SEMANTICS OF PROBABILISTIC PROGRAMS

Dexter Kozen

IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

Abstract. Two complementary but equivalent semantic interpretations of a high level probabilistic programming language are given. One of these interprets programs as partial measurable functions on a measurable space. The other interprets programs as continuous linear operators on a Banach space of measures. It is shown how the ordered domains of Scott and others are embedded naturally into these spaces. Two general results about probabilistic programs are proved.

1. Introduction

Probabilistic algorithms have undergone a recent surge of interest among computer scientists (see references). In this paper we provide a formal denotational semantics for a simple yet powerful class of probabilistic programs. There are several reasons why this should prove worthwhile:

1) A. Yao (1977) and M. Rabin (1976) have grouped research in probabilistic algorithms into two disparate areas, which Yao has termed the *distributional approach* and the *randomized approach*. In the former, the algorithm itself is deterministic, the input is assumed to satisfy some fixed distribution, and the average behavior of the program is studied with respect to that distribution (see e.g. Knuth (1973), Karp (1976), Yao and Yao (1978)). In the latter, the input is fixed, but the program is allowed to make stochastic moves (see e.g. Rabin (1976), Solovay and Strassen (1977), Gill (1974), Adleman (1978), Miller (1975)). Yao (1977) established a connection between the two approaches by defining a measure of complexity based on each and proving their equivalence. In the semantics given here, the two approaches are completely unified.

2) Until now, the theoretical models of probabilistic computation have been relatively low-level from a programming language point of view (decision trees in Yao (1977), probabilistic Turing machines in Gill (1974), directed graphs in Gouda and Manning (1976), probabilistic finite automata in Paz (1971)). However, high level probabilistic languages have been in use since the earliest versions of FORTRAN (Backus et al. (1957)) and BASIC (see Kurtz (1978)), both of which had a random number facility. Here we consider a version of **while** programs, a higher level language more closely resembling modern procedure-oriented languages. Besides being easier to program, the **while** program model is more versatile. All the models mentioned above restrict us to a limited class of *discrete* probability measures, whereas there are certainly situations in which continuous distributions are more realistic, for instance in modeling economic systems or population growth. The semantics herein makes no distinction between discrete and continuous distributions.

3) Until now, analysis of probabilistic programs has been largely ad hoc. This may be all right for the low-level models above, since the discrete distributions they use are well understood (e.g. a "random graph" is usually taken to have every possible edge with probability 1/2, certainly an expedient choice, but not necessarily the most realistic). In the general case, sums are replaced by abstract integrals, combinatorics is replaced by analysis, and intuition is more likely to break down. It is therefore worthwhile to develop an assertion language and a system of proof rules, for the same good reasons as in the deterministic case; but proof rules do not make sense without a viable semantics, so this paper may be considered a first step in that direction.

4) Finally, and most importantly, this work recasts the usual Scott-Strachey least fixed point semantics in a unexpected mold: the theory of linear operators in Banach spaces. It is shown how the partially ordered domains of Scott (1970) and others, which originally may have appeared contrived, are in fact embedded as substructures of more conventional mathematical structures which have been studied since the 1930s. Specifically, we interpret programs and data as elements of the ordered Banach spaces of Birkhoff (1938) and Kakutani (1941). Banach spaces are normed vector spaces which are complete with respect to the metric induced by their norm. Such spaces have a wide range of applications: they form the basis of statistical mechanics, functional analy-

sis, and ergodic theory, and are also used in the study of Markov processes and differential equations. Their theory is a sublime marriage of analysis and algebra, and is among the richest of all mathematics. It therefore seems worthwhile to point out their relationship to programming language semantics, thereby putting powerful techniques at our disposal.

Section 2 of the paper is devoted to the relevant definitions and elementary results of linear analysis and probability theory, in an effort to make the paper as self-contained as possible.

In section 3 we describe the syntax of probabilistic **while** programs, which are like deterministic **while** programs (simple assignments $x_i := f(x_1,...,x_n)$, composition, conditional tests, **while** loops), except they allow calls on a random number generator (x_i := RANDOM). We then give two equivalent semantic definitions, 1 and 2. Semantics 1 is more operational and more intuitive. It is based on classical probability theory as found in Feller (1968) and Chung (1974), and is probably the first attempt one would make, especially if she were a probabilist. In semantics 1, program variables are interpreted as partial random variables defined on a probability space and taking values in a domain X. Programs with n variables are interpreted as measurable functions from X^n to X^n. The input to the program is a sequence of n random variables. The output is again a sequence of n random variables, obtained by composing the program with the input. Semantics 2 is the more denotational; it more closely resembles Scott-Strachey least fixed point semantics, since it involves partially ordered domains, namely the partially ordered Banach spaces of Birkhoff (1938) and Kakutani (1941). In Semantics 2, the input to a program S is a probability distribution μ on X^n and the output is a subprobability distribution $S(\mu)$. Programs are interpreted as continuous linear transformations on the space of distributions.

The chief advantage of Semantics 2 over Semantics 1 is that many problems of overspecification are avoided. As a result, properties concerning the probabilistic behavior of the program are more easily expressed. Another advantage of Semantics 2 is that the actual specification of the semantics itself is concise and elegant. The semantics of the **while** loop is obtained by solving a simple operator equation in $S' \cap P'$. The equation is obtained by unwinding the loop once. It turns out to be an example of one well studied by functional analysts, and it appears in many texts (e.g. Collatz, p. 358). The existence of a solution is immediate from standard theorems of functional analysis.

We also show in section 3 how an ordered domain of partial functions is embedded naturally into an ordered Banach space.

In section 4 we prove a general theorem allowing the constructions of section 3 in higher types.

In section 5 we give two results on probabilistic programs, illustrating the use of semantics 1 and 2. The first result gives a condition for program equivalence. It says that *if two programs agree whenever the input satisfies a discrete distribution, then they are equivalent.* In other words, a program's behavior is completely determined by its behavior on discrete distributions. A discrete distribution has all its mass on countably many points. Not all distributions are approximated by a sequence of discrete distributions; if this were the case, the result would be trivial, which it is not. The result is rather a manifestation of the discrete nature of programs. As a consequence of this theorem, programs can now be proved equivalent by considering their action on discrete inputs only, which can be represented by coutable sums as opposed to abstract integrals. Thus, *combinatorics can replace analysis and integration theory in equivalence proofs.*

The second result of section 5 is a concise characterization of all possible solutions of the defining equation for **while** loops. As a corollary we show that the condition *on all inputs, the loop is executed infinitely often with probability 0* is necessary and sufficient for the uniqueness of the solution.

2. Background and notation

In this section we establish notation and review some basic results that will be used in later sections. For a more careful treatment, the following texts are recommended: measure theory: Halmos (1950); probability theory: Feller (1968), Chung (1974); linear analysis: Dunford and Schwartz (1958). In addition, *Lattice Theory* by Garrett Birkhoff (1967) is an excellent introduction to partially ordered linear spaces.

2.1 Measure and probability

It is assumed that the reader is familiar with the following concepts: σ-algebra, measurable set, measurable space, cartesian product of measurable spaces, measurable function, measure, positive measure, simple function, integral, probability measure, probability space, sample point, joint distribution, independence, conditional probability.

R denotes the real numbers and ω denotes the nonnegative integers. **R** and ω also denote the measurable spaces $(\mathbf{R},L)$, $(\omega,P\omega)$, where L is the class of Lebesgue measurable sets and $P\omega$ is the power set of ω.

Let (X,M) be a measurable space. We will often denote (X,M) by just X. B, C denote measurable sets. If $B\in M$, $\bar{B}$ denotes its complement $X - B$. A *measurable partition* of $B\in M$ is a family of pairwise disjoint measurable sets whose union is B. If $B\in M$, then $\chi_B:X\rightarrow\{0,1\}$ denotes the characteristic function of B.

The cartesian product of α copies of (X,M) is denoted (X^α,M^α) or just X^α. If α is countable, then M^α is the smallest σ-algebra containing all *rectangles* $\prod_{n<\alpha} B_n$, each $B_n\in M$.

