Distributing probability over nondeterminism

DANIELE VARACCA[1†] and GLYNN WINSKEL[2]

[1] *Department of Computing, Imperial College London.*
[2] *Computer Laboratory, Cambridge University.*

February 2006

We study the combination of probability and nondeterminism from a categorical point of view. In category theory, nondeterminism and probability are represented by suitable monads. Those two monads do not combine well, as they are. To overcome this problem we introduce the notion of indexed valuations. This notion is used to define a new monad that can be combined with the usual nondeterministic monad via a categorical distributive law. We give an equational characterization of our construction. We discuss the computational meaning of indexed valuations, and we show how they can be used by giving a denotational semantics of a simple imperative language.

1. Introduction

Nondeterminism and probability are computational effects whose semantics has been thoroughly studied. The combination of the two appears to be essential in giving models for concurrent processes (Vardi, 1985; Hansson, 1991; Segala and Lynch, 1995). Denotationally, nondeterminism is handled by the notion of powerdomain functor in a suitable category of domains (Plotkin, 1983), while probabilistic behaviour is handled by the powerdomain of valuations (Jones and Plotkin, 1989; Jones, 1990; Kirch, 1993). They happen to be monads, thus fitting the general idea, introduced by Moggi, of monads as models for computational effects (Moggi, 1991). As many other computational monads, they are freely generated from suitable equational theories (Plotkin and Power, 2002).

There are various ways of combining two monads. When they are freely generated from equational theories, we can first combine the theories in some way and then freely generate a new monad. In (Hyland et al., 2002) three main ways of combining theories are identified: sum, commutative combination, distributive combination. In the first case, the two equational theories are combined by joining the operations and the equations, without adding new equations. In the second case, one adds equations expressing that every operation of one theory commutes with every operation of the other theory. In the third case, one adds equations expressing distributivity of every operation of one theory over every operation of the other theory. This last approach can sometimes be followed more categorically using the notion of distributive law (Beck, 1969). The

[†] This work was carried out while the first author was PhD student at BRICS, Aarhus, DK.

leading example is given by the theory of abelian groups and the theory of monoids. Their distributive combination (distributing the monoid over the group) yields the theory of rings. The free ring monad can also be obtained by giving a categorical distributive law between the free abelian group monad and the free monoid monad.

The study of the operational semantics of systems combining probability and nondeterminism suggests that, in some cases, probabilistic choice should distribute over nondeterministic choice (Morgan et al., 1994; Bandini and Segala, 2001). As we will explain, there is no categorical distributive law between the nondeterministic monad and the probabilistic monad. Two solutions are possible at this point.

We can still form the distributive combination of the equational theories and generate a new monad. This is the path followed by Tix (Tix, 1999; Tix et al., 2005) and Mislove (Mislove, 2000) who, independently, define the notion of geometrically convex powerdomain $\mathcal{P}_{TM}$. When X is a real cone (a structure similar to a vector space), $\mathcal{P}_{TM}(X)$ is, roughly speaking, the set of all *convex* subsets of X. The nondeterministic choice is interpreted as union followed by convex closure. We will briefly recall this construction later.

The other possibility, the one we follow in this work, is to modify the definition of one of the monads, so as to allow the existence of a categorical distributive law. Analysing the reasons behind the failure of the distributive law, we are led to define the notion of an *indexed valuation*. Mathematically, indexed valuations arise as a free algebra for an equational theory obtained from the theory of valuations by removing one equation. In the category of sets, there exists a distributive law between indexed valuations and the finite nonempty powerset. Moreover, indexed valuations have an interesting concrete representation. The price we pay is that, since we modify the equational characterisation, indexed valuations do not satisfy all the laws usually satisfied by probability distributions.

Besides their categorical justification, indexed valuations have a computational meaning, which we present by giving semantics to an imperative language containing both random assignment and nondeterministic choice. The operational semantics is given in terms of probabilistic automata. Such a model comes equipped with a notion of a scheduler for resolving the nondeterminism. In the literature there are two main notions of scheduler: deterministic and probabilistic. Using indexed valuations, we give a denotational semantics which is adequate with respect to deterministic schedulers. A semantics in terms of the Tix-Mislove construction, instead, is adequate with respect to probabilistic schedulers.

Finally we briefly sketch the various possible ways to extend the construction above to some category of domains.

This work is part of the first author's PhD thesis (Varacca, 2003). An extended abstract appeared as part of (Varacca, 2002).

2. Background

In this section we outline some of the mathematical notions we need for our work. We assume a working knowledge of Category Theory (MacLane, 1971) and Universal Algebra (Cohn, 1981). All notions we need are dealt with in detail in Chapter 2 of (Varacca, 2003). The notation we use should be self-explanatory.

2.1. *Monads*

A *monad* on a category $\mathbf{C}$ is an endofunctor $T : \mathbf{C} \to \mathbf{C}$ together with two natural transformations, $\eta^T : Id_{\mathbf{C}} \to T$, the *unit*, and $\mu^T : T^2 \to T$, the *multiplication*, satisfying the following axioms:

— $\mu^T \circ T\eta^T = Id_T = \eta^T T \circ \mu^T$;
— $\mu^T \circ T\mu^T = \mu^T \circ \mu^T T$.

If T is a monad and if $f : X \to T(Y)$, the *Kleisli extension* $f^\dagger : T(X) \to T(Y)$ is defined as $T(X) \xrightarrow{T(f)} T(T(Y)) \xrightarrow{\mu^T_Y} T(Y)$. The *Kleisli Category* of the monad T, denoted by $\mathbf{C}_T$ has the same objects as $\mathbf{C}$, and $X \xrightarrow{f} Y$ in $\mathbf{C}_T$ if and only if $X \xrightarrow{f} T(Y)$ in $\mathbf{C}$. The identity is the unit of the monad, while composition is defined using the Kleisli extension. Monads can equivalently be defined using the Kleisli extension as a primitive notion and deriving the multiplication by $\mu^T_X := Id^\dagger_{T(X)}$.

In a category $\mathbf{C}$, an *algebra* for an endofunctor F is an object A together with a morphism $k : F(A) \to A$. An algebra for a monad (T, η^T, μ^T) is an algebra (A, k) for the functor T, satisfying the following compatibility axioms:

— $k \circ \eta^T_A = Id_A$;
— $k \circ T(k) = k \circ \mu^T_A$.

Algebras for a monad (T, η^T, μ^T) in $\mathbf{C}$ form a category $\mathbf{C}^T$, where a morphism $(A, k) \xrightarrow{\phi} (A', k')$ is given by a morphism $A \xrightarrow{\phi} A'$ such that $\phi \circ k = k' \circ T(\phi)$.

Every adjunction $(F, G, \eta, \epsilon) : \mathbf{C} \to \mathbf{D}$ generates a monad on $GF : \mathbf{C} \to \mathbf{C}$, with $\eta^{GF} := \eta$ and $\mu^{GF} := G\epsilon F$. Conversely, given a monad T, there is an adjunction $F^T \dashv U^T : \mathbf{C} \to \mathbf{C}^T$, where U^T is the forgetful functor sending an algebra to its carrier

$$\begin{tikzcd}
 & (X,k) \arrow[d, "\phi"'] & & X \arrow[d, "\phi"] \\
U^T\colon & (X',k') & \mapsto & X'
\end{tikzcd}$$

while F^T sends an object to its multiplication (the *free algebra*)

$$\begin{tikzcd}
 & X \arrow[d, "\phi"'] & & (TX,\mu^T_X) \arrow[d, "T(\phi)"] \\
F^T\colon & X' & \mapsto & (TX',\mu^T_{X'})\,.
\end{tikzcd}$$

This adjunction generates precisely the monad (T, η^T, μ^T).

Suppose we have an adjunction $(F, G, \eta, \epsilon) : \mathbf{C} \to \mathbf{D}$ generating a monad (T, η^T, μ^T). Such a monad generates the adjunction $(F^T, U^T, \eta^T, \epsilon^T)$. There is a "comparison" functor $K : \mathbf{D} \to \mathbf{C}^T$, defined as

$$\begin{tikzcd}
 & D \arrow[d, "f"'] & & (G(D),G(\epsilon_D)) \arrow[d, "G(f)"] \\
K\colon & D' & \mapsto & (G(D'),G(\epsilon_{D'}))
\end{tikzcd}$$

satisfying $U^T K = G$ and $KF = F^T$. An adjunction is *monadic* if the comparison functor K defined above is an equivalence of categories. A functor is monadic if it is the right adjoint of a monadic adjunction.

2.2. *Free algebras and monads*

Given a category $\mathbf{C}$ of algebraic structures defined by an equational theory, the forgetful functor $U : \mathbf{C} \to \mathbf{SET}$ has a left adjoint F and is monadic. This means that UF supports a monad structure in $\mathbf{SET}$ and that $\mathbf{C}$ is equivalent to the category of UF-algebras. In such a case the monad UF is called the *free algebra* for the structures in $\mathbf{C}$. It is equivalently characterised by the following *universal property* (where, as customary, we omit the mention of the forgetful functor): there exists a family of functions $\eta_X : X \to F(X)$, such that for every set X, for every structure Z of $\mathbf{C}$, and for every function $f : X \to Z$, there exists a unique morphism $\overline{f} : F(X) \to Z$ (the "homomorphic extension") that satisfies $\overline{f} \circ \eta_X = f$. We say that $F(X)$ is the *free algebra* over X (with respect to $\mathbf{C}$).

In the following, we present some relevant examples.

2.3. *The nondeterministic monad*

Assume that $\uplus$ (*formal union*) represents some kind of nondeterministic choice operator: $A \uplus B$ offers to the environment the choice between A and B. Usually such an operator satisfies the following equations:

— $A \uplus B = B \uplus A$;
— $A \uplus (B \uplus C) = (A \uplus B) \uplus C$;
— $A \uplus A = A$.

Since $\uplus$ is associative and commutative, we can introduce the following convention. If X is a set where $\uplus$ is defined, and $(x_i)_{i \in I}$ is a finite family of elements of X, we write

$$\biguplus_{i \in I} x_i$$

to denote the formal union of all x_i's. A similar convention will be used for the operation $\oplus$ (*formal sum*) which will also be associative and commutative (but not idempotent).

