Comparative semantics for a process language with probabilistic choice and non-determinism

Citation for published version (APA):

Document status and date:
Published: 01/01/1998

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.
Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 04. Feb. 2024
Comparative semantics for a process language with probabilistic choice and non-determinism

J.I. den Hartog

Abstract

In this report a comparative semantics is given for a language $\mathcal{L}_p$ containing probabilistic and non-deterministic choice. The effects of interpreting these operators as local or global are investigated. For three of the possible combinations an operational model and a denotational model are given and compared. First models for local probabilistic choice and local non-deterministic choice are given using a generative approach. By adjusting these models slightly models for global probability and local non-determinism are obtained. Finally models for local probability and global non-determinism are presented using a stratified approach. For use with the denotational models a construction of a complete ultra-metric space of finite multisets is given.

1 Introduction

The goal of this paper is to construct comparative semantics for a language combining non-determinism and probabilistic choice. The main interest is the interplay between these two concepts. Since many of the interesting properties of probabilistic systems are properties that will hold eventually, or will hold on average, one wants to be able to describe infinite behaviors of processes. The use of metric semantics gives a convenient way to work with infinite objects. Fixed point and distance arguments can be used, which do not rely on finiteness. Also of interest are infinite sequences of probabilistic choices. To adequately reason about such sequences as a whole, distributions are used.

The modeling of probability has been the subject of various papers. The usual approach, when dealing with probability, is to replace non-deterministic choice by probabilistic choice. In [17], [28] and [3] this approach is followed. However, interpreting all choices as probabilistic choices does not seem to be appropriate, especially when there is also parallel composition. When replacing non-deterministic choice with probabilistic choice, parallel composition will either become probabilistic, or needs to be restricted.

A more general approach is to view probability and non-determinism as two distinct phenomena, which could be modeled separately. This approach is used here. Both non-deterministic choice and probabilistic choice are included in the same language. The term non-determinism is used exclusively for the non-determinism caused by the first (non-probabilistic) kind of choice. The term probability is used for the different options created by the second, probabilistic, kind of choice.
Two different interpretations of choice, local and global, are investigated. Local and global non-deterministic choice have been studied extensively. For instance in [5], [19] and more recently in [2] where local choice is called static choice and global choice is called dynamic choice. For probabilistic choice, the interpretation as local or global has not been investigated much.

The difference between the two interpretations of choice is the influence of the environment. The environment is formed by the statements in parallel with the current statement. Local choice is made independent of the actions of the environment. Global choice may be influenced by the environment. Global choice cooperates with the environment and will avoid (direct) deadlock when possible. To avoid deadlock it is necessary to know what communication options the environment offers. Global choice delays its choice until the options of the environment are available.

Both non-deterministic choice and probabilistic choice can be interpreted as local or as global choice, leading to four different interpretations and four different comparative semantics. The comparative semantics consists of giving an operational model, a denotational model and relating both models. For the semantics modeling metric domains are used. An overview of and introduction in metric semantics can be found in [6]. For the denotational model for global interpretation of probability a branching domain for probabilistic processes is required. For this purpose the functor $M_f$ is introduced that, for a space $S_f$, gives the space $M_f(S)$ of finite multisets over $S$. For the operational models a linear domain for probabilistic processes is needed. For this purpose the functor $\mu$ is introduced. For a space $S$, $\mu(S)$ is the space of all probability distributions over $S$. See section 4 for details.

The contributions of this paper consists of the construction of a metric domain of multisets, through the functor $M_f$, especially suitable for the branching description of finite discrete probabilistic choices. The functor $M_f$ is new. Using domains constructed with this functor, denotational semantics are given.

As a basis for the operational semantics, transition system specifications are introduced which describe finite probabilistic choice without needing distributions. These transition system specifications rely on implicit counting of multiple transitions, resulting in intuitive rules for handling probabilistic choice. This way of dealing with probabilistic choice in transition systems is also new.

The structure of this report is as follows. After the mathematical preliminaries in section 2, the language $L_p$ is defined in section 3. In section 4 the domain constructors used in this paper are described. Sections 5 and 6 give models for the local interpretation of non-determinism. Section 7 gives a model for global non-deterministic choice with local probabilistic choice. Finding a model for the combination of both global non-deterministic choice and global probabilistic choice is currently an open question. Some issues concerning this question are mentioned in section 8. The following table illustrates the different interpretations and their place in this report.
Closely related is the work done in [27] where the combination of non-determinism and probability is also investigated. However, in this work the non-determinism is still seen as something that can be refined to probability, only the exact chances are not known. It seems that the generative transition systems used in sections 5 and 6 have at least the same expressibility as the probabilistic automata used in [27]. In [27] these probabilistic automata are the starting point of the discussion whereas in this report, constructing the transition system for a statement in the language is also an important step.

In [17] reactive, generative and stratified models are given for a calculus PCCS. In this calculus, based on Milner’s SCCS [24] the non-deterministic choice has been replaced by probabilistic choice and the parallel composition is a synchronous product. The different semantics are given using SOS and bisimulation. In [22] bisimulation for reactive probabilistic processes has been studied in a testing setting. A testing algorithm which can distinguish non-bisimilar processes with probability $1 - \epsilon$ (for any given $\epsilon > 0$) is given.

In [3] a calculus PrACP, derived fromACP (from [8]) is introduced. In the calculus PrACP the operator for choice `+' is replaced by a probabilistic version `+$p'. The parallel composition `||' is also made probabilistic by replacing it with an operator `||t,a'.

In [28] a calculus PCSP is introduced. The calculus PCSP contains a probabilistic choice `$p\cap$' which replaces the non-deterministic choice of CSP. In PCSP there is also a restricted form of external choice and there are two kinds of parallel composition. The first kind of parallel composition is a lockstep synchronized composition $P||Q$, in which $P$ and $Q$ have to make the same transition at the same time in each step. The second, less restricted, kind of parallel composition is $P_A||B_Q$ in which $P, Q$ synchronize on actions from $A \cap B$, $P$ does independent action from $A \setminus B$ and $Q$ does independent transition from $B \setminus A$. The external choice is restricted to a choice between processes with disjoint sets of possible first steps. For instance an (external) choice between $a$ followed by $P_1$ and $a$ followed by $P_2$ is not allowed. A first step analysis is done in [28] to find which statements are allowed.

In [21] a reactive model is given for a language containing probabilistic choice and non-determinism. In this model the probabilistic choice is guarded by an action. The behavior of a process after performing an action $a$ is a probabilistic choice between several alternatives. A limitation to the non-deterministic choice is that there must be disjunct starting actions and the parallel composition must synchronize in every step. Especially the limitation on the parallel composition seems strong.
2 Mathematical preliminaries

In this section some notation and lemmas that are needed in the sequel are given. The notions of a compact set, a complete metric space and an ultra-metric space are assumed known (See e.g. [6]).

Definition 2.1 Let \((M_1, d_1)\) and \((M_2, d_2)\) be metric spaces.

(a) A function \(f : M_1 \rightarrow M_2\) is called non-expansive if \(d_2(f(x), f(y)) \leq d_1(x, y)\) for all \(x, y \in M\).

(b) A function \(f : M_1 \rightarrow M_2\) is called contractive if there exists an \(\alpha < 1\) such that \(d_2(f(x), f(y)) \leq \alpha \cdot d_1(x, y)\) for all \(x, y \in M\).

(c) The space of all non-expansive functions from \(M_1\) to \(M_2\) is denoted by \(M_1 \rightarrow M_2\). The metric defined on \(M_1 \rightarrow M_2\) is

\[
d(f, g) = \sup \{d_2(f(m), g(m)) \mid m \in M\}.
\]

(d) Let \(B_\varepsilon(m) = \{m' \in M \mid d(m, m') < \varepsilon\}\). The family of open sets \(O_\varepsilon\) is defined by:

\[
O_\varepsilon = \{O \text{ open} \mid m \in O \Rightarrow B_\varepsilon(m) \subseteq O\}.
\]

Note that in an ultra-metric space \(M\), \(B_\varepsilon(m) \subseteq O_\varepsilon\) for all \(m \in M\).

Often a metric space \((M, d)\) is denoted simply by \(M\). The metric \(d\) is assumed to be clear from the context. The contractive functions used are usually \(\frac{1}{2}\)-contractive, i.e. the \(\alpha\) in the definition above is \(\frac{1}{2}\).

For a set \(A\) the metric space \(A^\infty\) consists of all (finite or infinite) sequences of elements of \(A\) with the following distance

\[
d(w_1, w_2) = \inf \{2^n \mid w_1[n] = w_2[n]\}.
\]

\([w_1]\) is the prefix of \(w\) of length \(n\). The metric space \(A^*\) is the subspace of \(A^\infty\) of all finite sequences.

Banach’s theorem provides an important tool for metric semantics. It states that any contractive function on a complete ultra-metric space has a unique fixed point, a fact that will be used frequently in definitions and proofs below.

Theorem 2.2 (Banach) Let \((M, d)\) be a complete metric space. If \(f : M \rightarrow M\) contractive, then there exists a unique point \(x \in M\) for which \(f(x) = x\). This \(x\) is denoted by \(\text{fix}(f)\). If \(y\) is an arbitrary point of \(M\) then \(\lim_{n \to \infty} f^n(y) = \text{fix}(f)\) (where \(f^n\) is the \(n\)-th iteration of \(f\)).
2.1 Transition systems

A transition system $T = (\text{Conf}, \text{Obs}, \rightarrow)$ can be used to define operational semantics. The transition system represents a virtual machine that can take steps from one configuration to another. Each step produces observable output $(a \in \text{Obs})$. The transition system for each statement in the language will be defined in the style of SOS [25] using a transition system specification. With some abuse of language the transition system specification $T$ will be referred to as the transition system. The notation $(x \in X)$ will be used for the set $X$ with a typical element $x$. Similarly $x(\in X)$ is used for the element $x$ which is a typical element of the set $X$.

**Definition 2.3** A transition system (or more precisely a transition system specification) $T$ is a four-tuple $T = (\text{Conf}, \text{Obs}, \rightarrow, \text{Spec})$ where,

(a) $(c \in \text{Conf}$ is a set of configurations, $(a \in \text{Obs}$ is a set of observations and $\rightarrow$, the transition relation, is a subset of $\text{Conf} \times \text{Obs} \times \text{Conf}$). An element of $\text{Conf} \times \text{Obs} \times \text{Conf}$ is called a transition.

(b) $\text{Spec}$ is a set of $n(\geq 1)$ axioms and $m(\geq 0)$ rules.

An axiom is a construct of the form $c \overset{a}{\rightarrow} c'$. It specifies that all tuples of the form $(c, a, c')$ belong to $\rightarrow$. An axiom can be seen as a rule with no premise.

A rule is a construct:

$$
\frac{c_1 \overset{a_1}{\rightarrow} c_1' \ldots c_k \overset{a_k}{\rightarrow} c_k'}{c \overset{a}{\rightarrow} c'}
$$

It specifies that if $(c_i, a_i, c_i') \in \rightarrow \ (i = 1, \ldots, k)$ then also $(c, a, c') \in \rightarrow$.

(c) A proof tree in a transition system is a finite tree for which every node is the conclusion of a rule in $\text{Spec}$ and its sub-nodes are the premises of this rule. The leaves of the tree are axioms. The transition at the root of a proof tree is said to be derived from the specification (by the proof tree).

$$
\frac{c \overset{\xi}{\rightarrow} E}{c; a \overset{\xi}{\rightarrow} a \quad \xi \overset{\xi}{\rightarrow} E}
$$

$$
\frac{c; a \parallel \xi \overset{\xi}{\rightarrow} a}{(c; a \parallel \xi) \overset{b}{\rightarrow} a}
$$

A proof tree in $T_{ps}$ (section 7)

(d) $\rightarrow$ is the subset of $\text{Conf} \times \text{Obs} \times \text{Conf}$ consisting exactly the transitions that can be derived from the specification, i.e. for which there is a proof tree in $\text{Spec}$.

For a transition there may be several different proof trees. Often the number of proof trees is not relevant, but for probabilistic statements the number of
proof trees will turn out to be important. The notation \( c \xrightarrow{a_1} \ldots \xrightarrow{a_n} c' \) (for \( n > 0 \)) will denote that there are \( n \) distinct proof trees for the transition \( c \xrightarrow{a} c' \).

The shorthand

\[
\frac{c_1 \xrightarrow{a_1} c'_1 \ldots c_k \xrightarrow{a_k} c'_k}{\bar{c}_1 \xrightarrow{a} \bar{c}_1'}
\]

\[
\ldots
\]

\[
\bar{c}_n \xrightarrow{a} \bar{c}_n'
\]

is used for the \( n \) rules

\[
\frac{c_1 \xrightarrow{a_1} c'_1 \ldots c_k \xrightarrow{a_k} c'_k}{\bar{c}_i \xrightarrow{a} \bar{c}_i'} \quad (i = 1, \ldots, n).
\]

### 2.2 Generative and stratified models

The models used are of two types. The terminology is due to [17]. The first type of model will be called generative. Generative models assign probability to actions such that the sum of probabilities for all actions is one or less. To find the probability of an action successive choices are combined into one. In the presence of non-determinism this means that the non-determinism has to be combined, somehow, with the probabilistic choice. This is achieved by resolving non-determinism before probability.

The second kind of model will be called stratified. Stratified models assign probability to statements. This means that a probability may be assigned to a statement which is itself probabilistic or non-deterministic. Successive choices are not combined. A statement to which a probability is assigned may be non-deterministic. As the probability and non-determinism do not have to be combined, the non-determinism need not be resolved before the probability.

The domain used here is not a purely stratified domain. It will not be necessary to completely stratify the probability. A non-deterministic choice is not combined with a following probabilistic choice, but successive probabilistic choices can still be combined. (This is not the case in [17] which deals, among others, with restriction.) Reactive models as introduced in [17] are not used.

### 3 The syntax of the language \( \mathcal{L}_p \)

The language \( \mathcal{L}_p \) is based on a set of atomic actions \( \text{Act} \) which will be ranged over by \( a \). The atomic actions are divided into a set of internal actions \( \text{IAct} \) ranged over by \( b \) and a set of synchronization actions \( \text{Sync} \) ranged over by \( c, \bar{c} \). For each synchronization action \( c \in \text{Sync} \) there is a (unique) complementary action \( \bar{c} \) in \( \text{Sync} \) with which \( c \) can synchronize. The complementary action for \( \bar{c} \) is \( c \), i.e., \( \bar{\bar{c}} = c \). The synchronization of the actions \( c \) and \( \bar{c} \) results in a special internal action \( \tau \). Other than this the atomic actions are left without an interpretation. The meaning of an action \( b \) will simply be \( b \) itself.

The atomic actions are supplemented with the probability that this action occurs. This results in the set \( \text{PA} \text{ct} \) of pairs in \([0,1] \times \text{Act} \) and \( \text{PI} \text{Act} \) of pairs in
The variable $\alpha$ ranges over $PAct$, $\beta$ ranges over $PIAct$ and $\rho, \sigma, \kappa$ range over $[0,1]$. The symbol $\delta$ is used to denote deadlock. Furthermore the following notation is used: $PIAct_\delta = PIAct + [0,1] \times \{\delta\}$, $IAct^\infty_\delta = IAct^\infty + IAct^\ast \cdot \{\delta\}$.

To describe recursion a set $PVar$ of procedure variables is used. The variable $x$ ranges over $PVar$.

