Mixing Up Nondeterminism and Probability: a preliminary report [1]

J.I. den Hartog, E.P. de Vink

Faculty of Mathematics and Computer Science
Vrije Universiteit
Amsterdam, The Netherlands

Abstract

For a process language with both nondeterministic and probabilistic choice, and a form of failure a transition system is given from which, in a modular way, various operational models corresponding to various interpretations of nondeterminism and probability can be obtained. The effect of failure of one component for the system as a whole is treated differently in each interpretation. The same approach is followed for an extension of the language with a parallel operator. The adopted concurrency model is of a distributed nature and assumes that progress is guaranteed if nonfailing components exist. To this end the notion of a take-over of a failing component is incorporated in the transition system. It is shown that the modular way in which the transition system can yield different semantical models applies to this setting as well.

1 Introduction

In this paper the interplay of probabilistic and nondeterministic choice is studied in the context of two process languages, a nonconcurrent and a concurrent one. Various operational, denotational and axiomatic models are proposed in the literature to describe the probabilistic operator of PCCS (cf. [8]). See, e.g., [15,18,19,9,22,17]. Often the nondeterminacy is removed or restricted when treating probability, and the parallel operator is interpreted as a synchronous product. In a number of these references, though, both probability on the one hand, and nondeterminism and/or concurrency on the other are treated on equal footing. In principle, various interpretations are possible in the modeling of these concepts, especially with respect to the recovery from

[1] This paper is an updated version of the paper presented at the PROBMIV'98 workshop, Indianapolis.

Preprint submitted to Elsevier Preprint 17 May 1999

failure. However, a systematic study of this topic is lacking so far. Below a first step toward this direction is taken, which also incorporates a treatment of a parallel operator.

We propose to take one single transition system (specification) as a starting point, in which non-determinism, probability and parallelism are coded explicitly. We use the term transition system (specification) for the set of axioms and rules describing the possible transitions of each statement. One can vary in the interpretation of nondeterminism and the interpretation of probability by tuning the way in which meanings, are obtained from the transition system. The meaning or operational semantics of a statement is given as a sets of distributions. We distinguish between local, "global-don't care" and "global-don't know" nondeterminism versus unconditional, "conditional-don't care" and "conditional-don't know" probability. Different combinations of these induce different semantical models. We show that all the 3×3 semantical models are distinct, and provide typical examples.

The distinction of local versus global indeterminacy is treated extensively in the concurrency literature, cf. [13] for example. More recently, in the context of process algebra, [2] proposes to distinguish the so-called static and dynamic aspects of nondeterminism as well, leading to a notion of partial choice. In our setup probability reveals yet another dimension of nondeterministic choice, which in the field of logic programming is known as *don't care* and *don't know* nondeterminism.

In the local interpretation of nondeterminism a choice is made independent of the environment. In contrast, in a global setting, the law $s \Box \mathit{fail} = s$ holds, for each program s. A branch should not be taken if it immediately results in deadlock. But what precisely is immediate? Below we consider, what we refer to as, the "don't care"-case and the "don't know"-case. In the "don't care"-case the next branching point of making another choice (be it probabilistic or nondeterministic) is considered as a next moment. In this interpretation, e.g., the statement $\mathit{fail} \Box \mathit{fail}$ does not deadlock immediately. There is still the internal activity of selecting one of the possibilities ahead. In the "don't know"-case first probability and nondeterminism are resolved. As a consequence the interpretation of $a \Box (\mathit{fail} \Box \mathit{fail})$ is different in the "don't care"-case and "don't know"-case. In the former there is no recovery from failure once control is transferred to the alternative $\mathit{fail} \Box \mathit{fail}$, whereas in the latter the alternative a can still be selected when this failure occurs.

For the probabilistic choice a similar distinction is made; we distinguish unconditional, "conditional-don't care" and "conditional-don't know" probability. In the unconditional interpretation each draw from a probabilistic choice is acceptable, failing or not. It is accepted unconditionally. For the conditional case one can vary in the amount of look-ahead that the probabilistic choice comes equipped with. This leads to different meanings for $a \oplus_{1/2} \mathit{fail}$ and

$a \oplus_{1/2} (\mathit{fail} \Box \mathit{fail})$, for example. (Here the operator $\oplus_{1/2}$ denotes a probabilistic choice taking both the left and right operand with probability 1/2.)

For the probabilistic choice there are, as coined in [9], the interpretations of reactive, generative and stratified probability. We have chosen to deal with a stratified model as it is the most general one. We expect no difficulties for the translation of the semantical models to, e.g., the generative setting. As a consequence the process languages that we treat feature multi-branching operations $\Box_{i=1}^{n} s_i$ for nondeterministic choice and $\oplus_{i=1}^{n} \rho_i \cdot s_i$ for the probabilistic choice, with the implicit assumption that the ρ_i's add up to 1. The first language that we consider, provides an explicit construct to generate deadlock, viz. the statement *fail* mentioned already. Additionally, in the second process language the operation of parallel composition is incorporated. The parallel operator comes along with its own form of indeterminacy, not necessarily the same as for the nondeterministic choice. We chose the "distributed" interpretation of concurrency, i.e., if one of the parallel components can make progress, then the whole system can. As a consequence we will have $a \| \mathit{fail} = a; \mathit{fail}$. A uniprocessor interpretation is feasible as well, but seems less interesting and is therefore not discussed here.

Below we adopt the interleaving model for concurrency. However, in order to do justice to the distributed nature of the parallelism, we consider here the case that failure of a single component should not cause failure of the system as a whole. Therefore, we make explicit that first one component, referred to as the primary component, is selected. If the primary component fails, i.e. does not produce any actions, it will at a certain point be overtaken by another component. This take-over is modeled explicitly in our setup as well. So, a failing primary component can pass control to another parallel component.

In the two transition systems below, resolving nondeterministic and probabilistic choices is made explicit by steps in the transition system. In the transition system for $\mathcal{L}_1$, the language with parallel composition, the selection of the primary component and the occurrence of take-over are also modeled by a step. Choices are finite and only a finite number of choices have to be resolved before the next action is clear. Also the number of take-overs before an action is finite. In other words, the transition systems are finitely branching and free of internal divergence.

The definitions of the operational semantics given below are circular. We claim that they can be justified by using so-called higher-order transformations. These transformations are contractions on a metrically complete function space. By Banach's Theorem they have a unique fixed point, which is the object specified by the equations of the original definitions. The nice point of using metric spaces as underlying mathematical structure is the close connection between measure theory and metric topology. We refer to [16,4] for a further discussion on the use of complete metric spaces and Banach's Theorem in the area of programming language semantics. The advantage of using metric spaces is also reflected in a number of semantical investigations related to probability. See [7,17,5] for example. The mathematical background is sketched in the next section. For our study we have used a construction for obtaining a metric space of measures from a given metric space that is borrowed from [23,11]. A similar construction is available for valuations, cf. [3].

The paper contributes a first step towards a systematic comparison of the various ways to combine probabilistic choice with more classical process algebra constructions. For the treatment of the parallel operator in the context of failure we propose a notion of take-over. The metric techniques to obtain semantical models from one single transition system constitute a uniform approach with a modular setup, in which nondeterminism, probability and take-over can be modeled conveniently. The remainder of this paper is organized as follows: Section 2 introduces notations and defines the semantical domain. The first process language $\mathcal{L}_0$ is treated in Section 3. There its syntax, transition system and operational models are given. It is shown that the semantic equivalences of these models are independent. In Section 4 the process language $\mathcal{L}_0$ is extended with a parallel operator to the language $\mathcal{L}_1$. The transition system and operational models are adapted accordingly. It is argued that the independence of the interpretations remains. In Section 5 the semantical models are further illustrated with typical examples. The paper ends with some concluding remarks.

2 Mathematical preliminaries

Let $\mathcal{A}$ be an alphabet and δ a fresh symbol ($\delta \notin \mathcal{A}$). We use $\mathcal{A}_\delta^\infty$ to denote the collection of all finite string over $\mathcal{A}$ possibly ending in δ ($\mathcal{A}_\delta^*$) and all infinite strings over $\mathcal{A}$ ($\mathcal{A}^\omega$), i.e. $\mathcal{A}_\delta^\infty = \mathcal{A}^* \cup \mathcal{A}^* \cdot \delta \cup \mathcal{A}^\omega$. The empty string is ϵ; a nonempty string w can be written as δ or $a \cdot w'$ for suitable $a \in \mathcal{A}$, $w' \in \mathcal{A}_\delta^\infty$. The prefix of length n of a string w, denoted by $w[n]$, is ϵ if $w = \epsilon$ or $n = 0$ and is $a \cdot (w'[n-1])$ if $w = a \cdot w', n > 0$. The set $\mathcal{A}_\delta^\infty$ is endowed with the Baire-distance d_B, given by

$$d_B(w_1, w_2) = \inf\{\, 2^{-n} \mid w_1[n] = w_2[n] \,\},$$

which turns $\mathcal{A}_\delta^\infty$ into a complete metric space. For $w \in \mathcal{A}_\delta^\infty$, $w \uparrow$ denotes the set of all string in $\mathcal{A}_\delta^\infty$ having w as a prefix.