A model for the above theory is a *semilattice*. The category of semilattices is denoted by $\mathbf{SLAT}$. It is well known that the free semilattice functor can be concretely represented as (is naturally isomorphic to) the finite nonempty powerset functor $P : \mathbf{SET} \to \mathbf{SLAT}$, where the symbol $\uplus$ is interpreted as union. If X is a set, Z is a semilattice and $f : X \to Z$ is a function, the unique homomorphic extension $\overline{f} : P(X) \to Z$ is defined by

$$\overline{f}(Y) = \biguplus_{y \in Y} f(y)\,.$$

The corresponding monad $P : \mathbf{SET} \to \mathbf{SET}$ has the following unit and multiplication

$$\begin{array}{rcl} \eta^P_X(x) & = & \{x\}\,; \\ \mu^P_X(\mathcal{S}) & = & \bigcup \mathcal{S}\,. \end{array}$$

2.4. *The probabilistic monad*

Assume that $\oplus_p$ represents a probabilistic choice operator of some programming language: $A \oplus_p B$ is choosing A with probability p and B with probability $(1-p)$. This operator comes with an equational theory: for example it usually satisfies $A \oplus_p A = A$, because the choice between two equivalent possibilities is considered to be the same as not making any choice at all. Note that this assumes that the act of making the choice is invisible: the coin is always flipped behind one's back.

We are going to study a more general equational theory, that subsumes the one of the probabilistic choice.

Definition 2.1. A *real cone* is an algebra for the following equational theory in the category $\mathbf{SET}$ (the reason for the numbering will be apparent in the sequel).

(1) $A \oplus B = B \oplus A$;
(2) $A \oplus (B \oplus C) = (A \oplus B) \oplus C$;
(3) $A \oplus \underline{0} = A$;
(4) $0A = \underline{0}$;
(5) $1A = A$;
(6) $p(A \oplus B) = pA \oplus pB, \;\; p \in [0, +\infty[$;
(7) $p(qA) = (pq)A, \;\; p, q \in [0, +\infty[$;
(13) $(p + q)A = pA \oplus qA, \;\; p, q \in [0, +\infty[$.

We call $\mathbf{RCONE}$ the category of real cones and homomorphisms.

In a real cone, the probabilistic choice operator is coded as *convex combination*: $pA \oplus (1-p)B$. We choose to deal with the more general notion of real cone because its equational theory is nicer than the theory of the probabilistic choice. We are now going to characterise concretely the free real cone.

Definition 2.2. A *discrete valuation* on a set X is any function $v : X \to [0, +\infty]$.

The *support* of a discrete valuation v on X is the set

$$Supp(v) := \{x \in X \mid v(x) > 0\} \,.$$

The set of discrete valuations on X is denoted by $V_\infty(X)$. A discrete valuation v on a set X is a *discrete probability distribution* if $\sum_{x \in X} v(x) = 1$. The set of discrete probability distributions on X is denoted by $V^1_\infty(X)$. Discrete valuations taking values in $[0, +\infty[$ are called *weightings* (Jonsson et al., 2001). A *finite valuation* is a weighting whose support is finite. The set of finite valuations over a set X is denoted by $V(X)$. For each $x \in X$, the finite valuation η_x defined by

$$\eta_x(y) = \begin{cases} 1 & y = x; \\ 0 & y \neq x; \end{cases}$$

is called a *point valuation*.

Two operations of sum and scalar product are defined pointwise on $V(X)$:

$$v \oplus w(x) = v(x) + w(x);$$

$$pv(x) = p(v(x)), \;\; p \in [0, +\infty[.$$

If X is a set, the set $V(X)$ with the pointwise operations defined above is a real cone. Moreover:

Proposition 2.3. The finite valuations over a set X form the free real cone over X.

Let $f : X \to R$ be a function, where R is a real cone. Define the family of functions $\eta^V_X : X \to V(X)$ by $\eta^V_X(x) = \eta_x$ The unique real cone homomorphism $\overline{f} : V(X) \to R$ which extends f is defined as follows:

$$\overline{f}(\nu) = \bigoplus_{x \in Supp(\nu)} \nu(x) f(x)\,.$$

The multiplication of the generated monad is defined as:

$$\mu^V_X(\Xi)(x) = \bigoplus_{\nu \in Supp(\Xi)} \Xi(\nu)\nu(x)\,.$$

2.5. *Distributive Laws*

A general way for combining two monads is by defining a *distributive law* (Beck, 1969). Suppose we have two monads (T, η^T, μ^T), (S, η^S, μ^S) on some category. A *distributive law* of S over T is a natural transformation $d : ST \overset{\cdot}{\longrightarrow} TS$ satisfying the following axioms:

- $d \circ \eta^S T = T\eta^S$;
- $d \circ S\eta^T = \eta^T S$;
- $d \circ \mu^S T = T\mu^S \circ dS \circ Sd$;
- $d \circ S\mu^T = \mu^T S \circ Td \circ dT$.

A distributive law defines a monad on the functor TS. If $d : ST \overset{\cdot}{\longrightarrow} TS$ is a distributive law, then $\big(TS, \eta^T\eta^S, (\mu^T\mu^S) \circ TdS\big)$ is a monad.

$$\begin{tikzcd}
TSTS \arrow[r, "\cdot", "TdS"'] & TTSS \arrow[r, "\cdot", "\mu^T\mu^S"'] & TS
\end{tikzcd}$$

A *monad morphism* between T and S is a natural transformation $\alpha : T \overset{\cdot}{\longrightarrow} S$ which suitably commutes with units and multiplications. A *lifting* of the monad T to the category of S-algebras is a monad $(\tilde{T}, \eta^{\tilde{T}}, \mu^{\tilde{T}})$ on $\mathbf{C}^S$, such that, if $U^S : \mathbf{C}^S \to \mathbf{C}$ is the forgetful functor then

- $U^S\tilde{T} = TU^S$;
- $U^S\eta^{\tilde{T}} = \eta^T U^S$;
- $U^S\mu^{\tilde{T}} = \mu^T U^S$.

Beck has proved the following theorem (Beck, 1969).

Theorem 2.4. Suppose we have two monads (T, η^T, μ^T), (S, η^S, μ^S) on some category $\mathbf{C}$. Then the existence of each one of the following implies the existence of the other two:

1 A distributive law $d : ST \overset{\cdot}{\longrightarrow} TS$.

2 A multiplication $\mu : TSTS \overset{\cdot}{\longrightarrow} TS$, such that

- $(TS, \eta^T\eta^S, \mu)$ is a monad;
- the natural transformations $\eta^T S : S \overset{\cdot}{\longrightarrow} TS$ and $T\eta^S : T \overset{\cdot}{\longrightarrow} TS$, are monad morphisms;

— the following middle unit law holds:

$$\begin{tikzcd}
TS \arrow[r, "T\eta^S\eta^T S"] \arrow[dr, "Id_{TS}"'] & TSTS \arrow[d, "\mu"] \\
 & TS\,.
\end{tikzcd}$$

3 A lifting $\tilde{T}$ of the monad T to $\mathbf{C}^S$.

The way to obtain (2) from (1) has been sketched above. To obtain a lifting from a distributive law we define $\tilde{T}(A, \sigma)$ as the S-algebra

$$\begin{tikzcd}
ST(A) \arrow[r, "d_A"] & TS(A) \arrow[r, "T(\sigma)"] & T(A)\ .
\end{tikzcd}$$

Conversely if we have the multiplication μ we can define d by

$$\begin{tikzcd}
ST \arrow[r, "\eta^T TS\eta^S"] & TSTS \arrow[r, "\mu"] & TS\ .
\end{tikzcd}$$

If we have a lifting $\tilde{T}$, we define d by

$$\begin{tikzcd}
ST \arrow[r, "TS\eta^S"] \arrow[dr, "d"'] & STS \arrow[r, equal] & U^SF^STU^SF^S \arrow[r, equal] & U^SF^SU^S\tilde{T}F^S \arrow[d, "U^S\epsilon\tilde{T}F^S"] \\
 & TS \arrow[r, equal] & TU^SF^S \arrow[r, equal] & U^S\tilde{T}F^S\,,
\end{tikzcd}$$

where ϵ is the counit of the adjunction $F^S \dashv U^S$.

The correctness of the above constructions is shown by several diagram chases (Beck, 1969).

When the two monads involved are the free algebras of some equational theory, an equational distributive law may or may not correspond to a categorical distributive law. We discuss this issue in the next section.

3. The failure of the distributive law

Distributing probabilistic choice over nondeterministic choice amounts to the following equation:

$$A \oplus_p (B \uplus C) = (A \oplus_p B) \uplus (A \oplus_p C)\ .$$

Intuitively this expresses indifference to whether the environment chooses before or after the probabilistic choice is made. Once we accept the distributive law, the extra *convexity* law (Bandini and Segala, 2001)

$$A \uplus B = A \uplus B \uplus (A \oplus_p B) \uplus (B \oplus_p A)$$

must be also accepted, because

$$A \uplus B = (A \uplus B) \oplus_p (A \uplus B) = (A \oplus_p A) \uplus (B \oplus_p B) \uplus (A \oplus_p B) \uplus (B \oplus_p A).$$

If the equational distributive law corresponded to a categorical distributive law, by Beck's theorem (Theorem 2.4) the nondeterministic monad would lift to the category of algebras for the probabilistic monad. In the category $\mathbf{SET}$ this means that the powerset monad would lift to the category of real cones. The convexity law suggests that this is not possible, because not all sets satisfy the convexity law. In fact, the following general theorem says that the obvious definition of the operations for the powerset cannot satisfy $A \oplus_p A = A$. Suppose we have an equational theory. Take a model X for it. We can extend every operation f of arity n to the subsets of X by

$$f(X_1, \ldots, X_n) = \{f(x_1, \ldots, x_n) \mid x_i \in X_i, i \in I_n\}.$$

Theorem 3.1 (Gautam, 1957). A necessary and sufficient condition for the operations defined on the powerset of X to satisfy an equation of the theory is that each individual variable occurs at most once on each side of the equation.