\textbf{Definition 3.1} The syntax of the language $\mathcal{L}_p$ is given by

\begin{itemize}
  \item[(a)] $s (\in Stat) ::= a \mid x \mid (s;s) \mid (s+s) \mid (s||s) \mid (s +_p s)$
  \item[(b)] $g (\in GStat) ::= a \mid (g;s) \mid (g + g) \mid (g||g) \mid (g +_p g)$
  \item[(c)] $(D \in Decl = PVar \rightarrow GStat )$
  \item[(d)] $(\pi \in L_p = Decl \times Stat )$
\end{itemize}

The basic ingredients for the language are the actions $a (\in Act)$ and procedure variables $x (\in PVar)$. A declaration $D (\in Decl)$ gives the body $D(x)$ for a procedure $x$. The body must consist of a guarded statement, i.e. recursion is restricted to guarded recursion.

In the sequel one fixed declaration $D$ will be assumed and $D$ will be dropped from the notation, i.e. $s \in \mathcal{L}_p$ will be written instead of $(D, s) \in \mathcal{L}_p$.

The syntactical operators used to build statements from this basis are explained below.

The statement $s_1; s_2$ denotes sequential composition. It behaves like $s_1$ until $s_1$ terminates and then it continues by behaving like $s_2$.

The statement $s_1 + s_2$ denotes a non-deterministic choice. The non-determinism can be viewed as being local or global. If non-determinism is local it does not depend on the environment. The environment cannot force one alternative to occur. Using this view the statement $s = s_1 + s_2$ describes that $s$ can behave like $s_1$ or it can behave like $s_2$. This means that the statement can be implemented in two ways either by $s_1$ or by $s_2$ (i.e. $s$ has two possible behaviors). Since the choices do not depend on the environment they can be made beforehand (illustrating the link with static choice from [2]).

Global non-determinism takes the environment into account. If one of the alternatives starts with communication with the environment then this alternative will only be chosen if the environment is willing to communicate. This makes it possible for the environment to determine which alternative is chosen. Using this view a statement $s = s_1 + s_2$ denotes that $s$ can behave like $s_1$ and it can behave like $s_2$. This means that in an implementation of $s$ both alternatives have to be present. The choice cannot be made beforehand (by the implementer) but can only be made when the actions of the environment are present, since the environment might force one of the alternatives. The choice is made only when the first action of one of the alternatives actually executes (illustrating the link with dynamic choice from [2]).
The statement $s_1 \parallel s_2$ denotes parallel composition and also causes non-determinism. If the non-determinism is local, a failed synchronization attempt (from either $s_1$ or $s_2$) will result in deadlock. A failed synchronization attempt is an action $c$ from $s_1$ without a matching action $\bar{c}$ from $s_2$. If the non-determinism is global then synchronization is only initiated if it will succeed. If synchronization is not possible, the statement that wants to synchronize will wait until it is possible or it chooses another alternative. The only possibility of deadlock in this case is if all statements are waiting for a synchronization that cannot take place.

The statement $s_1 +_p s_2$ denotes probabilistic choice. The argument $\rho$ is assumed to be strictly between 0 and 1. For probability the effects of being local or global are slightly different. When the probability is local, a statement like $s_1 +_p s_2$ will behave like an unconditional, independent probabilistic choice between behaving like $s_1$ (with probability $\rho$) and behaving like $s_2$ (with probability $1 - \rho$). This means that once a coin has been flipped the decision is final. The probability of behaving like $s_1$ is $\rho$, independent of $s_1$, $s_2$ and the environment.

When the probability is interpreted as being global the statement $s_1 +_p s_2$ will behave like a conditional, dependent probabilistic choice between behaving like $s_1$ and behaving like $s_2$. If one of the statements, say $s_1$, starts with a synchronization that fails (i.e. $s_1$ deadlocks) then it is allowed to backtrack over the coin flip. The decision becomes final only when the first action has happened. The result is that that $s_1 +_p s_2$ behaves like $s_1$ if $s_2$ deadlocks, like $s_2$ if $s_1$ deadlocks and like a probabilistic choice between $s_1$ and $s_2$ only if neither $s_1$ nor $s_2$ deadlock. The probability of behaving like $s_1$ depends on $s_1$, $s_2$ and the environment.

In the transition system for the generative model (i.e. the generative transition system) the auxiliary operators '$\parallel$' and '$|$' are used. For this transition system the extended language $L_p^+$ is used. The language $L_p^+$ is obtained from $L_p$ by adding the auxiliary operators to $\text{Stat}$:

$$s \in \text{Stat} : := \ldots \mid (s \parallel s) \mid (s|s).$$

The statement $s_1 \parallel s_2$ denotes parallel composition in which the first action has to be done by $s_1$. (Synchronizing with $s_2$ is not allowed.) The statement $s_1 | s_2$ denotes parallel composition in which the first action is synchronization between $s_1$ and $s_2$.

The configurations of the transition system consist of a resumption with a declaration, i.e. $\text{Conf} = \text{Decl} \times \text{Res}$. As with programs the declarations are dropped from the notation. A resumption is either a statement $s$ or a special symbol $E$ denoting a finished computation. For the generative transition system the statement will be from $L_p^+$, for the stratified transition system the statement will be from $L_p$:

$$r \in \text{Res} = s \mid E.$$
The set of observations will be different for the generative and stratified transition systems. For each transition system the set of observations will be given separately.

Structural induction is not sufficient for the proofs in the following sections. The rule handling recursion will have a premise that is syntactically more complex than the conclusion. This rule is allowed because the limitation to guarded recursion prevents problems. The proofs can be given by using induction on the weight function introduced below. (Induction on a weight function was introduced in [20] and the systematic use is due to Van Breugel [12].)

**Definition 3.2** \( \text{wgt: } \text{Res} \to \mathbb{N} \) is given by

\[
\begin{align*}
\text{wgt}(E) & = 0 \\
\text{wgt}(a) & = 1 \\
\text{wgt}(x) & = \text{wgt}(D(x)) + 1 \\
\text{wgt}(s_1 \cdot s_2) & = \text{wgt}(s_1) + 1, \text{ and similarly for } '||' \\
\text{wgt}(s_1 || s_2) & = 1 + \text{wgt}(s_1 \cdot s_2) + \text{wgt}(s_2 \cdot s_1) + \text{wgt}(s_1 || s_2) \\
\text{wgt}(s_1 + s_2) & = 1 + \text{wgt}(s_1 \cdot s_2) \\
\text{wgt}(s_1 +_p s_2) & = 1 + \text{wgt}(s_1) + \text{wgt}(s_2), \text{ and similarly for } '+_p' \text{ and } '|' .
\end{align*}
\]

That the weight function \( \text{wgt} \) is well defined is easy to see by structural induction, first on guarded statements and then on all resumptions. The restriction of the function \( \text{wgt} \) to stratified resumptions (without the additional operators '||' and '|') and to \( \mathcal{L}_p \) will also be called \( \text{wgt} \).

For the language \( \mathcal{L}_p \) three operational and three denotational models will be introduced in sections 5, 6 and 7. First the domains that are used to define these models are introduced in section 4.

### 4 Domain equations

This section describes the use of domain equations in defining domains and introduces some functors. These functors are used for specifying domains in subsections 5.3 and 7.3.

In sections 5, 6 and 7 several different domains are needed for the semantical models. The domains are defined using domain equations. A domain equation is an equation of the form \( M = \mathcal{F}(N) \), where \( \mathcal{F} \) is a functor, \( N \) a (metric) domain and \( M \) is the domain defined by the equation. Recursive domain equations are solved up to isomorphism, yielding instead the domain equation (or more accurately a domain isometry) \( M \simeq \mathcal{F}(M) \). The method of solving recursive domain equations over metric spaces comes from [7], [1] and [26]. First some basic notions is introduced and then the functors used are given. More on domain equations can be found in e.g. [6].

**Definition 4.1** Let \( \text{CUMS} \) denote the category of all complete ultra-metric spaces with non-expansive functions as morphisms.
(a) A functor \( F : \text{CUMS} \rightarrow \text{CUMS} \) is called locally non-expansive if for all
\( X,Y \in \text{CUMS} \) the mapping \( (X \xrightarrow{1} Y) \rightarrow (F(X) \xrightarrow{1} F(Y)) \) such that
\( f \mapsto F(f) \) is a non-expansive mapping.

(b) A functor \( F : \text{CUMS} \rightarrow \text{CUMS} \) is called locally contractive if for all \( X,Y \in \text{CUMS} \) the mapping \( (X \xrightarrow{1} Y) \rightarrow (F(X) \xrightarrow{1} F(Y)) \) such that \( f \mapsto F(f) \) is a
contractive mapping.

The theorem below is a special case of a result which can be found in [26].

**Theorem 4.2** A locally contractive functor \( F : \text{CUMS} \rightarrow \text{CUMS} \) has a unique
fixed point.

Several combinations of locally contractive functors with locally non-expansive functors result in a locally contractive functor. Due to this most functors used need only be locally non-expansive.

The following standard functors are used: constant, \( \alpha \)-identity \((id_\alpha)\), disjoint union \((+)^\dagger\), product \((\times)^\dagger\) and nonempty compact power domain \((\mathcal{P}_{nc}\cdot))\). The constant functor and the \( \alpha \)-identity functor are locally contractive. The rest of these functors are locally non-expansive. More on these functors can be found in e.g. [6]. In addition the following functors are introduced here,

1. finite multisets \((\mathcal{M}_f)\),
2. distributions \((\mu)\).

These two functors are described in more detail below.

The multiset functor is combined with the powerset functor when defining the domains. For linear domains the multiset functor is replaced by the distribution functor. By alternating the order of the use of the multiset functor and the powerset functor different domains are obtained. By applying the powerset functor to the result of applying the multiset functor a generative domain is obtained. By reversing this order a stratified domain is obtained. This is worked out in subsections 5.3 and 7.3.

**4.1 Multisets, the functor \( \mathcal{M}_f \)**

The different options created by non-determinism are described by sets of alternatives. Having the same non-deterministic option twice is the same as having it once. For a probabilistic option this is not true. If \( a \) can happen with probability \( \rho \) one way and \( a \) can happen with probability \( \rho \) another way, the total probability of \( a \) happening is not \( \rho \) but \( 2\rho \). This means that multiple occurrences cannot be identified (as is done with non-determinism). Instead of using sets, multisets are used to describe the probability. Finite multisets suffice since the probabilistic choice is finite.

For a space \( S \) the finite multiset functor gives the space of all finite multisets over \( S \). The way the multisets are coded is derived from the codings used in
pomsets and event-structures, see for example [18], [15] and [23]. The work here
has been inspired by [31], [14] in which a metric approach to pomsets is followed.
A countably infinite set $\chi$, called the base set, is chosen. A natural example
is $\mathbb{N}$. In event-structures and pomsets elements of the base set $\chi$ are called
the events or nodes. A partial labeling of $\chi$ with elements of $S$ can describe a
multiset over $S$. Since several labelings describe the same multiset, a multiset
is defined as an equivalence class of labelings.

**Definition 4.3** The space of multisets over $S$ is given by

$$
\mathcal{M}_{f}(S) = \mathbb{L}(S)/\sim \quad \mathbb{L}(S) = \chi \rightarrow (S + \{\ast\}), \text{ with finite support}
$$

The symbol $\ast$ is used to indicate undefinedness. The support of a function is the
part of the base set on which the function is defined, i.e. not equal to $\ast$. Two
labelings $L_1, L_2 : \chi \rightarrow (S + \{\ast\})$ are related ($L_1 \sim L_2$) if there exists a bijection
$\Phi : \chi \rightarrow \chi$ such that $L_1 \circ \Phi = L_2$. A class of labelings with representative $L$ is
denoted by $\mathbb{L}$.

**Example 4.4** The multiset containing the element $a$ twice and the element $b$
once can be represented by the following functions, using $\mathbb{N}$ as the base set. The
function $f$ is defined on $1, 2$ and $3$ only, $g$ is defined on $2, 4$ and $5$ only.

$$
\begin{align*}
  f(1) &= a & g(4) &= a & f(x) : a & b & a & \ast & \ast & \ast \\
  f(2) &= b & g(5) &= a & x : 1 & 2 & 3 & 4 & 5 & \ldots \\
  f(3) &= a & g(2) &= b & g(x) : \ast & b & \ast & a & a & \ast
\end{align*}
$$

If $S$ is a metric space equipped with metric $d$ then $\mathcal{M}_{f}(S)$ is equipped with
the following metric.

**Definition 4.5** The distance of two multisets $\mathbb{L}_1, \mathbb{L}_2$ is given by

$$
\begin{align*}
  d(\mathbb{L}_1, \mathbb{L}_2) &= \min \{d_L(L, L') \mid L \in \mathbb{L}_1, L' \in \mathbb{L}_2\} \\
  &= \min \{d_L(L_1 \circ \Phi, L_2) \mid \Phi : \chi \rightarrow \chi \text{ is a bijection}\}.
\end{align*}
$$

(This definition is illustrated in example 4.9.)

The distance between $\ast$ and $\ast$ in $S + \{\ast\}$ is one by definition. By taking
a bijection $\Phi$ that takes the support of $L_2$ to outside the support of $L_1$, the
distance $d(L_1 \circ \Phi, L_2)$ is one. This shows that the distance $d$ on $\mathcal{M}_{f}(S)$ is
bounded by one (even if the distance on $S$ is not). The only bijection $\Phi$ that
can give $d_L(L_1 \circ \Phi, L_2) < 1$, is one that takes the support of $L_2$ to the support
of $L_1$. The behavior of $\Phi$ on the rest of $\chi$ is irrelevant to the distance. There
are only finitely many different ways to take the support of $L_2$ to the support
of $L_1$ since both sets are finite. This means that the $\min \{\ldots\}$ in the definition
does indeed exist; the distance on $\mathcal{M}_{f}(S)$ is a well-defined function. That $d$ is
indeed a metric is straightforward to show.

**Lemma 4.6** If $S$ is a complete ultra-metric space then $\mathcal{M}_{f}(S)$ is also a com-
plete ultra-metric space.
**Proof** That \((M_f(S),d)\) is an ultra-metric space is easy to see.

To find the distance between two multisets one can fix one of the representations and only vary the other. For all \(L_1\) it holds that
\[
d(L_1,L_2) = \min\{d_L(L_1,L') | L' \in \mathcal{L}_2\},
\]
Also, unlike with sets, two multisets which are close together have the same number of elements. This means that a Cauchy sequence of multisets will (from a certain point onwards) consist of multisets with the same number of elements. Keeping these two things in mind, completeness is straightforward to prove. □

By defining \(M_f\) on non-expansive functions (the arrows of the category CUMS) \(M_f\) becomes an endo-functor on CUMS.

**Definition 4.7** Let \(A,B \in \text{CUMS}\) and \(f : A \rightarrow B\) then \(M_f(f) : M_f(A) \rightarrow M_f(B)\) is defined by
\[
\mathcal{M}_f(f) (L) = f^* \circ L
\]
where \(f^*(m) = \begin{cases} f(m) & \text{if } m \in A \\ * & \text{if } m = * \end{cases}\).

For well-definedness it needs to be shown that the definition does not depend on the choice of the representative \(L\) and that \(M_f(f)\) is a non-expansive function. Both are again straightforward. The following lemma prepares for the use of the functor \(M_f\) in domain equations.

**Lemma 4.8** The functor \(M_f\) is locally non-expansive.

**Proof** Let \(X,Y \in \text{CUMS}, f,g : X \rightarrow Y\) and \(M \in \mathcal{M}_f(A)\). Choose a labeling \(L\) from the class \(\mathcal{M}\) then:
\[
d(\mathcal{M}_f(f)(M),\mathcal{M}_f(g)(M)) = d(f^* \circ L,g^* \circ L) \leq d(f^*,g^*) = d(f,g)
\]
Since this equality holds for every \(M \in \mathcal{M}_f(A)\) it follows that \(d(\mathcal{M}_f(f),\mathcal{M}_f(g)) \leq d(f,g)\) □

In the remainder the coding of the multisets is suppressed. Multisets are seen as sets that may contain an element more than once. The following is used to denote a finite multiset over \(S\):
\[
\{ [s_1, s_2, \ldots, s_k] \}
\]
for \(s_i \in S\). An element \(s_i\) may occur more than once. For example: \([a,a]\). As an abbreviation a multiplicity may be assigned to an element. \([3\cdot s]\) is short for \([s,s,s]\). The union of two multisets is denoted by \(\sqcup\).
\[
\{ [s_1, s_2, \ldots, s_k] \} \sqcup \{ [r_1, \ldots, r_l] \} = \{ [s_1, s_2, \ldots, s_k, r_1, \ldots, r_l] \}
\]
For example:

\[ \{a,b,a\} \cup \{a,b,c\} = \{a,b,a,a,b,c\} = \{3a,2b,1c\} \].

