JOURNAL OF COMPUTER AND SYSTEM SCIENCES 22, 328–350 (1981)

Semantics of Probabilistic Programs

DEXTER KOZEN

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

Revised January 5, 1981

This paper presents two complementary but equivalent semantics for a high level probabilistic programming language. 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. We use the semantics to prove a general result about probabilistic programs, namely, that a program's behavior is completely determined by its action on fixed inputs.

1. INTRODUCTION

Probabilistic computation has recently become an important topic of investigation in theoretical computer science. Major areas of activity include average-case analysis of algorithms, stochastic programming such as probabilistic primality testing, and the study of probabilistic machine models and reductions. In this paper we provide a formal semantics for a class of probabilistic programs. There are several reasons why this should prove worthwhile:

(1) Yao (1977) and Rabin (1976) have grouped research in probabilistic algorithms into two areas, which Yao has termed the *distributional approach* and the *randomized approach*. In the former, the program is deterministic, the input varies according to some probability 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 can make stochastic moves (see, e.g., Gill (1974), Miller (1975), Rabin (1976), Solovay and Strassen (1977), Adleman (1978)). Yao (1977) established a connection between the two approaches by defining a measure of complexity based on each and proving their equivalence. The formal semantics herein provides a common framework in which the two approaches are unified.

(2) Until now, models used in the study of probabilistic complexity 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)). These machines are severely limited in the way they use probability and do not offer the programmer

0022-0000/81/030328–23$02.00/0

much versatility. For example, randomness is available only in simple discrete distributions. 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. The BASIC random number generator uses a continuous distribution: it gives a random real number between 0 and 1 with uniform probability. This is useful in computing over the real numbers, since continuous distributions can often be more realistic, for instance in modeling economic systems or population growth. Here we consider **while** programs with a random assignment $x := \textbf{random}$, an idealized language that more closely resembles these high-level programming languages. Neither the distribution of the random number generator nor the distribution of the input is restricted in any way. In particular, there is no distinction drawn between discrete and continuous distributions.

(3) Up to now, probabilistic algorithms have been analyzed largely by ad hoc methods. This is acceptable for simple discrete distributions, since they are fairly well understood (e.g., a "random graph" is usually taken to have every edge with probability 1/2, certainly an expedient choice, but not necessarily the most realistic). In general, sums are replaced by integrals, combinatorics is replaced by analysis, and intuition is more likely to fail. In such cases it is useful to have a formal deductive system, but a viable semantics is a necessary first step. Ramshaw (1979) has taken a further step by proposing a deductive system and proving it sound with respect to this semantics.

(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, programs and data are interpreted as elements of the ordered Banach spaces of Birkhoff (1938) and Kakutani (1941). These spaces have a wide range of applications: statistical mechanics, functional analysis, ergodic theory, Markov processes, and differential equations. Their theory is a rich combination of analysis and algebra and has been the subject of dozens of volumes published over the past 50 years. It therefore seems worthwhile to point out their relationship to programming language semantics, thereby putting this well-developed theory at our disposal.

In order to be as self-contained as possible, and for ease of reference, Section 2 contains the basic definitions and elementary results of linear analysis and probability theory that are relevant to this paper.

In section 3 we describe the syntax of probabilistic **while** programs, which are like deterministic **while** programs (simple assignments

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

composition, conditional tests, **while** loops), except that they also allow calls on a random number generator

$$x_i := \textbf{random}.$$

We then give two equivalent semantic definitions, 1 and 2. Semantics 1 is closer to classical probability theory as found in Feller (1968) and Chung (1974), and is more likely to be the result of a first attempt at describing probabilistic programs formally, since it is more operational and more intuitive. Semantics 2 is more denotational and 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 this semntics, programs are interpreted as continuous linear operators on a Banach space of distributions. We prove the equivalence of the two semantics and argue that Semantics 2 expresses properties of the probabilistic behavior of programs at a more appropriate level of abstraction.

In Section 4 we demonstrate the connection between Scott–Strachey least fixed point semantics and probabilistic semantics by showing how an ordered domain of partial functions is embedded naturally into an ordered Banach space.

In Section 5 we show how to extend the semantics to all higher functional types.

In Section 6 we prove a result about probabilistic programs illustrating the use of the formalism developed in previous sections. The result gives a sufficient condition for program equivalence. It says that *if two programs agree whenever the input is fixed, then they are equivalent.* In order words, a program's behavior is completely determined by its behavior on inputs whose distribution is a point mass. This result may be considered a manifestation of the discrete nature of programs. One of its consequences is that programs can be proved equivalent by considering their action on discrete inputs only, which can be represented by countable sums instead of abstract integrals. Thus combinatorics can replace analysis and integration theory in equivalence proofs.

2. Background and Notation

We will use the notation and terminology of the following books: measure theory: Halmos (1950); probability theory: Feller (1968), Chung (1974); linear analysis: Dunford and Schwartz (1958). In addition, Birkhoff (1967) is an excellent introduction to partially ordered vector spaces. Some of the basic definitions and standard notation from these fields are reviewed below.

$\mathbb{R}$ denotes the real numbers, $\mathbb{R}^+$ the nonnegative real numbers, and ω the nonnegative integers.