Equations satisfying the above condition are called *affine*. The equation $A \oplus_p A = A$ is not affine, thus cannot be satisfied in the powerset. This would not exclude the possibility of lifting the operations in a different way, thus obtaining another distributive law. However, it turns out that there is no distributive law at all between the two monads. If (P, η^P, μ^P) is the finite nonempty powerset monad, and (V, η^V, μ^V) is the finite valuation monad in the category $\mathbf{SET}$, we have

Proposition 3.2. There is no distributive law of V over P.

Proof. See the Appendix. □

Our solution consists in changing the definition of probabilistic monad by removing the equation $A \oplus_p A = A$. In our presentation, the probabilistic monad is generated by the theory of real cones and the probabilistic choice $A \oplus_p B$ is coded as convex combination $pA \oplus (1 - p)B$. We remove the equation $pA \oplus qA = (p + q)A$ from the theory of real cones. In the category $\mathbf{SET}$, the monad freely generated by the new equational theory is called the *finite indexed valuation* monad IV. By Theorem 3.1, we can lift the operations to the powerset, thus obtaining a distributive law. The next section is devoted to this construction.

4. Indexed valuations

In this section we present the definition of the indexed valuation monad in the category $\mathbf{SET}$, and we show the existence of the categorical distributive law between indexed valuations and the finite nonempty powerset.

4.1. *Definition*

We first introduce the concrete characterisation of our construction.

Definition 4.1. Let X be a set. A *discrete indexed valuation* (DIV) on X is a pair (Ind, v) where $Ind : I \to X$ is a function and v is a discrete valuation on I, for some set I.

Note that we do not require that Ind be injective. This is indeed the main point of this construction: we want to divide the probability of an element among its indices. One possible interpretation is that indices in I represent computations, while elements of X represent observations. The semantics we present in Section 6 will confirm this intuition.

We shall also write x_i for $Ind(i)$ and p_i for $v(i)$. A discrete indexed valuation $\xi := (Ind, v)$ will also be denoted as $(x_i, p_i)_{i\in I}$.

We are now going to define an equivalence relation on the class of DIVs. Recall that $Supp(v) = \{i \in I \mid v(i) > 0\}$.

Definition 4.2. Let $(Ind, v) = (x_i, p_i)_{i\in I}$ and $(Ind', w) = (y_j, q_j)_{j\in J}$ be two discrete indexed valuations. We set

$$(Ind, v) \sim (Ind', w)$$

if and only if there exists a bijection $h : Supp(v) \to Supp(w)$ such that

$$\forall i \in Supp(v).\ y_{h(i)} = x_i\ ,$$

$$\forall i \in Supp(v).\ q_{h(i)} = p_i\ .$$

This says that two DIVs are equivalent up to renaming of the indices, and that only indices in the support matter. Indices are used to distinguish between different computations, but their precise character is unimportant. What is important is to keep track of the different computations there are, and how they relate to observations. Moreover, we may as well ignore computations with probability 0.

From now on we will use the term “discrete indexed valuations” to denote equivalence classes under $\sim$.

Given a set X and an infinite cardinal number α we define the set $IV_\alpha(X)$ as follows:

$$IV_\alpha(X) := \{(x_i, p_i)_{i\in I} \mid |I| < \alpha\}/\sim\ .$$

It is easy to realise that $IV_\alpha(X)$ is indeed a set. For every cardinal number $\beta < \alpha$, choose a set I_β such that $|I_\beta| = \beta$. The class $\{I_\beta \mid \beta < \alpha\}$ is a set. And clearly $IV_\alpha(X)$ is a quotient of $\bigcup_{\beta<\alpha} X^{I_\beta} \times [0, +\infty]^{I_\beta}$. In particular $IV_{\aleph_0}(X)$ is the set of discrete indexed valuations whose indexing set is finite.

Definition 4.3. A *finite indexed valuation* on X is an element of $IV_{\aleph_0}(X)$ for which $p_i < +\infty$ for all indices $i \in I$. The set of finite indexed valuations on X is denoted by $IV(X)$.

The construction above can be extended to a functor $IV : \mathbf{SET} \to \mathbf{SET}$ as follows. If $f : X \to Y$ then

$$IV(f)([(x_i, p_i)_{i\in I}]_\sim) := [(f(x_i), p_i)_{i\in I}]_\sim\ .$$

It is easy to check that this construction is well defined (i.e. does not depend on the representative).

From now on we will drop the explicit mention of equivalence classes, and work with representatives to simplify the reading.

4.2. *Equational characterisation*

We define two operations on discrete indexed valuations.

Definition 4.4. Let $\nu := (Ind, v) = (x_i, p_i)_{i\in I}$, $\xi := (Ind', w) = (y_j, q_j)_{j\in J}$ be two DIVs on a set X. Assume that $I \cap J = \emptyset$. This is not restrictive, because we can always reindex.

We define $\nu \oplus \xi$ to be $(Ind \cup Ind', v \cup w)$. For $p \in [0, +\infty[$ we define $p\nu$ to be $(x_i, pp_i)_{i \in I}$. With $\underline{0}$ we denote the DIV whose indexing set is empty.

Note, in particular, that when $p \in]0, 1[$, we have $p\nu \oplus (1-p)\nu \not\sim \nu$, because the indexing sets do not have the same cardinality.

Consider the following equational theory:

(1) $A \oplus B = B \oplus A$;
(2) $A \oplus (B \oplus C) = (A \oplus B) \oplus C$;
(3) $A \oplus \underline{0} = A$;
(4) $0A = \underline{0}$;
(5) $1A = A$;
(6) $p(A \oplus B) = pA \oplus pB \;\; p \in [0, +\infty[$;
(7) $p(qA) = (pq)A \;\; p, q \in [0, +\infty[$.

These axioms are almost the ones defining a real cone. The difference is that we have dropped the equation $(p+q)A = (pA \oplus qA)$.

Definition 4.5. A *real quasi-cone* is an algebra for the equational theory (1)–(7) in the category $\mathbf{SET}$. The category of real quasi-cones is denoted by $\mathbf{QCONES}$.

Proposition 4.6. The finite indexed valuations over a set X form the free real quasi-cone over X.

Proof. For any set X, it is clear that $IV(X)$ with the operations defined above is a quasi-cone. Define the family of functions $\eta^{IV}_X : X \to IV(X)$ by

$$\eta^{IV}_X(x) = (x, 1)_{* \in \{*\}} .$$

Let Q be a quasi-cone and let $f : X \to Q$ be a function. We have to show that there is a unique quasi-cone homomorphism $\overline{f} : IV(X) \to Q$ such that $\overline{f}(\eta^{IV}_X(x)) = f(x)$. The homomorphism condition forces us to define

$$\overline{f}(x_i, p_i)_{i \in I} = \bigoplus_{i \in I} p_i f(x_i) .$$

Equations (1)–(4) guarantee that the definition does not depend on the representative for $(x_i, p_i)_{i \in I}$. Equation (5) guarantees that $\overline{f}(x, 1) = f(x)$. The homomorphism condition for the sum (and $\underline{0}$) are obvious, while for the scalar product we have to use equations (6) and (7). □

The above proposition tells us that the functor IV extends to a monad. Its multiplication is as follows:

$$\begin{array}{rccc} \mu^{IV}_X : & IV(IV(X)) & \to & IV(X) \\ & (((x_{i_\lambda}, p_{i_\lambda})_{i_\lambda \in I_\lambda}, \pi_\lambda)_{\lambda \in \Lambda}) & \mapsto & (x_j, q_j)_{j \in J} \, ; \end{array}$$

where

$$J = \biguplus_{\lambda \in \Lambda} I_\lambda \, , \;\; q_j = p_j \pi_\lambda \text{ if } j \in I_\lambda \, .$$

To simplify the definition of μ, recall that a DIV is in fact an equivalence class. We can therefore assume that $I_\lambda = I$ for every $\lambda \in \Lambda$ because we can always reindex and add indices with probability 0. Therefore

$$((x_{i_\lambda}, p_{i_\lambda})_{i_\lambda \in I_\lambda}, \pi_\lambda)_{\lambda \in \Lambda} \sim ((x_i^\lambda, p_i^\lambda)_{i \in I}, \pi_\lambda)_{\lambda \in \Lambda} \,.$$

This allows us to use a simpler expression for the multiplication:

$$\mu_X^{IV} \left(((x_i^\lambda, p_i^\lambda)_{i \in I}, \pi_\lambda)_{\lambda \in \Lambda}\right) = (x_i^\lambda, \pi_\lambda p_i^\lambda)_{(i,\lambda) \in I \times \Lambda} \,.$$

4.3. *The distributive law*

Since all the equations in the theory of real quasi-cones are affine, Gautam's theorem guarantees that the operations lift to the powerset. Such a lifting boils down to a lifting of the finite nonempty powerset monad P to the category of real quasi-cones (IV-algebras). This, by Theorem 2.4, guarantees the existence of a distributive law $d : IV \circ P \stackrel{\cdot}{\longrightarrow} P \circ IV$.

We construct this lifting explicitly, in order to show the correspondence of the categorical distributive law with the equational one. Recall that a semilattice is a model of the following theory:

(8) $A \uplus B = B \uplus A$;
(9) $A \uplus (B \uplus C) = (A \uplus B) \uplus C$;
(10) $A \uplus A = A$.

We have seen that the finite nonempty powerset is the free semilattice. Consider now the combined equational theory (1)–(10) augmented with the following equations:

(11) $p(A \uplus B) = pA \uplus pB$;
(12) $A \oplus (B \uplus C) = (A \oplus B) \uplus (A \oplus C)$.

Equations (11)–(12) express that the probabilistic operators distribute over the nondeterministic one.

Definition 4.7. A *quasi-cone semilattice* is a model of the theory (1)–(12). The corresponding category is denoted by $\mathbf{QCS}$.