Clearly multiset union is a non-expansive operation.

**Example 4.9** The following represent multisets over \(a, b^\infty:\)

\[
\begin{align*}
M_1 &= \{ab, ab, b\} \\
M_2 &= \{ab, abb, b\}
\end{align*}
\]

The distance between these multisets is \(\frac{1}{4}\). This can be seen by ‘pairing’ the elements as follows: The first \(ab\) from \(M_1\) is compared with \(ab\) from \(M_2\), the second \(ab\) from \(M_1\) is compared with \(abb\) from \(M_2\) and \(b\) from \(M_1\) is compared with \(b\) from \(M_2\). The maximum distance that is obtained in this way is \(\frac{1}{4}\).

The formal approach is to choose a representation of \(M_1\), say \(f\), below, and find a representation \(g\) of \(M_2\) with minimum distance to \(f\). The two possible choices for \(g\) are the one below and \(g\) below with \(g(1)\) and \(g(2)\) reversed.

\[
\begin{align*}
f(1) &= ab & g(1) &= ab & f(x): & ab & ab & b & * \\
f(2) &= ab & g(2) &= abb & x: & 1 & 2 & 3 & \ldots \\
f(3) &= b & g(3) &= b & g(x): & ab & abb & b & *
\end{align*}
\]

The distance between the functions \(f\) and \(g\) is \(\frac{1}{4}\).

Multisets over \(A + A \times X\) are used (for several different spaces \(A\) and \(X\)). The following short-hand notation is used for these multisets.

**Definition 4.10**

\[
[a, \{a, x\}] = \{a_i \mid i \in I\} \cup \{\{a_j, x_j\} \mid j \in J\}
\]

In this notation \(I\) and \(J\) are assumed to be disjoint index sets, \(a_I\) is a function from \(I\) to \(A\) and \(a_I(i)\) is written as \(a_i\) (i.e. \(a_I\) is the sequence \((a_i)\, i \in I\)). \(\{b, x\}\rangle_J\) is a function from \(J\) to \(B \times X\). \(\langle b, x\rangle_J\rangle(j)\) is written as \(\langle b, x\rangle_J\).

Clearly there are many different notations \([a, \{a, x\}]\) that can be used to represent the same multiset; however, all of these will give the same result when used (definitions are independent of representation). This aspect can and will be ignored.

A primed version \([a', \{a, x\}]\) with obvious definition is also used. The index sets \((I, J, I'\text{ and } J')\) are assumed to be pairwise disjoint. Functions are lifted to sequences by applying them to each element, i.e. \(F(x_i) = (F(x_i))_{i \in I}\) for any index set \(I\). Using this convention, saying \(y_i = F(x_i)\) is short for saying \(\forall i \in I: y_i = F(x_i)\).

**Example 4.11** Let \(I = \{1, 2, 3\}, J = \{4, 5\}, a_1 = a, a_2 = a, a_3 = b, a_4 = a, x_4 = x, a_5 = c, x_5 = y\) then

\[
[a, \{a, x\}] = \{a, a, b, \langle a, x\rangle, \langle c, y\rangle\}.
\]
If a multiset can contain the special symbol \( \delta \) labeled with a probability \( \rho \) the similar notation \([a, (a, x)], \kappa\) is used. \([a, (a, x)], \kappa\) is a multiset over \(A_i + A \times X\) (which contains the element \(\delta\) at most once).

\[
[a, (a, x)], \kappa = \{a; i \in I\} \cup \{\{a_j, x_j\}; j \in J\} \cup \{\kappa \delta; \kappa > 0\}
\]

The same conventions as for \([a, (a, x)]\) are used. Additionally \(0 \leq \kappa \leq 1\). The situation \(\kappa = 0\) indicates that \(\delta\) is not in the multiset.

The metric on multisets is significantly different from the (Hausdorff) distance on sets. Two sets which are close in the Hausdorff sense can still have distance 1 if interpreted as multisets (with each element once).

**Example 4.12** Take for \(S\) the space \(\{a, b\}^{\infty}\) consisting of all (possibly infinite) sequences of \(a\)’s and \(b\)’s, then the following holds:

\[
d([a, aa, aab], [a, aa]) = \frac{1}{4} \quad \text{in } P_f(S)
\]

\[
d([a, aa, aab], [a, aa]) = 1 \quad \text{in } M_f(S).
\]

The second distance is 1 since the first multiset contains more elements than the second. No matter the representation, one element of the first multiset will always be compared with \(*\).

As an aside, the following relation exists: If two multisets are close then so are the sets obtained by forgetting multiplicity. Also if two finite sets are close then there are two multisets which contain the same elements as the sets (but possibly more than once) that are also close. In the example one could take the second multiset to be \(([a, aa, aa]).

Caution has to be observed when defining functions that return multisets. If such a function \(f\) has to be contractive in an argument \(x\), care has to be taken that the number of elements in \(f(x)\) is independent of \(x\). The following form of definition, not unusual when working with sets, does not result in a contractive function \(f\) when using multisets.

**Example 4.13** The function \(f : M_f(S) \to M_f(S)\) (with \(S\) again \(\{a, b\}^{\infty}\)) given by

\[
f(M) = \{a \cdot x; x \in M\}
\]

is not contractive, since

\[
d(f([a, b]), f([a, b])) = d([aa, ab], [aa]) = 1 = d([a, b], [a, b])
\]

Definitions of the form given in the example are typically used in linear domains. To describe probability in a linear domain, distributions are employed. Multisets are only used to describe probabilistic choices in a branching fashion.
4.2 Distributions, the functor \( \mu \)

In this subsection the functor \( \mu \) is defined. The functor \( \mu(\cdot) \) yields, given \( S \), the space of probability distributions over \( S \). Distributions are used to describe probabilistic choices in a linear domain. Finite multisets turn out to be insufficient for this purpose. An infinite sequence of probabilistic choices cannot be described in a linear fashion by a finite multiset. An example is given in subsection 5.3 were the functor \( \mu \) is used in specifying a linear domain.

The functor \( \mu \) is that same as the functor \( M_{1}^{*} \) introduced in \([30]\) restricted to \( CUMS \). A more detailed description and more general results can be found there. Also in \([4]\) a functor yielding distributions (called evaluations in \([4]\)) is given on the category \( CUMS \). A comparison with set-theoretic and complete partial order approaches can be found there.

**Definition 4.14** Let \( X \) be a metric space.

(a) A \( \sigma \)-algebra \( A \) over \( X \) is a collection of subsets which is closed under complement and countable union.

(b) The collection \( B(X) \) of Borel sets over \( X \) is the least \( \sigma \)-algebra containing all open sets.

(c) A (Borel) probability measure on \( X \) is a function \( \mu : B(X) \to [0,1] \) such that \( \mu(X) = 1 \) and \( \mu \) is \( \sigma \)-additive, i.e. \( \mu(\bigcup_{i \in I} A_{i}) = \sum_{i \in I} \mu(A_{i}) \) for countable number of disjoint sets \( A_{i} \).

(d) A probability measure \( \mu \) is said to have compact support if \( \mu \) vanishes outside some compact set \( K \), i.e. \( \mu(X \setminus K) = 0 \) thus \( \mu(K) = 1 \).

(e) \( \mu(X) \) denotes the space of all probability measures with compact support on \( X \).

If the space \( X \) is a metric space, a metric can also be given on \( \mu(X) \).

**Lemma 4.15** Let \( (M,d) \) be a complete ultra-metric space and \( \mu, \nu \in \mu(M) \). Define the distance \( d \) on \( \mu(M) \) by

\[
d(\mu, \nu) = \inf \{ \varepsilon > 0 \mid \forall O \in \mathcal{O} : \mu(O) = \nu(O) \}
\]

then \( (\mu(M), d) \) is a complete ultra-metric space.

The proof of this lemma can be found in the full version of \([30]\). The compactness of the support is needed for completeness of \( \mu(M) \). The distance \( d \) on \( \mu(M) \) satisfies the equation

\[
d(\mu, \nu) = \inf \{ \varepsilon > 0 \mid \forall m \in M : \mu(B_{\varepsilon}(m)) = \nu(B_{\varepsilon}(m)) \}.
\]

This description of the distance is used in the remainder.

The only step still needed to turn \( \mu \) into a functor on \( CUMS \) is to define \( \mu \) on non-expansive functions (the arrows of \( CUMS \)).
Lemma 4.16 Let $M, N$ be complete ultra-metric spaces and $f: M \to N$ a non-expansive function. Define $\mu(f): \mu(M) \to \mu(N)$ by

$$\mu(f) = \lambda \mu. \lambda B. \mu(f^{-1}(B))$$

then $\mu$ is a locally non-expansive functor on CUMS.

The proof of this lemma can again be found in the full version of [30]. When working with distributions over sets of sequences, the following notation is useful. Let $B$ be a subset of $A^\infty$ then the set $B/b = \{ w \mid aw \in B \}$. If $B$ is a Borel set then $B/b$ is also a Borel set.

In subsection 5.3 and 7.3 domains are built using the functors introduced in this section.

5 Local non-determinism and local probability

In this section a generative model for $L_p$ is given. In this model both the non-determinism and the probability are interpreted as being local. The following example is used to show the effect of the interpretation on the interplay between non-determinism and probability. The example is also used in the following sections.

$s = (c_1 + c_2)(c_1 + \rho c_2)$

When both choices are interpreted locally, they will make the decision independent of the environment. Both non-determinism and probability are present. The statement can deadlock. For example if the non-deterministic alternative $c_2$ is chosen then there is a probability of $\rho$ that the statement will deadlock by choosing $c_1$ as the probabilistic option.

A linear operational semantics and a branching denotational semantics are given and compared.

5.1 The transition system

The operational semantics for $L_p$ is defined using a generative transition system. The configurations are as described in section 3. The model used is generative in the sense that a probability is bound to an action (as opposed to being bound to a statement). For some configurations the sum of the probabilities of all transitions may be less than one. The remaining probability is interpreted as the probability of deadlock.

To be able to assign a probability to $a_1$ in the next statement $s$, it is necessary to first resolve the non-determinism. If the first alternative is chosen then the probability of $a_1$ is $\rho$, otherwise it is 0 (assuming distinct $a_1, a_2, a_3, a_4$).

$s = (a_1 + \rho a_2) + (a_3 + \sigma a_4)$

However, after resolving the non-determinism and finding the probabilities of each action it has to be clear which actions belong to which non-deterministic
option. The auxiliary action $\iota$ is used to distinguish between the different non-deterministic alternatives. Each non-deterministic choice is made explicit by the appearance of the $\iota$ action.

The observations in the transition system are either actions with a probability assigned to them or the auxiliary action $\iota$ (without a probability).

**Definition 5.1** The generative transition system is given by $T_p = (\text{Decl} \times \text{Res}, \text{PA} \cup \{\iota\}, \rightarrow, \text{Spec})$. A transition $((D|r), a, (D|r')) \in \rightarrow$ is written as $r \xrightarrow{a} r'$.

The rest of this subsection is devoted to the definition of $\text{Spec}$.

The axiom (Act) expresses that action $a$ can take an $a$ transition with probability one and then be finished. The Rule (Rec) takes care of recursion by body replacement. The rule (Seq) states that $s_1;s_2$ behaves like $s_1$ until $s_1$ is done (the case that $r = E$) and after that behaves like $s_2$ (since $E;s_2 = s_2$).

The $\theta$-transition can be either an auxiliary $\iota$ transition or a probabilistic $\alpha$ transition.

- $a \xrightarrow{1} s \quad \text{(Act)}$
- $s_1 \xrightarrow{\theta} r$
- $x \xrightarrow{\theta} r$ if $D(x) = s_1$
- $s_1 ; s_2 \xrightarrow{\theta} r ; s_2$ \quad \text{(Seq)}

The non-determinism is caused by the $+$ and $\parallel$ operators. As described earlier resolving the non-determinism is done explicitly. This is achieved by the axioms (Choice), (Intro $\parallel$) and (Intro $|$).

- $s_1 + s_2 \xrightarrow{1} s_1$
- $s_1 + s_2 \xrightarrow{1} s_2$ \quad \text{(Choice)}
- $s_1 \parallel s_2 \xrightarrow{1} s_1 \parallel s_2$
- $s_1 \parallel s_2 \xrightarrow{1} s_1 \parallel s_1$
- $s_1 \parallel s_2 \xrightarrow{1} s_1 \parallel s_2$ \quad \text{(Intro-$\parallel$)}
- $s_1 \parallel s_2 \xrightarrow{1} s_1 | s_2$
- $s_1 \parallel s_2 \xrightarrow{1} s_1 \parallel s_2$ \quad \text{(Intro-$|$)}

The statement $s_1 \parallel s_2$ first resolves the non-determinism in $s_1$. After the first step this results in $s \parallel s_2$. The left merge is still maintained since resolving non-determinism is not considered a real step. After all non-determinism in $s_1$ has been resolved $s_1 \parallel s_2$ then takes the same action as $s_1$. Afterwards the execution continues with the parallel composition $\parallel$ of the resulting resumption $r$ and $s_2$.

The configuration $s_1 | s_2$ resolves non-determinism in both $s_1$ and $s_2$ and then tries to synchronize on the first step. Failure to synchronize will result in deadlock. Because of the possibility of deadlock the probability that a statement $s_1|s_2$ takes any step may be less than one.
\[
\begin{align*}
&s_1 \xrightarrow{1} s \\
&s_1 \parallel s_2 \xrightarrow{1} s \parallel s_2 \quad \text{(Leftmerge 1)} \\
&s_1 \parallel s_2 \xrightarrow{\alpha} s_2 \quad \text{(Sync 1)} \\
&s_1 \xrightarrow{\alpha} r \\
&s_1 \parallel s_2 \xrightarrow{\alpha} r \parallel s_2 \quad \text{(Leftmerge)} \\
&s_1 \xrightarrow{\rho \sigma \tau} r_1 \quad s_2 \xrightarrow{\sigma \tau} r_2 \\
&s_1 \parallel s_2 \xrightarrow{\rho \sigma \tau} r_1 \parallel r_2 \quad \text{(Sync)}
\end{align*}
\]

The statement \( s_1 +_\rho s_2 \) first resolves the non-determinism in \( s_1 \) and \( s_2 \) in any order. If all non-determinism in both \( s_1 \) and \( s_2 \) has been resolved, \( s_1 +_\rho s_2 \) acts like \( s'_1 \) with probability \( \rho \) and like \( s'_2 \) with probability \( 1 - \rho \).

\[
\begin{align*}
&s_1 \xrightarrow{1} s \\
&s_1 +_\rho s_2 \xrightarrow{1} s +_\rho s_2 \\
&s_2 +_\rho s_1 \xrightarrow{1} s_2 +_\rho s \\
&s_1 \xrightarrow{\sigma \tau \alpha} s \\
&s_1 +_\rho s_2 \xrightarrow{\rho \sigma \tau \alpha} s \\
&s_2 +_\rho s_1 \xrightarrow{(1 - \rho)\sigma \tau \alpha} s \\
&s_1, s_2 \xrightarrow{i} \\
&s_2 +_\rho s_1 \xrightarrow{(\rho)^i} s
\end{align*}
\]

With these last rules there is a subtle point that has to be taken into account. It may be possible to derive the same transition more than once. The easiest example is

\[
s = a +_\frac{1}{2} a
\]

Here the transition \( s \xrightarrow{\frac{1}{2} a} \text{E} \) can be derived twice. This is not the same as having the transition once, because the total probability of \( s \) performing an action should be one, not a half. This requires extra care when defining the operational semantics.