The collection of the Borel sets $\mathcal{B}(x)$ over a metric space X, is the least sigma algebra containing the open sets. A Borel probabilistiy measure on the metric space X is a function $\mu : B(X) \to [0,1]$ which is countabely additive for disjoint sets and is 1 for the whole space X. The set PMS (probability measures over sequences) consists of all Borel probability measures on $\mathcal{A}_\delta^\infty$ with compact support. The distance d_M on PMS is given by

$$d_M(\mu_1, \mu_2) = \inf\{\, 2^{-n} \mid \forall w \in \mathcal{A}_\delta^\infty\colon |w| \leq n \Rightarrow \mu_1(w \uparrow) = \mu_2(w \uparrow) \,\}.$$

In (the technical report version of) [23] one finds a theorem stating that such a construction preserves completeness. So, in particular, *PMS* is a complete metric space.

The set *SPMS* (sets of probability measures over sequences) consists of all nonempty and compact subsets of *PMS*. We assume *SPMS* to come equipped with the Hausdorff-distance d_H, which is given by

$$\begin{aligned} d_H(S_1, S_2) = \inf\{\, \alpha > 0 \mid & \forall x \in S_1\, \exists y \in S_2 [d(x,y) < \alpha], \\ & \forall y \in S_2\, \exists x \in S_1 [d(x,y) < \alpha] \,\}\,. \end{aligned}$$

It is well-known that the collection of nonempty compacta with the Hausdorff-distance is complete whenever the underlying metric space is complete. Thus, in our situation, *SPMS* is a complete metric space.

The Dirac-measure for the string w is denoted by 1_w. So $1_w(B) = 1$ if $w \in B$, and $1_w(B) = 0$ otherwise, for any Borel-set $B \subseteq \mathcal{A}_\delta^\infty$. Dirac-measures are Borel probability measures of compact support, viz. the singletons $\{w\}$. If $\mu_1, \ldots, \mu_n$ are probability measures ($n \geq 1$) and $\rho_1, \ldots, \rho_n$ positive real numbers such that their sum equals one, the balanced sum $(\rho_1 \cdot \mu_1) \oplus \cdots \oplus (\rho_n \cdot \mu_n)$, assigning $\rho_1 \cdot \mu_1(B) + \cdots + \rho_n \cdot \mu_n(B)$ to any Borel-set B of $\mathcal{A}_\delta^\infty$, is a probability measure as well. The notation $\oplus$ is used instead of $+$ to signify that this is a weigthed sum with weigths summing up to one, as the space *PMS* is not closed under general addition of functions. For $\mu \in \mathit{PMS}$ and $a \in \mathcal{A}$, the Borel probability measure $a \cdot \mu$ is defined by $(a \cdot \mu)(B) = \mu(\{\, w \mid a \cdot w \in B \,\})$. The operation $a \cdot \mu$ is a standard construction for a measure μ called the measure along pref_a, where pref_a is prefixing with a.

A mapping $f \colon M_1 \to M_2$ from a metric space M_1 to a metric space M_2 is called nonexpansive if $d_2(f(x), f(y)) \leq d_1(x,y)$ for all $x, y \in M_1$. The collection of all nonexpansive mappings from M_1 to M_2 is denoted by $M_1 \to_1 M_2$. The distance of uniform convergence d_F given by

$$d_F(f_1, f_2) = \sup\{\, d_2(f_1(x), f_2(x)) \mid x \in M_1 \,\}$$

makes $M_1 \to_1 M_2$ a complete metric space provided the metric space M_2 is complete (cf. [4]). A function $g \colon M_1 \to M_2$ is called contractive whenever $d_2(g(x), g(y)) \leq \alpha d_1(x,y)$ for all $x, y \in M_1$, where α is a constant, $0 \leq \alpha < 1$. The classical result, referred to as Banach's Fixed Point Theorem (used below for the justification of the definition of the operational models) states that any contraction $g \colon M \to M$ on a complete metric space M has a unique fixed point $\mathit{fix}(g)$.

The measures d_B on strings, d_H on sets and d_F on nonexpansive functions are standard constructions in metric semantics. More details can be found in, e.g., [4]. See [23] for more details on Borel probability measures and on d_M.

3 The language $\mathcal{L}_0$: choice, chance and failure

The first language provides a fail-statement and multi-ary nondeterministic and probabilistic choice. We assume that a declaration D, which gives a statement for each procedure variable, is fixed. The fixed declaration D is used instead of an explicit recursive construct such as $\mu x.s$. For a procedure variable x, the declaration D gives the body $D(x)$. The advantage of this approach is that the definition is unfolded stepwise and there is no need for a fixed point argument on the syntactical level. We sometimes use the notation $x \Leftarrow s$ to denote that $D(x) = s$. More details and examples can be found in e.g. [4]. A transition system with so-called ν-steps and ρ-steps for nondeterministic and probabilistic choice is given. From the transition system operational models of differing distinguishing power are obtained in Definition 3.5.

Definition 3.1 *Let Act and PVar be two primitive classes of actions and procedure variables, respectively, ranged over by a and x. The language $\mathcal{L}_0$ is the syntactic class of statements, with typical element s, given by*

$$s ::= a \mid \mathit{fail} \mid x \mid s; s \mid \square_{i=1}^{n} s_i \mid \oplus_{i=1}^{n} \rho_i \cdot s_i$$

where $n \geq 2$, $0 < \rho_i < 1$, $(i = 1, \ldots, n)$ and $\rho_1 + \cdots + \rho_n = 1$.

Actions a, procedure variables x and sequential compositions $s_1; s_2$ are interpreted as usual. The declaration D gives the body $D(x)$ for a procedure variable x. The body must consist of a guarded statement, i.e. recursion is restricted to guarded recursion. This is a technical detail and not elaborated any further. The fail-statement *fail* embodies abnormal inaction or failure. There will be no activity after its execution. The statement $\square_{i=1}^{n} s_i$ denotes nondeterministic choice. It is assumed that the bound n is at least 2. The statement selects on s_i, with $1 \leq i \leq n$, and after that acts exactly like s_i. For a binary nondeterministic choice the notation $s_1 \square s_2$ is employed.

The statement $\oplus_{i=1}^{n} \rho_i \cdot s_i$ denotes probabilistic choice or chance. Again the bound n is at least 2. Each number ρ_i is strictly between 0 and 1 and the sum of all the ρ_i's equals 1. For the execution of the statement $\oplus_{i=1}^{n} \rho_i \cdot s_i$ with probability ρ_i the statement s_i is selected and executed. For the binary probabilistic choice we write $s_1 \oplus_\rho s_2$ which executes s_1 with probability ρ and s_2 with probability $1 - \rho$.

Next we present the transition system $\mathcal{T}_0$ for $\mathcal{L}_0$. In Definition 3.5 we obtain from this single transition system various semantical models. The basic idea is that a statement in $\mathcal{L}_0$ is either deterministic, nondeterministic, probabilistic or failing. A statement is in exactly one of these modes, which is (modulo body replacement for procedure variables) determined by the form of the statement. The special symbol E, also referred to as the empty statement, is introduced to represent a terminated computation.

In order to have a separation of concerns we distinguish several types of arrows in the transition system $\mathcal{T}_0$. The type is in fact determined by the label. (A similar technique is employed in [19].) The labels of the transitions out of a statement can either be: an action for the deterministic case, the special symbol ν for the nondeterministic case, or a real number ρ in the open interval $(0,1)$ for the probabilistic case. The absence of any transition covers the case of deadlock for a proper statement; the empty statement E has no transitions but is postulated to be terminating.

Definition 3.2 *Distinguish two fresh symbols* E *and* ν. *The transition system* $\mathcal{T}_0$ *for* $\mathcal{L}_0$ *has the set* $\mathcal{L}_0 \cup \{\mathrm{E}\}$ *as its configurations,* $Act \cup \{\nu\} \cup (0,1)$ *as its labels (r denotes any configuration, and* λ *denotes any label), and the following axiom and rules:*

$$a \xrightarrow{a} \mathrm{E} \qquad \textit{(Act)}$$

$$\frac{D(x) \xrightarrow{\lambda} r}{x \xrightarrow{\lambda} r} \quad \textit{(Rec)} \qquad\qquad \frac{s_1 \xrightarrow{\lambda} r}{s_1; s_2 \xrightarrow{\lambda} r; s_2} \quad \textit{(Seq)}$$

$$\Box_{i=1}^{n}\, s_i \xrightarrow{\nu} s_i \quad \textit{(Choice)} \qquad\qquad \oplus_{i=1}^{n}\, \rho_i \cdot s_i \xrightarrow{\rho_i} s_i \quad \textit{(Chance)}$$

A statement consisting of a single action a performs this a and then terminates. Recursion is handled by unfolding. The transitions for a procedure variable x are precisely those of its body, i.e. the statement $D(x)$. If the first component s_1 of a sequential composition $s_1; s_2$ can do a λ-transition, the statement $s_1; s_2$ itself can do a λ-transition as well. However, the configuration r_1 of the first transition is modified with s_2 resulting in $r_1; s_2$. In case r_1 equals E, then by convention the configuration $r_1; s_2$ should be read as s_2. Selection of an alternative in a nondeterministic choice is made explicit via a ν-transition. Likewise, the selection of a component of a probabilistic choice is signaled by a ρ-transition, where the value ρ is precisely the probability of the component to be chosen. Note that for the fail-statement there is no axiom or rule. As a consequence, *fail* has no transitions. A similar approach, though for probabilistic transitions only, can be found in [9] and [19].