To show that P lifts to a monad in the category of real quasi-cones we show that it is left adjoint of the forgetful functor $U : \mathbf{QCS} \to \mathbf{QCONES}$. The first observation is that when Z is a real quasi-cone, then $P(Z)$ is in $\mathbf{QCS}$. By defining sum and multiplication pointwise, it is not difficult to verify that all the equations (1)–(12) are satisfied. Then we have to show the following universal property: for every real quasi-cone Z, for every quasi-cone semilattice W and for every real quasi-cone homomorphism $f : Z \to W$ there exists a unique extension $\overline{f} : P(Z) \to W$ which is a quasi-cone semilattice homomorphism, and for which $\overline{f}(\{z\}) = f(z)$.

$$\begin{tikzcd}
Z \arrow[d, "\eta^P"'] \arrow[dr, "f"] & \\
P(Z) \arrow[r, dashed, "\overline{f}"'] & W
\end{tikzcd}$$

The homomorphism condition forces us to define, for $Y \subseteq_{fin} Z$,

$$\overline{f}(Y) = \bigcup_{z \in Y} f(z) \,,$$

which gives us uniqueness. It is routine to verify that it is well defined and a homomorphism.

Since the extension is defined exactly as the extension of the monad P in the category of sets, what we have defined is indeed a lifting of P. Using Theorem 2.4, we deduce the existence of the distributive law.

Concretely, for every set X the component $d_X : IV(P(X)) \to P(IV(X))$ is defined as follows:

$$d_X\left((S_i, p_i)_{i\in I}\right) = \{(h(i), p_i)_{i\in I} \mid h : I \to X,\ h(i) \in S_i\}\ .$$

Note, in particular, the use of the “choice function” h.

A direct proof of the fact that the above family of functions is a distributive law can be found in (Varacca, 2003).

5. The convex powerset

Another solution for combining the nondeterministic and probabilistic monad consists in forming the distributive combinations of the theories, thus freely generating a new monad. The convexity law suggests a way of representing this construction concretely. This section is inspired by the work of Tix and Mislove, although they are concerned with DCPOs, while we work here in the category $\mathbf{SET}$.

5.1. *Finitely generated convex sets*

Recall that a *real cone* is a real quasi-cone satisfying the extra equation

(13) $(p+q)A = pA \oplus qA$.

Definition 5.1. A subset X of a real cone is *convex* if for every $x, y \in X, p \in [0,1]$, we have $px \oplus (1-p)y \in X$. Given a set X, its *convex closure* $\overline{X}$ is the smallest convex set containing X. A convex set X is *finitely generated* if there exists a finite set X_0 such that $X = \overline{X_0}$. Given a finite set I, elements $x_i, i \in I$, of a real cone and nonnegative real numbers $p_i, i \in I$, such that $\sum_{i\in I} p_i = 1$, the element $\bigoplus_{i\in I} p_i x_i$ is said to be a *convex combination* of the x_i.

The following result is standard.

Proposition 5.2. For a set X, we have that $\overline{X}$ is the set of convex combinations of elements of X.

Definition 5.3. For a real cone Z we define

$$P_{TM}(Z) = \{Y \subseteq Z \,|\, Y \text{ is convex and finitely generated}\}\ .$$

5.2. *Equational characterisation*

We characterise the functor P_{TM} as a free construction.

Definition 5.4. A *real cone-semilattice* is a model for the theory (1)–(13). The corresponding category is called $\mathbf{RCS}$.

Given a real cone Z, we define the following operations on $P_{TM}(Z)$:

— $pY := \{py \mid y \in Y\}$;
— $Y \oplus Y' := \{y \oplus y' \mid y \in Y, y' \in Y'\}$;
— $\underline{0} := \{\underline{0}\}$;
— $Y \uplus Y' := \overline{Y \cup Y'} = \{py \oplus (1-p)y' \mid p \in [0,1],\ y \in Y, y' \in Y'\}$.

The above operations are well defined: if Y, Y' are convex sets, it is easy to show that the sets $pY, Y \oplus Y', Y \uplus Y$ are also convex; if Y_0, Y'_0 are finite generators for Y, Y' then pY_0 is a finite generator for pY, $Y_0 \oplus Y'_0$ is a finite generator for $Y \oplus Y'$ and $Y_0 \cup Y'_0$ is a finite generator for $Y \uplus Y'$.

The above operations satisfy (1)–(13) so as to make $P_{TM}(Z)$ a real cone-semilattice. The only nontrivial ones to verify are (12)-(13): here is where convexity is needed.

We now show the universal property characterising freeness. For every real cone Z, real cone-semilattice H and real cone homomorphism $f : Z \to H$, there exists a unique $\mathbf{RCS}$-morphism $\overline{f} : P_{TM}(Z) \to H$ such that $\overline{f}(\{z\}) = f(z)$. For every $Y \in P_{TM}(Z)$ let Y_0 be one of its finite generators. Then define

$$\overline{f}(Y) := \biguplus_{y \in Y_0} f(y)\,.$$

We need to show that the above definition does not depend on the chosen finite generator. This requires some lemmas.

Proposition 5.5. In a real cone-semilattice, if w is a convex combination of y, y' then

$$y \uplus y' = y \uplus y' \uplus w\,.$$

Proof. Let $w = py \oplus (1-p)y'$. Then

$$\begin{array}{rcl} y \uplus y' & = & p(y \uplus y') \oplus (1-p)(y \uplus y') \\ & = & y \uplus y' \uplus (py \oplus (1-p)y') \uplus (py' \oplus (1-p)y) \\ & = & y \uplus y' \uplus (py \oplus (1-p)y') \end{array}$$

as in any semilattice, if $x = x \uplus x' \uplus x''$, then $x = x \uplus x'$. □

Lemma 5.6. Let H be a real cone-semilattice, let Y_0, Z_0 be finite subsets of H. If $\overline{Y_0} = \overline{Z_0}$, then $\biguplus Y_0 = \biguplus Z_0$

Proof. We prove this for the simple case where $Y_0 = \{y, y'\}$, $Z_0 = \{z, z'\}$. The general case can be proved in a similar way. We want to prove that $y \uplus y' = z \uplus z'$. It is enough to prove that $y \uplus y' = y \uplus y' \uplus z \uplus z'$, which, by symmetry, implies our result. Note that, from the assumption, z, z' must be convex combinations of y, y'. The statement is thus a consequence of Proposition 5.5. □

Now pick two different finite generators Y_0, Y'_0 for Y. We want to prove that $\biguplus f(Y_0) = \biguplus f(Y'_0)$. Since f is a homomorphism of real cones we have that $\overline{f(Y_0)} = f(\overline{Y_0}) = f(Y)$. Therefore $\overline{f(Y_0)} = \overline{f(Y'_0)}$. By Lemma 5.6 we have $\biguplus f(Y_0) = \biguplus f(Y'_0)$.

It is easy to verify that $\overline{f}$ respects the operations, using the equational distributive laws (11)-(12), and the fact that f is already a homomorphism of real cones. Moreover the homomorphism condition implies uniqueness.

We have thus proved the following:

Proposition 5.7. The operator P_{TM} with the operations as above defines a functor $\mathbf{RCONE} \to \mathbf{RCS}$ which is left adjoint of the forgetful functor.

The combination of the two adjunctions

$$\begin{tikzcd}\mathbf{SET} \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathbf{RCONE} \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathbf{RCS} \arrow[l, shift left]\end{tikzcd}$$

gives rise to a monad in $\mathbf{SET}$. Note that the monad P_{TM} on $\mathbf{RCONE}$ is not a lifting of the monad P, because, in general, convex sets are not finite. Therefore the monad $P_{TM} \circ V$ on $\mathbf{SET}$ is not obtained from any distributive law $V \circ P \stackrel{\cdot}{\longrightarrow} P \circ V$.

6. Semantics of programs

We give an example of how to use the constructions of the previous sections by giving a denotational semantics to a simple imperative language with probabilistic and nondeterministic primitives. We give the language an operational semantics in terms of a simplified version of probabilistic automata. We present two denotational semantics: one in terms of indexed valuations and the standard powerset, the other in terms of the standard valuations and the convex powerset. We show adequacy theorems relating the first semantics to *deterministic* schedulers, and the second semantics to *probabilistic* schedulers. Finally we discuss the computational intuition lying behind the mathematics.

6.1. *Probabilistic automata*

Probabilistic automata were introduced as such in (Segala, 1995). Their relationships with other probabilistic models are well known (Bartels et al., 2003; Stoelinga, 2002). We are going to adapt that general framework to our needs. We recall that if Y is a subset of $V^1_\infty(X)$, by $\overline{Y}$ we denote the set of convex combinations of elements of Y.

Let $P_\perp(X)$ denote $P(X) \cup \{\emptyset\}$. A *probabilistic automaton* on a set of states X is a function $k : X \to P_\perp(V^1_\infty(X))$ together with an initial state $x_0 \in X$. We will use the notation of (Herescu and Palamidessi, 2000): whenever $\nu \in k(x)$ we write

$$x(\xrightarrow{p_i} x_i)_{i \in I}$$

where $x_i \in X, i \neq j \implies x_i \neq x_j$, and $\nu(x_i) = p_i$. We also write $\xrightarrow{p} x$ for $(\xrightarrow{p} x)_{i \in \{*\}}$. A *finite path* of a probabilistic automaton is an element in $(X \times V^1_\infty(X))^* X$, written as $x_0 \nu_1 x_1 \ldots \nu_n x_n$, such that $\nu_i(x_i) > 0$. The path is *deterministic* if $\nu_{i+1} \in k(x_i)$. It is *probabilistic* if $\nu_{i+1} \in \overline{k(x_i)}$. The last state of a path s is denoted by $l(s)$. The probability of a path $s := x_0 \nu_1 x_1 \ldots \nu_n x_n$ is defined as

$$\Pi(s) = \prod_{1 \leq i \leq n} \nu_i(x_i)\,.$$

A *probabilistic scheduler* for a probabilistic automaton k is a partial function

$$\mathcal{S} : (X \times V^1_\infty(X))^* X \to V^1_\infty(X)$$

such that, if $k(l(r)) \neq \emptyset$, then $\mathcal{S}(r)$ is defined and $\mathcal{S}(r) \in \overline{k(l(r))}$. Equivalently, a probabilistic

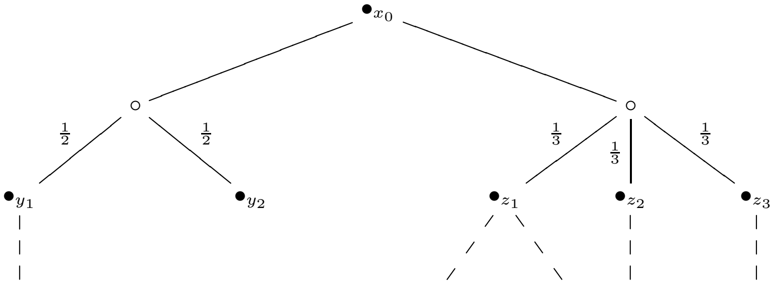

Fig. 1. A probabilistic automaton

scheduler can be defined as a partial function

$$\mathcal{S} : (X \times V^1_\infty(X))^* X \to V^1(V^1_\infty(X)) \ ,$$

requiring that $Supp(\mathcal{S}(r)) \subseteq k(l(r))$.