**Example 5.2** Let \( D(x) = (a +_\rho b) \vdash \) then \( a \xrightarrow{1 + a} \text{E} \), \( b \xrightarrow{1 + b} \text{E} \) and \( c \xrightarrow{1 + c} \text{E} \) by axiom
Before giving the operational semantics some properties of the transition system are stated. First some notation is introduced and then several properties of the transition system can be shown by induction on the complexity measure \( \text{wgt} \) introduced before.

**Definition 5.3** For any statement \( s \) the following notation is introduced.

(a) That \( s \) can take one or more \( i \) steps and end up in \( s' \) is denoted by \( s \xrightarrow{+i} s' \), i.e. \( s \xrightarrow{+i} s' \) when \( s \xrightarrow{1} s' \) or \( s \xrightarrow{+i} s'' \) and \( s'' \xrightarrow{+i} s' \).

(b) If \( s \) can take zero or more \( i \) steps and end up in \( s' \), \( s \xrightarrow{0} s' \) is written, i.e. \( s \xrightarrow{0} s' \) when \( s \xrightarrow{+1} s' \) or \( s \equiv s'' \).

(c) Also, \( s \xrightarrow{\cdot} s' \) is used to denote that \( s' \) is a normal form of \( s \) with regards to \( i \) steps, i.e. \( s \xrightarrow{\cdot} s' \) and \( s' \xrightarrow{\cdot} \).

(d) The probability that \( s \) takes an internal step is written as \( \Pi(s) \),

\[
\Pi(s) = \sum_{s \xrightarrow{\cdot} s'} p + \sum_{s \xrightarrow{+1} e} p.
\]

(e) In \( s \Rightarrow_\circ S \) the probabilistic options of \( s \) that belong to one non-deterministic alternative are grouped together. To be precise, \( s \Rightarrow_\circ S \) if \( s \xrightarrow{\cdot} s_0 \) and

\[
S = \{ n \cdot \beta \mid s_0 \xrightarrow{n} \beta \cdot E \} \cup \{ n \cdot \langle \beta, s' \rangle \mid s_0 \xrightarrow{n} \beta \cdot E \} \\
\cup \{ (1 - \Pi(s_0)) \cdot \delta \mid \Pi(s_0) < 1 \}.
\]

(f) Comparable to this \( s \Rightarrow_D S \) is defined as, \( s \Rightarrow_D S \) when \( s \xrightarrow{\cdot} s_0 \) and

\[
S = \{ n \cdot \alpha \mid s_0 \xrightarrow{n} \alpha \cdot E \} \cup \{ n \cdot \langle \alpha, s' \rangle \mid s_0 \xrightarrow{n} \alpha \cdot E \}.
\]

(g) The successor set of \( s \), \( S(s) = \{ S \mid s \Rightarrow_D S \} \).
The notation \( s \xrightarrow{s'} \) indicates that \( s' \) is one of the non-deterministic alternatives of \( s \). The notation \( s \Rightarrow S \) denotes that \( S \) contains all probabilistic alternatives for a single non-deterministic alternative of \( s \). Each probabilistic transition that can be derived \( n \) times \( (s_0 \xrightarrow{a} r) \) will also occur \( n \) times in the multiset \( S \). This ensures the correct handling of statements like \( a + \frac{1}{2} a \). If the total probability is less than the one the ‘missing’ probability \( (1 - \prod(s)) \) is interpreted as a chance of deadlock and \( \delta \) is introduced.

In \( s \Rightarrow S \), \( S \) is a multiset over \( PIA \) + \( PIA \times Stat \). The shorthand \([a_I, \{a, x\}], \kappa]\) introduced after definition 4.10 is used. For this specific case the space \( A \) is \( PIA \) and the space \( X \) is \( Stat \). An element \( \beta \in PIA \) is a pair of an internal action \( a \) and a probability \( \rho \).

\[
[\beta, \{\beta, s\} J, \kappa] = \{\{\beta | i \in I\} \cup \{(j, s_j) | j \in J\} \cup \{\kappa * \delta | \kappa > 0\} = \{\rho_i * b_i | i \in I\} \cup \{(j, s_j) | j \in J\} \cup \{\kappa * \delta | \kappa > 0\}.
\]

The notation \( s \Rightarrow_D S \) is like \( s \Rightarrow_O S \) except that unmatched synchronization actions are not interpreted as deadlock but maintained in the output. This notation is used when comparing the operational and the denotational model.

In \( s \Rightarrow_D S \), \( S \) is a multiset over \( PIA + PIA \times Stat \). The shorthand \([a_I, \{a, x\}], \kappa]\) introduced in definition 4.10 is used. For this specific case the space \( A \) is \( PIA \) and the space \( X \) is \( Stat \).

\[
[a_I, \{a, s\} J] = \{\alpha_{i, j} | i \in I\} \cup \{(j, s_j) | j \in J\} = \{\rho_i * a_{i, j} | i \in I\} \cup \{(j, s_j) | j \in J\}.
\]

The successors of a statement are divided into a set of multisets. Each multiset \( S \) in the successor set \( S(s) \) of \( s \) gives all probabilistic alternatives for one non-deterministic alternative. The auxiliary \( \alpha \) actions that are necessary to reach the non-deterministic alternative are not explicitly present. Their purpose was to be able to split the successors of \( s \) into the multisets contained in \( S(s) \). There is, in general, more than one way to reach the same multiset. This, however, is irrelevant. For non-deterministic alternatives it is not relevant how often they occur, so the non-deterministic options form a set as opposed to a multiset.

The following properties of the transition system can now be shown to hold. The proofs are based on induction on the weight. The weight function \( \text{wgt} \) introduced in section 3 is used for this.

**Lemma 5.4** For \( T_p \) the following holds.

(a) If \( s \xrightarrow{a} s' \) then \( \text{wgt}(s') < \text{wgt}(s) \) and \( s \xrightarrow{E} \).

(b) The auxiliary \( \alpha \) steps and \( \alpha \) steps are mutually exclusive, i.e. if \( s \xrightarrow{a} s' \) then \( s \xrightarrow{\alpha} r \) and if \( s \xrightarrow{\alpha} r \) then \( s \xrightarrow{a} s' \).

(c) \( T_p \) is strongly normalizing for \( \alpha \)-steps (no infinite sequence \( s \xrightarrow{\alpha} s_1 \xrightarrow{\alpha} s_2 \xrightarrow{\alpha} \ldots \) exists).
(d) $T_p$ is finitely branching, that is, for all $s \in \mathcal{L}_p$:

1. $S(s)$ is a finite set and,
2. $S$ is a finite multiset for each $S \in S(s)$.

(e) If $s \Rightarrow [\beta_1, (\beta, s)] \Rightarrow [\beta_1, (\beta, s)] \Rightarrow [\beta_1, (\beta, s)] \Rightarrow \ldots$, then the sum of probabilities of elements of $[\beta_1, (\beta, s)] \Rightarrow [\beta_1, (\beta, s)] \Rightarrow [\beta_1, (\beta, s)] \Rightarrow \ldots$ is 1. (In this sum the probability of an element occurring $n$ times is counted $n$ times).

$$\sum_{i \in I} p_i + \sum_{j \in J} p_j + \kappa = 1$$

Proof

(a) That $s \not\Rightarrow E$ is directly clear from the transition system, the rest can be shown by induction on $\text{wgt}(s)$. Three cases, the others are similar:

- $[s_1 + s_2]$ If $s_1 + s_2 \overset{1}{\Rightarrow} s'$ then $s' = s_1$ or $s' = s_2$. Therefore $\text{wgt}(s_1 + s_2) = 1 + \text{wgt}(s_1) + \text{wgt}(s_2) > \text{wgt}(s_1) + \text{wgt}(s_2) \geq \text{wgt}(s')$.

- $[s_1; s_2]$ If $s_1; s_2 \overset{1}{\Rightarrow} s'$ then $s_1 \overset{1}{\Rightarrow} s'_1$ and $s' = s'_1; s_2$. Since $\text{wgt}(s'_1) \overset{1}{=} \text{wgt}(s_1)$;

- $\text{wgt}(s_1; s_2) = \text{wgt}(s_1) + 1 > \text{wgt}(s'_1) + 1 = \text{wgt}(s'_1; s_2) = \text{wgt}(s')$.

- $[s_1 + \rho s_2]$ If $s_1 + \rho s_2 \overset{1}{\Rightarrow} s'$ then $s_1 \overset{1}{\Rightarrow} s'_1$, $s' = s'_1 + \rho s_2$ or $s_2 \overset{1}{\Rightarrow} s'_2$, $s' = s_1 + \rho s'_2$.

In the first case: $\text{wgt}(s_1 + \rho s_2) = 1 + \text{wgt}(s_1) + \text{wgt}(s_2) \overset{1}{=} \text{wgt}(s'_1) + \text{wgt}(s_2) = \text{wgt}(s')$.

In the second case: $\text{wgt}(s_1 + \rho s_2) = 1 + \text{wgt}(s_1) + \text{wgt}(s_2) \overset{1}{=} \text{wgt}(s'_1) + \text{wgt}(s'_2) = \text{wgt}(s')$.

(b) Directly clear from the transition system.

(c) Direct consequence of part (a).

(d) Clear by induction on $\text{wgt}(s)$ using part (a).

(e) Clear from the definition of $s \Rightarrow_0 S$. □

With these properties in place the operational semantics can be found from the transition system. First the domains which are used to define the operational and denotational semantics are defined.

5.3 Domains for generative models

The domains that are used have three levels. The basic level is that of actions, possibly followed by a process. On top of the action level there are the levels of non-determinism and probability. For the generative models probability is assigned to actions, so the first level on top of the action level is that of probability. The probabilistic choice is described using multisets in a branching domain.
and using distribution in a linear domain. The non-deterministic level is put on top of the probabilistic level. The non-determinism is described using sets.

<table>
<thead>
<tr>
<th>Generative Domain</th>
<th>Example(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top level ($\mathbb{P}$)</td>
<td>$\mathbb{P}_{nce}(\mathbb{Q})$</td>
</tr>
<tr>
<td>Middle level ($\mathbb{Q}$)</td>
<td>$\mathbb{M}_d(\mathbb{R})$</td>
</tr>
<tr>
<td>Action level ($\mathbb{R}$)</td>
<td>$A + A \times \mathbb{R}$</td>
</tr>
</tbody>
</table>

Structure of a (branching) generative domain.

A linear domain, a mixed domain and two branching domains are given below. A model using a linear, mixed or branching domain are called a linear, mixed or branching model, respectively. In the style of [16] these models can be ordered by their distinguishing power. The situation here, however, is a lot simpler. There are two branching models, a denotational model and an operational-like model which distinguishes fewer statements than the denotational model. The term operational-like model is used for a model based on the transition system but defined on another domain than the operational domain. The mixed model is a version of the operational model in which the non-determinism is described in a linear fashion but the probability in a branching way. As one might expect the mixed model distinguishes less than the branching models and the linear model less than the mixed model.

The linear model is used for the operational meanings. When a machine executes a statement only the resulting actions are visible. The choices a machine makes cannot be observed, only their result. This is reflected in a linear domain, where the moment of choice is not visible.

When composing statements the moments of choice are relevant to the behavior of the composed system. For the denotational semantics, which is based on composing meanings, a branching domain, which does include the moments of choice, is used. This branching information is actually only needed if choice is interpreted as global choice, as for local choice the moment of choice is irrelevant. The mixed domain remembers the moments of choice for probability, but not the moments of choice for non-determinism. The mixed domain is used for the comparison of the (branching) denotational semantics and the (linear) operational semantics.

In the operational semantics deadlock is modeled explicitly. The symbol $\delta$ is included in the domain to model deadlock. The only cause of deadlock in this report is failure to synchronize. The sets modeling the non-deterministic choice can be assumed to be non-empty since there is always at least one alternative, possibly to deadlock with probability one.

There are two possible causes for synchronization to fail. There may be an unmatched synchronization action or an actual attempt to synchronize may fail. The first situation does not lead to deadlock in the denotational model. The statement may still be composed with a statement that matches the synchronization action. The second situation will also fail in the denotational semantics and is modeled by a total probability of less than one. The statement $c \cdot b$ can be modeled by $\{ \{\}\}$ (singleton the empty multiset) which indicates that the
statement only has one option, to deadlock with probability one.

In the branching domain the information about the moment of choice is incorporated. The non-determinism is described using non-empty finite sets. Every element in the set is one possible non-deterministic alternative. A technical problem with the space of finite sets is that this space is not complete. To solve this problem the space is expanded to contain all non-empty compact sets. The functor that yields this space is \( P_{nco}(\cdot) \).

As described before, multiple occurrences of probabilistic options should not be identified and therefore multisets rather than sets are used to describe the probability. These multisets are also finite. The functor giving the space of nonempty finite multisets is \( M_f \). The space of finite multisets is compact so no expansion is required here.

The action level consists of a set of possible actions \( \text{Act} \) which may be followed by a process, i.e., an element of the domain \( P_d \). In the generative model the actions are labeled with the probability of this action occurring (number in \([0,1]\)), yielding \( PAct \) as the set of possible actions. The pairing of an action with the process that may follow is done by \( x \). The + provides the choice between a single action or an action followed by a process.

The following domain equations describe the branching domain used for the denotational semantics.

\[
\begin{align*}
P_d &\simeq P_{nco}(Q_d) \\
Q_d &\simeq M_f(\mathbb{R}_d) \\
\mathbb{R}_d &\simeq PAct + PAct \times id_{P}(P_d)
\end{align*}
\]

In the operational semantics only the internal actions are allowed (also labeled with a probability) giving \( PAct \) as the set of possible actions. Unmatched synchronization actions lead to deadlock. The following domain equations describe the domain for a branching operational-like semantics.

\[
\begin{align*}
P_b &\simeq P_{nco}(Q_b) \\
Q_b &\simeq M_f(\mathbb{R}_b) \\
\mathbb{R}_b &\simeq PAct_b \cup PAct \times P_b
\end{align*}
\]

The branching operational-like semantics is used to compare the operational and the denotational semantics.

In the mixed domain the moment of probabilistic choice is remembered. The moment of non-deterministic choice is abstracted away by collecting all non-determinism. This yields the following domain equations for a mixed operational-like semantics.

\[
\begin{align*}
P_m &= P_{nco}(Q_m) \\
Q_m &\simeq M_f(\mathbb{R}_m) \\
\mathbb{R}_m &\simeq PAct_b \cup PAct \times id_{P}(Q_m)
\end{align*}
\]
The mixed operational-like semantics is used for the comparison of the operational and the denotational semantics. Since all non-deterministic choices are collected, the non-determinacy may be no longer finite because there may be infinitely many choices in the execution of a statement. (See the example below.) The compact sets capture exactly the infinite non-deterministic choice obtained from collecting finite non-deterministic choices. The expansion from finite to compact subsets is needed here for more than technical reasons.

\[
\begin{align*}
  s &= x \\
  D(x) &= (a + b);x
\end{align*}
\]

The execution of \( s \) is a not ending sequence of choosing between \( a \) and \( b \). Collecting all choices results in a choice from any infinite sequence of \( a \)'s and \( b \)'s. This is clearly not a finite choice.