A *measure* on (X,M) is a countably additive, (finite) real valued function defined on M. Measures are denoted μ, ν. The set of all measures on (X,M) is denoted $B(X,M)$ and the set of positive measures (those assuming only nonnegative values) is denoted P.

If X and Y are two measurable spaces, and if μ, ν are measures on X and Y, respectively, then the *product* of μ and ν, denoted $\mu\times\nu$, is the unique measure on the cartesian product $X\times Y$ such that $(\mu\times\nu)(B\times C) = \mu(B)\nu(C)$ for all rectangles $B\times C$.

The *variation* or *absolute value* of μ is the map $|\mu|:M\rightarrow\mathbf{R}$ defined by

$$|\mu|(B) = \sup_{\pi} \sum_{i\in\pi} |\mu(B_i)|$$

where the supremum is taken over all countable measurable partitions π of B. $|\mu|$ is itself a measure (Dunford and Schwartz, Lemma 7, p. 128). μ is positive iff $\mu=|\mu|$. The *total variation* of a measure μ is the nonnegative real number $|\mu|(X)$, denoted $\|\mu\|$.

A *probability measure* is a positive measure with $\mu(X) = 1$. A *subprobability measure* or *distribution* is a positive measure with $\mu(X) \leq 1$. If μ is a measure and B is a measurable set, μ_B denotes the measure which annihilates μ outside of B; i.e.,

$$\mu_B(A) = \mu(A\cap B).$$

Thus if μ is a probability measure, the measure $\mu_B/\mu(B)$ gives the conditional probability of an event, given event B. The map e_B is the map taking μ to μ_B.

A measure is *discrete* if all its weight is on at most countably many points, i.e. if there exists a countable measurable set B such that $\mu = \mu_B$. If B consists of a single point, then μ is called a *point mass*. A measure is *continuous* if $\mu(B) = 0$ for all countable B.

Let (Ω,F,P) be a probability space and (X,M) a measurable space. A *partial random variable* is a partial measurable function $x:(\Omega,F,P)\rightarrow(X,M)$. In this paper, the term *random variable* will always mean partial random variable. The symbols x,y denote random variables. μ_x denotes the subprobability measure on (X,M) induced by x·

$$\mu_x(B) = P(x^{-1}(B)),$$

representing the probability of the event "$x\in B$". If x is total, then μ_x is a probability measure.

A *random vector* is a list of random variables $x_i:(\Omega,F,P)\rightarrow(X,M)$, $i \leq \alpha$, with identical domains. Thus a random vector is a random variable from (Ω,F,P) into the cartesian product (X^α,M^α). If $\bar{x} = x_1,x_2,\ldots$ is a random vector, then the subprobability measure $\mu_{\bar{x}}$ on (X^α,M^α) induced by $\bar{x}$ is called the *joint distribution* of the random variables $x_1,x_2,\ldots$. Two random variables x,y are *independent* if their joint distribution is exactly the product distribution $\mu_x\times\mu_y$.

2.2 Partially ordered linear spaces

A (real) *normed linear space B* is a vector space over **R** with norm $\|\ \|$. The norm induces a metric on B: the distance between x and y is $\|x-y\|$. If B is complete with respect to this metric, then B is called a *Banach space*. Otherwise unqualified, the words *complete, closed, open, continuous, bounded*, etc. refer to this metric. The notations Int(A), Cl(A), Ker(T), Im(T) refer respectively to the interior of a set A, the closure of A, the kernel of a linear transformation T, and the image of the entire domain under T. I denotes the identity and 0 the zero transformation.

If B, C are two normed linear spaces and if $T:B\rightarrow C$ is a linear transformation, T is *bounded* if

$$\sup_{\|x\|=1} \|T(x)\|$$

is finite. It is well known that a linear transformation is bounded if and only if it is continuous. The space of continuous linear transformations from B to C, denoted $(B\rightarrow C)$, is a normed linear space under pointwise addition and scalar multiplication, with the *uniform norm*

$$\|T\| = \sup_{\|x\|=1} \|T(x)\|,$$

so called because it characterizes uniform convergence of sequences of functions. Moreover, if C is a Banach space, then $(B\rightarrow C)$ is. An element of $(B\rightarrow B)$ is called a *linear operator* on B and an element of $(B\rightarrow\mathbf{R})$ is called a *functional* on B.

The *positive cone* P of a normed linear space B is a subset of B satisfying the two properties

$$x,y \in P \text{ and } a,b \geq 0 \rightarrow ax + by \in P$$

$$x, -x \in P \rightarrow x = 0 .$$

The first property says that P is closed under positive operations, and the second says that P is not too wide. For example, P might be the set of vectors in $\mathbf{R}^n$ with nonnegative coefficients, or the set of functions taking on only nonnegative values in the space of continuous real valued functions on some interval.

P induces a partial order $x \leq y$ iff $y-x \in P$; P is then the set of x such that $x \geq 0$ (hence the term *positive*). The order $\leq$ is *conditionally complete* if every set x_α with an upper bound has a least upper bound $\sup_\alpha x_\alpha$. There is a duality principle in force: the map $x \rightarrow -x$ is a linear automorphism taking $\leq$ to $\geq$. Addition and scalar multiplication are order continuous, i.e.

$$x + \sup_\alpha x_\alpha = \sup_\alpha x + x_\alpha$$

$$a \sup_\alpha x_\alpha = \sup_\alpha ax_\alpha, \quad a \geq 0$$

in the sense that if one side exists, then so does the other and they are equal.

A *directed set* is a subset A of B such that any pair of elements in A has an $\leq$-upper bound in A.

The pair (B,P) is a *vector lattice* if every pair $x,y \in B$ has a $\leq$-least upper bound $x \vee y$. The definition would be equivalent with $\wedge$ in place of $\vee$, by duality. Vector lattices are distributive as lattices, and addition and scalar multiplication distribute over $\vee$ and $\wedge$:

$$x + (y \vee z) = (x + y) \vee (x + z)$$

$$a(x \vee y) = ax \vee ay, \quad a \geq 0$$

$$a(x \vee y) = ax \wedge ay, \quad a \leq 0$$

and dually. Also,

$$x + y = x \vee y + x \wedge y.$$

Every x in a vector lattice can be decomposed uniquely into $x = x^+ - x^-$, where $x^+ \wedge x^- = 0$. The *absolute value* of x, denoted $|x|$, is defined to be $x^+ \vee x^- = x^+ + x^-$. The following properties hold in all vector lattices:

$$|x| \geq 0, \text{ and } |x| = 0 \rightarrow x = 0$$

$$|x - y| = x \vee y - x \wedge y$$

$$x \vee y = \tfrac{1}{2}(x + y + |x-y|).$$

If B is a Banach space and (B,P) is a vector lattice such that order and norm are related by the properties

$$\| |x| \| = \|x\|, \text{ and}$$

$$\|x+y\| = \|x\| + \|y\| \text{ for positive } x,y,$$

then (B,P) is called an *(L)-space*. A consequence of the last two properties is

$$\|x\| = \|x^+\| + \|x^-\|.$$

(L)-spaces were first defined by Birkhoff (1938) and Kakutani (1941), and have since served in numerous applications, such as ergodic theory, statistical mechanics, and solution of differential equations. The set $B(X,M)$ of measures on (X,M), together with its positive cone P of positive measures, form an (L)-space, with addition and scalar multiplication defined pointwise:

$$(\mu + \nu)(B) = \mu(B) + \nu(B)$$

$$(a\mu)(B) = a\mu(B).$$

The norm is the total variation norm described in 2.1.

3. Probabilistic while programs

In this section we describe a class of probabilistic programs called **while** programs and give two approaches to their interpretation: a more operational semantics 1 and a more denotational semantics 2. The semantics 1 is a straightforward, perhaps more intuitive interpretation we might ascribe to **while** programs at first. It is more closely related to classical probability theory as found in Feller (1968) or Chung (1974). It interprets the program variables as random variables and programs as mappings from random variables to random variables. The semantics 2, on the other hand, is closer to Scott-Strachey style semantics, as it involves partially ordered domains and least fixed points of monotone maps. The domains in question are the partially ordered Banach spaces of Birkhoff and Kakutani, as described in section 2.2.