2.1. *Measure and Probability*

A *measurable space* is a pair (X, M) where X is a set and M is a σ-algebra of subsets of X, i.e., M is a Boolean algebra of subsets of X closed under countable union. Elements of M are called *measurable sets* or *events* and are denoted B, C,.... $\sim B$ denotes the complement of B in X. A function $f\colon (X, M) \to (Y, N)$ is *measurable* provided $f^{-1}(B) \in M$ whenever $B \in N$.

Let (X_n, M_n) be a sequence of measurable spaces and let $\Pi_n X_n$ be the direct product of the X_n with projections $\pi_i: \Pi_n X_n \to X_i$. The *cartesian product* $\Pi_n(X_n, M_n)$ is the space $(\Pi_n X_n, M)$, where M is the smallest σ-algebra containing all *cylinders* $\pi_i^{-1}(B)$, $B \in M_i$. If the number of X_n is countable, then M is the smallest σ-algebra containing all *rectangles* $\Pi_n B_n$, each $B_n \in M_n$. The cartesian product of α copies of (X, M) is denoted $(X^\alpha, M^{(\alpha)})$.

A *measure* or *distribution* μ on (X, M) is a function $M \to \mathbb{R}$ that is *countably additive*, i.e., if B_n are a countable set of pairwise disjoint elements of M, then $\mu(\bigcup_n B_n) = \Sigma_n \mu(B_n)$. It follows that $\mu(A \cup B) \leqslant \mu(A) + \mu(B)$ and $\mu(0) = 0$. A measure μ is *positive* if $\mu(B) \geqslant 0$ for all $B \in M$. It is a *probability measure* if it is positive and $\mu(X) = 1$ and a *subprobability measure* if it is positive and $\mu(X) \leqslant 1$.

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 $B \times C$.

Let (X, M) be a measurable space, $B \in M$. The *characteristic function* of B is the function $\chi_B: X \to \{0, 1\}$ such that $\chi_B(x) = 1$ iff $x \in B$. A *measurable partition* of B is a family of pairwise disjoint measurable sets whose union is B. A *simple function* $f: X \to \mathbb{R}$ is one of the form $\Sigma_{B \in \pi} a_B \chi_B$ where $a_B \in \mathbb{R}$ and π is a finite measurable partition of X.

If μ is a measure and $B \in M$, let μ_B denote the measure $\mu_B(A) = \mu(A \cap B)$. *The conditional probability* relative to B is given by the measure $\mu_B/\mu(B)$.

Every measure can be decomposed into its positive and negative parts: to every measure μ there correspond unique positive measures μ^+ and μ^- such that $\mu^+ = \mu_B$ and $\mu^- = -\mu_{\sim B}$ for some $B \in M$ (Halmos 1950, Theorem B, p. 123). This is called the *Jordan decomposition* of μ. The measures μ^+ and μ^- are called called the *positive* and *negative variation* of μ respectively. The measure $|\mu| = \mu^+ + \mu^-$ is called the *total variation* or *absolute value* of μ. The *total variation norm* is a map $\| \|: \mathbf{B} \to \mathbb{R}^+$ associating with each measure μ the nonnegative real number $\|\mu\| = |\mu|(X)$.

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 $\mu = \mu_{\{x\}}$ and $\mu(\{x\}) = 1$, then μ is called a *point mass*. A measure is *continuous* if $\mu(B) = 0$ for all countable B. Every measure can be represented uniquely as the sum of a discrete measure and a continuous measure.

A *measure space* (X, M, μ) is a measurable space equipped with a measure. A *probability space* is a measure space (X, M, μ) where μ is a probability measure. A *random variable* is a (partial) measurable function whose domain is a probability space. The domain of a random variable is called the *sample space* and its range is called the *value space*.

A random variable $x: (X, M, \mu) \to (Y, N)$ induces a subprobability measure $\mu \circ x^{-1}$ on (Y, N):

$$\mu \circ x^{-1}(B) = \mu(x^{-1}(B)).$$

If x is total, then $\mu \circ x^{-1}$ is a probability measure. When the sample space is

understood, we occasionally use the more intuitive notation "$\Pr(x \in A)$" and say "the probability that x lies in A" to denote the value of $\mu \circ x^{-1}(A)$.

A *random vector* is a list of random variables

$$x_i\colon (X, M, \mu) \to (Y_i, N_i)$$

with the same domain. Equivalently, a random vector is a random variable from (X, M, μ) into the cartesian product $\Pi(Y_i, N_i)$. If $x = x_1, x_2, \ldots$ is a random vector, then the subprobability measure $\mu \circ x^{-1}$ on $\Pi(Y_i, N_i)$ induced by x is called the *joint distribution* of the random variables $x_1, x_2, \ldots$. Two random variables x, y defined on the same sample space (X, M, μ) are *independent* if their joint distribution is exactly the product distribution $(\mu \circ x^{-1}) \times (\mu \circ y^{-1})$. In other words, x and y are independent if $\Pr(x \in A \text{ and } y \in B) = \Pr(x \in A) \cdot \Pr(y \in B)$.