Definition 3.3 *A statement* s *is said to be*

(i) deterministic if s *has exactly one transition which is of the form* $s \xrightarrow{a} r$*;*

(ii) nondeterministic if s *has transitions of the form* $s \xrightarrow{\nu} s'$ *exclusively;*

(iii) probabilistic if s *has transitions of the form* $s \xrightarrow{\rho} s'$ *exclusively;*

(iv) failing if s has no transitions.

A statement s is called live if it does not fail, i.e. s is live if there exists a transition $s \xrightarrow{\lambda} r$.

It is easy to show by induction on a suitable weight function that each statement s is determinsitic, nondeterministic, probabilistic or s fails. The limitation to guarded recursion is essential here.

In case s is nondeterministic, we write $s \Rightarrow S$ if $S = \{\!|\, s' \mid s \xrightarrow{\nu} s' \,|\!\}$, i.e. the multiset of statements reachable in exactly one ν-step. In case s is probabilistic, we write $s \Rightarrow T$ if $T = \{\!|\, \langle \rho, s' \rangle \mid s \xrightarrow{\rho} s' \,|\!\}$, i.e. the multiset of probability/statement pairs for which s' is reachable from s via exacly one ρ-transition. Note that the use of multisets, rather than sets, is important in the context of probability. The number of elements corresponds with the number of prooftrees of transitions with s at the left-hand side. For example, we have $a \oplus_{\frac{1}{2}} a \Rightarrow \{\!|\, \frac{1}{2} \cdot a, \frac{1}{2} \cdot a \,|\!\}$ by using rule (Chance) for $i = 1$ and for $i = 2$.

Next we discuss how to obtain semantical models from the transition system $\mathcal{T}_0$ for the various interpretations of the choice operators $\Box$ and $\oplus$. Any interpretation of the nondeterministic choice can arbitrarily be combined with an interpretation for the probabilistic choice and visa versa. As there are three different interpretations of both nondeterministic choice and probabilistic choice, there are nine different semantical models to be considered.

The nondeterministic choice can be local or global. In the latter case there is a further distinction between "don't care" and "don't know" nondeterminism. In the "don't care"-case a branch, i.e. any component s_i of a nondeterministic choice $\Box_{i=1}^{n} s_i$, can be taken if it is not failing. So the component s_i has an a-step, ν-step or ρ-step. Once the component s_i is chosen, the other options, $s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n$, are dispensed with. For the "don't know"-case any component of a nondeterministic choice can be selected. However, if from $\Box_{i=1}^{n} s_i$ the component s_i is selected, the options $s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n$ are stored. If the computation for s_i fails before executing any action, i.e. before performing an a-step, one of the stored alternatives is selected and takes over control.

The probabilistic choice can either be unconditional or conditional. The unconditional probabilistic is similar to the local nondeterministic choice; a component is selected regardless of its possible steps. In the conditional interpretation of the probabilistic choice the selection of a component is done more carefully. Conditional probability is similar to global nondeterminism. For the "don't care"-case, a failing component $\rho_i \cdot s_i$ will not be selected. The probability ρ_i of the option is proportionally added to the other components. In the "don't know"-case an option will not have effect if it will not, eventually, produce an action. As for the "don't know"-case of global nondeterminism, the other options are stored when exploring a certain branch and reinvoked later when this exploration fails to perform any a-step.

Conceptually one can consider discarding of options simultaneously for all failing and/or non-productive components. This gives the same result as removing them one by one. So, for "don't care" probabilistic choice, we will have that

$$\oplus_{i=1}^{n} \rho_i \cdot s_i = \oplus_{i \in I} \frac{\rho_i}{R} \cdot s_i$$

where $I = \{\, i \mid 1 \leq i \leq n,\ s_i \text{ live} \,\}$ and R is the scaling factor given by $R = \sum_{i \in I} \rho_i$.

In order to be able to handle the "don't know" version of global nondeterministic choice and conditional probabilistic choice, the other alternatives should be stored when selecting an option in order to recover from a deadlock. To this end we introduce alternatives, which are essentially stacks of multisets of statements.

Definition 3.4 *The collection Alt of alternatives, with typical element A, is given by the clause $A ::= \Delta \mid S \triangleright A \mid T \triangleright A$ where Δ is a fresh symbol, called the empty alternative, S is a nonempty multiset of statements s, and T is a nonempty multiset of probability-statement pairs $\langle \rho, s \rangle$.*

The empty alternative Δ denotes that there are no more alternatives available. The two other cases are $S \triangleright A$, $T \triangleright A$. Here S consists of statements which apparently still remain from the resolution of some nondeterministic choice $\square_{i=1}^{n} s_i$. Likewise, T consists of the alternatives still open for the resolution of a probabilistic choice $\oplus_{i=1}^{n} \rho_i \cdot s_i$. Since this resolution also depends on the probability with which the branch is taken pairs $\langle \rho, s \rangle$ are used for this. Below we will employ the constructions $(S \backslash s) \triangleright A$ and $(T \backslash \langle \rho, s \rangle) \triangleright A$ for deletion of one element from the topmost multiset. The convention $\emptyset \triangleright A = A$ will be employed in case $S \backslash s$ or $T \backslash \langle \rho, s \rangle$ becomes the empty multiset, so that the resulting stacks are still of the proper format. Occasionally we write a statement s in $s \triangleright A$ instead of the one-element multiset $\{\!|\, s \,|\!\}$ containing it. Similar for a probability/statement pair $\langle \rho, s \rangle$.

We distinguish the abbreviations *loc*, *gdc* and *gdk* for the three forms of nondeterminism, and *unc*, *cdc* and *cdk* for the three forms of probability. Hence, for example, $\mathcal{O}_{gdc,unc}$ is the model with "global-don't care" nondeterministic choice and unconditional probabilistic choice.

Definition 3.5 *The semantical mapping $\mathcal{O}_{i,j}[\![\cdot]\!] \colon \mathcal{L}_0 \to SPMS$, for i = loc, gdc, gdk and j = unc, cdc, cdk, is given by*

$$\mathcal{O}_{i,j}[\![s]\!] = \mathcal{O}_{i,j}(s)[\Delta]$$

where the mapping $\mathcal{O}_{i,j}\colon (\mathcal{L}_0 \cup \{\mathrm{E}\}) \to Alt \to SPMS$ *is defined as follows:*

(a) $\mathcal{O}_{i,j}(s_0)[A] = \{\, a \cdot \mu \mid s_0 \xrightarrow{a} r, \mu \in \mathcal{O}_{i,j}(r)[\Delta] \,\}$ *for* s_0 *deterministic,*

(b) – $\mathcal{O}_{loc,j}(s_0)[A] = \bigcup \{\, \mathcal{O}_{loc,j}(s)[A] \mid s \in S \,\}$,

– $\mathcal{O}_{gdc,j}(s_0)[A] = \bigcup \{\, \mathcal{O}_{gdc,j}(s)[A] \mid s \in S' \,\}$ *if* S' *is nonempty,*

$\mathcal{O}_{gdc,j}(s_0)[A] = \mathcal{O}_{gdc,j}(\mathsf{fail})[A]$ *otherwise,*

– $\mathcal{O}_{gdk,j}(s_0)[A] = \bigcup \{\, \mathcal{O}_{gdk,j}(s)[(S \backslash s) \rhd A] \mid s \in S \,\}$

for s_0 *nondeterministic, with* $s_0 \Rightarrow S$ *and* $S' = \{\!|\, s \in S \mid s \text{ } live \,|\!\}$,

(c) – $\mathcal{O}_{i,unc}(s_0)[A] = \{\, \oplus_{\langle \rho, s \rangle \in T} \ \rho \cdot \mu_s \mid \mu_s \in \mathcal{O}_{i,unc}(s)[A] \ (\mathit{for} \ \langle \rho, s \rangle \in T) \,\}$,

– $\mathcal{O}_{i,cdc}(s_0)[A] = \{\, \oplus_{\langle \rho, s \rangle \in T'} \ \frac{\rho}{R} \cdot \mu_s \mid \mu_s \in \mathcal{O}_{i,cdc}(s)[A] \ (\mathit{for} \ \langle \rho, s \rangle \in T') \,\}$
if T' *is nonempty,*

$\mathcal{O}_{i,cdc}(s_0)[A] = \mathcal{O}_{i,cdc}(\mathsf{fail})[A]$ *otherwise,*

– $\mathcal{O}_{i,cdk}(s_0)[A] = \{\, \oplus_{\langle \rho, s \rangle \in T} \ \rho \cdot \mu_s \mid \mu_s \in \mathcal{O}_{i,cdk}(s)[(T \backslash \langle \rho, s \rangle) \rhd A]$
$(\mathit{for} \ \langle \rho, s \rangle \in T) \,\}$

for s_0 *probabilistic, with* $s_0 \Rightarrow T$ *and* $T' = \{\!|\, \langle \rho, s \rangle \in T \mid s \text{ } live \,|\!\}$ *and* $R = \sum_{\langle \rho, s \rangle \in T'} \rho$,

(d) $\mathcal{O}_{i,j}(s_0)[\Delta] = \{1_\delta\}$,

$\mathcal{O}_{i,j}(s_0)[S \rhd A] = \bigcup \{\, \mathcal{O}_{i,j}(s)[(S \backslash s) \rhd A] \mid s \in S \,\}$,

$\mathcal{O}_{i,j}(s_0)[T \rhd A] = \{\, \oplus_{\langle \rho, s \rangle \in T} \ \frac{\rho}{R(T)} \cdot \mu_{\langle \rho, s \rangle} \mid \mu_{\langle \rho, s \rangle} \in \mathcal{O}_{i,j}(s)[(T \backslash \langle \rho, s \rangle) \rhd A]$
$(\mathit{for} \ \langle \rho, s \rangle \in T) \,\}$,

for s_0 *failing, with* $R(T) = \sum_{\langle \rho, s \rangle \in T} \rho$,

(e) $\mathcal{O}_{i,j}(\mathrm{E})(A) = \{1_\epsilon\}$.