A *deterministic* scheduler is a probabilistic scheduler that does not make use of the convex combinations. That is for a deterministic scheduler we have $\mathcal{S}(r) \in k(l(r))$.

Now given a state $x \in X$ and a scheduler $\mathcal{S}$ for k, we consider the set $\mathcal{B}(k, \mathcal{S})$ of maximal paths, obtained from k by the action of $\mathcal{S}$. That is the paths $x_0\nu_1x_1 \dots \nu_nx_n$ such that for every $i < n$, $\nu_{i+1} = \mathcal{S}(x_0\nu_1 \dots x_i)$, and $k(x_n) = \emptyset$. A deterministic scheduler generates deterministic paths, a probabilistic scheduler generates probabilistic paths.

A good way of visualising probabilistic automata is by using alternating trees (Hansson, 1991). Figure 1 shows an example of an alternating tree, where black nodes represent states, while hollow nodes represent probability distributions. The use of trees instead of graphs is a way of keeping track of the paths: a deterministic scheduler is thus a function that, for every black node, chooses one of its hollow sons.

6.2. *A simple imperative language*

We present a small imperative language $\mathbf{L}$. It has the following (abstract) syntactic categories:

— integers $\mathbf{Num}$, ranged over by n;
— locations $\mathbf{Loc}$, ranged over by X;
— finite probability distributions over integers $\mathbf{Prob}$, ranged over by χ;
— arithmetical expressions $\mathbf{Aexp}$, ranged over by a;
— boolean expressions $\mathbf{Bexp}$, ranged over by b;
— commands $\mathbf{Comm}$, ranged over by c.

The (abstract) BNF for the last three syntactic categories are as follows:

$$\begin{array}{lll} a & ::= & n \mid X \mid a + a \mid a - a \mid a * a\,; \\ b & ::= & \mathbf{true} \mid \mathbf{false} \mid a \leq a \mid \neg b \mid b \wedge b\,; \\ c & ::= & \mathbf{skip} \mid X := a \mid X := \chi \mid c; c \mid \mathbf{if}\ b\ \mathbf{then}\ c\ \mathbf{else}\ c \mid c\ \mathbf{or}\ c\,. \end{array}$$

We also need the notion of *state*. A state is a function $\sigma : \mathbf{Loc} \to \mathbf{Num}$. We call Σ the set of states. We call any pair $\langle c, \sigma \rangle$ a *configuration*. We denote the set of all configurations by Γ. The

$$\langle \mathbf{skip}, \sigma\rangle \xrightarrow{1} \langle \epsilon, \sigma\rangle$$

$$\langle X := a, \sigma\rangle \xrightarrow{1} \langle \epsilon, \sigma[n/X]\rangle \quad \text{where } n = [\![a]\!]\sigma$$

$$\langle X := \chi, \sigma\rangle (\xrightarrow{\chi(n)} \langle \epsilon, \sigma[n/X]\rangle\ )_{n\in\mathbf{Num}}$$

$$\frac{\langle c,\sigma\rangle (\xrightarrow{p_i}\langle c_i,\sigma_i\rangle)_{i\in I}}{\langle c;c',\sigma\rangle(\xrightarrow{p_i}\langle c_i;c',\sigma_i\rangle)_{i\in I}}$$

$$\langle \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma\rangle \xrightarrow{1} \langle c_1, \sigma\rangle \quad \text{if}[\![b]\!]\sigma = \mathbf{false}$$

$$\langle \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma\rangle \xrightarrow{1} \langle c_0, \sigma\rangle \quad \text{if}[\![b]\!]\sigma = \mathbf{true}$$

$$\frac{\langle c,\sigma\rangle(\xrightarrow{p_i}\gamma_i)_{i\in I}}{\langle c\ \mathbf{or}\ c',\sigma\rangle(\xrightarrow{p_i}\gamma_i)_{i\in I}} \qquad \frac{\langle c',\sigma\rangle(\xrightarrow{p_j}\gamma_j)_{j\in J}}{\langle c\ \mathbf{or}\ c',\sigma\rangle(\xrightarrow{p_j}\gamma_j)_{j\in J}}$$

Fig. 2. Operational semantics of **L**

set Γ is ranged over by γ. To make the notation more uniform we introduce (at the metalevel) the empty command ϵ. We use it with the following meaning:

$$\langle \epsilon, \sigma\rangle \equiv \sigma\ , \quad \epsilon; c \equiv c; \epsilon \equiv c\ .$$

Consequently, we extend the notion of configuration so that a state σ is a configuration $\langle c, \sigma\rangle$ where $c = \epsilon$.

It is straightforward to define the value of arithmetic and boolean expressions in a given state so as to have

$$[\![a]\!]\sigma \in \mathbf{Num} \ \text{ and } \ [\![b]\!]\sigma \in \{\mathbf{true}, \mathbf{false}\}\ .$$

6.3. *The operational semantics*

The operational semantics of **L** is given in terms of probabilistic automata on the set of configurations. For every configuration γ_0 we have the probabilistic automaton $\mathcal{M}\gamma_0 = (\Gamma, k, \gamma_0)$ where k is defined inductively using the rules in Figure 2.

Definition 6.1. Let $\mathcal{S}$ be a scheduler for $\mathcal{M}\langle c, \sigma\rangle$. To simplify the notation we say that $\mathcal{S}$ is a scheduler for $\langle c, \sigma\rangle$. We define $\mathcal{B}(c, \sigma, \mathcal{S})$ to be the set of maximal paths of $\mathcal{M}\langle c, \sigma\rangle$ generated by $\mathcal{S}$. We define $\mathit{Val}(\mathcal{S}, c, \sigma)$ to be the probability distribution such that

$$\mathit{Val}(\mathcal{S}, c, \sigma)(\sigma') = \sum_{\substack{s\in\mathcal{B}(c,\sigma,\mathcal{S})\\ l(s)=\sigma'}} \Pi(s)\ .$$

We define $\mathit{Ival}(\mathcal{S}, c, \sigma)$ to be the discrete indexed valuation

$$(l(s), \Pi(s))_{s\in\mathcal{B}(c,\sigma,\mathcal{S})}\ .$$

The last definition is a formalisation of the intuitive interpretation of indexed valuations. Here the indexing set is the set of paths (the computations) while the elements considered are the final states (the observations).

$$
\begin{array}{rcl}
[\![\mathbf{skip}]\!]\sigma & = & \{(\sigma, 1)\} \\
[\![X := a]\!]\sigma & = & \{(\sigma[n/X], 1)\} \text{ where } n = [\![a]\!]\sigma \\
[\![X := \chi]\!]\sigma & = & \{(\sigma[n/X], \chi(n))_{n \in Supp(\chi)}\} \\
[\![c_0; c_1]\!] & = & [\![c_1]\!]^{\dagger} \circ [\![c_0]\!] \\
[\![c_0 \text{ } \mathbf{or} \text{ } c_1]\!]\sigma & = & [\![c_1]\!]\sigma \cup [\![c_0]\!]\sigma \\
[\![\mathbf{if} \; b \; \mathbf{then} \; c_0 \; \mathbf{else} \; c_1]\!]\sigma & = & \begin{cases} [\![c_0]\!](\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{true} \\ [\![c_1]\!](\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{false} \end{cases}
\end{array}
$$

Fig. 3. Denotational semantics of **L** using indexed valuations

$$
\begin{array}{rcl}
[\![\mathbf{skip}]\!]_{TM}\sigma & = & \{\eta_{\sigma}\} \\
[\![X := a]\!]_{TM}\sigma & = & \{\eta_{\sigma[n/X]}\} \text{ where } n = [\![a]\!]\sigma \\
[\![X := \chi]\!]_{TM}\sigma & = & \{\lambda\sigma' \in \Sigma. \begin{cases} \chi(n) & \text{if } \sigma' = \sigma[n/X] \\ 0 & \text{otherwise} \end{cases} \} \\
[\![c_0; c_1]\!]_{TM} & = & [\![c_1]\!]^{\dagger}_{TM} \circ [\![c_0]\!]_{TM} \\
[\![c_0 \text{ } \mathbf{or} \text{ } c_1]\!]_{TM}\sigma & = & [\![c_1]\!]_{TM}\sigma \uplus [\![c_0]\!]_{TM}\sigma \\
[\![\mathbf{if} \; b \; \mathbf{then} \; c_0 \; \mathbf{else} \; c_1]\!]_{TM}\sigma & = & \begin{cases} [\![c_0]\!]_{TM}(\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{true} \\ [\![c_1]\!]_{TM}(\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{false} \end{cases}
\end{array}
$$

Fig. 4. Denotational semantics of **L** using the convex powerset

One could prove directly (by structural induction on the commands) that $Val(\mathcal{S}, c, \sigma)$ is a probability distribution. However, this is also a consequence of the adequacy theorem.

