes the adoption of rather complex and non-standard notion of equivalence. The problem of defining weak equivalences being even more hard. Typical questions, especially in the context of stratified approaches, are: Does the weak equivalence considered abstract with respect to τ actions no matter which is the priority level? In the affirmative case are such τ at different levels treated in an uniform way? The machinery in [12], somehow exploiting the uniformity given by the adoption of an unstratified approach, answers positively to both questions.

In our approach we do not have the problem to evaluate the priority of actions upon synchronization and to express transitions conditioned on the environment because we apply priority only to τ_h actions that are not affected by the environment, thus getting unconditioned τ transitions that can be treated just as in weak bisimulation. We do not see a particular advantage in expressing priorities before applying the synchronization context: anyway all the transitions, even unprioritized ones, have to be preserved (because we cannot know the behavior of the context in advance) and the additional effort has to be done to carry the information about the dependency of the transition from the context until the context is applied, (or, more precisely, the actions taking priority are restricted outside of such context).

In [12] the following philosophical idea is implemented: if we consider terms such that the use of priorities is "confined" into subterms restricting all the actions used internally to express the priority mechanisms, then equivalence over such subterms turns out to be ordinary weak bisimulation and we can find an ordinary CCS term equivalent to them. In this paper a different philosophical idea is, instead, considered: a term is assumed to represent a complete system only if all the actions that it employs are "complete". In standard process algebra an action can be regarded as complete when it is a reduction, e.g. in CCS an action becomes complete when, from being a potential a, it becomes an actual unrestrictable τ by means of syn-

chronization. When we additionally consider priorities (and probabilities), we need a further step in order for an action to become complete: besides being complete with respect to synchronization it must subsequently be also quantified. Only when this happens it actually becomes a reduction from which we can abstract in weak equivalence. Our approach allows us to extend the use of ordinary weak bisimulation and observational congruence to priorities: actions that are not complete (in terms of synchronization, i.e. a_h, or quantification, i.e. τ_h) are treated like ordinary non-τ actions, while the only actions from which we abstract are the quantified ones (our prioritized τ). In this sense our approach allows us to manage in a standard way not only the behaviour of terms such that the actions involved in the priority mechanisms are internally restricted, but the standard treatment encompasses any term of the prioritized calculus.

Finally, concerning the notions of local versus global priorities considered in [8], referring to the effect of prioritized actions to range over a limited scope, e.g. a sequential process, (local priorities) or over the whole term (global priorities), our approach treats (generated) reductions has having global priority. However we showed that we can implement global priority with a special operational rule for the parallel operator, different from that in [8], that exploits the distinction between successful termination and failure (see Sect. 2.4): as a consequence constraining equivalence to be sensible to divergence of prioritized τ transitions (like in [8]) is not needed to get a congruence; thus, in our case, standard observational congruence can simply be adopted.

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.

[2] J.C.M. Baeten, J. Bergstra, and J.W. Klop, *"Syntax and defining equations for an interrupt mechanism in process algebra"*, *Fundamenta Informaticae*, 9:127168, 1986.

[3] M. Bravetti, *"Expressing Priorities, External Probabilities and Time in Process Algebra via Mixed Open/Closed Systems"*, Technical report UBLCS-2007-18, Department of computer science, University of Bologna, June 2007.

[4] M. Bravetti and M. Bernardo. *"Compositional asymmetric cooperations for process algebras with probabilities, priorities, and time"*, in Proc. of the *1st Workshop on Models for Time-Critical Systems* (MTCS 2000) , ENTCS 39(3), 2000

[5] 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.

[6] J. Camilleri, G. Winskel, *"CCS with priority choice"*, *Information and Computation*, 116(1):2637, 1995.

[7] R. Cleaveland, M.C.B. Hennessy, *"Priorities in process algebra"*, *Information and Computation*, 87(1/2):5877, 1990.

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

[9] N. Lynch, R. Segala, F. Vaandrager, *"Observing Branching Structure through Probabilistic Contexts"*, To appear in *Siam Journal on Computing*.

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

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

[12] I. Phillips, *"CCS with Priority Guards"*, to appear in *Journal of Logic and Algebraic Programming*.

[13] 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.