In the linear domain the moment of probabilistic choice should not be present. To abstract from the moment of probabilistic choice all probability is collected. As with the non-deterministic choice this might result in an infinite choice. Consider the example below which can be seen as an infinite sequence of fair coin tosses.

\[
\begin{align*}
  s &= x \\
  D(x) &= (a + \frac{1}{2} b);x
\end{align*}
\]

There are infinitely many possible sequences of actions. However, each of these sequences has probability 0 of occurring. (The probability of the first action occurring is \( \frac{1}{2} \), the probability of the first two actions occurring is \( \frac{1}{4} \), etc. So the probability of the complete trace occurring is 0.) This indicates that (multi)sets are insufficient to describe the behavior the statement \( s \). Distributions (on sequences of actions) are used instead. The domain \( \mathbb{P} \) described by the following domain equations is used for the (linear) operational semantics.

\[
\begin{align*}
  \mathbb{P} &= \mathcal{P}_{nc}(\mathcal{Q}) \\
  \mathcal{Q} &= \mu(\mathbb{P}) \\
  \mathbb{R} &= \mathcal{I}(\mathcal{P})
\end{align*}
\]

In the next subsections these domains are used for a comparative semantics for the language \( \mathcal{L} \) with local interpretation of both kinds of choice. In section 6 these domains are used again, there for a comparative semantics for the language \( \mathcal{L} \) with local interpretation of non-deterministic choice and a global interpretation of probabilistic choice.

### 5.4 Operational semantics

The operational semantics uses the generative linear domain, \( \mathbb{P} \), described in the previous subsection.
The operational semantics of a statement \( s \) is a set of distributions. The probability that the sequence of actions that \( s \) takes are in a given set \( B \) depends on the non-deterministic choices now and in the future. If \( s \overset{\alpha}{\rightarrow} s_0 \Rightarrow S \) the current non-deterministic choice is the alternative \( s_0 \). The possible future non-deterministic choices are collected in \( O(s') \) for each \( \rho \ast b, s' \) in \( S \). Making the future non-deterministic choices consist of picking a distribution \( \mu \) from \( O(s') \) for each \( s' \). Each possible combination of these distributions will yield a possible distribution for \( s_0 \).

In the definition below making the current non-deterministic choice is described by the first equation. Making all future non-deterministic choices and combining the chosen distribution to one new distribution is described by the second equation. In the equations below the notation introduced in definition 4.10, definition 5.3 and the remarks following these definitions is used. Additionally, \( \Delta_x = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \)

**Definition 5.5** The operational semantics \( O \) is given by

\[
O(s) = \bigcup_{s \Rightarrow O s} \hat{O}(S)
\]

\[
\hat{O}([\beta_1, \langle \beta, s \rangle J, \kappa]) = \{ \lambda B: \sum_{i \in I} \rho_i \cdot \Delta_i \in B + \sum_{j \in J} \rho_j \cdot \mu_j(B/b_j) + \kappa \cdot \Delta_i \in B \mid \mu_j \in O(s_j) \}.
\]

The first equation should be clear. In the second equation there are three possible situations. If \( \rho \ast b \in S \) it contributes \( \rho \) to the probability of a sequence in \( B \) occurring whenever \( b \in B \). If \( \delta \in B \) then the deadlock component \( \kappa \) contributes to the probability of \( B \). If \( \langle \rho \ast b, s \rangle \in S \) it will contribute to the probability of a sequence in \( B \) occurring only if there is a sequence in \( B \) that starts with a \( b \) and \( s \) can do the rest of the sequence. The chance that \( s \) will do the rest of the sequence depends on the non-deterministic choices made in \( s \). If the non-deterministic choices result in a distribution \( \mu \) then the probability of doing the rest of a sequence in \( B \) that starts with a \( b \) \( \mu(B/b) \). The total contribution of \( \langle \rho \ast b, s \rangle \) becomes \( \rho \cdot \mu(B/b) \).

The definition can be justified by showing that it is the fixed point of a higher-order transformation.

**Lemma 5.6** Let \( Sem = Stat \xrightarrow{1} \mathbb{P}_c \) and let \( \Phi: Sem \rightarrow Sem \) be given by

\[
\Phi(\psi)(s) = \bigcup_{s \Rightarrow O s} \hat{\Phi}(\psi)(S)
\]

\[
\hat{\Phi}(\psi)([\beta_1, \langle \beta, s \rangle J, \kappa]) = \{ \lambda B: \sum_{i \in I} \rho_i \cdot \Delta_i \in B + \sum_{j \in J} \rho_j \cdot \mu_j(B/b_j) + \kappa \cdot \Delta_i \in B \mid \mu_j \in \psi(s_j) \}.
\]

then \( \Phi \) has a unique fixed point, i.e. there is exactly one function \( O \) satisfying the equations in definition 5.5.
Proof  Sufficient to show that \( \Phi \) is a contractive function. Using Banach’s theorem this gives that \( \Phi \) has a unique fixed point.

Note that
\[
B_{\frac{1}{\lambda}}(a \cdot w) / b = \begin{cases} 
\emptyset & \text{if } a \neq b \\
B_{s}(w) & \text{otherwise}.
\end{cases}
\]

Let \( \psi_1, \psi_2 \in \text{Sem} \) with \( d(\psi_1, \psi_2) < \epsilon \) be given. Then \( d(\psi_1(s), \psi_2(s)) < \epsilon \) for all \( s \in \text{Stat} \). This gives that for all \( \mu_1 \in \psi_1(s) \) there is a \( \mu_2 \in \psi_2(s) \) such that \( d(\mu_1, \mu_2) < \epsilon \).

To show that \( d(\Phi(\psi_1), \Phi(\psi_2)) \leq \frac{1}{2} \epsilon \) one has to show that for all \( s \in \text{Stat} \) and all \( \mu \in \Phi(\psi_1)(s) \) there is a \( \mu' \in \Phi(\psi_2)(s) \) with a distance to \( \mu \) of \( \frac{1}{2} \epsilon \) or less. The reverse (being able to find a \( \mu \) for each \( \mu' \)) holds by symmetry.

Let \( s \in \text{Stat} \) and \( \mu \in \Phi(\psi_1)(s) \) then
\[
\mu = \lambda B \sum_{i \in I} \rho_i \cdot \Delta_{i \in B} + \sum_{j \in J} \rho_j \cdot B_{j / b} + \kappa \cdot \Delta_{i \in B}
\]
for some \( \mu_j \in \psi_1(s_j) \), \( (j \in J) \). For each \( \mu_j \) there is a \( \mu_j' \in \psi_2(s) \) such that \( d(\mu_j, \mu_j') < \epsilon \). Define
\[
\mu' = \lambda B \sum_{i \in I} \rho_i \cdot \Delta_{i \in B} + \sum_{j \in J} \rho_j \cdot B_{j / b} + \kappa \cdot \Delta_{i \in B}.
\]
Then
\[
\mu(B_{\frac{1}{\lambda}}(aw)) - \mu'(B_{\frac{1}{\lambda}}(aw)) = \sum_{j \in J} \rho_j \cdot (\mu_j(B_{\frac{1}{\lambda}}(aw) / b_j) - \mu_j'(B_{\frac{1}{\lambda}}(aw) / b_j))
\]
\[= \sum_{j \in J \land b_j = a} \rho_j \cdot (\mu_j(B_{s}(w)) - \mu_j'(B_{s}(w)))
\]
\[= 0.
\]
This gives that for each \( \mu \in \Phi(\psi_1)(s) \) there is a \( \mu' \in \Phi(\psi_2)(s) \) such that \( d(\mu, \mu') \leq \frac{1}{2} \epsilon \). Since this follows for all \( s \in \text{Stat} \), \( d(\Phi(\psi_1), \Phi(\psi_2)) \leq \frac{1}{2} \epsilon \). \( \square \)

This completes the definition of the operational semantics. Lemma 5.6 illustrates the use of a higher-order function to show correctness of reflexive definitions. The same technique can be used for other definitions below but the higher-order functions are not given explicitly.

5.5 Denotational semantics

When checking properties of a statement one wants to be able to decompose the statement in parts and work with the parts of the statement. This requires a way of finding how the parts of the statement compose. One wants to describe the meaning of a statement based on the meaning of the parts of a statement (compositionality principle). The denotational semantics will use the compositionality principle to give the meaning of a statement. For each of the
syntactical operators (p.e. "\|\|") a semantical counterpart is defined which builds the (denotational) meaning of a composed system (p.e. \( s_1 \| s_2 \)).

The operational semantics is not compositional. The statements \( c_1 \) and \( c_2 \) have the same operational behavior, but within the context \( \square \| c_1 \) they behave differently. (Here \( \square \) indicates the place in the context where a statement can be substituted.) This means that the operational behavior does not contain sufficient information to be able to compose statements. For the denotational model extra information about a statement is maintained. Information about unmatched synchronization actions and about moments of choice is added. As mentioned before the branching information is not really needed for local interpretation of choice, only for global interpretation of choice. Unmatched synchronization actions are visible and do not result in deadlock. The domain \( \mathbb{P}_d \) as described in subsection 5.3 is used. On the domain \( \mathbb{P}_d \) the semantical operators are defined as follows:

**Definition 5.7** All operators are non-expansive functions that take pairs of (denotational) processes and yield a single process. \( Op = \mathbb{P}_d \times \mathbb{P}_d \rightharpoonup \mathbb{P}_d \)

(a) The operator \( + \in Op \) is defined by

\[
p_1 + p_2 = p_1 \cup p_2
\]

(b) The operator \( +_\rho \in Op \) is defined by

\[
p_1 +_\rho p_2 = \{ q \in p_1, q_2 \subseteq p_2 \}
q_1 +_\rho q_2 = \rho q_1 \cup (1 - \rho) q_2.
\]

where \( \rho \) is short for \( \{ \rho \sigma \ast a \mid \sigma \ast a \in q \} \cup \{ \rho \sigma \ast a, p \mid \sigma \ast a, p \in q \} \).

(c) The operator \( ; \in Op \) is defined by

\[
p_1 ; p_2 = \{ q_1 ; q_2 \mid q_1 \in p_1 \}
q ; p = \{ \{ \alpha, p \mid \alpha \in q \} \cup \{ \alpha, p' ; p \mid \alpha, p' \in q \} \}.
\]

(d) The operator \( \| \in Op \) is defined by

\[
p_1 \| p_2 = p_1 \| p_2 \cup p_1 \| p_2 \cup p_1 \| p_2
q \| p = \{ \{ \alpha, p \mid \alpha \in q \} \cup \{ \alpha, p' ; p \mid \alpha, p' \in q \} \}
q_1 \| q_2 = \{ \rho \sigma \ast \tau \mid p \ast c \in q_1, \sigma \ast \tau \in q_2 \}
\cup \{ \rho \sigma \ast \tau, p \mid \rho \ast c, p \in q_1, \sigma \ast \tau \in q_2 \}
\cup \{ \rho \sigma \ast \tau, p \mid \rho \ast c, p \in q_1, \sigma \ast \tau \in q_2 \}
\cup \{ \rho \ast c, p \mid \rho \ast c, p \in q_1, \sigma \ast \tau \in q_2 \}.
\]
Note that for $p_1|p_2$ may have a total probability of less than one. It may even be the empty multiset. The ‘missing’ probability is interpreted as a probability of deadlock.

The definitions are reflexive but can be shown to be correct using metric machinery by defining each operator as the fixed point of a contractive higher order function $\Omega: O_p \times O_p \rightarrow O_p$.

The operators introduced above are non-expansive by definition. For the operator ‘;’ this can be strengthened. The operator ‘;’ is non-expansive in its first component and contractive in its second, i.e.

$$d(p_1; p_2, p_1'; p_2') \leq \max\{d(p_1, p_1'), \frac{1}{2}d(p_2, p_2')\}.$$ 

To justify the following definition, $D$ can be defined as the fixed point of a higher-order mapping. Contractiveness of ‘;’ in its second argument is required for contractiveness of this higher-order mapping.

**Definition 5.8** $D: \mathcal{L}_P \rightarrow \mathbb{P}_d$ is given by

\[
D(a) = \{\{1*a\}\} \\
D(x) = D(D(x)) \\
D(s_1 op s_2) = D(s_1) op D(s_2).
\]

A single action $a$ acts like $a$ with probability one. Recursion is handled by body replacement and the semantical operator $op$ is used to give the meaning of any statement built using the syntactic construct $op$. The operator $op$ is one of operations ‘+’, ‘+$’, ‘;’ or ‘||’.

**Example 5.9** Let $D(x) = (a +_\rho b)c$ then

\[
D(x) = D(D(x)) = D((a +_\rho b)c) = D(a +_\rho b)D(c) \\
= D(a) +_\rho D(b)\{\{1*c\}\} = \{\{1*a\}\} +_\rho \{\{1*b\}\};\{\{1*c\}\} \\
= \{\{\rho*a, (1-\rho)*b\}\};\{\{1*c\}\} \\
= \{\{\rho*a, \{\{1*c\}\}\}, (1-\rho)*b, \{\{1*c\}\}\}\}
\]
5.6 Comparing the operational and denotational semantics

The denotational semantics maintains more information than the operational semantics. As described in the previous section this is necessary to achieve compositionality. The branching structure and unmatched synchronization actions are still in the semantics. To compare the denotational and operational semantics a series of operational-like intermediate semantics are introduced. Step by step the extra information needed for the denotational semantics is added, starting from the operational semantics.

The first step is to introduce an operational-like semantics on a mixed domain. This adds the information about the probabilistic branching structure.

**Definition 5.10** The mixed operational-like semantics $O_m$ is given by

$$O_m(s) = \{ [\beta_1, \langle \beta, q \rangle, j, k] \mid q_j \in O(s_j), s \Rightarrow_O [\beta_1, \langle \beta, s \rangle, j, k] \}.$$  

First a non-deterministic alternative $[\beta_1, \langle \beta, s \rangle, j, k]$ is chosen by resolving the current non-determinism. For each probabilistic option in $[\beta_1, \langle \beta, s \rangle, j, k]$, all future non-deterministic choices are made by selecting a $q_j \in O(s_j)$. By combining all the selected $q$, the multiset $[\beta_1, \langle \beta, q \rangle, j, k]$ is obtained. Each $[\beta_1, \langle \beta, q \rangle, j, k]$ represents an option of $s$ in which all non-determinism has been resolved.

The mixed domain can be related to the linear domain by the following abstraction function that removes the probabilistic branching structure.

**Definition 5.11** The abstraction function $\text{abs}_1 : \mathbb{P}_m \rightarrow \mathbb{P}_l$ is defined as:

$$\text{abs}_1(p) = \{ \text{abs}_1'(q) \mid q \in p \}$$

$$\text{abs}_1'([\beta_1, \langle \beta, q \rangle, j, k])(B) = \sum_{i \in I} \rho_i \cdot \Delta_{i \in B} + \sum_{j \in J} \rho_j \cdot \text{abs}_1'(q_j)(B/b_j) + \kappa \cdot \Delta_{i \in B}$$

The abstraction function $\text{abs}_1$ is only well-defined on a subspace of the domain $\mathbb{P}_m$. This is not a problem since $O_m$ only yields results within this subspace. The operational semantics is an abstraction of the mixed model introduced above. If from the result $O_m(s)$ the probabilistic branching structure is removed, the operational semantics of $s$ is obtained.

**Lemma 5.12** $O(s) = \text{abs}_1(O_m(s))$.

The next step is to introduce an operational-like model on a branching domain. The information about the non-deterministic branching structure is added. The branching operational-like model on the domain $\mathbb{P}_b$ as introduced in subsection 5.3 is defined below.