6.4. *Two adequate denotational semantics*

The denotational semantics

$$[\![c]\!] : \Sigma \to P(IV(\Sigma))$$

is defined in Figure 3. The indexed valuation $(\sigma, p)_{* \in \{*\}}$ is denoted as (σ, p). If X, Y are sets and $f : X \to P(IV(Y))$ then $f^{\dagger} : P(IV(X)) \to P(IV(Y))$ is the Kleisli extension of f for the monad $P \circ IV$.

There is a very tight correspondence between the denotational and the operational semantics.

Theorem 6.2 (Adequacy). Let c be a command of **L** and ν be a finite indexed valuation in $IV(\Sigma)$. Then $\nu \in [\![c]\!]\sigma$ if and only if there exists a scheduler $\mathcal{S}$ for $\langle c, \sigma \rangle$ s.t. $\nu = Ival(\mathcal{S}, c, \sigma)$.

Proof. See the Appendix. □

The main feature of Theorem 6.2 is the use of deterministic schedulers. A semantics in terms of the convex powerset functor is adequate with respect to probabilistic schedulers.

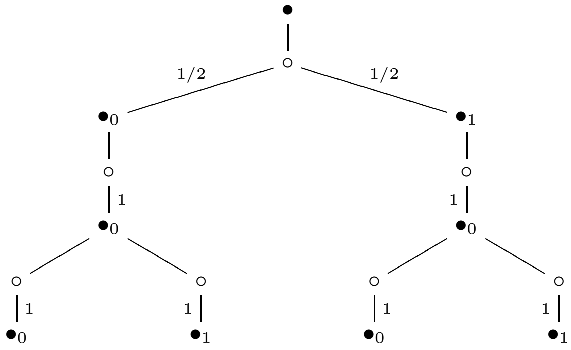

Fig. 5. The automaton of $c_1; c_2; c_3$

The new denotational semantics $[\![c]\!]_{TM} : \Sigma \to P_{TM}(V(\Sigma))$ is defined in Figure 4. Here, if X, Y are sets and $f : X \to P_{TM}(V(Y))$ then $f^\dagger : P_{TM}(V(X)) \to P_{TM}(V(Y))$ is the Kleisli extension of f for the monad $P_{TM} \circ V$.

Theorem 6.3 (Adequacy). Let c be a command of $\mathbf{L}$ and ν be a finite valuation in $V(\Sigma)$. Then $\nu \in [\![c]\!]_{TM}\sigma$ if and only if there exists a probabilistic scheduler $\mathcal{S}$ for $\langle c, \sigma\rangle$ s.t. $\nu = \mathit{Val}(\mathcal{S}, c, \sigma)$

Proof. See the Appendix. □

6.5. *Discussion*

We have seen the mathematical reasons why there is no distributive law between the functors P and V. We can exemplify this with a program in our language. Suppose the denotation of a command c is to be defined as a function $[\![c]\!] : \Sigma \to P(V(\Sigma))$. If we want it to be compositional, we have to define $[\![c_1; c_2]\!]$ in terms of $[\![c_1]\!], [\![c_2]\!]$. The first intuitive idea would be to define it as

$$[\![c_1; c_2]\!](\sigma) = \left\{\lambda\sigma'. \sum_{\sigma'' \in \Sigma} \nu(\sigma'') \cdot h(\sigma'')(\sigma') \mid \nu \in [\![c_1]\!](\sigma), h : \Sigma \to V(\Sigma), h(\sigma'') \in [\![c_2]\!](\sigma'')\right\}.$$

However, this definition would make sequential composition non-associative. To see this with an example, let

— c_1 be the command $X := \chi$, where $\chi(0) = 1/2, \chi(1) = 1/2$;
— c_2 be the command $X := 0$;
— c_3 be the command $X := 0$ **or** $X := 1$;

and consider the program $c_1; c_2; c_3$.

In this example we can assume there are only two states: $\Sigma = \{0, 1\}$. For $i = 0, 1$ we have

— $[\![c_1]\!](i) = \{\frac{1}{2}\eta_0 + \frac{1}{2}\eta_1\}$;
— $[\![c_2]\!](i) = \{\eta_0\}$;
— $[\![c_1; c_2]\!](i) = \{\eta_0\}$;
— $[\![c_3]\!](i) = \{\eta_0, \eta_1\}$;
— $[\![c_2; c_3]\!](i) = \{\eta_0, \eta_1\}$.

If we read $c_1; c_2; c_3$ as $c_1; (c_2; c_3)$, then $[\![c_1; c_2; c_3]\!](i) = \{\eta_0, \frac{1}{2}\eta_0 + \frac{1}{2}\eta_1, \eta_1\}$.
If we read $c_1; c_2; c_3$ as $(c_1; c_2); c_3$, then $[\![c_1; c_2; c_3]\!](i) = \{\eta_0, \eta_1\}$.

In the second case the function h (which roughly speaking does the job of the scheduler), when choosing a valuation in $[\![c_3]\!](0)$ does not "remember" that the process has reached the state 0 by two different paths. Therefore we miss one valuation in the final set. When the denotation is given in terms of indexed valuations, the function h is given enough information to remember this. Indeed, in the case of indexed valuations, h chooses with regard to at the paths, rather than only with regard to the states.

However, a memoryless scheduler can simulate the combination of schedulers with memory by flipping a coin. That is why a semantics that is adequate with respect to probabilistic schedulers does not need to be given in terms of indexed valuations (see the proofs in the Appendix for more details). A similar phenomenon was recently observed in the context of Stochastic Games (Chatterjee et al., 2004).

7. Indexed valuations and domains

In this section, we sketch how to extend the above discussion to domain theory. Recall that a domain is a partial order with lubs of directed subsets (and some extra properties). When doing universal algebra on domains, instead of equational theories we have inequational theories, while operations and homomorphisms are required to be continuous (Abramsky and Jung, 1994).

Consider the equational theory of semilattices. In the category of continuous domains its free algebra functor is known as the *Plotkin powerdomain*. However, we can also modify the equational theory by adding an extra inequation. If we add $A \sqsubseteq A \uplus B$, we obtain the theory of *join-semilattices*. The free join-semilattice functor is known as the *Hoare powerdomain*. If instead we add the inequality $A \uplus B \sqsubseteq A$, we obtain the theory of *meet-semilattices*. The corresponding free algebra functor is known as the *Smyth powerdomain*. Being freely generated by a theory, all the above functors give rise to monads.

In the category of continuous domains, the theory of real cones generates the monads of *continuous valuations* (Jones and Plotkin, 1989; Jones, 1990; Kirch, 1993). In this category, we can weaken the theory of real cones by removing the equation $pA \oplus qA = (p+q)A$, or by transforming it into an inequation: $pA \oplus qA \sqsupseteq (p+q)A$ or $pA \oplus qA \sqsubseteq (p+q)A$. Also, we can decide whether the scalar multiplication is continuous with the respect to the domain of scalars $[0, +\infty]$, or not. All these choices have an important effect on the corresponding monad. Some results were already presented in (Varacca, 2002; Varacca, 2003), where the existence of certain distributive laws is shown, and the relation between indexed valuations and continuous valuations is studied. A recent related work is (Mislove, 2005).

A detailed study of all the cases, with particular attention to a concrete characterisation of the monads, is the subject of ongoing work.

Acknowledgments

Thanks to Andrzej Filinski who showed us how the nondeterministic and probabilistic monads are combined in ML. Thanks to Michael Mislove, Gordon Plotkin, Luigi Santocanale and Zhe

Yang for useful discussions. Daniele Varacca acknowledges the contribution of the Danish National Research Foundation, and of the EPSRC grant GR/T04724/01. Glynn Winskel acknowledges the contribution of the EPSRC grant GR/T22049/01.

Appendix: Proofs

We produce here the proofs of three results: Proposition 3.2, which states the lack of a distributive law between the nondeterministic and the probabilistic monad, and the two adequacy theorems, Theorem 6.2 and Theorem 6.3.

Proof of Proposition 3.2 The idea for this proof is due to Gordon Plotkin[†]. Assume that $d : VP\overset{\cdot}{\longrightarrow}PV$ is a distributive law in the category **SET**. Consider the set $X := \{a, b, c, d\}$. Take $\Xi := \frac{1}{2}\eta_{\{a,b\}} + \frac{1}{2}\eta_{\{c,d\}} \in VP(X)$. We try to find out what $R := d_X(\Xi)$ is.

Let $Y := \{a, b\}$. Consider:

$$f : X \to Y \quad f : \begin{cases} a & \mapsto & a \\ b & \mapsto & b \\ c & \mapsto & a \\ d & \mapsto & b \end{cases}$$

$$f' : X \to Y \quad f' : \begin{cases} a & \mapsto & a \\ b & \mapsto & b \\ c & \mapsto & b \\ d & \mapsto & a\,. \end{cases}$$

We have that $VP(f)(\Xi) = \eta_Y = VP(f')(\Xi)$. Consider the naturality diagram for f:

$$\begin{tikzcd}
\Xi \arrow[r, mapsto, "d_X"] \arrow[d, mapsto, "VP(f)"'] & R \arrow[d, mapsto, "PV(f)"] \\
\eta_Y \arrow[r, mapsto, "d_Y"'] & S.
\end{tikzcd}$$

One of the unit laws for d tells us that $S := d_Y(\eta_Y) = \{\eta_a, \eta_b\}$. Therefore, considering the functorial action of PV, we must have that

$$\emptyset \neq R \subseteq \big\{p\eta_a + (1-p)\eta_c \mid p \in [0,1]\big\} \cup \big\{q\eta_b + (1-q)\eta_d \mid q \in [0,1]\big\}\,.$$

Consider the same diagram for f':

$$\begin{tikzcd}
\Xi \arrow[r, mapsto, "d_X"] \arrow[d, mapsto, "VP(f')"'] & R \arrow[d, mapsto, "PV(f')"] \\
\eta_Y \arrow[r, mapsto, "d_Y"'] & S.
\end{tikzcd}$$

This tells us that

$$\emptyset \neq R \subseteq \big\{p'\eta_a + (1-p')\eta_d \mid p' \in [0,1]\big\} \cup \big\{q'\eta_b + (1-q')\eta_c \mid q' \in [0,1]\big\}\,.$$

[†] Personal communication

Combining these pieces of information, we conclude that R must be a nonempty subset of $\{\eta_a, \eta_b, \eta_c, \eta_d\}$.