3.1 Syntax

The programs we consider are **while** programs over the variables $x_1,\ldots,x_n$. Syntactically, they consist of five types of statements:

(3.1.1) *simple assignment*

$x_i := f(x_1,\ldots,x_n)$

(3.1.2) *random assignment*

$x_i :=$ RANDOM

(3.1.3) *composition*

S;T

(3.1.4) *conditional*

if B **then** S **else** T **fi**

(3.1.5) ***while** loop*

while B **do** S **od**

When there is no ambiguity, **fi** and **od** will be omitted.

3.2 The semantics 1

Under semantics 1, program variables and random numbers are random variables $(\Omega,F,P) \rightarrow (X,M)$. Informally, a sample program execution consists of first picking a sample point in Ω, simultaneously determining the values of the program variables and ω random numbers, which are placed on an infinite stack. The program then executes deterministically. Each time x_i := RANDOM is executed, the next random number is popped from the stack.

More formally, let (Ω,F,P) be a probability space and let (X,M) be a measurable space. Let $(X^{n+\omega},M^{n+\omega})$ be the cartesian product of $n+\omega$ copies of (X,M). The first n components represent the n program variables and the last ω represent the infinite stack of random numbers.

The B which may occur as the conditional test in 3.1.4 and 3.1.5 may be any measurable set $B \in M^n$. The f in 3.1.1 may be any partial measurable function $X^n \rightarrow X$. All the most common functions in **R**, such as + or log, are measurable.

Under these conditions, each program S denotes a partial measurable function $f_S:X^{n+\omega} \rightarrow X^{n+\omega}$, as follows:

(3.2.1) *simple assignment.* If $f:X^n \rightarrow X$ is a measurable function, the simple assignment 3.1.1 denotes the measurable function $X^{n+\omega} \rightarrow X^{n+\omega}$ which takes sequence

$a_1,\ldots,a_n,a_{n+1},\ldots$

to sequence

$a_1,\ldots,a_{i-1},f(a_1,\ldots,a_n),a_{i+1},\ldots,a_n,a_{n+1},\ldots$

(3.2.2) *random assignment.* The statement

x_i : = RANDOM

denotes the measurable function which takes sequence

$a_1,\ldots,a_n,a_{n+1},\ldots$

to sequence

$a_1,\ldots,a_{i-1},a_{n+1},a_{i+1},\ldots,a_n,a_{n+2},\ldots$

That is, the infinite stack $a_{n+1},a_{n+2},\ldots$ of random numbers is popped, and the top element is assigned to x_i.

(3.2.3) *composition.* The program S;T denotes the composition $f_T \circ f_S$.

(3.2.4) *conditional.* The conditional statement 3.1.4 denotes the measurable function which on input $\bar{a}$ gives

$f_S(\bar{a})$ if $\bar{a} \in B \times X^\omega$

$f_T(\bar{a})$ if $\bar{a} \notin B \times X^\omega$.

(3.2.5) ***while** loop.* The while statement 3.1.5 denotes the measurable function which on input $\bar{a}$ gives

$f_S{}^n(\bar{a})$ if n is least such that $f_S{}^n(\bar{a}) \notin B \times X^\omega$,

undefined if no such n exists.

The specification is completed by giving a sequence $y_{n+1},y_{n+2},\ldots$ of independent, identically distributed random variables $(\Omega,F,P) \rightarrow (X,M)$ (for the random number generator). If the input is a sequence of random variables $x_1,\ldots,x_n$, we also require that $y_{n+1},y_{n+2},\ldots$ be independent of $x_1,\ldots,x_n$. The result of applying program S to $x_1,\ldots,x_n$ is the first n components of the random vector $f_S \circ (x_1,\ldots,x_n,y_{n+1},\ldots)$.

There are some noteworthy problems with this approach, mostly centered on the fact that modeling probabilistic values as random variables results in needless overspecification. For example, the particular random number assigned to x_i in the random assignment depends on the path of execution up to that point, whereas the probabilistic behavior of the program is independent of this, since the y_i are independent and identically distributed. Along the same lines, the probabilistic flow of the program, based on conditional tests in 3.1.4 and 3.1.5, do not depend on the random vectors $\bar{x}$ themselves, but only on their distributions. Finally, if we are studying the average behavior of a deterministic program with respect to some input distribution, we are usually given only the distribution and not some random variable satisfying it. In such cases we would be forced to *invent* a sample space (Ω,F,P) and an input vector $\bar{x}:(\Omega,F,P) \rightarrow (X^n,M^n)$ satisfying that distribution. This complaint also applies to the random number generator $y_{n+1},\ldots$. These observations suggest a new approach in which random vectors with the same distribution are identified, and programs are interpreted as mappings from *distributions to distributions*, instead of from random vectors to random vectors. In so amending semantics 1, the (X^ω,M^ω) tail constructed to accomodate the random number generator becomes superfluous.

3.3 The semantics 2

In this semantics, a program S maps distributions μ on (X^n,M^n) to distributions $S(\mu)$ on (X^n,M^n). It is helpful to think of μ as a fluid mass distributed throughout X^n. Execution of the program causes this mass to flow, splitting apart at the conditional tests, flowing together again after the two sides of the conditional have been executed. In while loops, the mass μ flows around and around the loop; at each cycle, the part of the mass which occupies $\overline{B}$ breaks off and freezes, and the rest goes around the loop again. Part of the mass may go around infinitely often. This characterization is conceptually helpful in many situations; for example, suppose μ is a probability distribution (i.e. $\mu \geq 0$ and $\mu(X^n) = 1$). What is the probability that program S halts when its input satisfies distribution μ? It is exactly $S(\mu)(X^n)$, the probability of the universal event X^n upon output.

Let (X,M) be a measurable space and let $B = B(X^n,M^n)$ be the set of measures on (X^n,M^n), as defined in section 2. If we define addition and scalar multiplication on B by

$$(\mu + \nu)(B) = \mu(B) + \nu(B)$$
$$(a\mu)(B) = a(\mu(B)), \; a \in \mathbf{R}$$

then the total variation $\| \ \|$ is a norm and B forms a real Banach space as described in section 2. In addition, if P is the positive cone of B (recall $P = \{\mu \mid \mu(B) \geq 0 \text{ for all } B\}$), with induced partial order $\mu \leq \nu$ iff $\nu - \mu \in P$, then (B,P) becomes a conditionally complete vector lattice and even an (L)-space, as defined in section 2. The set of measures satisfying $\|\mu\| \leq 1$ is the *closed unit ball* S of B. Thus the distributions are the elements of $S \cap P$.

In semantics 2, programs will be interpreted as mappings $S \cap P \rightarrow S \cap P$. It will turn out, however, that each program will extend *uniquely* to a *linear transformation* $B \rightarrow B$. Moreover, this extension will be *continuous* with respect to the metric induced by $\| \ \|$. As outlined in section 2, the space $B' = (B \rightarrow B)$ of continuous linear transformations or *operators* $B \rightarrow B$ forms a real Banach space under the *uniform norm*

$$\|T\| = \sup_{\|\mu\| = 1} \|T(\mu)\|$$

and pointwise addition and scalar multiplication. Thus programs may be interpreted as elements of this space.

Define P' as the set of $T \in B'$ which preserve P. It is not hard to show that P' is a positive cone, and so induces a partial order $\leq$ on B'. The elements of P' are exactly the continuous linear transformations $T: B \rightarrow B$ which are *isotone* with respect to the order $\leq$ in B, i.e.

$$\mu \leq \nu \;\rightarrow\; T(\mu) \leq T(\nu).$$

Also, $S \leq T$ if and only if $S(\mu) \leq T(\mu)$ for all $\mu \in P$. These observations are easy consequences of the properties of partially ordered linear spaces listed in section 2.

In addition to P', define S' as the set of $T \in B'$ which preserve S. By the definition of the uniform norm, S' is exactly the closed unit ball of B'. By construction, programs will preserve both S and P, thus they may be interpreted as elements of $S' \cap P'$.