**Definition 5.13** The function $O_b : \mathcal{L}_p^+ \rightarrow \mathbb{P}_b$ is given as

$$O_b(s) = \{ [\beta_1, \langle \beta, p \rangle, j, k] \mid p_j = O_b(s_j), s \Rightarrow_O [\beta_1, \langle \beta, s \rangle, j, k] \}.$$

29
A non-deterministic option $[\beta_I, \langle \beta, s \rangle_J, \kappa]$ is selected. For each probabilistic element of this option the semantics is found. For elements of the form $\beta$ or $\rho \ast \delta$ this is simply the element itself. For elements of the form $\langle \beta, s \rangle$ this is $\langle \beta, O_b(s) \rangle$.

The branching domain can be related to the mixed domain by the following abstraction function that removes the non-deterministic branching structure.

**Definition 5.14** The abstraction function $\text{abs}_2 : \mathbb{P}_b \rightarrow \mathbb{P}_m$ is defined as:

$$\text{abs}_2(p) = \bigcup_{[\beta_I, \langle \beta, p \rangle_J, \kappa] \in p} \{ [\beta_I, \langle \beta, q \rangle_J, \kappa] \mid q_J \in \text{abs}_2(p_J) \}.$$ 

The mixed model is an abstraction of the branching model introduced above. If from $O_b(s)$ the non-deterministic branching structure is removed, the mixed semantics of $s$ is obtained.

**Lemma 5.15** $O_m(s) = \text{abs}_2(O_b(s)).$

The last step is to introduce an operational-like semantics on the denotational domain and to show that this model coincides with the denotational semantics. The unmatched synchronization actions are added and deadlock is removed. The intermediate model $O^*$ on the denotational domain is defined as:

**Definition 5.16** The function $O^* : \mathcal{L}_p^+ \rightarrow \mathbb{P}_d$ is given as

$$O^*(s) = \{ [\alpha_I, \langle \alpha, p \rangle_J] \mid p_J = O^*(s_J), s \Rightarrow p [\alpha_I, \langle \alpha, s \rangle_J] \}.$$ 

The denotational domain can be related to the branching operational style domain by the following abstraction function that removes the communication actions and introduces deadlock where needed.

**Definition 5.17** The function $\text{abs}_3 : \mathbb{P}_d \rightarrow \mathbb{P}_b$ is defined as:

$$\text{abs}_3(p) = \{ \text{abs}_3(q) \mid q \in p \}$$

$$\text{abs}_3(q) = \{ [\beta] \mid [\beta] \in q \} \cup \{ [\langle \beta, \text{abs}_3(p) \rangle] \mid [\langle \beta, p \rangle] \in q \}$$

$$\Pi(q) = \sum_{\rho \in q} \rho + \sum_{(\rho h, p) \in q} \rho.$$ 

$\Pi(q)$ gives the probability that the next action in $q$ is an internal action.

The model on the branching domain is an abstraction of the model on the denotational domain. If in $O^*(s)$ the synchronization actions and ‘missing’ probability are replaced by deadlock, $O_b(s)$ is obtained.

**Lemma 5.18** $O_b = \text{abs}_3 \circ O^*.$
That \( \mathcal{D} \) and \( \mathcal{O}^* \) coincide (on \( \mathcal{L}_{p} \)) can be checked using the higher-order function \( \Phi^* \) implicitly used to define \( \mathcal{O}^* \). It is sufficient to show that \( \Phi^*(\mathcal{D}) = \mathcal{D} \). Because the fixed point of \( \Phi^* \) is unique, this implies that \( \mathcal{D} = \mathcal{O}^* \).

The denotational semantics is only defined on \( \mathcal{L}_{p} \) where \( \mathcal{O}^* \) is defined on \( \mathcal{L}_{p}^* \). The higher-order function \( \Phi^* \) expects an argument that is defined on \( \mathcal{L}_{p}^* \). This is why \( \mathcal{D} \) is expanded to \( \mathcal{L}_{p}^* \), visually by adding

\[
\mathcal{D}(s_1 \parallel s_2) = \mathcal{D}(s_1) \parallel \mathcal{D}(s_2) \\
\mathcal{D}(s_1 \mid s_2) = \mathcal{D}(s_1) \mid \mathcal{D}(s_2).
\]

**Lemma 5.19** \( \mathcal{D} \) is a fixed point of \( \Phi^* \), i.e. \( \Phi^*(\mathcal{D})(s) = \mathcal{D}(s) \).

**Proof** By induction on \( \mathrm{wgt}(s) \). Only the two most interesting cases are given.

\[
\begin{align*}
\mathcal{D}(s_1 + \rho \ s_2) &= \mathcal{D}(s_1) + \rho \mathcal{D}(s_2) \\
&= \Phi^*(\mathcal{D})(s_1) + \rho \Phi^*(\mathcal{D})(s_2) \\
&= \{ [\alpha, (\alpha, p)] \mid p = \mathcal{D}(s), s_1 \Rightarrow_D [\alpha, (\alpha, s)] \} + \rho \{ [\alpha, (\alpha, p)] \mid p = \mathcal{D}(s), s_2 \Rightarrow_D [\alpha, (\alpha, s)] \} \\
&= \{ [\alpha, (\alpha, p)] \mid p = \mathcal{D}(s), s_1 \Rightarrow_D [\alpha, (\alpha, s)] \} \\
&= \mathcal{D}(s_1) + \rho \mathcal{D}(s_2) \\
\mathcal{D}(s_1 + s_2) &= \mathcal{D}(s_1) + \mathcal{D}(s_2) \\
&= \Phi^*(\mathcal{D})(s_1) + \Phi^*(\mathcal{D})(s_2) \\
&= \{ [\alpha, (\alpha, p)] \mid p = \mathcal{D}(s), s_1 \Rightarrow_D [\alpha, (\alpha, s)] \} \\
&= \{ [\alpha, (\alpha, p)] \mid p = \mathcal{D}(s), s_2 \Rightarrow_D [\alpha, (\alpha, s)] \} \\
&= \{ [\alpha, (\alpha, p)] \mid p = \mathcal{D}(s), s_1 \Rightarrow_D [\alpha, (\alpha, s)] \} \cup \{ [\alpha, (\alpha, p)] \mid p = \mathcal{D}(s), s_2 \Rightarrow_D [\alpha, (\alpha, s)] \} \\
&= \mathcal{D}(s_1) + \mathcal{D}(s_2) \\
&= \Phi^*(\mathcal{D})(s_1 + s_2).
\end{align*}
\]

Combining the results in this section gives that the operational semantics is an abstraction of the denotational semantics.

**Theorem 5.20** The operational semantics is an abstraction of the denotational semantics, \( \mathcal{O} = \text{abs} \circ \mathcal{D} \).
\textbf{Proof} Define \( \text{abs} = \text{abs}_1 \circ \text{abs}_2 \circ \text{abs}_3 \) then by the previous lemmas \( \mathcal{O} = \text{abs} \circ \mathcal{D} \):

\[
\begin{align*}
\mathcal{O} &= [\text{lemma } 5.12] \; \text{abs}_1 \circ \mathcal{O}_m \\
 &= [\text{lemma } 5.15] \; \text{abs}_1 \circ \text{abs}_2 \circ \mathcal{O}_b \\
 &= [\text{lemma } 5.18] \; \text{abs}_1 \circ \text{abs}_2 \circ \text{abs}_3 \circ \mathcal{O}^* \\
 &= [\text{lemma } 5.19] \; \text{abs} \circ \mathcal{D}.
\end{align*}
\]

The result obtained in theorem 5.20 gives that \( \mathcal{D} \) is correct with respect to \( \mathcal{O} \) (i.e. \( \mathcal{D}(s_1) = \mathcal{D}(s_2) \Rightarrow \mathcal{O}(s_1) = \mathcal{O}(s_2) \)). Correctness gives that if \( \mathcal{D} \) identifies two statements then \( \mathcal{O} \) will certainly identify them. The following example shows that the reverse does not hold. \( \mathcal{O} \) identifies more statements than \( \mathcal{D} \).

\textbf{Example 5.21} Take \( s_1 = c_1 \) and \( s_2 = c_2 \) then

\[
\begin{align*}
\mathcal{O}(s_1) &= \mathcal{O}(s_2) \\
\mathcal{D}(s_1) &\neq \mathcal{D}(s_2)
\end{align*}
\]

That \( \mathcal{O}_b \) lies in between \( \mathcal{O}_m \) and \( \mathcal{D} \) can be seen from the lemmas 5.15 and 5.18. Likewise, that \( \mathcal{O}_m \) lies in between \( \mathcal{O} \) and \( \mathcal{O}_b \) can be seen from the lemmas 5.12 and 5.15. For each step there are easy examples that show that the inclusions are strict.

\[
\begin{tikzpicture}
\node (D) {\( \mathcal{D} = \mathcal{O}^* \)};
\node (abs3) [below of=D, yshift=-1cm] {\( \text{abs}_3 \)};
\node (O_b) [below of=abs3, yshift=-1cm] {\( \mathcal{O}_b \)};
\node (abs2) [below of=O_b, yshift=-1cm] {\( \text{abs}_2 \)};
\node (O_m) [below of=abs2, yshift=-1cm] {\( \mathcal{O}_m \)};
\node (abs1) [below of=O_m, yshift=-1cm] {\( \text{abs}_1 \)};
\node (O) [below of=abs1, yshift=-1cm] {\( \mathcal{O} \)};
\draw[->] (D) -- (abs3);
\draw[->] (abs3) -- (O_b);
\draw[->] (O_b) -- (abs2);
\draw[->] (abs2) -- (O_m);
\draw[->] (O_m) -- (abs1);
\draw[->] (abs1) -- (O);
\end{tikzpicture}
\]

This completes the comparative semantics for the combination of the local interpretation of non-determinism and the local interpretation of probability.

6 Local non-determinism and global probability

The goal in this section is to give the probabilistic choice a global interpretation. The probabilistic choice will wait to see what the environment has to offer before committing itself to an option. Or looking at it differently, it is possible to
backtrack over a coin toss if the choice made would result in deadlock. For the canonical example
\[ s = (c_1 + c_2)(c_1 + \rho c_2) \]
this would yield the following. The statement \( s \) has two non-deterministic alternatives, synchronizing on \( c_1 \) or synchronizing on \( c_2 \). Both options will give \( \tau \) (with probability one). In this example there is no longer any probability. The global probabilistic choice is determined by the local non-deterministic choice. The probabilistic choice is delayed until after the non-deterministic choice is resolved. Options that would fail (result in direct deadlock) are left out of consideration.

6.1 Transition system and operational semantics

The transition system that was introduced in the previous section first resolved the non-deterministic choice and then the probabilistic choice. This means that the probabilistic choice already waits for the environment to make its (non-deterministic) choices. The only thing that remains to be done is to remove the unmatched synchronization actions. The difference with the previous section is created by the way these actions are removed. Instead of being interpreted as failure, the unmatched actions are interpreted as possibilities that are not used. The transition system \( T_p \) can be used again.

The unmatched actions are removed. If other alternatives exist then an unmatched action does not cause deadlock but is simply ignored. The probabilistic choice will avoid deadlock. If the sum of probabilities of allowed actions is less than one, the missing probability should not be interpreted as deadlock but as an option that will be avoided. The notation \( s \Rightarrow_o S \) introduced in the previous section specifies that \( S \) contains all probabilistic options of one non-deterministic option of \( s \). If \( s \Rightarrow_o [\beta_i, \langle \beta, s \rangle_j, \kappa] \) the available options for \( s \) are doing \( \beta_i \) (for some \( i \in I \)) or doing \( \beta_j \) and continuing with \( s_j \) (for some \( j \in J \)). The chance that an option that deadlocks is chosen is \( \kappa \) i.e. the missing probability is \( \kappa \). Since deadlocking options will be avoided the probability of the other options should be normalized to one. The normalization can be done by dividing by \( 1 - \kappa \).

The models given here are again generative. A probability is assigned to an action. The level of non-determinism comes on top of the probabilistic level. This is the same construction as was used in section 5. The resulting domains are the same. For the branching denotational domain the interpretation is different, as described in the next subsection. For the other domains also the interpretation remains the same.

The operational semantics will be given using a linear domain. The domain is the same as the domain \( P_l \) introduced in subsection 5.3.
Definition 6.1

\[ \mathcal{O}(s) = \bigcup_{x \in \mathcal{O}(S)} \mathcal{O}(s) \]

\[ \mathcal{O}(s) = \{ \lambda B \cdot \Delta_{i \in B} \} \text{ if } \kappa = 1, \text{ otherwise} \]

\[ \mathcal{O}(s) = \{ \mu J \in \mathcal{O}(s) \} \cdot \{ \mu J \in \mathcal{O}(s) \} \]

This definition is similar to the definition 5.5. There are only small differences that are caused by the different interpretation of unmatched synchronization actions and missing probability. The probabilities are normalized and there is no deadlock if other options exist. If no other options exist, then deadlock will also occur in this model.

6.2 Denotational semantics, relating \( \mathcal{O}(s) \) and \( \mathcal{D} \)

It is not possible to know which synchronization attempts will succeed and which will fail without knowing the environment. Therefore the denotational semantics maintains the unmatched synchronization actions in the outputs. This means that unmatched synchronization actions can not cause deadlock. The other situation resulting in deadlock in the previous section was missing probability.

In section 5 this missing probability was interpreted as a probability of deadlock. Here, however, such deadlock would be avoided by the probabilistic choice. Instead of the missing probability being interpreted as deadlock, the probability that is present should be interpreted as relative to the total probability present. If the only action available is \( a \) labeled with \( \frac{1}{2} \) then the probability of \( a \) happening is \( 1 \). If there are two actions, \( a \) labeled with \( \frac{1}{2} \) and \( b \) labeled with \( \frac{1}{2} \), then the probability of \( a \) occurring would be \( \frac{1}{2} \) and the probability of \( b \) occurring would be \( \frac{1}{2} \).

Other than the different interpretation of multisets of labeled actions, the denotational domain is the same as in section section 5. The semantical operators are also the same as the operators introduced in definition 5.7. By choosing an initial label of 1 for the actions, the function \( \mathcal{D} \) is the same as in the previous section, as it is built from the same base using the same operators. The denotational semantics gives the same result as in in the previous section but this result is interpreted differently.

The comparison of the operational and denotational model is similar to the comparison done in the previous section. An operational-like model is defined on the mixed time domain \( \mathbb{P}_m \), on the branching time domain \( \mathbb{P}_b \) and on the domain \( \mathbb{P}_d \). Suitable abstraction functions are used to relate each of these models. Finally the operational-like model on the denotational domain is shown to coincide with the denotational semantics.

\(^{1}\)The interpretation of labels being relative to the sum of all labels leaves the initial value undetermined. Starting with an initial label of, for instance, \( \frac{1}{2} \) would yield the same interpretation of the resulting multiset.
The only step which is repeated here is the introduction of an abstraction function from the denotational to the branching operational domain. Since the interpretation of an element of the denotational domain is different than before, the abstraction function taking the denotational meanings to branching time operational meanings, will also be different. The interpretation of the operational domains has not changed, so the abstraction functions used before should still be valid. The only thing that changes is the operational-like models. The operational-like models will not be worked out here. The models can be constructed as is done in section 5, with a normalization added as with \( \mathcal{O}(\ell) \) above.

The abstraction function \( \text{abs}_4 \) replaces the abstraction function \( \text{abs}_3 \)

**Definition 6.2** \( \text{abs}_4 : \mathbb{P}_d \to \mathbb{P}_b \) is given by

\[
\text{abs}_4(p) = \{ \text{abs}_4(q) \mid q \in p \}
\]

\[
\text{abs}_4(q) = \begin{cases} 
\{ [1 + \delta] \} & \Pi(q) = 0 \\
\{ [\beta] \Pi(q) \mid \beta \in q \} \cup \{ [\beta] \Pi(q), \text{abs}_4(p) \} & \langle \beta, p \rangle \in q \}
\end{cases}
\]

where \( \Pi(q) \) as in definition 5.17.