Now let $Z := \{a, c\}$. Consider

$$f'' : X \to Z \;\; f'' : \begin{cases} a & \mapsto & a \\ b & \mapsto & a \\ c & \mapsto & c \\ d & \mapsto & c\,. \end{cases}$$

We have that $VP(f'')(\Xi) = \frac{1}{2}\eta_{\{a\}} + \frac{1}{2}\eta_{\{c\}}$. Let us look at the naturality diagram for f'':

$$\begin{tikzcd} \Xi \arrow[r, mapsto, "d_X"] \arrow[d, mapsto, "VP(f'')"'] & R \arrow[d, mapsto, "PV(f'')"] \\ \frac{1}{2}\eta_{\{a\}} + \frac{1}{2}\eta_{\{c\}} \arrow[r, mapsto, "d_Z"'] & T. \end{tikzcd}$$

Since $T = PV(f'')(R)$, then T must be a nonempty subset of $\{\eta_a, \eta_c\}$. But the other unit law for d tells us that $T = d(\frac{1}{2}\eta_{\{a\}} + \frac{1}{2}\eta_{\{c\}}) = \{\frac{1}{2}\eta_a + \frac{1}{2}\eta_c\}$. Contradiction. □

Before proving Theorem 6.2, we need to look concretely at the Kleisli extension of the monad $P \circ IV$ generated by the distributive law. Let $f : X \to P(IV(Y))$ be a function. Consider a finite set of indexed finite valuations $A \in P(IV(X))$. Let $f^\dagger : P(IV(X)) \to P(IV(Y))$ be the Kleisli extension of f. We want to evaluate $f^\dagger(A)$.

When A is a singleton $\big\{(x_i, p_i)_{i\in I}\big\}$, we have that

$$f^\dagger\Big(\big\{(x_i, p_i)_{i\in I}\big\}\Big) = \bigoplus_{i\in I} p_i f(x_i)\,.$$

By induction on the size of I one can prove that

$$\bigoplus_{i\in I} p_i f(x_i) \quad = \quad \Big\{(y^i_j, p_i q^i_j)_{(j,i)\in J\times I} \,|\, h : I \to IV(Y),\ h(i) = (y^i_j, q^i_j)_{j\in J} \in f(x_i)\Big\}\,.$$

(Since I and all $f(x_i)$ are finite, it is not restrictive to assume that all the valuations involved are indexed by the same set J.) The functions h above are “choice” functions: For every $i \in I$, h chooses an indexed valuation in $f(x_i)$.

For general A we have

$$f^\dagger(A) = \bigcup_{\xi\in A} f^\dagger(\{\xi\})\,.$$

Proof of Theorem 6.2 By structural induction. The nontrivial case is the sequential composition. A maximal path of $\mathcal{M}\langle c_0; c_1, \sigma\rangle$ is the concatenation of a maximal path r of $\mathcal{M}\langle c_0, \sigma\rangle$ together with a maximal path t in $\mathcal{M}\langle c_1, l(r)\rangle$, renaming the configurations of the first part. Therefore a scheduler $\mathcal{S}$ for $\langle c_0; c_1, \sigma\rangle$ can be thought of as a scheduler $\mathcal{S}_0$ for $\langle c_0, \sigma\rangle$ together with schedulers $\mathcal{S}_r$ for $\langle c_1, l(r)\rangle$ for every maximal path r of $\mathcal{M}\langle c_0, \sigma\rangle$.

By the induction hypothesis we have $(l(r), \Pi(r))_{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)} \in [\![c_0]\!]\sigma$ and for every r, $(l(t), \Pi(t))_{t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)} \in [\![c_1]\!]l(r)$. We have to show that

$$(l(s), \Pi(s))_{s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})} \in [\![c_1]\!]^\dagger([\![c_0]\!]\sigma)\,.$$

Recalling the characterisation of $f^\dagger$, it is enough to show that

$$(l(s),\Pi(s))_{s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})} \in [\![c_1]\!]^\dagger\left(\{\,(l(r),\Pi(r))_{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}\,\}\right)\;.$$

Let us define the choice function $h:\mathcal{B}(c_0,\sigma,\mathcal{S}_0)\to IV(\Sigma)$ as

$$h(r) = (l(t),\Pi(t))_{t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)} \in [\![c_1]\!]l(r)\,.$$

Therefore by the characterisation of $f^\dagger$:

$$(l(t),\Pi(r)\Pi(t))_{\substack{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)\\ t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)}} \in [\![c_1]\!]^\dagger\left(\{\,(l(r),\Pi(r))_{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}\,\}\right)\;.$$

Since a path in $\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})$ is the concatenation of a path r in $\mathcal{B}(c_0,\sigma,\mathcal{S}_0)$ together with a path t in $\mathcal{B}(c_1,l(r),\mathcal{S}_r)$, we have

$$(l(t),\Pi(r)\Pi(t))_{\substack{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)\\ t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)}} = (l(s),\Pi(s))_{s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}\,.$$

Conversely suppose $\nu\in[\![c_1]\!]^\dagger([\![c_0]\!]\sigma)$. By the characterisation of the Kleisli extension, there exist $(\sigma_i,p_i)_{i\in I}\in[\![c_0]\!]\sigma$ and $h:I\to IV(\Sigma)$, $h(i)=(y^i_j,q^i_j)_{j\in J}\in[\![c_1]\!]\sigma_i$ such that

$$\nu = (y^i_j,p_iq^i_j)_{(j,i)\in J\times I}\;.$$

By the induction hypothesis there exists a scheduler $\mathcal{S}_0$, such that

$$I=\mathcal{B}(c_0,\sigma,\mathcal{S}_0), p_r=\Pi(r), \sigma_r = l(r)\,.$$

And for every $r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)$, there is a scheduler $\mathcal{S}_r$ such that

$$h(r) = (l(t),\Pi(t))_{t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)}\,.$$

Combining $\mathcal{S}_0$ with the $\mathcal{S}_r$ we obtain a scheduler $\mathcal{S}$ for $\langle c_0;c_1,\sigma\rangle$. In order to obtain an overall scheduler $\mathcal{S}$, formally we have to define it also for the paths not in $\mathcal{B}(c_0,\sigma,\mathcal{S}_0)$. But this choice can be arbitrary, because it does not influence the definition of $\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})$. Therefore

$$\nu = (y^i_j,p_iq^i_j)_{(j,i)\in J\times I} = (l(t),\Pi(r)\Pi(t))_{\substack{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)\\ t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)}}$$

$$= (l(s),\Pi(s))_{s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}\;.$$

□

Before proving Theorem 6.3 we need to look concretely at the Kleisli extension of the monad $P_{TM}\circ V$ defined in Section 5.

Take $f:X\to P_{TM}(V(Y))$, say $f(x)=B_x$. We have that $f^\dagger:P_{TM}(V(X))\to P_{TM}(V(Y))$ is defined as

$$f^\dagger(A) = \biguplus_{\xi\in A_0}\bigoplus_{x\in X}\xi(x)B_x\,.$$

We have following characterisation:

$$f^\dagger(A) = \bigcup_{\xi\in A}\bigoplus_{x\in X}\xi(x)B_x = \left\{\bigoplus_{x\in X}\xi(x)h(x)\,|\,h:X\to V(Y), h(x)\in B_x,\,\xi\in A\right\}\,.$$

In order to prove it, let’s call

— $V := \bigcup_{\xi \in A_0} \bigoplus_{x \in X} \xi(x) B_x$;
— $U := \biguplus_{\xi \in A_0} \bigoplus_{x \in X} \xi(x) B_x$;
— $W := \bigcup_{\xi \in A} \bigoplus_{x \in X} \xi(x) B_x$.

Remember that $U = \overline{V}$, by definition. We have to prove that $U = W$.

Clearly $V \subseteq W$. Moreover W is convex:

$$p \bigoplus_{x \in X} \xi(x)h(x) \oplus (1-p) \bigoplus_{x \in X} \xi'(x)h'(x) = \bigoplus_{x \in X} p\xi(x)h(x) \oplus (1-p)\xi'(x)h'(x)\,.$$

Define $\xi'' = p\xi \oplus (1-p)\xi' \in A$, and $h''(x) = \frac{p\xi(x)}{\xi''(x)}h(x) + \frac{(1-p)\xi'(x)}{\xi''(x)}h'(x)$. Since B_x is convex, then $h''(x) \in B_x$. (If $\xi''(x) = 0$ then $h''(x)$ can be set equal to any element of B_x.) We have

$$\xi''(x)h''(x) = p\xi(x)h(x) \oplus (1-p)\xi'(x)h'(x)\,.$$

Therefore $U \subseteq W$.

For the other direction take $\bigoplus_{x \in X} \xi(x)h(x)$. We know that $\xi = \bigoplus_{i \in I} p_i \xi_i$ with $\xi_i \in A_0$. So

$$\bigoplus_{x \in X} \xi(x)h(x) = \bigoplus_{x \in X} \bigoplus_{i \in I} p_i \xi_i(x)h(x) = \bigoplus_{i \in I} p_i \bigoplus_{x \in X} \xi_i(x)h(x)$$

which is a convex combination of elements of V.

It is worth making the following observation. Suppose $f : X \to P_{TM}(V(Y))$ is such that the range of f contains only probability distributions (rather than general finite valuations). Suppose W is a finitely generated convex sets containing only probability distributions. Then, it is not difficult to show that $f^\dagger(W)$ contains only probability distributions. This observation can be used to show that the denotational semantics of **L** uses probability distributions only.

Proof of Theorem 6.3 By structural induction. Note that the probabilistic schedulers are necessary for the semantics of the nondeterministic choice, because the operator $\uplus$ is defined as union followed by convex closure.

Again the nontrivial case is sequential composition. Take a scheduler $\mathcal{S}$ for $\langle c_0; c_1, \sigma \rangle$. Such an $\mathcal{S}$ can be thought of as a scheduler $\mathcal{S}_0$ for $\langle c_0, \sigma \rangle$ together with schedulers $\mathcal{S}_r$ for $\langle c_1, l(r) \rangle$ for every maximal path in r of $\mathcal{M}\langle c_0, \sigma \rangle$.