2.2. *Partially Ordered Normed Vector Spaces*

A *norm* on a vector space $\mathbf{B}$ is a map $\| \|\colon \mathbf{B} \to \mathbb{R}^+$ such that

$$\begin{aligned} \|x\| &= 0 && \text{iff } x = 0, \\ \|ax\| &= |a|\,\|x\| && \text{for all scalars } a, \\ \|x + y\| &\leqslant \|x\| + \|y\|. \end{aligned}$$

The norm induces a metric on $\mathbf{B}$: the distance between x and y is $\|x - y\|$. If $\mathbf{B}$ is complete with respect to this metric, then $\mathbf{B}$ is called a *Banach space.*

If $(\mathbf{B}, \| \|), (\mathbf{C}, \| \|)$ are two normed vector spaces and if $T\colon \mathbf{B} \to \mathbf{C}$ is a linear transformation, T is $\| \|$-*bounded* if

$$\sup_{\mathbf{S}} \|T(x)\| < \infty,$$

where the supremum is taken over the closed unit sphere $\mathbf{S} = \{x \mid \|x\| \leqslant 1\}$. A linear transformation is $\| \|$-bounded if and only if it is continuous with respect to the metric induced by $\| \|$. The space of $\| \|$-bounded linear transformations from $\mathbf{B}$ to $\mathbf{C}$ is a normed vector space under pointwise addition and scalar multiplication, with the *uniform norm*

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

so called because it characterizes uniform convergence of sequences of functions. A $\| \|$-bounded linear transformation $\mathbf{B} \to \mathbf{B}$ is called a *linear operator* on $\mathbf{B}$. If $\mathbf{B}$ is a Banach space, then so is the space of linear operators.

A *positive cone* in a normed vector space $\mathbf{B}$ is a distinguished subset $\mathbf{P}$ of $\mathbf{B}$ satisfying the two properties

$$x, y \in \mathbf{P} \text{ and } a, b \geqslant 0 \quad \to \quad ax + by \in \mathbf{P}$$

$$x, -x \in \mathbf{P} \to x = 0.$$

For example, **P** might be the set of vectors in $\mathbb{R}^n$ 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 on **B**: $x \leqslant y$ iff $y - x \in \mathbf{P}$. **P** is then the set of $x \geqslant 0$ (hence the term *positive*).

With respect to the order $\leqslant$, addition and scalar multiplication are *order continuous*, i.e.,

$$x + \sup_\alpha x_\alpha = \sup_\alpha (x + x_\alpha),$$

$$\sup_\alpha a x_\alpha = a \sup_\alpha x_\alpha, \qquad a \geqslant 0$$

in the sense that if one side exists, then so does the other and they are equal. The following are equivalent: $x \leqslant y$ for some $y \in \mathbf{P}$; x is in the linear span of **P**; x can be written as the difference of two positive elements.

A *directed set* is a subset A of **B** such that any pair of elements in A has an $\leqslant$-upper bound in A. An *interval* is a set $[x, y] = \{z \mid x \leqslant z \leqslant y\}$. A set is *order-bounded* if it is contained in an interval.

(B, P) is a *vector lattice* if every pair $x, y \in \mathbf{B}$ has a $\leqslant$-least upper bound or *join* $x \vee y$. Equivalently, **B** is a vector lattice if every pair $x, y \in \mathbf{B}$ has a greatest lower bound or *meet* $x \wedge y$. Vector lattices are distributive, addition and scalar multiplication distribute over $\vee$ and $\wedge$, and

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

Every x in a vector lattice can be written uniquely as the difference of two positive elements x^+, x^- whose meet is 0. This is called the *Jordan decomposition* of x. The *absolute value* of x, denoted $|x|$, is defined as

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

For any x, y in a vector lattice,

$$x^+ = x \vee 0 \qquad \text{and} \qquad x^- = (-x) \vee 0,$$

$$x \vee y = (x - y)^+ + y,$$

$$x \wedge y = -(-x \vee -y),$$

$$|x| \geqslant 0, \qquad \text{and} \qquad |x| = 0 \text{ iff } x = 0,$$

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

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

A vector lattice **(B, P)** is *conditionally complete* if every set of elements of **B** with an $\leqslant$-upper bound has a least upper bound.

Let T be a linear transformation between vector latices. T is *isotone* if $x \leqslant y$ implies $T(x) \leqslant T(y)$. Equivalently, T is isotone if T maps the positive cone of the domain into the positive cone of the range. The set of isotone maps forms a positive cone in the space of linear transformations, and this induces an order $\leqslant$ as above. T is said to be *order bounded* if it maps order-bounded sets to order-bounded sets. If the range of T is conditionally complete, then the following are equivalent: T is order bounded; T is in the linear span of the isotone linear transformations; T has a Jordan decomposition $T = T^+ - T^-$ (Birkhoff, 1967, p. 366). The set of order-bounded linear transformations from a vector lattice to a conditionally complete vector lattice is again a vector lattice under the definition

$$\begin{aligned}(S \vee T)(y) &= \sup_{0 \leqslant x \leqslant y} S(x) + T(y-x), \qquad y \geqslant 0,\\ (S \vee T)(y) &= (S \vee T)(y^+) - (S \vee T)(y^-).\end{aligned} \tag{2.2.1}$$

If $\mathbf{B} = (\mathbf{B}, \mathbf{P}, \| \|)$ is both a Banach space and a vector lattice such that order and norm are related by the properties

$$\begin{aligned}\| |x| \| &= \|x\|,\\ 0 \leqslant x \leqslant y &\to \|x\| \leqslant \|y\|,\end{aligned}$$

then $\mathbf{B}$ is called a *Banach lattice.*

If (X, M) is a measurable space, the set $\mathbf{B}(X, M)$ of measures on (X, M) with the cone $\mathbf{P}$ of positive measures and total variation norm $\| \|$ forms a Banach lattice, with addition scalar multiplication defined pointwise:

$$\begin{aligned}(\mu + \nu)(B) &= \mu(B) + \nu(B),\\ (a\mu)(B) &= a\mu(B).\end{aligned}$$

3. Probabilistic **while** Programs

In this section we describe a class of probabilistic programs called *probabilistic **while** programs* and give two equivalent approaches to their interpretation.