The abstraction function removes the unmatched synchronization actions and normalizes the probabilities. Deadlock is introduced where needed.

The following can be shown by introducing the operational-like semantics as described above.

**Theorem 6.3** The operational semantics is an abstraction of the denotational semantics.

**Proof** Define \( \text{abs} = \text{abs}_1 \circ \text{abs}_2 \circ \text{abs}_4 \) then \( \mathcal{O}(\ell) = \text{abs} \circ \mathcal{D} \) can be shown in the same way as was done in section 5. \( \square \)

The result obtained in theorem 6.3 gives that \( \mathcal{D} \) is correct with respect to \( \mathcal{O}(\ell) \). The different distinguishing powers of the models are related in the same way as before. \( \mathcal{O}(\ell) \) identifies more statements than \( \mathcal{O}_m(\ell) \) which identifies more statements than \( \mathcal{O}_b(\ell) \). The denotational semantics, \( \mathcal{D} \), identifies the least number of statements.

As the denotational model is the same as in the previous section, both models certainly identify exactly the same statements. For the operational models this is not the case. The operational model introduced in this section identifies more statements than the operational model introduced in the previous section. The following example shows why.

**Example 6.4** Let \( s_1 = b \) and \( s_2 = b + \frac{1}{2} c \) then

\[
\mathcal{O}(s_1) \neq \mathcal{O}(s_1) \\
\mathcal{O}(\ell)(s_1) = \mathcal{O}(\ell)(s_1).
\]

The statements \( s_1 \) and \( s_2 \) are identified by \( \mathcal{O}(\ell) \) since the extra option \( c \) would deadlock and is therefore ignored.
That the operational model $\mathcal{O}^{(g)}$ identifies all the statements that the operational model of the $\mathcal{O}$ identifies can be seen by introducing an abstraction function that removes $\delta$ if there are other options and normalizes the probabilities.

$$D = \mathcal{O}^*$$

This completes the comparative semantics for the combination of the local interpretation of non-determinism and the global interpretation of probability.

### 7 Global non-determinism and local probability

The only difference between the two models presented so far is how they deal with failed synchronization attempts. The transition systems and denotational models even coincide. This close relation is a consequence of the decision to resolve non-determinism and then resolve probability. A similar close relation cannot be expected when the non-determinism is interpreted globally instead of the probability. This would require the probability to be somehow resolved first. This conflicts with the choice of resolving non-determinism first.

When probability is resolved before non-determinism then at the moment the probability is resolved it is not always known what the next action will be. The following example shows this.

$$a + \rho (b + c)$$

A probability has to be assigned to the process $b + c$ instead of to a single action. The use of a generative model seems inappropriate. A stratified model, which assigns probability to processes, seems more natural.

In the canonical example\(^2\)

$$(c_1 + c_2) | (c_1 + \rho c_2)$$

\(^2\)Although $|$ is not explicitly present in $\mathcal{L}_p$, it expresses part of the behavior of $|$. 

36
there are two (probabilistic) options. Synchronize on $s_1$ with probability $\rho$ or synchronize on $s_2$ with probability $1 - \rho$.

In the example there is no non-determinism: the local probabilistic choice determines the global non-deterministic choice. A non-deterministic choice is delayed until the probabilistic choice has been resolved. Options that would fail (result in direct deadlock) are left out of consideration.

7.1 The transition system

To be able to describe the assignment of probabilities to processes, making a probabilistic decision is viewed as an explicit action. The probability will occur as an action in the transition system. A statement $s_1 + \rho \cdot s_2$ can take a $\rho$ transition to $s_1$ or a $1 - \rho$ transition to $s_2$. In general, if a statement can make $\rho$ transition to $r$ this means that a probabilistic choice has to be made and with probability $\rho$ the statement will behave like $r$ after this choice.

The configurations for the transition system are the same as before. The observations are different. As probabilities are no longer assigned to actions, the actions will no longer have a label giving their probability. Instead probabilities occur as transitions themselves. The non-deterministic choice does not need to be made explicit by the special action any more. The role of the $\triangleright$ (dividing the possible actions into groups) is now incorporated into the $\rho$ transitions.

$$
Obs = Act \cup [0, 1]
$$

In the following transition system negative premises are used several times. Correctness of the transition system can be shown using induction on the weight function $wgt$.

Definition 7.1 The transition system $T_{ps}$ is given by $T_{ps} = (Decl \times Res, Act \cup [0, 1], \rightarrow, Spec)$. A transition $((D|r), a, (D|r')) \in \rightarrow$ is written as $r \xrightarrow{a} r'$. Spec is given in two parts. The first part concerns the transitions labeled with actions.

- $a \xrightarrow{\triangleright} E$  \hspace{1cm} (Act)
- $s_1 \xrightarrow{a} r$
  - $x \xrightarrow{a} r$
  - $s_1; s_2 \xrightarrow{a} r; s_2$
    - if $D(x) = s_1$  \hspace{1cm} (Rec)
    - if $s_1 \xrightarrow{D(x)} r_1$
  - $s_1 \xrightarrow{a} r$
    - if $s_1 \xrightarrow{a\rho} r_1$
    - if $s_1 \xrightarrow{a\rho} r_1$

37
The second part concerns the transitions labeled with a probability $\rho \in [0, 1]$.

- $s_1 \xrightarrow{\rho} r$
  
  $s_1 \parallel s_2 \xrightarrow{\rho} r \parallel s_2$  
  if $s_2 \xrightarrow{\rho} s_1 \parallel r$  
  if $s_2 \xrightarrow{\rho} s_1 \parallel r$

- $s_1 \xrightarrow{\rho} r_1 \quad s_2 \xrightarrow{\rho} r_2$
  
  $s_1 \parallel s_2 \xrightarrow{\rho} r_1 \parallel r_2$

Note the reversal of the roles of ‘+’ and ‘\parallel’ compared with $T_p$. A single $a$ takes an $a$ transition and is done. There is no probability label. Recursion is handled by body replacement.

A probabilistic choice is resolved by a transition, labeled with the associated probability, to one of the alternatives. The probabilistic choice is resolved directly and no longer waits for the non-determinism to be resolved.

Resolving non-deterministic is no longer made explicit by the use of an $\iota$ action. Instead, resolving the non-determinism is done implicitly by the transition system. For a statement $s$ there may be more than one outgoing action transition. The non-determinism is resolved only after all probability (for the first step) has been resolved. This is expressed by the rules (Choice $\rho$)

Probability being resolved propagates through the non-deterministic choice as expressed by the rules (Choice $\rho_1$) and (Choice $\rho_2$). The probability on the left hand side is resolved first. Similar remarks hold for parallel composition.
The rules for sequential composition and recursion have the same form as before. Both rules have been split in two parts, one part dealing with actions the other with probabilities.

The transition is defined using negative premises. Showing the transition system is still well defined can easily be done by induction on wgt. For each rule in the transition system the weight of the left hand side of the conclusion is more than the weight of all the left hand sides of the premises.

### 7.2 Properties of the transition system

Analogous to the definition given for the generative transition system some notation is given. Note again the reversal of the roles of non-determinism and the probability also visible in this definition.

How often a probabilistic transition occurs is important. When combining consecutive probabilistic choices, the number of occurrences of a combined transition should be remembered. This makes the following definition more complicated than the corresponding one in section 5.

**Definition 7.2** For any statement s the following notation is introduced.

(a) That s can take one or more σ steps and end up in s′ in n different ways is denoted by s^0→_n s′. To be precise, when

1. s ^0→_1 s′
2. s ^0→_m s″ and s″ ^0→_k s with ρ = ρ_1 · ρ_2,

then s ^0→_[(m+k)] s′. If only 1 holds then s ^0→_1 s′ and if only 2 holds then s ^0→_[(m+k)] s′.

(b) If s can take zero or more σ steps and end up in s′ then s ^0→_n s′ is written, i.e. s ^0→_n s′ when ρ = 1, n = 1 and s ≡ s′ or when s ^0→_n s′.

(c) Also, s ^∞→_n s′ is used to denote that s′ is a normal form of s with respect to σ steps, i.e. s ^∞→_n s′ when s ^0→_n s′ and s′ ^0→_r.

(d) The successor (multi)set of s,

\[ S(s) = \{ n \cdot (\rho \cdot \{ a \mid s' \xrightarrow{r} s'' \} \cup \{ (a', s'') \mid s'' \xrightarrow{r} s''' \}) \mid s ^0→_n s' \}. \]

(e) If s ^0→_r, i.e. if there are no r, b such that s ^0→_r, then s is said to block.

The transitions of a statement are divided into a multiset of sets. Each set describes the non-deterministic alternatives for a single probabilistic alternative. If it turns out that a non-deterministic alternative is not possible (would
deadlock) then it is left out of consideration. Using the generative transition system there were two possible ways to obtain deadlock.

The first was by causing missing probability. This does not occur using the stratified transition system of this section. For a statement there are two possibilities. A statement (is non-probabilistic and) has the choice of several actions or the statement is probabilistic; it can choose between two probabilistic transitions. The probabilistic transition are \( \rho \) and \( 1 - \rho \) for some \( \rho \), so their sum is always one.

The second way to obtain deadlock was an unmatched synchronization action. If no other (non-deterministic) option is available, an unmatched synchronization action will also cause deadlock in the operational model given in section.

Lemma 7.3 For \( T_{ps} \) the following holds:

(a) If \( s \xrightarrow{\rho} s' \) then \( \text{wgt}(s') < \text{wgt}(s) \).

(b) \( T_{ps} \) is strongly normalizing for \( \rho \)-steps (no infinite sequence \( s \xrightarrow{\rho} s_1 \xrightarrow{\rho} s_2 \ldots \) exists).

(c) \( T_{ps} \) is finitely branching, that is, for all \( s \in \mathcal{L}_p \):
   1. \( S(s) \) is a finite multiset and,
   2. \( S \) is a finite set for each \( p \ast S \in S(s) \).

(d) The sum of probabilities of all probabilistic alternatives for each statement is one.

\[
\sum_{s \rightarrow_{\varnothing_n} s'} n \cdot \rho = 1
\]

Proof

(a) Clear by induction on \( \text{wgt}(s) \).

(b) Direct consequence of (a).

(c) The first part is clear using (b). The second part follows by induction on \( \text{wgt}(s) \).

(d) Part (b) allows the use of induction to the maximum number of \( \rho \) steps required to reach a normal form \( s' \). If no steps are required then \( \rho \) and \( n \) are one by definition of \( \mathcal{L}_n \). Otherwise use that if \( s \xrightarrow{\rho} s_1 \) then also \( s \xrightarrow{1-\rho} s_2 \) for some \( s_2 \) and these are the only transitions for \( s \) (which can be proven by induction \( \text{wgt}(s) \)). This gives:

\[
\sum_{s \rightarrow_{\varnothing_n} s'} n \cdot \rho = \sum_{s \rightarrow_{\varnothing_m} s'} m \cdot \rho_1 \cdot \sigma + \sum_{s \rightarrow_{\varnothing_k} s'} k \cdot (1 - \rho_1) \cdot \sigma
\]
This equality still holds if \( s_1 \) and \( s_2 \) (or any of their derivatives) are the same since the multiplicity of transitions is accounted for.

With these properties in place the operational semantics can be found from the transition system. First the domains which are used to define the operational and denotational semantics are defined.

### 7.3 Domains for stratified models

In the domains for the generative models the level describing probability is built directly on top of the actions. In a stratified model probability is assigned to a, possibly non-deterministic, process instead of to an action. To allow assigning a probability to a non-deterministic process, the probability level is built onto the level describing non-determinism. The non-deterministic level is built upon the actions.

<table>
<thead>
<tr>
<th>stratified domain</th>
<th>example(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>top level (( \mathbb{P} ))</td>
<td>( \mathcal{M}_f(\mathbb{Q}) )</td>
</tr>
<tr>
<td>middle level (( \mathbb{Q} ))</td>
<td>( \mathcal{P}_{ncc}(\mathbb{R}) )</td>
</tr>
<tr>
<td>action level (( \mathbb{R} ))</td>
<td>( A + A \times \mathbb{P} )</td>
</tr>
</tbody>
</table>

Structure of a (branching) stratified domain.

Compared with the structure of a (branching) generative domain, given in subsection 5.3, the top and middle level have been reversed.

The following domain equations describe a branching domain for the stratified operational semantics. A branching instead of a linear domain is used since it is, at present, not clear how to abstract away the moments of choice in the stratified domain. (See section 8 for some comments on this point.)

\[
\mathbb{P}_c \simeq \mathcal{M}_f([0,1] \times \mathbb{Q}_c^d) \\
\mathbb{Q}_c^d \simeq \mathcal{P}_{ncc}(\mathbb{R}_c^d) + \{ \delta \} \\
\mathbb{R}_c \simeq \text{IAct} \cup \text{IAct} \times \text{id}_{\mathbb{P}}(\mathbb{P}_c^d) .
\]

In the domain equations the functor \( \mathcal{P}_{ncc}(\cdot) \) is used as before. The space of finite sets is insufficient for the same reason.

The stratified model is used for modeling the global interpretation of non-determinism. Synchronization will only be attempted if it will succeed. The only possibility for deadlock is if there are no allowed actions whatsoever. In the operational model this is possible since only the internal actions are allowed. In the denotational model, however, all actions, also unmatched synchronization, actions are allowed. In the denotational model there is no deadlock. The following domain equations describe the domain used for the stratified denotational semantics,

\[
\mathbb{P}_d \simeq \mathcal{M}_f([0,1] \times \mathbb{Q}_d^d) \\
\mathbb{Q}_d^d \simeq \mathcal{P}_{ncc}(\mathbb{R}_d^d) \\
\mathbb{R}_d \simeq \text{Act} \cup \text{Act} \times \text{id}_{\mathbb{P}}(\mathbb{P}_d^d) .
\]
In the next subsections these domains are used for a comparative semantics for the language $L_p$ with global interpretation of non-deterministic choice and local interpretation of probabilistic choice.

### 7.4 Operational semantics

For the operational semantics the branching time domain $P_s$ as introduced in the previous subsection is used. A branching time instead of a linear time domain is used since it is, at present, not clear how to abstract away the moments of choice in the stratified domain. Some more remarks about the question of linearizing the stratified domain are made in section 8.

**Definition 7.4** The function $O^{(g)} : L_p^+ \rightarrow P_c^s$ is given as

$$O^{(g)}(s) = \{ n \cdot (\rho * \tilde{O}^{(g)}(s')) | s \sim_n s' \}$$

$$\tilde{O}^{(g)}(s) = \begin{cases} \{ \delta \} & \text{if } s \text{ blocks} \\ \{ b | s \not\xrightarrow{b} E \} \cup \{ (b, O^{(g)}(s')) | s \xrightarrow{b} s' \} & \text{otherwise.} \end{cases}$$

The set $\tilde{O}^{(g)}(s)$ is finite (lemma 7.3), so it is certainly compact. The multiset $O^{(g)}(s)$ is finite (lemma 7.3). Note that the number of elements of $O^{(g)}(s)$ depends only on the number of successors of $s$, and not on $\tilde{O}(s)$. This is important for the contractiveness of the higher-order function which can be used to justify this definition.

### 7.5 Denotational semantics

The domain used for the denotational semantics is $P_d^s$ as introduced in subsection 7.3. This domain is also a branching time domain. For the denotational model this is required. As argued before the only way of obtaining deadlock in this stratified model is by unmatched synchronization actions. Since the unmatched synchronization actions are still included, there is no deadlock in the denotational semantics.

The semantical counterparts of the syntactical constructs `+', `+\rho', `;' and `||' are given below.