By the induction hypothesis we have that $\mathit{Val}(\mathcal{S}_0, c_0, \sigma) \in [\![c_0]\!]_{TM}\sigma$ and that for every r, $\mathit{Val}(\mathcal{S}_r, c_1, l(r)) \in [\![c_1]\!]_{TM} l(r)$. We have to show that

$$\lambda\sigma'. \sum_{\substack{l(s)=\sigma' \\ s \in \mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}} \Pi(s) \in [\![c_1]\!]^\dagger_{TM}([\![c_0]\!]_{TM}\sigma)\,.$$

Recall the characterisation of the Kleisli extension: if $f : X \to P_{TM}(V(Y))$, then

$$f^\dagger(A) = \left\{ \bigoplus_{x \in X} \xi(x)h(x) \mid h : X \to V(Y), h(x) \in f(x), \xi \in A \right\}$$

To prove our claim it is then enough to show that

$$\lambda\sigma'. \sum_{\substack{l(s)=\sigma' \\ s \in \mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}} \Pi(s) \in [\![c_1]\!]^\dagger_{TM}\left(\{\, \mathit{Val}(\mathcal{S}_0, c_0, \sigma)\,\}\right)\,.$$

Let us define $h : \Sigma \to V(\Sigma)$ as

$$h(\sigma'') = \sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{\mathit{Val}(\mathcal{S}_0, c_0, \sigma)(\sigma'')} \; \mathit{Val}(\mathcal{S}_r, c_1, \sigma'') \,.$$

Remember that, by definition:

$$\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r) = \mathit{Val}(\mathcal{S}_0, c_0, \sigma)(\sigma'') \,.$$

Therefore

$$\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{\mathit{Val}(\mathcal{S}_0, c_0, \sigma)(\sigma'')} = 1 \;.$$

Since $[\![c_1]\!]_{TM}\sigma''$ is convex, then $h(\sigma'') \in ([\![c_1]\!]_{TM}\sigma'')$. Therefore, by the characterisation of the Kleisli extension,

$$\sum_{\sigma'' \in \Sigma} \mathit{Val}(\mathcal{S}_0, c_0, \sigma)(\sigma'') h(\sigma'') \in [\![c_1]\!]^{\dagger}_{TM} \left(\{ \; \mathit{Val}(\mathcal{S}_0, c_0, \sigma) \, \} \right) \;.$$

But

$$\sum_{\sigma'' \in \Sigma} \mathit{Val}(\mathcal{S}_0, c_0, \sigma)(\sigma'') h(\sigma'')(\sigma')$$

$$= \sum_{\sigma'' \in \Sigma} \left(\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r) \left(\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{\mathit{Val}(\mathcal{S}_0, c_0, \sigma)(\sigma'')} \; \mathit{Val}(\mathcal{S}_r, c_1, \sigma'')(\sigma') \right) \right)$$

$$= \sum_{\sigma'' \in \Sigma} \left(\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{\mathit{Val}(\mathcal{S}_0, c_0, \sigma)(\sigma'')} \left(\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r) \; \mathit{Val}(\mathcal{S}_r, c_1, \sigma'')(\sigma') \right) \right)$$

$$= \sum_{\sigma'' \in \Sigma} \left(\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{\mathit{Val}(\mathcal{S}_0, c_0, \sigma)(\sigma'')} \right) \left(\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r) \; \mathit{Val}(\mathcal{S}_r, c_1, \sigma'')(\sigma') \right)$$

$$= \sum_{\sigma'' \in \Sigma} \left(\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r) \; \mathit{Val}(\mathcal{S}_r, c_1, \sigma'')(\sigma') \right)$$

$$= \sum_{\sigma'' \in \Sigma} \left(\sum_{\substack{l(r)=\sigma'' \\ r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r) \left(\sum_{\substack{l(t)=\sigma' \\ t \in \mathcal{B}(c_1,\sigma'',\mathcal{S}_r)}} \Pi(t) \right) \right)$$

$$= \sum_{r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)} \Pi(r) \left(\sum_{\substack{l(t)=\sigma' \\ t \in \mathcal{B}(c_1,l(r),\mathcal{S}_r)}} \Pi(t) \right)$$

$$= \sum_{\substack{l(s)=\sigma' \\ s \in \mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}} \Pi(s)\,,$$

and the claim is proved. For the last step, note that a path $s \in \mathcal{B}(c_0; c_1, \sigma, \mathcal{S})$ is the concatenation of a path $r \in \mathcal{B}(c_0, \sigma, \mathcal{S}_0)$ together with a path $t \in \mathcal{B}(c_1, l(r), \mathcal{S}_r)$.

Conversely, suppose that $\nu \in [\![c_1]\!]^\dagger_{TM}([\![c_0]\!]_{TM}\sigma)$. Then there exist $\xi \in [\![c_0]\!]_{TM}\sigma$ and $h : \Sigma \to V(\Sigma)$ such that $h(\sigma'') \in [\![c_1]\!]_{TM}\sigma''$ and $\nu = \sum_{\sigma''} \xi(\sigma'')h(\sigma'')$. By the induction hypothesis there exist schedulers $\mathcal{S}_0$, $\mathcal{S}_{\sigma''}$ such that $\xi = Val(\mathcal{S}_0, c_0, \sigma)$, and $h(\sigma'') = Val(\mathcal{S}_{\sigma''}, c_1, \sigma'')$. As earlier with deterministic schedulers, we combine them to get a scheduler $\mathcal{S}$ such that $\nu = Val(\mathcal{S}, c_0; c_1, \sigma)$. Notice that in this case the combined scheduler has some memoryless character: it behaves the same for every subautomaton starting at a configuration $\langle c_1, \sigma'' \rangle$, regardless of the previous history.

□

References

Abramsky, S. and Jung, A. (1994). Domain theory. In *Handbook of Logic in Computer Science*, volume 3. Clarendon Press.

Bandini, E. and Segala, R. (2001). Axiomatizations for probabilistic bisimulation. In *Proceedings of 28th ICALP*, volume 2076 of *LNCS*, pages 370–381. Springer.

Bartels, F., Sokolova, A., and de Vink, E. (2003). A hierarchy of probabilistic system types. In *Electronic Notes in Theoretical Computer Science*, volume 82. Elsevier.

Beck, J. (1969). Distributive laws. In *Seminar on Triples and Categorical Homology Theory*, volume 80 of *Lecture Notes in Mathematics*, pages 119–140. Springer.

Chatterjee, K., de Alfaro, L., and Henzinger, T. A. (2004). Trading memory for randomness. In *Proceedings of first QEST*, pages 206–217.

Cohn, P. M. (1981). *Universal Algebra*. Reidel.

Gautam, N. (1957). The validity of equations of complex algebras. *Archiv für Mathematische Logik und Grundlagenforschung*, 3:117–124.

Hansson, H. (1991). *Time and Probability in Formal Design of Distributed systems*. PhD thesis, Uppsala University.

Herescu, M. and Palamidessi, C. (2000). Probabilistic asynchronous π-calculus. In *Proceedings of 3rd FoSSaCS*, volume 1784 of *LNCS*, pages 146–160. Springer.

Hyland, M., Plotkin, G. D., and Power, J. (2002). Combining computational effects: Commutativity and sum. In *Proceedings of IFIP TCS*, pages 474–484. Kluwer.

Jones, C. (1990). *Probabilistic Non-determinism*. PhD thesis, University of Edinburgh.

Jones, C. and Plotkin, G. D. (1989). A probabilistic powerdomain of evaluations. In *Proceedings of 4th LICS*, pages 186–195. IEEE Computer Society.

Jonsson, B., Larsen, K., and Yi, W. (2001). Probabilistic extensions of process algebras. In *Handbook of Process Algebras*. Elsevier.

Kirch, O. (1993). Bereiche und Bewertungen. Master's thesis, Technische Hochschule Darmstadt.

MacLane, S. (1971). *Categories for the Working Mathematician*. Springer.

Mislove, M. (2000). Nondeterminism and probabilistic choice: Obeying the law. In *Proceedings of 11th CONCUR*, volume 1877 of *LNCS*, pages 350–364. Springer.

Mislove, M. (2005). Discrete random variables over domains. In *Proceedings of 32nd ICALP*, volume 3580 of *LNCS*. Springer.

Moggi, E. (1991). Notions of computation and monads. *Information and Computation*, 93(1):55–92.

Morgan, C. et al. (1994). Refinement-oriented probability for CSP. Technical Report PRG-TR-12-94, Oxford University Computing Laboratory.

Plotkin, G. D. (1983). Domains. University of Edinburgh.

Plotkin, G. D. and Power, J. (2002). Notions of computation determine monads. In *Proc of 5th FOSSACS*, volume 2303 of *LNCS*, pages 342–256. Springer.

Segala, R. (1995). *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, M.I.T.

Segala, R. and Lynch, N. (1995). Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273. An extended abstract appears in *Proceedings of 5th CONCUR*, volume 836 of *LNCS*, pages 481–496. Springer.

Stoelinga, M. (2002). An introduction to probabilistic automata. *Bulletin of the European Association for Theoretical Computer Science*, 78:176–198.

Tix, R. (1999). *Continuous D-Cones: Convexity and Powerdomain Constructions*. PhD thesis, Technische Universität Darmstadt.

Tix, R., Keimel, K., and Plotkin, G. D. (2005). Semantic domains for combining probability and non-determinism. *Electronic Notes in Theoretical Computer Science*, 129:1–104.

Varacca, D. (2002). The powerdomain of indexed valuations. In *Proceedings of 17th LICS*. IEEE Computer Society.

Varacca, D. (2003). *Probability, Nondeterminism and Concurrency. Two Denotational Models for Probabilistic Computation*. PhD thesis, BRICS - Aarhus University. Available at www.brics.dk/DS/03/14.

Vardi, M. Y. (1985). Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of 26th FOCS*, pages 327–338.