3.1. *Syntax*

We consider **while** programs over the variables $x_1, \ldots, x_n$. Syntactically, there are five types of statements:

(3.1.1) *simple assignment.*

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

(3.1.2) *random assignment.*

$$x_i := \textbf{random}.$$

(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.**

The delimeters **fi** and **od** will usually be omitted.

3.2. *Semantics* 1

In trying to assign formal meaning to these programs, Semantics 1 would likely be the first idea to occur. It is closer to classical probability theory as found in Feller (1968) or Chung (1974), extends deterministic semantics more directly, and is somewhat more operational and intuitive. Unfortunately, there are numerous problems with this approach, as we shall see later.

Let us assume that all variables range over the same domain X (this assumption is for simplicity of presentation only and is not essential). Suppose X has a family of measurable sets M associated with it. The $B \in M$ are the sets for which it makes sense to ask, "What is the probability that $x_i \in B$?" For example, if $X = \mathbb{R}$, then the class of Lebesgue measurable sets is a good choice for M; if $X = \omega$, M should be the power set 2^ω.

Under Semantics 1, input variables $x_1, \ldots, x_n$ will be *random variables* on a fixed probability space (Ω, F, μ), i.e., each x_i is a measurable function $(\Omega, F, \mu) \to (X, M)$. The "random number generator" will be a countable sequence of independent, identically distributed random variables $y_0, y_1, \ldots$ on (Ω, F, μ), each y_j independent of the x_i. Informally, a sample program execution consists of first picking a sample point $\omega \in \Omega$, simultaneously determining the values of the input variables and countably many 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 and assigned to x_i.

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

We will allow the B appearing in (3.1.4) and (3.1.5) to be any measurable set $B \in M^{(n)}$. The f in (3.1.1) may be any partial measurable function $f: X^n \to X$. The restriction of measurability of B and f is necessary for technical reasons but sufficiently general for all practical purposes: in $\mathbb{R}$, all common functions such as $+$ or log are measurable, and all first-order definable sets in a language with $+$, $\cdot$, and $\leqslant$ are measurable.

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

(3.2.1) *simple assignment.* If $f: X^n \to X$ is a measurable function, the simple assignment (3.1.1) denotes the measurable function $X^{n+\omega} \to X^{n+\omega}$ 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 (3.1.2) 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 $a = a_1, a_2, \ldots$ gives

$$\begin{array}{ll} f_S(a) & \text{if } a \in B \times X^\omega, \\ f_T(a) & \text{otherwise.} \end{array}$$

(3.2.5) ***while** loop.* The while statement (3.1.5) denotes the partial measurable function which on input $a = a_1, a_2, \ldots$ gives

$f_S^n(a)$ where n is least number such that $f_S^n(a) \notin B \times X^\omega$, if such an n exists, undefined otherwise.

The specification is completed by giving a sequence $y_{n+1}, y_{n+2}, \ldots$ of independent, identically distributed random variables $(\Omega, F, \mu) \to (X, M)$ 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 the input $x_1, \ldots, x_n$ is the first n components of the random vector $f_S \circ (x_1, \ldots, x_n, y_{n+1}, \ldots)$.

The most noteworthy problem with this approach is that too much has to be

specified. For example, the particular random number assigned to x_i in the random assignment $x_i :=$ **random** 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. Moreover, the probabilistic flow of the program, based on tests in (3.1.4) and (3.1.5), does not depend on the random vector of inputs itself, but only on its distribution. 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 construct a sample space (Ω, F, μ) and an input vector x: $(\Omega, F, \mu) \to (X^n, M^{(n)})$ satisfying that distribution. This 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 $(X^\omega, M^{(\omega)})$ tail constructed to accommodate the random number generator will become superfluous.

3.3. *Semantics* 2

Semantics 2 is closer to Scott–Strachey semantics, since 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. In this semantics, a program S maps distributions μ on $(X^n, M^{(n)})$ to distributions $S(\mu)$ on $(X^n, M^{(n)})$ (we use the same symbol S for both the program and its meaning under Semantics 2).

The intuition behind this approach is as follows. The program variables $x_1, \ldots, x_n$ satisfy some joint distribution μ on input. We will forget the variables themselves and concentrate on the distribution μ. We can think of μ as a fluid mass distributed throughout X^n. This mass is concentrated more densely in some areas than others, depending on which inputs are more likely to occur. Execution of a simple or random assignment redistributes the mass in X^n. Conditional tests cause the mass to split apart, and the two sides of the conditional are executed on the two pieces. In the **while** loop (3.1.5), the mass goes around and around the loop; at each cycle, the part of the mass which occupies $\sim B$ breaks off and exits the loop, and the rest goes around again. Part of the mass may go around infinitely often. Thus, at any point in time, there are different pieces of the mass that occupy different parts of the program, and each piece is spread throughout the domain according to the simple and random assignments that have occurred in its history. Different pieces that have come to occupy the same parts of the program through different paths are accumulated. At certain points in time, parts of the mass find their way out of the program. The output distribution $S(\mu)$ is the sum of all the pieces that eventually find their way out. Thus the probability that program S halts on input distribution μ is $S(\mu)(X^n)$, the probability of the universal event X^n upon output.