Executing an action a causes commitment to the currently selected alternative. Any other remaining alternatives are removed. Resolving nondeterministic or probabilistic choice adds to the alternative if choice being resolved is of the "don't know" type. In case of failure, one of the alternatives is selected instead and failure is avoided until there are no alternatives left. In all other cases the alternatives remain unaltered.

The equations for the operational models $\mathcal{O}_{i,j}$ above are circular, i.e., the semantical mappings occur at both sides of the clauses. Definition 3.5 can be justified via a so-called higher-order transformation on a complete metric space. In the proof of contractiveness of this transformation absence of divergence of the transition system and its property of finite branching are used. Though straightforward, the proof is cumbersome and the details are omitted here. However, it is here that we benefit from using the metric approach to programming language semantics, where Banach's Fixed Point Theorem is instrumental in the justification of circular definitions. We mention the references [16,4,11] for more details of the application of this technique.

Lemma 3.6

(a) $\mathcal{T}_0$ *is finitely branching.*

(b) The transition system $\mathcal{T}_0$ *has no internal divergence, i.e., there are no statements* $s_0, s_1, \ldots$ *and* $\lambda_1, \lambda_2, \ldots$ *in* $\{\nu\} \cup (0,1)$ *such that* $s_0 \xrightarrow{\lambda_1} s_1 \xrightarrow{\lambda_2} \cdots$ *is an infinite computation.*

(c) Put $Sem = \mathcal{L}_0 \cup \{\mathrm{E}\} \to Alt \to_1 SPMS$. *The model* $\mathcal{O}_{i,j}$ *for* $i = loc, gdc, gdk$, $j = unc, cdc, cdk$ *can be characterized as the unique fixed point of a contraction* $\Phi_{i,j}$ *on the complete metric space* Sem.

Proof The last part of this lemma justifies the previous definition. That the operators $\Phi_{i,j}$ are contractions is shown by induction on a suitable weight function on (statement, alternative) pairs. For a deterministic statement, contractiveness of $\Phi_{i,j}$ follows from the contractiveness (in the argument μ) of the construction $a \cdot \mu$. All other cases are handled by induction.

We can now give the announced theorem showing that the various interpretations of nondeterministic and probabilistic choice give rise to independent models. The proof is elementary. We evaluate the operational semantics of the statements s_1 through s_5 below, to separate the models $\mathcal{O}_{loc,j}$, $\mathcal{O}_{gdc,j}$ and $\mathcal{O}_{gdk,j}$ (for $j = unc, cdc, cdk$). The statements s'_1 through s'_5 can be used in the same way to separate the models $\mathcal{O}_{i,unc}$, $\mathcal{O}_{i,cdc}$ and $\mathcal{O}_{i,cdk}$ (for $i = loc, gdc, gdk$ and using $\oplus$ as shorthand for $\oplus_{1/2}$).

Theorem 3.7 *The operational semantics* $\mathcal{O}_{i,j}$ *with* $i = loc, gdc, gdk$, $j = unc, cdc, cdk$ *are nine independent models for* $\mathcal{L}_0$, *i.e. equality in one model does not generally imply equality in any other model.*

Proof The statements s_1 through s_5 and s'_1 through s'_5 are given by

$$
\begin{array}{ll}
s_1 = a & s'_1 = a \\
s_2 = a \Box \mathit{fail} & s'_2 = a \oplus \mathit{fail} \\
s_3 = a \Box (\mathit{fail} \Box \mathit{fail}) & s'_3 = a \oplus (\mathit{fail} \Box \mathit{fail}) \\
s_4 = (a+b) \Box (\mathit{fail} \Box \mathit{fail}) & s'_4 = (a+b) \oplus (\mathit{fail} \Box \mathit{fail}) \\
s_5 = (a+b) \Box ((\mathit{fail} \Box \mathit{fail}) \oplus (\mathit{fail} \Box \mathit{fail})) & s'_5 = (a+b) \oplus (\mathit{fail} \oplus \mathit{fail}) .
\end{array}
$$

One can show that the meaning of s_1 through s_5 in the various models is given by:

	$\mathcal{O}_{loc,j}$	$\mathcal{O}_{gdc,j}$	$\mathcal{O}_{gdk,j}$
s_1	$\{1_a\}$	$\{1_a\}$	$\{1_a\}$
s_2	$\{1_a, 1_\delta\}$	$\{1_a\}$	$\{1_a\}$
s_3	$\{1_a, 1_\delta\}$	$\{1_a, 1_\delta\}$	$\{1_a\}$
s_4	$\{1_a, 1_b, 1_\delta\}$	$\{1_a, 1_b, 1_\delta\}$	$\{1_a, 1_b\}$
s_5	$\{1_a, 1_b, 1_\delta\}$	$\{1_a, 1_b, 1_\delta\}$	$\{1_a, 1_b, \frac{1}{2}1_a \oplus \frac{1}{2}1_b\}$.

Note that these results are independent of the kind of probability involved (i.e. independent of j). The statements s_1 and s_2 are not identified by $\mathcal{O}_{loc,j}$ while they are identified by both $\mathcal{O}_{gdc,j}$ and $\mathcal{O}_{gdk,j}$. Similarly $\mathcal{O}_{gdc,j}$ does not identify s_2 and s_3 while $\mathcal{O}_{loc,j}$ and $\mathcal{O}_{gdk,j}$ do. Finally s_4 and s_5 are not identified by $\mathcal{O}_{gdk,j}$ whereas $\mathcal{O}_{loc,j}$ and $\mathcal{O}_{gdc,j}$ do identify these statements. This shows that equality in one model cannot be derived from equality in a model with a different kind of non-deterministic choice.
One can show that the statements s'_1 through s'_5 have meanings

	$\mathcal{O}_{i,unc}$	$\mathcal{O}_{i,cdc}$	$\mathcal{O}_{i,cdk}$
s'_1	$\{1_a\}$	$\{1_a\}$	$\{1_a\}$
s'_2	$\{\frac{1}{2}1_a \oplus \frac{1}{2}1_\delta\}$	$\{1_a\}$	$\{1_a\}$
s'_3	$\{\frac{1}{2}1_a \oplus \frac{1}{2}1_\delta\}$	$\{\frac{1}{2}1_a \oplus \frac{1}{2}1_\delta\}$	$\{1_a\}$
s'_4	$\{\frac{1}{2}1_a \oplus \frac{1}{2}1_\delta, \frac{1}{2}1_b \oplus \frac{1}{2}1_\delta\}$	$\{\frac{1}{2}1_a \oplus \frac{1}{2}1_\delta, \frac{1}{2}1_b \oplus \frac{1}{2}1_\delta\}$	$\{1_a, 1_b, \frac{1}{2}1_a \oplus \frac{1}{2}1_b\}$
s'_5	$\{1_a \oplus 1_\delta, \frac{1}{2}1_b \oplus \frac{1}{2}1_\delta\}$	$\{1_a \oplus 1_\delta, \frac{1}{2}1_b \oplus \frac{1}{2}1_\delta\}$	$\{1_a, 1_b, \frac{1}{2}1_a \oplus \frac{1}{2}1_b,$ $\frac{1}{4}1_a \oplus \frac{3}{4}1_b, \frac{3}{4}1_a \oplus \frac{1}{4}1_b\}$.

Using similar reasoning as above it follows that equality in one model can not be derived from equality in a model with a different kind of probabilistic choice, completing the independence proof.

The key differences between the models also become apparent when looking at operational equalities that hold in the models. To demonstrate this, we look at the basic laws of process algebra (see e.g. [1,2,14]), and check which laws are satisfied in the different models. We write $s \simeq_{i,j} s'$ exactly when $\mathcal{O}_{i,j}(s) = \mathcal{O}_{i,j}(s')$. We simple write $s \simeq s'$ when $s \simeq_{i,j} s'$ holds in each of the models (i.e. for $i = loc, gdc, gdk$ and $j = unc, cdc, cdk$). Finally we use $s \simeq_{glob} s'$ to denote operational equality in all models with global nondeterminism and $s \simeq_{cond} s'$ to denote operational equality in all models with conditional probability.