Although B' is not necessarily a vector lattice, it is conditionally complete in the sense that any set of elements with an $\leq$-upper bound has a $\leq$-least upper bound. Dually, any set with a lower bound has a greatest lower bound. In particular, any pair of elements S, T bounded above have a join $S \vee T$ given by

$$(S \vee T)(\mu) = \sup_{0 \leq \nu \leq \mu} S(\nu) + T(\mu - \nu) \quad \text{for} \;\; \mu \in P$$
$$(S \vee T)(\mu) = (S \vee T)(\mu^+) - (S \vee T)(\mu^-) \quad \text{for} \;\; \mu \notin P$$

and meet $S \wedge T$ given by

$$S \wedge T = -(-S \vee -T).$$

Any pair S, T of positive elements has a least upper bound, since they are bounded above by S+T. The assumption that S and T are bounded above is necessary in the above definition; without it, we do not know whether the set $\{S(\nu) + T(\mu - \nu) \mid 0 \leq \nu \leq \mu\}$ has an upper bound at all (see Birkhoff (1967), p. 365).

More important for our purposes than conditional completeness, however, is the property: *If* S_α *is any directed set in* B *(or* B'*), and if the set* S_α *is metrically bounded (that is, all* $\|S_\alpha\| \leq c$ *for some constant* c*), then* S_α *has a least upper bound* $\sup_\alpha S_\alpha$. *Moreover, if the* S_α *are all positive, then their norms converge to the norm of their least upper bound.* This will allow us to obtain the semantics of the **while** loop without the usual appeal to order continuity, although this will be more useful in higher types than here. Since this property is a special case of Theorem 4.5 below, we defer its proof to the next section. A proof for the space B may be found in Birkhoff (1967, Theorem 21, p. 371).

We are now ready to describe the semantics 2. In order to understand definitions 3.3.1-3.3.5 of semantics 2, it is helpful to keep in mind the definitions 3.2.1-3.2.5 of semantics 1, which are more operational and perhaps closer to the reader's intuition. At each of the five steps, it is straightforward to verify that the two semantic definitions are equivalent, in the sense made precise by Theorem 3.3.9 below.

(3.3.1) *simple assignment.* If $f: X^n \rightarrow X$ is any measurable function, then the meaning of

$$x_i := f(x_1, \ldots, x_n)$$

is the operator in B' taking

$$\mu \;\rightarrow\; \mu \circ \hat{f}^{-1},$$

where $\hat{f}:X^n \to X^n$ is the measurable function

$$a_1,\ldots,a_n \to a_1,\ldots,a_{i-1},f(a_1,\ldots,a_n),a_{i+1},\ldots,a_n.$$

Since f is measurable, so is $\hat{f}$, thus $\mu\circ\hat{f}^{-1}$ is indeed a measure.

(3.3.2) *random assignment.* The meaning of

$$x_i := \text{RANDOM}$$

is the operator which takes μ to ν, where

$$\nu(B_1\times\ldots\times B_n) =$$
$$\mu(B_1\times\ldots\times B_{i-1}\times X\times B_{i+1}\times\ldots\times B_n)\rho(B_i),$$

where ρ is a fixed distribution, the distribution of the random number generator. Since M^n is generated by rectangles of the form $B_1\times\ldots\times B_n$, ν is well defined.

(3.3.3) *composition.* The meaning of the program S;T is the composition of operators $T\circ S$.

(3.3.4) *conditional.* The meaning of

if B **then** S **else** T

is the operator

$$S\circ e_B + T\circ e_{\bar{B}}.$$

Here e_B is the operator $\mu\to\mu_B$ and + is addition in B'.

(3.3.5) ***while*** *loop.* We want equivalence between 3.1.5, the program

while B **do** S,

and the program

(3.3.6) **if** $\bar{B}$ **then** I **else** S;**while** B **do** S **od fi** ,

obtained by unwinding the loop once. Accordingly, using the composition and conditional semantics already defined, the meaning of 3.1.5 must be a solution of

(3.3.7) $\quad W = e_{\bar{B}} + W\circ S\circ e_B.$

This is a case of a simple operator equation scheme studied in functional analysis, and there are many techniques available for its solution (see e.g. Collatz (1966), p. 358). The most common approach is to search for a fixed point of the affine transformation $\tau:B'\to B'$ defined by

$$\tau(W) = e_{\bar{B}} + W\circ S\circ e_B.$$

The existence of such a fixed point is relatively easy to establish, using Theorem 4.5. First, note that the affine map τ is isotone with respect to $\leq$ in B', and that τ preserves S'. If A is the set of elements of $S'\cap P'$ such that $W \leq \tau(W)$, then A is nonempty (it contains 0) and is closed under suprema of directed sets, by Theorem 4.5. By Zorn's Lemma, A has a maximal element W, so W must be a fixed point.

Once the existence of a solution to 3.3.7 has been established, it is easy to show that

$$W_0 = \inf\{W\in S'\cap P' \mid \tau(W) = W\}$$

is the unique least such solution in $S'\cap P'$. First, W_0 exists, since the set of fixed points of τ in $S'\cap P'$ is bounded below by 0, and since B' is conditionally complete. Since τ is isotone, $\tau(W_0) \leq \tau(W) = W$ for any fixed point W, so $\tau(W_0) \leq W_0$. By Tarski's theorem (see Birkhoff, exercise 6, p. 116), τ has a fixed point in the interval $[0,W_0]$. This fixed point must be W_0.

As is customary, it is the least fixed point W_0 which we take as the meaning of the program **while** B **do** S.

It may also be shown by a well known construction that the supremum of the sequence

$$\tau^n(0) = \sum_{k=0}^{n-1} e_{\bar{B}}\circ(S\circ e_B)^k$$

is exactly W_0, by showing that τ is order continuous. The present approach was used instead to illustrate a more general technique, which is still applicable even in the absence of order continuity. This is discussed further in the next section.

The following theorem asserts that the constructions above indeed give elements of $S'\cap P'$. The proof is by induction on program structure, treating each of the five cases 3.1.1-3.1.5 separately; it is quite straightforward and is left to the reader.

Theorem 3.3.8. If S is any **while** program, then the interpretation of S under semantics 2 is an operator in $S'\cap P'$; that is,

- S is a continuous linear transformation from the space $B(X^n,M^n)$ of measures on (X^n,M^n) to itself;
- S takes positive elements of B to positive elements;
- $\|S\| \leq 1$. □

Let $\bar{x}:(\Omega,F,P)\to(X^{n+\omega},M^{n+\omega})$ be any random vector such that the components $x_{n+1},x_{n+2},\ldots$ are independent of $x_1,\ldots,x_n$ and are themselves independent and identically distributed with distribution ρ, and let μ be the distribution on X^n induced by the first n components of $\bar{x}$. Then $\bar{x}$ has distribution $\mu\times\rho^\omega$. If program S is applied to $\bar{x}$ under semantics 1, the result is $f_S\circ\bar{x}$, with distribution $(\mu\times\rho^\omega)\circ f_S^{-1}$. In light of this, the following theorem asserts the equivalence of semantics 1 and semantics 2.

Theorem 3.3.9. Let S be a program. Then for all $\mu\in B(X^n,M^n)$, $B\in M^n$,

$$S(\mu)(B) = (\mu\times\rho^\omega)\circ f_S^{-1}(B\times X^\omega). \quad □$$

3.4 Encoding deterministic semantics

It is obvious how deterministic semantics is a special case of probabilistic semantics: eliminate the random assignment, and restrict input distributions to point masses. In fact, many of the partially ordered domains encountered in Scott-Strachey style denotational semantics are naturally embedded in partially ordered Banach spaces.

As an example, consider the domain $\mathrm{Pfn}(\omega \to \omega)$, the space of partial functions $\omega \to \omega$, with the usual ordering $\sqsubseteq$ and bottom (least defined) element $\perp$. We may embed this space into a partially ordered Banach space in such a way that $\sqsubseteq$ becomes $\leq$ and $\perp$ becomes 0, and the elements of $\mathrm{Pfn}(\omega \to \omega)$ are all members of $S' \cap P'$. This construction was in fact foreshadowed by Zeiger (1969).