**Definition 7.5** All semantical operators are elements of $O_P = P_d^s \times P_d^s \xrightarrow{1} P_d^s$.

(a) The operator $+ \in O_P$ is defined by:

$$p_1 + p_2 = \{ \rho \sigma * (q_1 + q_2) | \sigma * q_1 \in p_1, \rho * q_2 \in p_2 \}$$

$$q_1 +\rho q_2 = q_1 \cup q_2.$$

(b) The operator $+\rho \in O_P$ is defined by:

$$p_1 +\rho p_2 = \{ \rho \sigma * q | \sigma * q \in p_1 \} \cup \{ (1 - \rho) \sigma * q | \sigma * q \in p_2 \}$$

42
(c) The operator \( ; \in \mathcal{O}_p \) is defined by:

\[
\begin{align*}
p_1 ; p_2 &= \{ \rho \ast q ; p_2 \mid \rho \ast q \in p_1 \} \\
q_1 ; p &= \{ \langle a, p \rangle \mid a \in q \} \cup \{ \langle a, p' ; p \rangle \mid \langle a, p' \rangle \in p \}
\end{align*}
\]

(d) The operator \( \parallel \in \mathcal{O}_p \) is defined by:

\[
\begin{align*}
p_1 \parallel p_2 &= \{ \rho \ast q \parallel q_2 \mid \rho \ast q_1 \in p_1 , \rho \ast q_2 \in p_2 \} \\
q_1 \parallel q_2 &= \{ \langle a, \{ 1 \ast q \} \rangle \mid a \in q \} \cup \{ \langle a, \rho \parallel q \ast q_2 \rangle \mid \langle a, \rho \rangle \in p \} \\
q_1 \parallel q_2 &= \{ \langle \tau, p_1 \parallel p_2 \rangle \mid \langle \tau, p_1 \rangle \in q_1, \langle \tau, p_2 \rangle \in q_2 \} \\
&\quad \cup \{ \langle \tau, q_1 \rangle \mid \tau \in c_1, c \in q_2 \} \\
&\quad \cup \{ \tau \mid c \in q_1, c \in q_2 \}
\end{align*}
\]

If compared with the operators introduced in definition 5.7, the reversal of the roles of probability and non-determinism is visible in two ways. The levels of probability (multisets of labeled objects) and non-determinism (sets of objects) have been reversed for all the operators. The role of the + and +_\rho have also been reversed. Roughly speaking '+_\rho' is union on the first level while '+' is union on the second level. For \( p_1 \parallel p_2 \) the first level of choice is resolved both in \( p_1 \) and \( p_2 \). Next '||' and ']' are used to find all non-deterministic options. Note that \( q_1 || q_2 \) may be empty but \( q_1 \parallel q_2 \) is not. The set \( q_1 || q_2 \) is non-empty.

**Definition 7.6**  The denotational semantics \( \mathcal{D}^{(s)} \colon \mathcal{L}_p \rightarrow \mathbb{P}_d \) is given by

\[
\begin{align*}
\mathcal{D}^{(s)}(a) &= \{ 1 \ast \{ a \} \} \\
\mathcal{D}^{(s)}(x) &= \mathcal{D}^{(s)}(\mathcal{D}(x)) \\
\mathcal{D}^{(s)}(s_1 \mathcal{O}_p s_2) &= \mathcal{D}^{(s)}(s_1) \mathcal{O}_p \mathcal{D}^{(s)}(s_2).
\end{align*}
\]

A single action \( a \) acts like \( a \) with probability one. Recursion is handled by body replacement and the semantical operator \( \mathcal{O}_p \) is used to give the meaning of any statement built using the syntactic construct \( \mathcal{O}_p \). The operator \( \mathcal{O}_p \) is one of operations '+', '+_\rho', ']' or '||'.

### 7.6 Comparing \( \mathcal{O}^{(s)} \) and \( \mathcal{D}^{(s)} \)

The comparison of the operational model and the denotational model is less work than before since the operational model is also defined on a branching time domain. To compare the operational and denotational model an intermediate operational-like semantics \( \mathcal{O}^{(s)} \) is defined on the denotational domain \( \mathbb{P}_d \). That the operational semantics is an abstraction of the denotational semantics then follows using the following two facts. The operational semantics is an abstraction of this intermediate semantics. The intermediate semantics coincides with the denotational semantics.

Define \( \mathcal{O}^{(s)} \) on the domain \( \mathbb{P}_d \) as follows.

---

43
Definition 7.7 The function $O^{[*]} : \mathcal{L}^+ \rightarrow \mathbb{P}^s$ is given by

$$
O^{[*]}(s) = \{ n \cdot (\rho \ast \hat{O}^{[*]}(s')) | s \xrightarrow{\mathcal{L}^+_n} s' \} \\
\hat{O}^{[*]}(s) = \{ a | s \xrightarrow{a} E \} \cup \{ (a, O^{[*]}(s')) | s \xrightarrow{a} s' \}
$$

As usual this definition can be justified by showing that $O^{[*]}$ is the unique fixed point of a higher-order operator. The intermediate model keeps unmatched synchronization actions. Like in the denotational model there is no deadlock.

The only difference between the domains $\mathbb{P}^s_a$ and $\mathbb{P}^s_e$ is that in the first domain the unmatched synchronizations are still present and there is, therefore, no deadlock. The two domains can be related by the following abstraction function that removes unmatched synchronization actions and introducing deadlock where needed.

Definition 7.8 The function $\text{abs}^{[*]} : \mathbb{P}^a \rightarrow \mathbb{P}^e$ is defined as:

$$
\text{abs}^{[*]}(p) = \{ \rho \ast \text{abs}^{[*]}(q) | \rho \ast q \in p \} \\
\text{abs}^{[*]}(q) = \left\{\begin{array}{ll}
\{ b | b \in q \} \cup \{ \langle b, \text{abs}^{[*]}(p) \rangle | \langle b, p \rangle \in q \} & \text{if this set is non-empty}
\end{array}\right. \text{ otherwise.}
$$

The operational semantics $O^{[*]}$ is an abstraction of the intermediate semantics $O^{[*]}$. If from $O^{[*]}(s)$ the unmatched synchronization actions are removed then $O(s)$ is obtained.

Lemma 7.9 $O(s) = \text{abs}^{[*]}(O^{[*]}(s))$.

Since the intermediate operational model $O^{[*]}$ is also only defined on $\mathcal{L}$ there is no need to extend the denotational semantics. As in previous sections the proof that $O^{[*]} = \mathcal{D}^{[*]}$ is done by showing that $\mathcal{D}^{[*]}$ is a fixed point of the higher-order operator implicitly used to define $O^{[*]}$.

Lemma 7.10 $\mathcal{D}^{[*]}$ is the fixed point of $\Phi^*$ i.e. $\Phi^*(\mathcal{D}^{[*]}) = \mathcal{D}$.

Proof By induction on $\text{wgt}(s)$. Only the two most interesting cases are given.

$$
\mathcal{D}^{[*]}(s_1 +_\rho s_2) = \mathcal{D}^{[*]}(s_1) +_\rho \mathcal{D}^{[*]}(s_2) \\
= \Phi^*(\mathcal{D}^{[*]})(s_1) +_\rho \Phi^*(\mathcal{D}^{[*]})(s_2) \\
= \{ n \cdot (\rho \ast \hat{\Phi}^*(\mathcal{D}))(s') | s_1 \xrightarrow{\mathcal{L}^+_n} s' \} \\
\cup \{ m \cdot ((1 - \rho) \sigma \ast \hat{\Phi}^*(\mathcal{D}))(s') | s_1 \xrightarrow{\mathcal{L}^+_m} s' \} \\
= \{ k \cdot (\sigma \ast \hat{\Phi}^*(\mathcal{D}))(s') | s_1 +_\rho s_2 \xrightarrow{\mathcal{L}^+_k} s' \} \\
= \Phi^*(\mathcal{D}^{[*]})(s_1 +_\rho s_2) \\
\mathcal{D}^{[*]}(s_1 + s_2) = \mathcal{D}^{[*]}(s_1) + \mathcal{D}^{[*]}(s_2)
$$
\[ F^*(D_{(g')})(s_1) + F^*(D_{(g')})(s_2) \]
\[ = \{ \| n \cdot (\rho \ast F^*(D))(s_1') \| \mid s_1 \sim_{h} s_1' \} \]
\[ + \{ \| m \cdot (\sigma \ast F^*(D))(s_2') \| \mid s_2 \sim_{h} s_2' \} \]
\[ = \{ n m \cdot (\rho \ast F^*(D))(s_1') \cup (\sigma \ast F^*(D))(s_2') \mid s_1 \sim_{h} s_1', s_2 \sim_{h} s_2' \} \]
\[ = \{ k \cdot (\rho \ast F^*(D))(s_1') \mid s_1 + s_2 \sim_{h} s' \} \]
\[ = F^*(D_{(g')})(s_1 + s_2). \]

Combining the results of this subsection gives that the operational semantics is an abstraction of the denotational semantics.

**Theorem 7.11** \( O_{(g')} = \text{abs} \circ D_{(g')} \)

**Proof** \( O_{(g')} = [\text{lemma 7.9}] \text{abs} \circ O^*(g') = [\text{lemma 7.10}] \text{abs} \circ D_{(g')} \).

The theorem above gives correctness of \( D \) with respect to \( O_{(g')} \). That \( O_{(g')} \) identifies more statements than \( D \) is clear from the same example as before, \( s_1 = c_1 \) and \( s_2 = c_2 \).

No relation exists between the distinguishing power of the models in this section and the models in the previous sections. The first part of the example below shows that there are statements identified in all the models in section 5 and 6 but not by any of the models in this section. The second part of the example shows that the reverse is also the case.

**Example 7.12** Let in this example the actions \( a, b, c \) all be distinct elements of \( I \). 

1. Let \( s_1 = (a + \rho)(b + c) \) and \( s_2 = (a + \rho b) + (a + \rho c) \). The statements \( s_1 \) and \( s_2 \) are identified by \( D \) from section 5,

\[ D(s_1) = \{ \| \rho \ast a, (1 - \rho) \ast a \|, \| \rho \ast a, (1 - \rho) \ast b \| \} = D(s_2). \]

Since the statements are identified by \( D \), they are certainly identified by both \( O \) of section 5 and \( O_{(g)} \) of section 6. The operational semantics \( O_{(g')} \), and therefore also the denotational semantics \( D_{(g')} \), distinguishes between \( s_1 \) and \( s_2 \),

\[ O_{(g')}(s_1) = \{ \| \rho \ast a \}, (1 - \rho) \ast \{ b, c \} \} \]
\[ O_{(g')}(s_2) = \{ \| \rho^2 \ast a \}, (1 - \rho) \ast \{ a, c \}, \rho(1 - \rho) \ast \{ b, a \}, (1 - \rho)^2 \ast \{ b, c \} \} \]

2. Let \( s_1 = (a + \rho b) + c \) and \( s_2 = (a + c) + \rho (b + c) \). \( s_1 \) and \( s_2 \) are identified by the denotational semantics \( D_{(g')} \) of this section.

\[ D_{(g')}(s_1) = \{ \| \rho \ast \{ a, c \}, (1 - \rho) \ast \{ b, c \} \} = D_{(g')}(s_2). \]
The operational semantics $O^{(lg)}$ distinguishes between $s_1$ and $s_2$.

$$
O^{(lg)}(s_1) = \{ \rho \ast a, (1-\rho) \ast b, 1 \ast c \}
$$

$$
O^{(lg)}(s_2) = \{ \rho \ast a, (1-\rho) \ast b, \rho \ast a, (1-\rho) \ast c, \\
\rho \ast c, (1-\rho) \ast b, 1 \ast c \}
$$

This completes the comparative semantics for the combination of the local interpretation of non-determinism and the local interpretation of probability.

8 Conclusions

In this report comparative semantics for three different interpretations of the language $\mathcal{L}_p$ are given. There are several open questions yet to be answered. The first one concerns the semantics in section 7. The question is how to construct a linear time semantics based on the stratified transition system. A linear time domain that might be used for this is one of the form given below.

$$
\mathbb{P} = \mu(\mathbb{Q})
$$

$$
\mathbb{Q} = \mathcal{P}_{ncc}(\mathbb{R})
$$

$$
\mathbb{R} = I\text{Act}^c
$$

The main problem when going from a transition system to a meaning in this domain is how to divide the non-determinism. This is illustrated by the following example.

Suppose that in the transitions system, perhaps after doing some $\rho$ steps, there are two non-deterministic alternatives $a; s_1$ and $a; s_2$, both starting with the same action $a$. Given a measurable set $B \in \mathcal{B}(\mathbb{Q})$ clearly only the $q \in B$ that have sequences that start with $a$ contribute to the probability of something from $B$ happening. How the sequences in such a $q$ should be distributed over $s_1$ and $s_2$ is not clear.

Another open question is the combination of global probability with global non-determinism. Looking at the usual example

$$(c_1 + c_2)(c_1 + \rho c_2)$$

it is not clear what should happen. Using the interpretation that a global choice waits for the environment to make its choices, nothing would happen in
this example. Both the probabilistic choice and the non-deterministic choice would wait for the other choice to make a decision before proceeding.

The interpretation above results in unexpected deadlocks. The statement above did not deadlock in the models in section 6 nor in the models from section 7. Providing more flexibility in avoiding deadlock by making more choices global should certainly not result in more statements deadlocking.

The interpretation of a global choice as a choice that may be backtracked over if the chosen option fails, would yield a different result in this case. One of the choices would make a decision and the other would follow it. But then, which choice would get to make the decision? Several viable options exist but all have their drawbacks.

A model for both global non-determinism and global probability should somehow allow both probability and non-determinism to be resolved first, depending on the situation. Looking at the statements given in 7.12 this model would probably have to distinguish between $s_1$ and $s_2$ in both cases, yielding a model which identifies less statements than both the stratified and generative models given in this report.

A possible model for local probability with local non-determinism is given in section 5, this model is very close to the model with global probability. The difference is in the interpretation of the transition systems, not in the transition systems themselves. By a different interpretation of the model with global non-determinism another model for local probability with local non-determinism can be found. Example 7.12 shows that this model is not equivalent to the model given in section 5. (The example does not use the fact that non-determinism is global.) Perhaps a model can be found for local interpretation of both non-determinism and probability that identifies more statements than the models in section 6 and 7.

Another, more technical question, is how the use of the auxiliary $\xi$ steps in the generative transition system can be avoided. The $\xi$ is used to distinguish alternatives belonging to different non-deterministic alternatives. A way to avoid using $\xi$ is to somehow join together the transitions that belong to one non-deterministic alternative. Bundling together transitions can be done by using hyper-transition systems as introduced in [10] and [11] or probabilistic automata as introduced in [27]. Both these methods could use a (more) systematic approach to defining the possible transitions, for instance, an adjusted form of transition system specifications (i.e. in the style of SOS [23]) would be an option.

As a final topic for further work, the probability allowed in the probabilistic choice operator in this report are in $\{0, 1\}$. Also allowing the probability to be 0 or 1 is not interesting for the local interpretation of probability. The behavior of $s_1 +_0 s_2$ is simply that of $s_2$. For the global interpretation, however, the use of zero probability could be used to express precedence. $s_1 +_0 s_2$ behaves like $s_2$ (with probability 1) if possible, but like $s_1$ if $s_2$ deadlocks. This way of expressing precedence is also used in [9] and in [29]. To be able to handle situations like $s_1 +_1 (s_2 +_1 s_3)$ (where $s_1$ has precedence over $s_2$ and $s_2$ has
precedence over $s_3$) the probabilistic choice would have to be fully stratified. Subsequence probabilistic choices can no longer be combined as is done in this report.

Acknowledgments

I would like to thank Erik de Vink for his large contribution to this work and Jaco de Bakker for his helpful comments on preliminary versions of this paper.

I would also like to thank the members of the Amsterdam Coordination Group (ACG) for their suggestions given during presentations on the subject of this paper.

References