The following equational laws hold in each of the nine models.

$$x \Box y \;\simeq\; y \Box x$$

$$
\begin{aligned}
x \oplus_\rho y &\simeq y \oplus_{1-\rho} x \\
(x \Box y); z &\simeq x; z \Box y; z \\
(x \oplus_\rho y); z &\simeq x; z \oplus_\rho y; z \\
(x; y); z &\simeq x; (y; z)\ .
\end{aligned}
$$

The operators '$\Box$' and '$\oplus$' are symmetric (modulo a label change for the probabilistic choice) and left-distributive over ';'. These properties can be generalized for the n-ary choice operators. The operator ';' is commutative.

For all models we have

$$
\begin{aligned}
a \Box a &\simeq a \\
a \oplus_\rho a &\simeq a
\end{aligned}
$$

and for $i \neq gdk$ or $j = cdk$ the first law can be strengthened to

$$
x \Box x \simeq_{i,j} x\ .
$$

Idempotency of both choices holds for single actions. General idempotency of probabilistic choice does not hold in any of the models. The statement $(a \Box b) \oplus_\rho (a \Box b)$ is not identified with $a \Box b$ in any of the models. This is an instance of the more general phenomenon that the way nondeterministic choices are resolved may depend on the result of previous probabilistic decisions.

In the statement $a \oplus_\rho \mathit{fail}$ there is a single coin toss to decide whether a will be executed or whether the statement will fail. In the statement $(a \oplus_\rho \mathit{fail}) \Box (a \oplus_\rho \mathit{fail})$, there are two coins which can be tossed, resulting in two opportunities for obtaining a. In case of global don't know nondeterminism with unconditional probability this results in a higher probability of executing a. In the case with "global don't know" nondeterminism and "conditional don't care" probability the slightly more complicated example $a \oplus_\rho (\mathit{fail} \Box \mathit{fail})$ versus $(a \oplus_\rho (\mathit{fail} \Box \mathit{fail})) \Box (a \oplus_\rho (\mathit{fail} \Box \mathit{fail}))$ shows that general idempotency for nondeterministic choice does not hold. In the models with other local or don't care nondeterminism there is no way of exploiting the extra opportunities and general idempotency hold for nondeterministic choice. In models with "conditional don't know" probabilistic choice the chance of failure will always be zero or one and the extra opportunities do not matter. General idempotency for '$\Box$' again holds.

For $i \neq gdk$ or $j = cdk$ we have

$$
(x \Box y) \Box z \simeq_{i,j} x \Box (y \Box z)
$$

Due to the use of stratified probabilistic choice, the conditional probabilistic choices are not commutative. A generative probabilistic choice would be commutative (with appropriate label changes). See [9] for details on stratified versus generative models.

The nondeterministic choice is also interpreted as a stratified choice. With global don't know nondeterminism and unconditional probability, the semantics of $(a \oplus_\rho \mathit{fail}) \Box (b \Box c)$ contains $\rho \cdot 1_a \oplus (1-\rho) \cdot 1_c$, which is not present in the semantics of $((a \oplus_\rho \mathit{fail}) \Box b) \Box c$. After failure the first statement can revert to b or c. The second statement can only revert to b. As with the idempotency, the difference can only be observed with global don't know nondeterminism and unconditional or conditional don't care probability. In the other models commutativity for nondeterministic choice does hold.

$$\begin{array}{rcl} \mathit{fail}; x & \simeq & \mathit{fail} \\ \mathit{fail} \Box x & \simeq_{glob} & x \\ \mathit{fail} \oplus_\rho x & \simeq_{cond} & x \end{array}$$

No actions are executed after failure. For global and conditional models we have absorption of failure.

The differences observed show that even the basic equational laws can not be taken for granted in a setting with both probabilistic and nondeterministic choice. All laws have to be carefully checked for the precise types of nondeterministic and probabilistic choice that are considered.

4 The language $\mathcal{L}_1$: adding parallel composition

In this section the language $\mathcal{L}_0$ is extended with parallel composition. The parallel composition, or merge, of two statements $s \| s'$ is the arbitrary interleaving of the behaviours of s and s'. A distributed interpretation is adopted in the sense that there will be progress if either statement is able to make progress. Failure of one statement does not directly cause failure of the system as a whole.

The construction of the operational semantics for $\mathcal{L}_1$ follows the same path as is followed for $\mathcal{L}_0$. First the syntax and the transition system are defined. Based on the transition system the different operational semantics, depending on the interpretation of choice and chance, are introduced. Each step in this construction extends the corresponding step for the language $\mathcal{L}_0$.

The syntax of $\mathcal{L}_1$ is obtained by adding the operator $\|$, called merge, to the syntax of $\mathcal{L}_0$. Again a fixed declaration D is assumed, now yielding statements in $\mathcal{L}_1$. In configurations of the transition system the auxiliary operator $\lfloor\!\lfloor$, called leftmerge, is also used. The configuration $s \lfloor\!\lfloor s'$ describes a parallel composition in which s is the primary component, i.e. the component that is ready to begin first.

Definition 4.1 *The language $\mathcal{L}_1$ is the class of statements, with typical element s, given by*

$$s ::= a \mid \mathit{fail} \mid x \mid s; s \mid \Box_{i=1}^n s_i \mid \oplus_{i=1}^n \rho_i \cdot s_i \mid s \| s$$

where $n \geq 1$, $0 < \rho_i < 1$, $(i = 1, \ldots, n)$ and $\rho_1 + \cdots + \rho_n = 1$.

The auxiliary operator $\lfloor\!\lfloor$ is added to give a class $\mathcal{L}_1^+$ with typical element t, where t is given by the clause

$$t ::= s \mid t; s \mid t \lfloor\!\lfloor s.$$

Note that the auxiliary operator $\lfloor\!\lfloor$ is not allowed as an argument of non-deterministic or probabilistic choice. This is needed for technical reasons as explained after Lemma 4.5. It is possible for $\lfloor\!\lfloor$ on the left-handside of ; or $\lfloor\!\lfloor$ as can be seen by finding the transitions for $(a \| b) \| c$ and $(a \| b); c$ using the transition system given below.

Two new types of arrows are used in the transition system. Transitions labeled with the symbol π are used when parallel composition is resolved as described below. A transition labeled with take-over, τ, describes the event that one parallel component cannot continue and therefore the other component takes over. The transition system $\mathcal{T}_1$ is obtained from $\mathcal{T}_0$ by adding the axioms and rules given below.

Definition 4.2 *Distinguish two fresh symbols π and τ. The transition system $\mathcal{T}_1$ for $\mathcal{L}_1$ has the set $\mathcal{L}_1^+ \cup \{\mathrm{E}\}$ as its configurations and $Act \cup \{\nu, \pi, \tau\} \cup (0,1)$ as its labels (again r denotes a typical configuration, and λ denotes a typical label, furthermore ζ denotes any label other than an action). The rule (Seq) from $\mathcal{T}_0$ is adapted to*

$$\frac{t_1 \xrightarrow{\lambda} r}{t_1; s_2 \xrightarrow{\lambda} r; s_2} \qquad \textit{(Seq)}$$

and the following axioms and rules are added:

$$\begin{array}{l} s_1 \| s_2 \xrightarrow{\pi} s_1 \lfloor\!\lfloor s_2 \\ s_1 \| s_2 \xrightarrow{\pi} s_2 \lfloor\!\lfloor s_1 \end{array} \qquad \textit{(Merge)} \qquad\qquad \frac{t_1 \xrightarrow{a} r}{t_1 \lfloor\!\lfloor s_2 \xrightarrow{a} r \| s_2} \qquad \textit{(Leftmerge 1)}$$

$$\frac{t_1 \xrightarrow{\zeta} t}{t_1 \lfloor\!\lfloor s_2 \xrightarrow{\zeta} t \lfloor\!\lfloor s_2} \qquad \textit{(Leftmerge 2)} \qquad\qquad \frac{t_1 \textit{ fails}}{t_1 \lfloor\!\lfloor s_2 \xrightarrow{\tau} s_2; \textit{fail}} \qquad \textit{(Take Over)}$$

The rule (Seq) is extended so it can also deal with the new types of configurations in $\mathcal{L}_1^+$. The other rules from $\mathcal{T}_0$ are included literally as they are (the syntactic elements s, r and λ now refer to statements, resumptions and labels as defined in this section).

The two axioms (Merge) express that either component of the parallel composition is allowed to begin first, i.e. become the primary component, the $\|$ is resolved into $\lfloor\!\lfloor$. The statement $t_1 \lfloor\!\lfloor s_2$ will behave like t_1 until t_1 has taken an action or fails. The rule (Leftmerge 1) describes the situation in which the primary component t_1 takes an action. Rule (Leftmerge 2) describes the situation in which t_1 has to resolve choice, chance or parallelism or do a *take-over*. The primary component still has to do an action so the $\lfloor\!\lfloor$ remains on the right-handside of the conclusion in rule (Leftmerge 2).

The distributed interpretation of $\|$ implies that failure of one component does not stop the execution of the other components. If, in $t_1 \lfloor\!\lfloor s_2$, the component t_1 cannot take any steps, i.e. fails, the component s_2 will take over. This event is made explicit in the transition system by the use of a τ-transition as described by rule (Take Over). Of course, if a component fails, the parallel composition cannot terminate normally. After all other components terminate or fail, the parallel system as a whole will fail.