First, endow ω with a class of measurable sets. For this purpose we use the power set $\mathrm{P}\omega$. Let $B = B(\omega, \mathrm{P}\omega)$, the Banach space of measures. Elements of B may be viewed as formal sums

$$\sum_{x \in \omega} a_x x$$

where the coefficients a_x are real numbers such that

$$\sum_{x \in \omega} |a_x| \leq \infty.$$

The total variation norm is given by

$$\| \sum_{x \in \omega} a_x x \| = \sum_{x \in \omega} |a_x|.$$

Let P be the cone of positive measures and let S be the closed unit ball.

As usual, a "flat domain" is constructed from ω by appending a bottom element $\perp$ to ω and defining an order $\sqsubseteq$ on $\omega \cup \{\perp\}$ so that $\perp \sqsubseteq \perp \sqsubseteq x \sqsubseteq x$ for all $x \in \omega$, but no other inequality holds. Partial functions $\omega \to \omega$ may then be viewed as $\sqsubseteq$-isotone functions $\omega \cup \{\perp\} \to \omega \cup \{\perp\}$.

If points in ω are mapped to their corresponding point masses in B, and if $\perp$ is mapped to 0 in B, then the result is an embedding of $\omega \cup \{\perp\}$ into $S \cap P$ which takes $\sqsubseteq$ into $\leq$.

Let us identify each element of ω with its corresponding point mass in B, and think of ω as a subset of $S \cap P$. Under this identification, each function $t: \omega \cup \{\perp\} \to \omega \cup \{\perp\}$ becomes a partial function $S \cap P \to S \cap P$; the function t extends uniquely to a linear transformation $T: B \to B$ by taking

$$T(\sum_{x \in \omega} a_x x) = \sum_{x \in \omega} a_x t(x).$$

Moreover, it is a simple matter to verify that $\|T\|$ is bounded, and indeed $\|T\| \leq 1$; and that T preserves P. This says that T is in both the positive cone and the closed unit ball of the space B' of operators.

Under this embedding of $\mathrm{Pfn}(\omega \to \omega)$ in B', the totally undefined function on ω is mapped to 0, as desired; moreover, $\sqsubseteq$ in $\mathrm{Pfn}(\omega \to \omega)$ is mapped to $\leq$ in B'.

4. Extension to higher types

In this section we will prove a general theorem which allows the least fixed point construction in higher types. As a corollary we show that the intersection $S \cap P$ of the positive cone and the closed unit ball of all higher types forms a conditionally complete lattice.

This is done by defining the *positive uniform norm* $\| \ \|_P$ on higher types. This norm is equivalent to $\| \ \|$, in the sense that the two norms are bound to each other by a multiplicative factor; in fact, they are identical in B'. It is then shown, using the new norm $\| \ \|_P$, that

Theorem 4.5. Every type satisfies the property: any metrically bounded directed set S_α has a least upper bound $S = \sup_\alpha S_\alpha$. Moreover, if the S_α are all positive, then $\|S_\alpha\|_P \to \|S\|_P$.

This property, along with conditional completeness, allows the least fixed point construction of the previous section to be applied verbatim in all higher types.

In the process of proving Theorem 4.5 we will obtain some insight into the respective roles of order and norm. Scott and others used order exclusively. Their strategy was based on the view that "all semantically meaningful functions should be [order] continuous" (Lehmann, p. 123). However, there is no *a priori* reason for this restriction; it is motivated more by technical convenience than by practical experience. Although all elements of B' are order continuous (use Birkhoff, Theorem 21, p. 371), many potentially interesting operators in higher types are not. Theorem 4.5 gives a more general method of obtaining a fixed point which does not require order continuity, but only the weaker property of isotonicity.

Suppose C, D are Banach spaces with positive cones P_C, P_D and closed unit balls S_C, S_D. Suppose also that C is the closure of the linear span of P_C. As above, if E is the space $(C \to D)$, and if P_E is the set of elements of E mapping P_C into P_D (i.e. P_E is the set of maps which are isotone with respect to $\leq$ in C and D), then P_E is a positive cone in E.

Let a *type* be defined recursively as either B or the closed subspace of $(C \to D)$ generated by the positive cone, where C and D are types. Every type is at once a Banach space (being a closed linear subspace of $(C \to D)$) and a partially ordered space whose positive cone consists of the isotone operators.

In addition to the uniform norm

$$\|T\| = \sup_{S_C} \|T(x)\|$$

let us define a new norm $\| \, \|_P$ on types: $\| \, \|_P = \| \, \|$ in the base type B, and for higher types,

$$\|T\|_P = \sup_{S_C \cap P_C} \|T(x)\|_P.$$

Lemma 4.1. In any type, the norms $\| \, \|$ and $\| \, \|_P$ are equivalent; that is, there exists a constant k such that $\|T\|_P \leq k\|T\|$ and $\|T\| \leq k\|T\|_P$ for any T.

Proof. The proof is by induction on type structure. For the base type B, there is nothing to prove. Now suppose the two norms are equivalent in types C and D. Using Dunford and Schwartz, Theorem 18, p. 55, it is straightforward to show that $(C \to D)$ is complete in the metric induced by $\| \, \|_P$, so it is a Banach space with respect to this norm. It follows inductively from the definitions that $\|T\|_P \leq \|T\|$ for any T, thus the topology induced by $\| \, \|$ refines that induced by $\| \, \|_P$. But then the two topologies must coincide (Dunford and Schwartz, Theorem 5, p. 58; this is a consequence of the so-called "open mapping principle.") Since the $\| \, \|$-open unit ball $\mathrm{Int}(S)$ is open in the $\| \, \|_P$ topology, there is a $\| \, \|_P$ neighborhood of radius ϵ about 0 entirely contained in $\mathrm{Int}(S)$. Take $k = 1/\epsilon$. □

In the case of B', the two norms are in fact identical:

Lemma 4.2. $\|T\| = \|T\|_P$ for $T \in B'$.

Proof. Clearly $\|T\|_P \leq \|T\|$, and for any $T \in B'$, $\mu \in S$,

$$\begin{aligned}&\|T(\mu)\|\\ &\leq \|\mu^+\| \, \|T(\mu^+/\|\mu^+\|)\| + \|\mu^-\| \, \|T(\mu^-/\|\mu^-\|)\|\\ &\leq (\|\mu^+\| + \|\mu^-\|)\|T\|_P,\end{aligned}$$

and

$$\|\mu^+\| + \|\mu^-\| = \|\mu\| \leq 1,$$

since B is an (L)-space. □

Lemma 4.3. Every type satisfies the property: if $0 \leq S \leq T$, then $\|S\|_P \leq \|T\|_P$.

Proof. The proof is by induction on type structure. Since B is an (L)-space, $0 \leq \mu \leq \nu$ implies $\|\mu\| \leq \|\nu\|$. Now suppose the lemma holds for types C and D. Let $0 \leq S \leq T$, $S, T \in (C \to D)$. Then since $0 \leq S(x) \leq T(x)$ for all $x \in S \cap P$,

$$\begin{aligned}\|S\|_P &= \sup_{S \cap P} \|S(x)\|_P\\ &\leq \sup_{S \cap P} \|T(x)\|_P = \|T\|_P\end{aligned}$$

by induction hypothesis. □

It follows immediately from 4.1 and 4.3 that

Corollary 4.4. In any type, every order bounded set of positive elements is metrically bounded.

Theorem 4.5. Every type satisfies the property: any metrically bounded directed set S_α has a least upper bound $S = \sup_\alpha S_\alpha$. Moreover, if the S_α are all positive, then $\|S_\alpha\|_P \to \|S\|_P$.