More formally, let (X, M) be a measurable space and let $\mathbf{B} = \mathbf{B}(X^n, M^{(n)})$ be the set of measures on the cartesian product $(X^n, M^{(n)})$. $\mathbf{B}$ consists of all possible joint

distributions of the program variables $x_1, \ldots, x_n$, plus all their linear combinations, where addition and scalar multiplication are defined by

$$(\mu + \nu)(B) = \mu(B) + \nu(B),$$
$$(a\mu)(B) = a(\mu(B)), \qquad a \in \mathbb{R}.$$

If $\mathbf{P}$ is the set of positive measures and if $\| \ \|$ denotes the total variation norm

$$\|\mu\| = |\mu|\,(X^n),$$

then $(\mathbf{B}, \mathbf{P}, \| \ \|)$ is a conditionally complete Banach lattice as described in Section 2; that is, $(\mathbf{B}, \| \ \|)$ is a Banach space, or complete normed vector space, and $(\mathbf{B}, \mathbf{P})$ is a conditionally complete vector lattice under the order $\leqslant$ induced by $\mathbf{P}$, such that

$$\| |\mu| \| = \|\mu\|,$$
$$0 \leqslant \mu \leqslant \nu \to \|\mu\| \leqslant \|\nu\|.$$

The measures we are primarily interested in are the subprobability measures. These are the positive measures of norm at most 1, i.e., the elements of $\mathbf{S} \cap \mathbf{P}$, where $\mathbf{S} = \{\mu \mid \|\mu\| \leqslant 1\}$ is the closed unit ball of $\mathbf{B}$. The probability measures are the subprobability measures whose norm is exactly 1, i.e., they are elements of $\partial\mathbf{S} \cap \mathbf{P}$, where $\partial_{\mathbf{S}}$ is the boundary of $\mathbf{S}$.

Every program $\mathbf{S}$ will map a probability distribution into a subprobability distribution, thus it can be interpreted as a function $\partial\mathbf{S} \cap \mathbf{P} \to \mathbf{S} \cap \mathbf{P}$. It will turn out, however, that when this function is defined in a way consistent with Semantics 1 above, it will extend uniquely to a linear transformation $\mathbf{B} \to \mathbf{B}$. Moreover, this extension will be $\| \ \|$-bounded and therefore continuous with respect to the metric induced by $\| \ \|$. Thus each program will define a continuous linear transformation or *operator* $\mathbf{B} \to \mathbf{B}$.

The space $\mathbf{B}'$ of operators $\mathbf{B} \to \mathbf{B}$ forms a Banach space under the *uniform norm*

$$\|T\| = \sup_{\mu \in S} \|T(\mu)\|$$

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

A linear operator $T: \mathbf{B} \to \mathbf{B}$ is isotone with respect to the order $\leqslant$ in $\mathbf{B}$, i.e.,

$$\mu \leqslant \nu \to T(\mu) \leqslant T(\nu),$$

iff T preserves the positive cone $\mathbf{P}$. Denote by $\mathbf{P}'$ the set of isotone elements of $\mathbf{B}'$. Then $\mathbf{P}'$ is a positive cone in $\mathbf{B}'$, and so induces a partial order $\leqslant$ on $\mathbf{B}'$. $S \leqslant T$ with respect to this order if and only if $S(\mu) \leqslant T(\mu)$ for all $\mu \in \mathbf{P}$. (There is an interesting relationship between $\leqslant$ as defined above and the order $\sqsubseteq$ in Scott–Strachey least fixed point semantics: they are the same thing. This will be discussed further in Section 4.)

Besides $\mathbf{P}'$, define $\mathbf{S}'$ as the set $T \in \mathbf{B}'$ which preserve $\mathbf{S}$. By the definition of the

uniform norm, $\mathbf{S}'$ is exactly the closed unit ball of $\mathbf{B}'$. Since the linear operators described by programs will preserve both $\mathbf{S}$ and $\mathbf{P}$, they will be elements of $\mathbf{S}' \cap \mathbf{P}'$.

$\mathbf{B}'$ is not necessarily a vector lattice, however it is conditionally complete in the sense that any set of elements with an $\leqslant$-upper bound has a $\leqslant$-least upper bound. In particular, any pair S, T with an upper bound has a join $S \vee T$ given by (2.2.1). If S, $T \in \mathbf{P}'$, then $S \vee T$ exists, since S and T are bounded above by $S + T$.

More important for our purposes than conditional completeness, however, is the property: *any* $\|\ \|$*-bounded directed set* x_α *of positive elements has a least upper bound* $\sup_\alpha x_\alpha$; *moreover* $\|\sup_\alpha x_\alpha\| = \sup_\alpha \|x_\alpha\|$. This property holds in both $\mathbf{B}$ and $\mathbf{B}'$. Since it is a special case of Theorem 5.6 below, we defer its proof until Section 5. A proof for the space $\mathbf{B}$ may be found in Birkhoff (1967, Theorem 21, p. 371).