Clearly the new axioms and rules do not play a role for a resumption and a declaration both without parallel composition. This means that for statements that are also in $\mathcal{L}_0$ the transition system remains the same.

In $\mathcal{L}_0$ a statement was either failing, deterministic, non-deterministic or probabilistic. In $\mathcal{L}_1$ two new types of statements are present. A statement s is said to be parallel if $s \xrightarrow{\pi} s'$ for some statement s'. If there exists a statement s' such that $s \xrightarrow{\tau} s'$ then s is called a *take-over* statement. It is easy to check that again each statement is of exactly one of these types.

Alternatives for choices that are made have to be remembered in the "don't know" models. In $\mathcal{L}_1$ there is an extra complication compared to $\mathcal{L}_0$. An alternative from within the component will prevent the component from failing and should therefore take precedence over a take-over action. An alternative from outside the parallel composition should only be used if the whole parallel composition fails. As the parallel composition will only fail if all components fail, the take-over action should be tried before reverting to any top level alternative. Examples s_4 and s_5 in Section 5 illustrate the difference between inside and outside options.

In order to be able to make the distinction between inside and outside alternatives the notion of an alternative is extended.

Definition 4.3 *The collection Alt of alternatives, with typical element B, is given by the clause $B ::= \Delta \mid S \triangleright B \mid T \triangleright B \mid *B$, where Δ, S, T are as in Definition 3.4.*

Compared to the definition of alternatives in the previous section, the only addition is $*B$. The $*$ indicates that the alternatives in B are from outside the current parallel composition. A take-over action will be tried before options from $*B$. As take-over indicates failure of the primary component a take-over statement is not considered to be live.

Definition 4.4 *The semantical mapping $\mathcal{O}_{i,j}[\![\cdot]\!]\colon \mathcal{L}_1 \to SPMS$, for i = loc, gdc, gdk and j = unc, cdc, cdk, is given by*

$$\mathcal{O}_{i,j}[\![s]\!] = \mathcal{O}_{i,j}(s)[\Delta]$$

where the mapping $\mathcal{O}_{i,j}\colon Stat \cup \{\mathrm{E}\} \to Alt \to SPMS$ is defined as follows:

(a) $\mathcal{O}_{i,j}(s_0)[B] = \{\, a \cdot \mu \mid s_0 \xrightarrow{a} r, \mu \in \mathcal{O}_{i,j}(r)[\Delta] \,\}$ *for s_0 deterministic,*

(b) – $\mathcal{O}_{loc,j}(s_0)[B] = \bigcup\{\, \mathcal{O}_{loc,j}(s)[B] \mid s \in S \,\}$,

– $\mathcal{O}_{gdc,j}(s_0)[B] = \bigcup\{\, \mathcal{O}_{gdc,j}(s)[B] \mid s \in S' \,\}$ *if S' is nonempty,*

$\mathcal{O}_{gdc,j}(s_0)[B] = \mathcal{O}_{gdc,j}(s)[B]$ *if S' empty and $s \in S$,*

– $\mathcal{O}_{gdk,j}(s_0)[B] = \bigcup\{\, \mathcal{O}_{gdk,j}(s)[(S \backslash s) \triangleright B] \mid s \in S \,\}$

for s_0 nondeterministic, with $s_0 \Rightarrow S$ and $S' = \{\, s \in S \mid s$ live$\,\}$,

(c) – $\mathcal{O}_{i,unc}(s_0)[B] = \{\, \oplus_{\langle \rho,s\rangle \in T}\ \rho \cdot \mu_s \mid \mu_s \in \mathcal{O}_{i,unc}(s)[B]$ *(for $\langle \rho,s\rangle \in T$)* $\,\}$,

– $\mathcal{O}_{i,cdc}(s_0)[B] = \{\, \oplus_{\langle \rho,s\rangle \in T}\ \frac{\rho}{R} \cdot \mu_s \mid \mu_s \in \mathcal{O}_{i,cdc}(s)[B]$ *(for $\langle \rho,s\rangle \in T'$)* $\,\}$ *if T' is nonempty,*

$\mathcal{O}_{i,cdc}(s_0)[B] = \mathcal{O}_{i,cdc}(s)[B]$ *if T' empty and $\langle \rho,s\rangle \in T$,*

– $\mathcal{O}_{i,cdk}(s_0)[B] = \{\, \oplus_{\langle \rho,s\rangle \in T}\ \rho \cdot \mu_s \mid \mu_s \in \mathcal{O}_{i,cdk}(s)[(T \setminus \langle \rho,s\rangle) \triangleright B]$ *(for $\langle \rho,s\rangle \in T$)* $\,\}$

for s_0 probabilistic, with $s_0 \Rightarrow T$, where $T' = \{\, \langle \rho,s\rangle \in T \mid s$ live$\,\}$ and $R = \sum_{\langle \rho,s\rangle \in T'} \rho$,

(d) $\mathcal{O}_{i,j}(s_0)[\Delta] = \{1_\delta\}$,

$\mathcal{O}_{i,j}(s_0)[S \triangleright B] = \bigcup\{\, \mathcal{O}_{i,j}(s)[(S \setminus s) \triangleright B] \mid s \in S \,\}$,

$\mathcal{O}_{i,j}(s_0)[T \triangleright B] = \{\, \oplus_{\langle \rho,s\rangle \in T}\ \frac{\rho}{R(T)} \cdot \mu_s \mid \mu_s \in \mathcal{O}_{i,j}(s)[(T \setminus \langle \rho,s\rangle) \triangleright B]$ *(for $\langle \rho,s\rangle \in T$)* $\,\}$,

$\mathcal{O}_{i,j}(s_0)[*B] = \{1_\delta\}$,

for s_0 failing, with $R(T) = \sum_{\langle \rho,s\rangle \in T} \rho$,

(e) $\mathcal{O}_{i,j}(s_0)[B] = \bigcup\{\, \mathcal{O}_{i,j}(s)[*B] \mid s \in S \,\}$ *for s_0 parallel, with $s_0 \Rightarrow S$,*

(f) $\mathcal{O}_{i,j}(s_0)[\Delta] = \mathcal{O}_{i,j}(s_1)[\Delta]$

$\mathcal{O}_{i,j}(s_0)[S \triangleright B] = \bigcup\{\, \mathcal{O}_{i,j}(s)[(S \setminus s) \triangleright B] \mid s \in S \,\}$,

$\mathcal{O}_{i,j}(s_0)[T \triangleright B] = \{\, \oplus_{\langle \rho,s\rangle \in T}\ \rho \cdot \mu_s \mid \mu_s \in \mathcal{O}_{i,j}(s)[(T \setminus \langle \rho,s\rangle) \triangleright B]$ *(for $\langle \rho,s\rangle \in T$)* $\,\}$,

$\mathcal{O}_{i,j}(s_0)[*B] = \mathcal{O}_{i,j}(s_1)[B]$

for s_0 overtaking with $s_0 \xrightarrow{\tau} s_1$,

(g) $\mathcal{O}_{i,j}(\mathrm{E})[B] = \{1_\epsilon\}$.

Showing that $\mathcal{O}_{i,j}$ is well-defined can again be done by characterizing $\mathcal{O}_{i,j}$ as the unique fixed point of a contraction $\Phi_{i,j}$. This construction work exactly like before and requires that transition system $\mathcal{T}_1$ is finitely branching and divergence free. These properties of the transition system can be shown to hold by induction on a suitable weight function. The details are omitted.

Definition 4.4 is very similar to Definition 3.5. Besides the added clauses for parallel and take-over statements, there are only minor differences. For a failing statement, the definition is extended for $*B$. The options in B are available only after a τ-step, so they cannot be used to avoid deadlock.

The "global don't care" nondeterministic choice should work as follows. Any of the live options may be chosen. If no live options exist a take-over action from one of the options may occur. If no option can take a take-over action, the choice should fail.

The first clause for "global don't care" nondeterministic choice has not changed and clearly satisfies the case that there are live options. If no live options exist the second clause is used. An arbitrary option s is taken and the choice will behave like this option. For this definition to be correct, the choice of s must not matter, i.e. all option have to behave the same. The following lemma states that indeed all non-live options of a choice behave the same.

Lemma 4.5 *If one option of a non-deterministic choice is a take-over statement then all non-live options are take-over statements and moreover the resulting configuration after the take-over action is the same for all these options.*

This lemma holds since a take-over statement is a parallel composition with a failing primary component. The nondeterministic choice must be part of the primary component due to the syntactic restrictions. The choice made in the primary component will not change the "parallel environment" which will take over if the primary component fails.

The syntactic restriction that the auxiliary operator $\Vert\!\!\lfloor$ may not be used within an argument of nondeterministic choice is essential. Without this restriction, one could form e.g. the statement $(\mathit{fail} \Vert\!\!\lfloor\, a)\Box(\mathit{fail} \Vert\!\!\lfloor\, b)$ for which the lemma would not hold. Formal justification of Lemma 4.5 can be done can by inspection of the transition system and by using weight induction. The similar justification for "conditional don't care" probability is skipped. The adjustments made to the definition are all caused by take-over and parallel steps. For a statement that does not contain any parallelism, and, therefore, no take-over, the definition is the same as Definition 3.5. In the Theorem below the operational semantics from Section 3 is denoted by $\mathcal{O}^{(fail)}_{i,j}$ and the operational semantics from this section by $\mathcal{O}^{(par)}_{i,j}$.