Proof. The basis is given by Birkhoff (1967, Theorem 21, p. 371). Now suppose C and D are types satisfying the theorem, and let S_α be a metrically bounded directed set in $(C \to D)$. For any fixed $x \in C$, $x \geq 0$, the set $S_\alpha(x)$ is metrically bounded in D: $\|S_\alpha(x)\|_P \leq \|S_\alpha\|_P \|x\|_P$. Also, $S_\alpha(x)$ is a directed set. Thus by the induction hypothesis, $\sup_\alpha S_\alpha(x)$ exists for positive x; let us call this $S(x)$. Moreover if the S_α are all positive, then $\|S_\alpha(x)\|_P \to \|S(x)\|_P$. As addition and scalar multiplication are order continuous, S is linear on the positive cone P of C; and since C is generated by P, S extends uniquely to an operator defined on all of C, by Birkhoff (1967, Lemma 2, p. 365). It is not hard to show that S is the least upper bound $\sup_\alpha S_\alpha$. In addition, if the S_α are positive, then

$$\begin{aligned}\|S\|_P &= \sup_{S \cap P} \|S(x)\|_P\\ &= \sup_{S \cap P} \|\sup_\alpha S_\alpha(x)\|_P\\ &= \sup_{S \cap P} \sup_\alpha \|S_\alpha(x)\|_P\\ &= \sup_\alpha \|S_\alpha\|_P. \; □\end{aligned}$$

This theorem implies the existence of a fixed point in $S \cap P$ in all types, by the argument of section 3. However, in order to obtain the least fixed point by taking the infimum of fixed points, we need:

Corollary 4.6. Any type is conditionally complete.

Proof. Again, the proof is by induction on type. It is certainly true for B, so consider the higher type $(C \rightarrow D)$. By Birkhoff (1967, Theorem 17, p. 365) and the induction hypothesis, the positive cone of $(C \rightarrow D)$ is a vector lattice.

Now suppose S_α is a set of elements bounded above by S. We may assume without loss of generality that the S_α are all positive, since we can translate a cofinal subset of a directed set to the positive cone by adding a constant, without affecting $\leq$. We may also assume that the set of S_α is closed under the join operation, since the supremum is not affected. By 4.4, S_α is metrically bounded, so its supremum exists by 4.5.

The situation is similar for greatest lower bounds, by duality.

□

5. Two results on probabilistic programs

In this section we use some of the tools developed in the previous sections to prove two general results about probabilistic programs.

5.1 Discrete measures and program equivalence

In section 3 we showed that all probabilistic programs denote elements of $S' \cap P'$. Not all elements of $S' \cap P'$ are denoted by programs, however, so it is natural to look for a characterization of those which are. Theorem 5.1.1 below sheds some light on this question. It says that *all programs are completely determined by their behavior on inputs satisfying discrete distributions*. That is, if S and T are two programs whose outputs $S(\mu)$ and $T(\mu)$ agree whenever μ is discrete, then S = T under semantics 2.

It is *not* the case that any measure can be approximated by a sequence of discrete measures: the discrete measures form a closed linear subspace of B, so any convergent sequence of discrete measures converges to a discrete measure. In fact, any measure μ can be decomposed uniquely into its discrete and continuous parts $e_{disc}(\mu)$ and $\mu - e_{disc}(\mu)$. The projection e_{disc} which takes a measure into its discrete part is a continuous linear transformation in $S' \cap P'$, given by sup e_B where the supremum is taken over all countable measurable sets B. This supremum exists and is in $S' \cap P'$ by Theorem 4.5. The projection $I - e_{disc}$, also in $S' \cap P'$, takes a measure into its continuous part. There are certainly distinct elements of $S' \cap P'$ which agree on the discrete measures; I and e_{disc}, for example. Since I is given by a program, Theorem 5.1.1 says that there is no program to compute e_{disc}.

Since the subspace of discrete measures is the closure of the linear span of the point masses, and since programs are linear and continuous, Theorem 5.1.1 also says that the behavior of a program is completely determined by its behavior on point masses. Thus to check whether two programs are equivalent, we need only check whether they are equivalent whenever the input satisfies a point mass distribution, i.e. is constant with probability 1.

It is relatively easy to see why 5.1.1 holds in the deterministic case, i.e. in the absence of random assignments. In the usual deterministic semantics, a program S has only countably many halting computation paths, and each such path is given by a program S_i without conditional tests or **while** loops. Moreover, the set of inputs which follow that computation path is a measurable set B_i, since it is a Boolean combination of measurable sets occurring in conditional tests along the path. The complement of the union of these B_i are all the inputs on which S does not halt; call this set B_0. Then the set of all B_i forms a countable measurable partition π, and

$$S = \sum_\pi S_i \circ e_{B_i}.$$

We can use this characterization to construct discrete measures which account for all "behavior patterns", by picking a representative point from each partition element and assigning it a nonzero weight.

In the presence of a random assignment, however, the situation is somewhat more complicated. For one thing, no such notion of "countably many behavior patterns" exists, even if the distribution of the random number generator is discrete. For example, it is an easy exercise to construct a probabilistic program with only a fair coin for a random number generator which, given real number x with probability one, $0 \leq x \leq 1$, halts with probability exactly x. In this example, there are uncountably many behavior patterns, one for each $0 \leq x \leq 1$.

In general, the situation is even more complicated than this. The random number generator may satisfy any arbitrary distribution. The result of any call on the random number generator not only may be used for deciding which path to take in an execution, but also may be added, multiplied, or in general combined with any other random number or input in any (measurable) way. Nevertheless, it is still true that program behavior is determined by the discrete inputs:

Theorem 5.1.1. If S, T are programs such that $S(\mu) = T(\mu)$ for all discrete $\mu \in B(X^n, M^n)$, then $S = T$.

Proof. Suppose S and T agree on discrete measures. In order to show $S = T$, by linearity it suffices to show that S and T agree on an arbitrary positive measure μ.

According to the semantics 1 of section 3.2, S and T denote partial measurable functions $f_S, f_T : X^{n+\omega} \rightarrow X^{n+\omega}$, respectively, and according to Theorem 3.3.9, it suffices to show that

$$(\mu \times \rho^\omega) \circ f_S^{-1}(B \times X^\omega) = (\mu \times \rho^\omega) \circ f_T^{-1}(B \times X^\omega),$$

where $B \in M^n$ is arbitrary.

Let π be the finite measurable partition of $X^{n+\omega}$ generated by the measurable sets $f_S^{-1}(B \times X^\omega)$, $f_T^{-1}(B \times X^\omega)$. For each $x \in X^n$, $A \in \pi$, let

$$A_x = \{ y \in X^\omega \mid (x,y) \in A \}.$$

Then A_x is a measurable set in X^ω (Halmos, Theorem A, p. 141), and

$$(\mu \times \rho^\omega)(A) = \int_{X^n} \rho^\omega(A_x) d\mu$$

(Halmos, Theorem B, p. 144). Let $\epsilon \geq 0$ be arbitrarily small. By definition of integral, there is a simple function s_A with

(5.1.2) $0 \leq s_A(x) \leq \rho^\omega(A_x)$ for all x,

such that

$$\int_{X^n} \rho^\omega(A_x) d\mu - \int_{X^n} s_A(x) d\mu \leq \epsilon,$$

or in other words

(5.1.3) $(\mu \times \rho^\omega)(A) - \int_{X^n} s_A(x) d\mu \leq \epsilon.$

The simple function s_A is defined in terms of a finite measurable partition of X^n; by taking the least common refinement of these partitions over all $A \in \pi$, we may assume all the s_A are defined in terms of the same partition. Thus there is a finite measurable partition σ of X^n such that

$$s_A = \sum_{C \in \sigma} a_{A,C} \chi_C,$$

where $0 \leq a_{A,C}$.

Construct a discrete measure ν on X^n which agrees with μ on all elements of σ. This is done by choosing an element x_C from each $C \in \sigma$ and assigning it weight $\mu(C)$.