We are now ready to give 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. At each of the five steps, it is straightforward to verify that the two definitions are equivalent, in the sense made precise by Theorem 3.3.9 below. In the following, we use the symbol S for both a program and the linear operator it denotes.

(3.3.1) *simple assignment.* If S is the program $x_i := f(x_1, \ldots, x_n)$, where $f: X^n \to X$ is a measurable function, then the meaning of S is the linear operator

$$S(\mu) = \mu \circ F^{-1},$$

where $F: X^n \to X^n$ is the measurable function

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

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

(3.3.2) *random assignment.* If S is the program $x_i :=$ **random** then S denotes the linear operator

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

where ρ is a fixed distribution, the distribution of the random number generator. Since $M^{(n)}$ is generated by rectangles of the form $B_1 \times \cdots \times B_n$, $S(\mu)$ 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.* Let μ_B denote the measure $\mu_B(A) = \mu(A \cap B)$. If μ is a probability measure, the conditional probability relative to B is given by the normalized measure $\mu_B/\mu(B)$. Intuitively, the conditional test should work as follows. Suppose the input satisfies probability distribution μ. First membership of $x_1, \ldots, x_n$ in B is tested. This occurs with probability $\mu(B)$, and hence S is executed with this probability. However, once this branch has been taken, we know that the event $\sim B$ is impossible, but aside from this we have no more knowledge than before. Therefore S should be executed on the conditional probability distribution $\mu_B/\mu(B)$, giving

$S(\mu_B/\mu(B))$. Similarly, with probability $\mu(\sim B)$ the program T will be executed on $\mu_{\sim B}/\mu(\sim B)$ to give $T(\mu_{\sim B}/\mu(\sim B))$. After the conditional statement, the probability that $x_1,\ldots,x_n$ lies in A is the probability that either the "true" branch was taken and $x_1,\ldots,x_n$ lies in A after executing S, or the "false" branch was taken and $x_1,\ldots,x_n$ lies in A after executing T, or in symbols

$$\mu(B)\, S(\mu_B/\mu(B))(A) + \mu(\sim B)\, T(\mu_{\sim B}/\mu(\sim B))(A)$$
$$= (\mu(B)\, S(\mu_B/\mu(B)) + \mu(\sim B)\, T(\mu_{\sim B}/\mu(\sim B)))(A).$$

Using the fact that S and T are linear, this becomes

$$(S(\mu_B) + T(\mu_{\sim B}))(A).$$

Thus the semantics of the program **if** B **then** S **else** T is the operator

$$S \circ e_B + T \circ e_{\sim B},$$

where e_B is the operator $e_B(\mu) = \mu_B$ and $+$ is addition in $\mathbf{B}'$.

(3.3.5) ***while*** *loop.* We want equivalence between the program **while** B **do** S and the program

$$\textbf{if } \sim B \textbf{ then } I \textbf{ else } S; \textbf{ while } B \textbf{ do } S \textbf{ od fi}, \tag{3.3.6}$$

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

$$W = e_{\sim B} + W \circ S \circ e_B. \tag{3.3.7}$$

This is a case of a simple operator equation scheme studied in functional analysis, and well-established techniques are available for its solution (see, e.g., Collatz, 1966, p. 358). A common approach is to search for a fixed point of the affine transformation $\tau\colon \mathbf{B}' \to \mathbf{B}'$ defined by

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

We will use Theorem 5.6 to obtain a fixed point. First, note that τ preserves $\mathbf{S}'$ and $\mathbf{P}'$ and is isotone with respect to $\leqslant$ in $\mathbf{B}'$. If A is the set of elements of $\mathbf{S}' \cap \mathbf{P}'$ such that $W \leqslant \tau(W)$, then A is nonempty (it contains 0) and is closed under suprema of directed sets, by Theorem 5.6. By Zorn's Lemma, A has a maximal element W, so W must be a fixed point.

Once a solution to (3.3.7) has been found, it is easy to show that

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

is the unique least such solution in $\mathbf{S}' \cap \mathbf{P}'$. First, W_0 exists since $\mathbf{B}'$ is conditionally complete. Since τ is isotone, $\tau(W_0) \leqslant \tau(W) = W$ for any fixed point W, so

$\tau(W_0) \leqslant W_0$. By Tarski's theorem (see Birkhoff, 1967, exercise 6, p. 116), τ has a fixed point in the interval $[0, W_0]$. This fixed point must be W_0, since W_0 is the infimum of fixed points.

As is customary, we take the least fixed point W_0 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

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

is exactly W_0, by showing that τ is order continuous. The present approach was used instead to illustrate a more general technique, which applies even in the absence of order continuity. This is discussed further in Section 5.

The following theorem asserts that the constructions above indeed give elements of $\mathbf{S}' \cap \mathbf{P}'$. The proof is a straightforward induction on program structure, treating each of the five cases (3.1.1)–(3.1.5) separately. We leave the details of the proof to the reader.