Theorem 4.6 *The operational semantics of $\mathcal{L}_1$ is a consistent extension of the operational semantics of $\mathcal{L}_0$, i.e. $\mathcal{O}^{(par)}_{i,j} \mid \mathcal{L}_0 = \mathcal{O}^{(fail)}_{i,j}$.*

Proof The transition system for a statement $s \in \mathcal{L}_0$ given by $\mathcal{T}_0$ is the same as that given by $\mathcal{T}_1$. Moreover, the adjustments to the definition of $\mathcal{O}$ have no effect for statements of $\mathcal{L}_0$.

A direct consequence of Theorem 4.6 is that the result from Theorem 3.7 also holds for $\mathcal{L}_1$.

Corollary 4.1 *The operational semantics $\mathcal{O}_{i,j}$ with i = loc, gdc, gdk, j = unc, cdc, cdk are 9 independent models for $\mathcal{L}_1$.*

5 Examples

Let $s_1 = x$ with $x \Leftarrow (a \oplus_\rho \mathit{fail}); x$. In the models with conditional probability this statement will have 1_{a^ω} as the only possible distribution. In the unconditional model one may wonder about the probability that s_1 never fails. We calculate

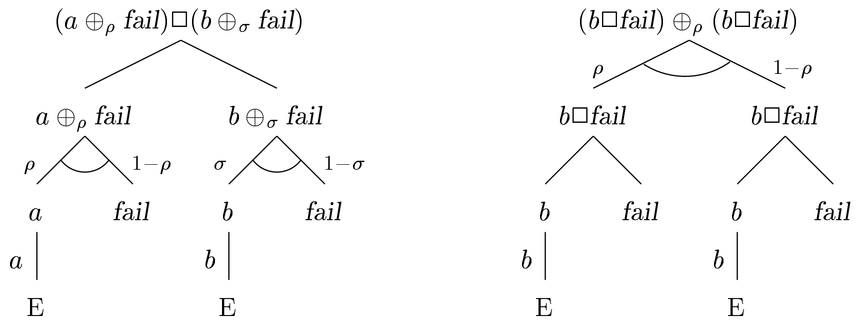
$$\begin{aligned}\mathcal{O}_{i,unc}(s_1)[\Delta] &= \mathcal{O}_{i,unc}((a \oplus_\rho \mathit{fail}); x)[\Delta] \\ &= \{\, \rho \cdot \mu_1 \oplus (1-\rho) \cdot \mu_2 \mid \mu_1 \in \mathcal{O}_{i,unc}(a; x)[\Delta], \\ &\qquad\qquad\qquad\qquad\qquad \mu_2 \in \mathcal{O}_{i,unc}(\mathit{fail}; x)[\Delta] \,\} \\ &= \{\, \rho \cdot (a \cdot \mu_1) \oplus (1-\rho) \cdot 1_\delta \mid \mu_1 \in \mathcal{O}_{i,unc}(x)[\Delta] \,\}\,.\end{aligned}$$

By a metric argument one can show that there is only one distribution μ in $\mathcal{O}_{i,unc}(s_1)[\Delta]$. For this distribution we have $\mu = \rho \cdot (a \cdot \mu) \oplus (1-\rho) \cdot 1_\delta$. The chance that s_1 will never fail is given by $\mu(Act^\infty)$, where $Act^\infty = Act^* \cup Act^\omega$ has no sequence ending in δ. Using the equation for μ above we see that

$$\begin{aligned}\mu(Act^\infty) &= \rho \cdot (a \cdot \mu(Act^\infty)) + (1-\rho) \cdot 1_\delta(Act^\infty) \\ &= \rho \cdot \mu(Act^\infty) + 0,\end{aligned}$$

so for any $\rho < 1$ the statement s_1 will eventually fail with probability one.

In the next examples both nondeterminism and probability are involved. Let $s_2 = ((a \oplus_\rho \mathit{fail}) \Box (b \oplus_\sigma \mathit{fail})$ and $s_3 = (b \Box \mathit{fail}) \oplus_\rho (b \Box \mathit{fail})$.

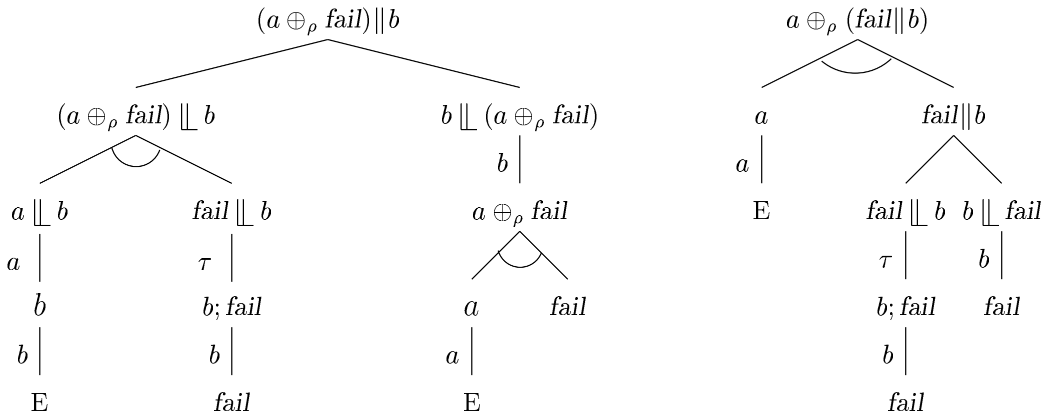


One can easily check that the operational semantics of s_2 is $\{\, 1_a, 1_b \,\}$ if probability is conditional. More interesting is the situation when probability is unconditional. With local or "don't care global" choice the result is $\{\, \rho \cdot 1_a \oplus (1-\rho) \cdot 1_\delta, \sigma \cdot 1_b \oplus (1-\sigma) \cdot 1_\delta \,\}$. For the local choice this should be clear. For the "global don't care" choice this result is obtained because both $a \oplus_\rho \mathit{fail}$ and $b \oplus_\sigma \mathit{fail}$ are live; they can make a probabilistic choice.

For the "don't know" case the operational semantics of s_2 is given by $\{\, \rho{\cdot}1_a{\oplus}(1{-}\rho)\sigma{\cdot}1_b{\oplus}(1{-}\rho)(1{-}\sigma){\cdot}1_\delta, \sigma{\cdot}1_b{\oplus}(1{-}\sigma)\rho{\cdot}1_a{\oplus}(1{-}\sigma)(1{-}\rho){\cdot}1_\delta \,\}$. The first of these distributions is obtained as follows: In the nondeterministic choice the first alternative is chosen and the second remembered. If the probabilistic choice results in a, an a is produced, otherwise the result is fail, which is replaced by the other alternative $b \oplus_\sigma \mathit{fail}$.

Similarly s_3 has $\{\, 1_b \,\}$ as the operational semantics if nondeterminism is global. If nondeterminism is local, both unconditional and "don't care conditional" choice result in the semantics $\{\, 1_b, \rho{\cdot}1_b{\oplus}(1{-}\rho){\cdot}1_\delta, \rho{\cdot}1_\delta{\oplus}(1{-}\rho){\cdot}1_b, 1_\delta \,\}$. The "don't know conditional" choice will be able to avoid failure if there is an alternative. This results in the operational semantics $\{\, 1_b, 1_\delta \,\}$. Note that the example s_3 shows that in the models with local nondeterminism $b\Box\mathit{fail} \neq (b\Box\mathit{fail}) \oplus_\rho (b\Box\mathit{fail})$. If *fail* is replaced by a then the inequality also holds in the models with global nondeterminism.

The next examples show the difference between options outside the parallel composition and options inside the parallel composition. We also use these examples to illustrate how the operational semantics is calculated using the definition. Let $s_4 = (a \oplus_\rho \mathit{fail}) \mathbin{|\!\lfloor} b$ and $s_5 = a \oplus_\rho (\mathit{fail} \mathbin{|\!\lfloor} b)$.