Now, by 5.1.2,

$$s_A(x_C) = a_{A,C} \leq \rho^\omega(A_{x_C}),$$

so

(5.1.4) $$\int_{X^n} s_A d\mu = \sum_{C \in \sigma} a_{A,C} \mu(C)$$
$$\leq \sum_{C \in \sigma} \rho^\omega(A_{x_C}) \mu(C)$$
$$= (\nu \times \rho^\omega)(A).$$

Also,

(5.1.5) $$(\nu \times \rho^\omega)(X^{n+\omega}) = (\nu \times \rho^\omega)(\bigcup_{A \in \pi} A)$$
$$= \sum_{A \in \pi} \sum_{C \in \sigma} \rho^\omega(A_{x_C}) \mu(C)$$
$$= \sum_{C \in \sigma} \mu(C) \sum_{A \in \pi} \rho^\omega(A_{x_C})$$
$$= \mu(X^n) \rho^\omega(X^\omega)$$
$$= (\mu \times \rho^\omega)(X^{n+\omega}).$$

By 5.1.3, 5.1.4, and 5.1.5, for any $A \in \pi$, $(\mu \times \rho^\omega)(A)$ and $(\nu \times \rho^\omega)(A)$ differ by no more than $|\pi| \epsilon$, where $|\pi|$ is the cardinality of π. Since $f_S^{-1}(B \times X^\omega)$ and $f_T^{-1}(B \times X^\omega)$ are each the disjoint union of two elements of π, we have that $(\mu \times \rho^\omega)(f_S^{-1}(B \times X^\omega))$ and $(\nu \times \rho^\omega)(f_S^{-1}(B \times X^\omega))$ differ by no more than $2|\pi|\epsilon$, and similarly for $f_T^{-1}(B \times X^\omega)$. By assumption, S and T agree on discrete measures, thus

$$(\nu \times \rho^\omega)(f_S^{-1}(B \times X^\omega)) = (\nu \times \rho^\omega)(f_T^{-1}(B \times X^\omega)),$$

by 3.3.9. Therefore, $(\mu \times \rho^\omega)(f_S^{-1}(B \times X^\omega))$ and $(\mu \times \rho^\omega)(f_T^{-1}(B \times X^\omega))$ differ by no more than $4|\pi|\epsilon$. As ϵ was arbitrary, the proof is complete. □

5.2 Overdefinition and the uniqueness of fixed points

In this section we give a necessary and sufficient condition for the uniqueness of the solution of 3.3.7 in $S' \cap P'$. Arguments of section 3 give at least one solution, W_0. However, there may be others: for example, *every* element of $S' \cap P'$ is a solution to 3.3.7 when $B = X^n$ and S is the identity. If there are two solutions to 3.3.7 in $S' \cap P'$ then there are a continuum of solutions, since {fixed points of $\tau\} \cap S' \cap P'$ is a convex set. Theorem 5.2.3 gives a picture of all these solutions.

Why should we be interested in alternative solutions of 3.3.7? The least fixed point definition of programs is often the least defined, and in a sense, it is better to have *some* definition than none at all, provided certain useful properties are not sacrificed. This occurs often in real programming languages, viz., call-by-name in ALGOL and outside-in evaluation in LISP. Let us call such interpretations *overdefined*. In probabilistic programs, **while** loops can sometimes be overdefined without sacrificing membership in $S' \cap P'$.

Call a program S *total* if $\|S\| = 1$, that is, on any input satisfying any probability distribution whatsoever, S halts with probability 1. Total programs correspond roughly to deterministic programs which halt on all inputs. In such cases, the meaning of the program is completely determined, and there is no room for overdefinition. More precisely, *if the program* **while** B **do** S *is total, then* W_0 *is the unique solution of 3.3.7 in* $S' \cap P'$. This is a special case of Corollary 5.2.5 below.

What of the cases in which **while** B **do** S has a nonzero probability of getting stuck in the loop on some input satisfying probability distribution μ? Call this probability the *residue* of **while** B **do** S on μ. In such cases overdefinition seems possible, but the action of **while** B **do** S on μ is forced with all but the residue probability, so we have only the residue at most to work with.

It will turn out that the following procedure is an acceptable method of overdefinition: on input μ, choose a random element of the universe (according to some predetermined distribution δ) and let that be the output of the program **while** B **do** S with the residue probability. This is proved in Corollary 5.2.4. Thus alternative solutions to 3.3.7 in $S' \cap P'$ are always possible as long as the residue is nonzero.

The following lemma gives the formal definition of residue.

Lemma 5.2.1. Let **while** B **do** S be an arbitrary but fixed **while** loop. There is a positive functional $\phi: B \to \mathbf{R}$, $\|\phi\| \leq 1$, defined by

$$\phi(\mu) = \lim_m (S \circ e_B)^m(\mu)(X^n).$$

Proof. Use Birkhoff (1967, Lemma 2, p. 365). □

Note that for positive μ,

$$\phi(\mu) = \inf_m \|(S \circ e_B)^m(\mu)\|.$$

The functional ϕ is called the *residue functional* of the while loop **while** B **do** S. The formal definition of ϕ, given in 5.2.1, captures the intuitive meaning as outlined in the preceding paragraphs; that is, for probability distributions μ,

$$e_B \circ (S \circ e_B)^m(\mu)(X^n)$$

gives the probability that the loop is executed at least m+1 times, thus $\phi(\mu)$ gives the probability that it is executed infinitely often.

Lemma 5.2.2. If $\mu \in P$, then $\phi(\mu) + \|W_0(\mu)\| \leq \|\mu\|$.

Proof. For $\mu \in P$, it is straightforward to prove by induction on m that

$$\|\sum_{k=0}^{m-1} e_{\bar{B}} \circ (S \circ e_B)^k(\mu) + (S \circ e_B)^m(\mu)\| \leq \|\mu\|,$$

using the properties of (L)-spaces; thus

$$\|\sum_{k=0}^{m-1} e_{\bar{B}} \circ (S \circ e_B)^k(\mu)\| \leq \|\mu\| - \|(S \circ e_B)^m(\mu)\|$$

$$\leq \|\mu\| - \phi(\mu).$$

Since

$$W_0 = \sup_m \tau^m(0)$$

$$= \sum_{k=0}^{m-1} e_{\bar{B}} \circ (S \circ e_B)^k,$$

we have

$$\|W_0(\mu)\| = \sup_m \|\sum_{k=0}^{m-1} e_{\bar{B}} \circ (S \circ e_B)^k(\mu)\|,$$

and the result follows. □

Theorem 5.2.3. $W_0 + W$ is a solution to 3.3.7 in $S' \cap P'$ if and only if $W \in S' \cap P'$ and $W = W \circ S \circ e_B$.

Proof. To say $W_0 + W$ is a solution to 3.3.7 is to say that it is a fixed point of τ, i.e.

$$W_0 + W = \tau(W_0 + W)$$

$$= e_{\bar{B}} + (W_0 + W) \circ S \circ e_B$$

$$= \tau(W_0) + W \circ S \circ e_B.$$

Since W_0 is a fixed point, it follows that $W_0 + W$ is a solution to 3.3.7 iff $W = W \circ S \circ e_B$.

Now suppose $W_0 + W$ is a solution to 3.3.7 in $S' \cap P'$. Since W_0 is the infimum of such solutions, $W_0 \leq W_0 + W$, thus $W \in P'$. Also, $W \in S'$ by 4.2 and 4.3.

Finally, suppose $W \in S' \cap P'$ and $W = W \circ S \circ e_B$. Clearly $W_0 + W \in P'$. To show it is also in S', we will show that for all positive μ,

$$\|W(\mu)\| \leq \phi(\mu).$$

The result will then follow from 4.2 and 5.2.2.

Since $W = W \circ S \circ e_B$, we have $W = W \circ (S \circ e_B)^n$ for all n, by composing on the right with $S \circ e_B$. Thus for all $\mu \geq 0$,

$$\|W(\mu)\| = \|W((S \circ e_B)^n(\mu))\|$$

$$\leq \|W\| \, \|(S \circ e_B)^n(\mu)\|$$

$$\leq \|(S \circ e_B)^n(\mu)\|,$$

since $\|W\| \leq 1$. The result follows from the definition of ϕ. □

The following corollary gives a large class of solutions to 3.3.7 in $S' \cap P'$, as described above.