THEOREM 3.3.8. *Let S be any* ***while*** *program over the variables $x_1, \ldots, x_n$, and let $\mathbf{B} = \mathbf{B}(X^n, M^{(n)})$ be the space of measures on $(X^n, M^{(n)})$. Under Semantics 2, S denotes a positive linear operator on $\mathbf{B}$ with $\|S\| \leqslant 1$.* ∎

Let $x: (\Omega, F, P) \to (X^{n+\omega}, M^{(n+\omega)})$ be any random vector such that the components $x_{n+1}, x_{n+2}, \ldots$ of x 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 x. Then x has distribution $\mu \times \rho^\omega$. If program S is applied to x under Semantics 1, the result is $f_S \circ x^{-1}$ 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 any program over $x_1, \ldots, x_n$. For all $\mu \in \mathbf{B}(X^n, M^{(n)})$, $B \in M^{(n)}$,*

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

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. When this is done, there appears a striking correspondence between the present formalism and the partially ordered domains encountered in Scott–Strachey denotational semantics.

Consider the domain $Pfn(\omega \to \omega)$ of partial functions $\omega \to \omega$, with the usual

ordering $\sqsubseteq$ and bottom (least defined) element $\perp$. We show that this space can be embedded in a partially ordered Banach space so that $\sqsubseteq$ becomes $\leqslant$ and $\perp$ becomes 0, and the elements of $Pfn(\omega \to \omega)$ are all members of $\mathbf{S}' \cap \mathbf{P}'$. The 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 2^ω. Let $\mathbf{B} = \mathbf{B}(\omega, 2^\omega)$ be the Banach space of measures. Elements of $\mathbf{B}$ may be viewed as formal sums

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

where the coefficients a_x are real numbers such that

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

The total variation norm is given by

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

Let $\mathbf{P}$ be the cone of positive measures and let $\mathbf{S}$ be the closed unit ball.

In the Scott–Strachey construction, a "flat domain" is constructed 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 $x \sqsubseteq y$ holds. Then a partial function $\omega \to \omega$ may be viewed as an $\sqsubseteq$-isotone function $\omega \cup \{\perp\} \to \omega \cup \{\perp\}$ which takes $\perp$ to $\perp$.

If each $y \in \omega$ is identified with its corresponding point mass $\sum \chi_{\{y\}}(x)\, \mathbf{x}$, where $\chi_{\{y\}}$ is the characteristic function of $\{y\}$, and if $\perp$ is identified with 0 in $\mathbf{B}$, then the result is an embedding of $\omega \cup \{\perp\}$ into $\mathbf{S} \cap \mathbf{P}$ which takes $\sqsubseteq$ into $\leqslant$. Under this embedding, each $\sqsubseteq$-isotone function $t: \omega \cup \{\perp\} \to \omega \cup \{\perp\}$ preserving $\perp$ becomes a partial function $t: \mathbf{S} \cap \mathbf{P} \to \mathbf{S} \cap \mathbf{P}$, and t extends uniquely to a linear transformation $T: \mathbf{B} \to \mathbf{B}$ by taking

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

Moreover, it is easy to verify that T is $\| \|$-bounded (indeed $\|T\| \leqslant 1$) and that T preserves $\mathbf{P}$. This says that T is in both the positive cone and the closed unit ball of the space $\mathbf{B}'$ of operators.

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

5. Extension to Higher Types

In this section we show how the least fixed point construction of the previous section extends naturally to higher types, in contrast to nondeterministic order semantics, in which the corresponding result is somewhat less natural.

The proof of Theorem 5.6 will show how order and norm interact. Scott and others, who used order exclusively, based their strategy on the view that "all semantically meaningful functions should be [order] continuous" (Lehmann, 1976, p. 123). However, this requirement makes the extension of Scott–Strachey semantics to higher types difficult. Besides, although all elements of $\mathbf{B}'$ are order continuous (use Birkhoff, 1967, Theorem 21, p. 371), many potentially interesting operators in higher types may not be. Theorem 5.6 gives a more general method of obtaining a fixed point which does not require order continuity but only isotonicity. Accordingly, as the restriction of order continuity is relaxed, the properties of the norm $\|\ \|$ take up the slack.

The space $\mathbf{B}(X, M)$ of measures on (X, M) with the norm $\|\ \|$ and positive cone $\mathbf{P}$ enjoys some very powerful properties: it is a Banach lattice, i.e., a Banach space under $\|\ \|$ and a vector lattice under $\leqslant$, satisfying the two properties

$$0 \leqslant x \leqslant y \rightarrow \|x\| \leqslant \|y\|, \tag{5.1}$$

$$\||x|\| = \|x\|. \tag{5.2}$$

Moreover, it satisfies two important properties that allow the least fixed point construction:

$$\text{every } \|\ \|\text{-bounded directed set has a supremum,} \tag{5.3}$$

if x_α is a directed set of positive elements with a supremum, then

$$\|\sup_\alpha x_\alpha\| = \sup_\alpha \|x_\alpha\|. \tag{5.4}$$

Note that (5.4) implies (5.1). Properties (5.1) and (5.3) together imply conditional completeness, i.e., every order-bounded set of elements has a supremum. To see this, let A be contained in the interval $[x, y] = \{z \mid x \leqslant z \leqslant y\}$. The set A' of finite joins of elements of A is also contained in $[x, y]$, and is directed; then $A' - x \subseteq [0, y - x]$, and by (5.1), every element of $A' - x$ has norm no greater than $\|y - x\|$. By (5.3), $A' - x$ has a supremum z, thus $z + x$ is the supremum of A.