In the models where probabilistic choice is "don't know conditional" we have

$$
\begin{aligned}
\mathcal{O}(s_4)[\Delta] &= \mathcal{O}((a \oplus_\rho \mathit{fail}) \mathbin{|\!\lfloor} b)[*\Delta] \;\cup\; \mathcal{O}(b \mathbin{|\!\lfloor} (a \oplus_\rho \mathit{fail}))[*\Delta] \\
&= (\mathcal{O}(a \mathbin{|\!\lfloor} b)[(\mathit{fail} \mathbin{|\!\lfloor} b) \triangleright *\Delta] \oplus_\rho \mathcal{O}(\mathit{fail} \mathbin{|\!\lfloor} b)[(a \mathbin{|\!\lfloor} b) \triangleright *\Delta]) \\
&\quad \cup\; b \cdot \mathcal{O}(a \oplus_\rho \mathit{fail})[\Delta] \\
&= (a \cdot \mathcal{O}(b)[\Delta] \oplus_\rho \mathcal{O}(a \mathbin{|\!\lfloor} b)[*\Delta]) \\
&\quad \cup\; b \cdot (\mathcal{O}(a)[\mathit{fail} \triangleright \Delta] \oplus_\rho \mathcal{O}(\mathit{fail})[a \triangleright \Delta]) \\
&= \{\, 1_{ab} \,\}{\oplus_\rho}\{\, 1_{ab} \,\} \cup\; b \cdot (\{\, 1_a \,\}{\oplus_\rho}\{\, 1_a \,\}) \\
&= \{\, 1_{ab}, 1_{ba} \,\}
\end{aligned}
$$

and

$$\begin{aligned}\mathcal{O}(s_5)[\Delta] &= \mathcal{O}(a)[\langle 1-\rho, \mathit{fail}\|b \triangleright \Delta\rangle] \oplus_\rho \mathcal{O}(\mathit{fail}\|b)[\langle \rho, a \triangleright \Delta\rangle] \\ &= \{\, 1_a \,\} \oplus_\rho (\mathcal{O}(\mathit{fail} \mathbin{\|\!\_} b)[*\langle \rho, a \triangleright \Delta\rangle] \;\cup\; \mathcal{O}(b \mathbin{\|\!\_} \mathit{fail})[*\langle \rho, a \triangleright \Delta\rangle]) \\ &= \{\, 1_a \,\} \oplus_\rho (\mathcal{O}(b; \mathit{fail})[\langle \rho, a\rangle \triangleright \Delta] \;\cup\; b \cdot \mathcal{O}(\mathit{fail})[\Delta]) \\ &= \{\, 1_a \,\} \oplus_\rho (b \cdot \mathcal{O}(\mathit{fail})[\Delta] \;\cup\; \{\, 1_{b\delta} \,\}) \\ &= \{\, \rho \cdot 1_a \oplus (1-\rho) \cdot 1_{b\delta} \,\}\,.\end{aligned}$$

In the models where probabilistic choice is unconditional the operational semantics for s_4 becomes $\{\, \rho \cdot 1_{ab} \oplus (1-\rho) \cdot 1_{b\delta}, \rho \cdot 1_{ba} \oplus (1-\rho) \cdot 1_{b\delta} \,\}$. For s_5 the operational semantics is again $\{\, \rho \cdot 1_a \oplus (1-\rho) \cdot 1_{b\delta} \,\}$. The examples s_4 and s_5 show the need for the operator $*$ as introduced in Section 4. In s_4 the alternative a should replace the *fail*. In s_5 the alternative a can only replace the parallel composition as a whole. The parallel composition will only fail in the first step if both components fail in the first step.

6 Concluding remarks and related work

Above we have shown how, for various interpretations, the concepts of nondeterminacy, probability, concurrency and failure can be dealt with in one single framework. The key idea is to make nondeterministic choice, probabilistic chance, parallel branching and recovery from failure by take-over explicit in the transition system. Subsequently, operational models can be characterized equationally; their definitions can be justified by metrical arguments. The resulting semantical models yield all different process equivalences; there are examples showing that all semantic equivalences are independent. However, the contribution of this paper unravels only a small portion of the interplay of the various computational concepts.

Further relationships of the proposed models and other types of process equivalences, especially simulations and bisimulations, both in an abstract setting [18,9,5,23] and in the context of I/O automata should be established [21,20,24]. In particular we are curious how our approach of explicit labels fits with the probabilistic I/O automaton model, since this might open the way to concrete applications. A strong-type bisimulation which is correct for each operational semantics, is easy to construct but uninformative. Too few statements are identified to make this notion useful. It is, however, not clear how to construct a weaker bisimulation which is correct for all models. For instance a “weak ν bisimulation” which does not distinguish between one and two ν steps is not correct for the “global don’t know” models. This means that different notions of bisimulation are required for the different models.

It will be interesting to find a complete taxonomy of mixtures of nondeterminism, probability and concurrency that covers the plethora of existing probabilistic models. The present paper constitutes only a small part of this.

For example, of the process algebraic laws of [14] some hold in all our models, some hold in some of our models, but law prA3 stating $s \oplus_\rho s = s$ does generally not hold in any of our models. (See Section 5, example s_3.) The question arises of what is a reasonable model for the various concepts under given circumstances. In [19] also nondeterministic steps are made explicit in the transition system, resulting in a tripartition of action, probabilistic and nondeterministic nodes. The treatment there is restricted to unconditional probabilistic choice, but incorporates a full CSP-style parallel operator. In [12] nondeterministic choice is modelled as continuously many probabilistic choices. In the model proposed in [20] nondeterminism is also supposed to be resolved into probabilistic choice. However in both approaches the semantics serves as a starting point for the development of a notion of refinement and of simulation, respectively, in a probabilistic setting. In [6], dealing with performance modelling, a similar route is taken. There continuous distributions for timing are employed and bisimulations and an axiomatic semantics are provided. In [10] a notion of probabilistic refusal testing dealing both with internal and external choice is proposed and additional fully abstract models are given. Of course, notions of bisimulation naturally arise from our operational models, but these are, as mentioned above, too fine and further work has to be done.

A last point relates to the extension of $\mathcal{L}_1$ with synchronization. One of the technical advantages of $\mathcal{L}_1$ is that failure is static. For synchronization this is completely different, since by progress of other parallel components an apparent synchronization failure might disappear. In contrast to the approach of [22] our framework, at least for the moment, fails to work for synchronization. It is plausible that a combination of (discrete and/or continuous) transition systems and more advanced measure-theoretical considerations will be necessary to handle this.

Acknowledgments We are grateful to Jaco de Bakker for useful comments on an earlier version of this paper, and acknowledge the feedback from the PROBMIV'98 workshop in Indianapolis.

References

[1] S. Andova. Process algebra with probabilistic choice. In *Proceedings ARTS'99*, Bamberg, 1999. LNCS. to appear.

[2] J.C.M. Baeten and J.A. Bergstra. Process algebra with partial choice. In B. Jonsson and J. Parrow, editors, *Proc. CONCUR'94*, pages 465–480. LNCS 836, 1994.

[3] C. Baier and M. Kwiatkowska. Domain equations for probabilistic processes (extended abstract). In *Proc. Express'97*. Electronic Notes in Theoretical Computer Science 7, 1997.

[4] J.W. de Bakker and E.P. de Vink. *Control Flow Semantics.* The MIT Press, 1996.

[5] R. Blute, J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. In *Proc. LICS'97*, pages 149–159, Warsaw, 1997.

[6] J.-P. Katoen D'Argenio, P.R. and E. Brinksma. A stochastic automata model and its algebraic approach. In *Proc. PAPM'97*, pages 1–16. University of Twente, Enschede, 1995. Appeared as CTIT Technical Report 97–14.

[7] A. Edalat. Domain theory and integration. *Theoretical Computer Science*, 151:163–193, 1995.

[8] A. Giacalone, C. Jou, and S.A. Smolka. Algebraic reasoning for probabilisitic concurrent systems. In M. Broy and C.B Jones, editors, *Proc. Working Conference on Programming Concepts and Methods*, pages 443–458. IFIP TC2, Sea of Gallilee, 1990.

[9] R.J. van Glabbeek, S.A. Smolka, and B. Steffen. Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, 121:59–80, 1995.

[10] C. Gregorio-Rodríguez and M. Núnez. Denotational semantics for probabilistic testing. In C. Baier, M. Huth, M. Kwiatkoswka, and M. Ryan, editors, *Preliminary proceedings PROBMIV'98*, pages 122–137. University of Birmingham, 1998. Appeared as technical report CSR–98–4.

[11] J.I. den Hartog. Comparative semantics for a process language with probabilistic choice and non-determinism. Technical Report IR–445, Vrije Universiteit, Amsterdam, 1998.

[12] He Jifeng, A. McIver, and K. Seidel. Probabilistic models for the guarded command language. Technical report, Oxford University Computing Laboratory, 1995. To appear in Science of Computer Programming.

[13] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall International, 1985.

[14] Baeten. J.C.M., J.A. Bergstra, and S.A. Smolka. Axiomatizing probabilistic processes: ACP with generative probabilities. *Information and Computation*, 121:234–255, 1995.

[15] C. Jones and G. Plotkin. A probabilistic powerdomain of evaluations. In *Proc. LICS'89*, pages 186–195. Asilomar, 1989.

[16] J.N. Kok and J.J.M.M. Rutten. Contractions in comparing concurrency semantics. *Theoretical Computer Science*, 76:179–222, 1990.

[17] M. Kwiatkowska and G. Norman. Probabilistic metric semantics for a simple language with recursion. In W. Penczek and A. Szalas, editors, *Proc. MFCS'96*, pages 419–430. LNCS 1113, 1996.

[18] K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.

[19] G. Lowe. Representing nondeterminism and probabilistic behaviour in reactive processes. Technical Report PRG–TR–11–93, Oxford University Computing Laboratory, 1993.

[20] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, MIT, 1995.

[21] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In B. Jonsson and J. Parrow, editors, *Proc. CONCUR'94*, pages 481–495. LNCS 836, 1994.

[22] K. Seidel. Probabilistic communicating processes. *Theoretical Computer Science*, 152:219–249, 1995.

[23] E.P. de Vink and J.J.M.M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proc. ICALP'97*, pages 460–470. LNCS 1256, 1997.

[24] Sue-Hwey Wu, S.A. Smolka, and E.W. Stark. Composition and behaviors of probabilistic I/O automata. *Theoretical Computer Science*, 176:1–38, 1997.