Corollary 5.2.4. Let δ be an arbitrary element of $S \cap P$. Then $W_0 + W$ is a solution of 3.3.7 in $S' \cap P'$, where

$$W(\mu) = \phi(\mu)\delta.$$

Proof. Clearly, $W_0 + W$ is in P', since both W_0 and W are.

To show $W_0 + W$ is in S', note

$$\| W_0 + W \| = \sup_{\mu \in S \cap P} \| W_0(\mu) + W(\mu) \| .$$

But for all $\mu \in S \cap P$,

$$\begin{aligned} &\| W_0(\mu) + W(\mu) \| \\ &\leq \| W_0(\mu) \| + \| W(\mu) \| \\ &\leq \| \mu \| , \end{aligned}$$

by 5.2.2. Finally, to show $W_0 + W$ is a solution of 3.3.7, by 5.2.3 it suffices to show that $W = W \circ S \circ e_B$. But this follows easily from the fact that $\phi = \phi \circ S \circ e_B$. □

By 5.2.3 and 5.2.4, we have a necessary and sufficient condition for the uniqueness of the solution W_0:

Corollary 5.2.5. W_0 is the unique solution to 3.3.7 in $S' \cap P'$ if and only if the residue functional ϕ vanishes identically. □

In other words, W_0 is the unique solution iff, on any input, **while** B **do** S loops infinitely often with probability 0.

6. Conclusion

This paper has only scratched the surface of formal analysis of probabilistic programs. Of the many possible directions for further work, probably the most beneficial would be the study of assertion languages which can express such statements as "this program halts with probability at least p" and "this program takes at most n^k steps with probability 2^{-k}," and systems of axioms and proof rules which might then be shown sound and complete relative to analysis. Ramshaw (1979) has made considerable progress in this direction.

One interesting side effect of establishing proof rules would be that *probabilistic programs could be used to formulate and solve problems of probability theory in a more natural way.* Many interesting topics in probability theory deal with iterated experiments (random walk, Markov chains, etc.) and probabilists bend over backwards to remove all traces of this dynamic aspect from the theory. However, it is quite straightforward to express a random walk problem as a halting problem for a probabilistic program. This seems much more natural than the usual formulation of the problem in terms of a random variable on the set of infinite sequences of moves.

Another area for further investigation concerns correct program transformations. Clearly all correct deterministic program transformations are also correct for probabilistic programs, but a moment's reflection tells us that, in the presence of a random number generator, there may be many more.

Finally we ask for a characterization of the computable distributions. Is there a "Church's thesis" of computable distributions? If so, can they be computed by probabilistic **while** programs, say with only a fair coin for a random number generator?

This research has raised an interesting point regarding nondeterminism. Although both nondeterministic and probabilistic semantics are extensions of deterministic semantics, neither may be encoded in the other, at least in any obvious way. For example, if a nondeterministic **while** loop has arbitrarily long computations, then it must have an infinite conputation, by König's Lemma; whereas a probabilistic **while** loop may run any given number of steps with nonzero probability, yet terminate with probability one. If this is the case, why has so much effort been spent developing nondeterministic semantics to the exclusion of probabilistic semantics? Scott-style semantics does not extend easily to encompass nondeterminism. None of the powerdomain constructions (Plotkin (1976), Lehmann (1976)) are truly satisfying. On the other hand, the probabilistic construction herein extends deterministic semantics quite simply and naturally. The advantage of the probabilistic over the nondeterministic is not only technical, but also very practical: in a large operating system, although there may be an extremely small probability of a certain sequence of events, nondeterministic models must treat that sequence on an equal footing with any other sequence. It may be more reasonable to ignore that possibility entirely and save the overhead required to check for that special case. Perhaps it is time to speak less of total correctness or partial correctness and more of correctness with high probability.

References

Adleman, L. Two theorems on random polynomial time. Proc. 19th Symp. on Foundations of Computer Science, Ann Arbor, Oct. 1978, 75-83.

Backus, J. Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Comm. ACM* 21:8 (1978), 613-641.

-----, et al. The FORTRAN automatic coding system. Proc. Western Joint Computer Conf., Los Angeles, Feb. 1957, 188-198.

Birkhoff, G. Dependent probabilities and the spaces (L). *Proc. N.A.S.* 24 (1938), 154-159.

-----. *Lattice theory.* 3rd ed. Amer. Math. Soc. Colloquium Publications, v. 25, Providence, 1967.

Chung, K.L. *A course in probability theory.* 2nd ed. Academic Press, New York, 1974.

Collatz, L. *Functional analysis and numerical mathematics.* Academic Press, New York, 1966.

Dunford, N. and J. Schwartz. *Linear operators.* v. 1. Interscience, 1958.

Feller, W. *An introduction to probability theory and its applications.* v. 1, 3rd ed. Wiley, New York, 1968.

Floyd, R. and R. Rivest. Expected time for selection. *Comm. ACM* 18 (1975), 165-172.

Gill, J. Computational complexity of probabilistic Turing machines. Proc. 6th ACM Symp. on Theory of Computing, May 1974, 91-95.

Gouda, M. and E. Manning. Probabilistic cost machines. in: *Algorithms and complexity*, ed. J.F. Traub, Academic Press, New York, 1976, 462.

Greibach, S.A. *Theory of program structures: schemes, 'semantics, and verification*. Springer Verlag, New York, 1975.

Halmos, P. *Measure theory*. Van Nostrand, New York, 1950.

Kakutani, S. Concrete representation of abstract (L)-spaces and the mean ergodic theorem. *Ann. of Math.* 42 (1941), 523-537.

Karamardian, S., ed. *Fixed points: algorithms and applications*. Academic Press, New York, 1977.

Karp, R. Probabilistic analysis of combinatorial search. in: *Algorithms and complexity*, ed. J.F. Traub, Acacdemic Press, New York, 1976, 1-20.

Knuth, D. *Art of computer programming: sorting and searching*. v. 3. Addison Wesley, Reading, Mass., 1973.

Kurtz, T.E. BASIC. *SIGPLAN Notices* 13:8 (1978), 103-118.

Lehmann, D. Categories for fixpoint semantics. Proc. 17th IEEE Symp. on Foundations of Computer Science, Oct. 1976, 122-126.

Manna, Z. *Mathematical theory of computation*. McGraw Hill, New York, 1974.

Miller, G. Riemann's hypothesis and tests for primality. Proc. 7th ACM Symp. on Theory of Computing, May 1975, 234-239.

Paz, A. *Introduction to probabilistic automata*. Academic Press, New York, 1971.

Plotkin, G. A powerdomain construction. *SIAM J. Comput.* 5 (1976), 452-487.

Rabin, M.O. Probabilistic algorithms. in: *Algorithms and complexity*, ed. J.F. Traub, Academic Press, New York, 1976, 21-40.

Ramshaw, L.H. *Formalizing the Analysis of Algorithms*. Ph.D. Thesis, Computer Science, Stanford University, June 1979.

Scott, D. Outline of a mathematical theory of computation. Proc. 4th Princeton Conf. on Info. Sci. and Sys., Princeton, 1970, 169-176.

-----, and C. Strachey. Towards a mathematical semantics for computer languages. Tech. mono. PRC6, Oxford Univ., August 1971.

Solovay, R. and V. Strassen. Fast Monte Carlo tests for primality. *SIAM J. Comput.* 6 (1977), 84-85.

Swaminathan, S., ed. *Fixed point theory and its applications*. Academic Press, New York, 1976.

Vuillemin, J. Proof techniques for recursive programs, Ph.D. thesis, Stanford Univ., 1973.

Yao, A. Probabilistic computations: toward a unified measure of complexity. Proc. 18th IEEE Symp. on Foundations of Computer Science, Providence, Oct. 1977, 222-227.

-----, and F. Yao. On the average case complexity of selecting the kth best. Proc. 19th IEEE Symp. on Foundations of Computer Science, Ann Arbor, Oct. 1978, 280-289.

Zeiger, H.P. Formal models for some features of programming languages. Proc. 1st ACM Symposium on Theory of Computing, Marina del Rey, Calif., May 1969, 211-215.