The space of operators on $\mathbf{B}(X, M)$ does not satisfy all these properties; indeed, it is not even necessary that two elements have an upper bound. However, if we restrict our attention to the order-bounded operators, i.e., those that map order-bounded sets to order-bounded sets, then it is the case that every pair of elements S, T has a supremum given by (2.2.1). In fact, the following three conditions are equivalent: S is order bounded; S is an element of the linear span of the positive cone; S^+, S^-, and $|S|$ exist. Thus redefining $\mathbf{B}'$ to be the set of $\|\ \|$-bounded, order-bounded linear transformations, and using the norm

$$\|S\|_+ = \||S|\| \tag{5.5}$$

instead of $\|\ \|$, $\mathbf{B}'$ becomes a Banach lattice and satisfies (5.3) and (5.4) (this is a special case of Theorem 5.6 below). The restriction of the space of operators to the

order-bounded ones and the use of $\|\ \|_+$ instead of $\|\ \|$ constitute no loss of generality for all practical purposes, since programs are always positive and hence order-bounded, and $\|\ \|$ and $\|\ \|_+$ agree on the positive cone.

Define a *type* recursively as either the space of measures $(\mathbf{B}(X, M), \mathbf{P}, \|\ \|)$ or the space $(\mathbf{C} \to \mathbf{D}, \mathbf{P}', \|\ \|_+)$ of order-bounded, $\|\ \|_+$-bounded linear transformations $\mathbf{C} \to \mathbf{D}$, where $(\mathbf{C}, \mathbf{P}, \|\ \|)$ and $(\mathbf{D}, \mathbf{P}, \|\ \|)$ are types. Here $\mathbf{P}'$ is the positive cone of isotone elements of $\mathbf{C} \to \mathbf{D}$.

THEOREM 5.6. *Every type is a Banach lattice satisfying* (5.3) *and* (5.4).

Proof. The proof is by induction on type structure. The base type $(\mathbf{B}(X, M), \mathbf{P}, \|\ \|)$ is a Banach lattice (Birkhoff, 1967, Corollary 1, p. 374) and satisfies (5.3) and (5.4) (Birkhoff, 1967, Theorem 21, p. 371). For the induction step, we need to show that if $(\mathbf{C}, \mathbf{P}, \|\ \|)$ and $(\mathbf{D}, \mathbf{P}, \|\ \|)$ satisfy the theorem, then so does $(\mathbf{C} \to \mathbf{D}, \mathbf{P}', \|\ \|_+)$. Since (2.2.1) above defines the supremum of two elements, $(\mathbf{C} \to \mathbf{D}, \mathbf{P}')$ is a vector lattice; also, it is easily verified that $\|\ \|_+$ is a norm. Then 5.2 is satisfied trivially.

To show 5.1, define

$$\|S\|_P = \sup_{x \neq 0} \|S(|x|)\| / \||x|\|.$$

LEMMA. *For all* $S \geqslant 0$, $\|S\|_P = \|S\|$.

Proof. Clearly $\|S\|_P \leqslant \|S\|$. Now $x \leqslant |x|$ for any $x \in \mathbf{C}$, and since $S \geqslant 0$, $S(x) \leqslant S(|x|)$. Similarly $S(-x) \leqslant S(|-x|) = S(|x|)$, thus $-S(|x|) \leqslant S(x) \leqslant S(|x|)$. It follows that $0 \leqslant |S(x)| \leqslant S(|x|)$, and since $\mathbf{D}$ satisfies (5.1) and (5.2), $\|S(x)\| \leqslant \|S(|x|)\|$. Thus

$$\begin{aligned}
\|S\| &= \sup_{x \neq 0} \|S(x)\| / \|x\|, \\
&\leqslant \sup_{x \neq 0} \|S(|x|)\| / \|x\|, \\
&= \sup_{x \neq 0} \|S(|x|)\| / \||x|\| \qquad \text{since } \mathbf{C} \text{ satisfies (5.2)}, \\
&= \|S\|_P. \quad \blacksquare
\end{aligned}$$

Thus $\mathbf{C} \to \mathbf{D}$ satisfies (5.1), since if $0 \leqslant S \leqslant T$ then $S(|x|) \leqslant T(|x|)$ for all x, therefore $\|S\|_P \leqslant \|T\|_P$.

Now we show that $\mathbf{C} \to \mathbf{D}$ satisfies (5.3) and (5.4). Suppose S_α is a $\|\ \|_+$-bounded directed set in $\mathbf{C} \to \mathbf{D}$. Since translation preserves order, we may assume without loss of generality that the S_α are all positive. For any fixed $x \in \mathbf{C}$, $x \geqslant 0$, the set of all $\mathbf{S}_\alpha(x)$ is $\|\ \|$-bounded in $\mathbf{D}$:

$$\|S_\alpha(x)\| \leqslant \|S_\alpha\|\, \|x\|.$$

Also, $S_\alpha(x)$ is a directed set. Since **D** satisfies (5.3) and (5.4), $\sup_\alpha(S_\alpha(x))$ exists for positive x, and

$$\|\sup_\alpha(S_\alpha(x))\| = \sup_\alpha \|S_\alpha(x)\|.$$

Define

$$S(x) = \sup_\alpha(S_\alpha(x)), \qquad x \geqslant 0,$$

$$S(x) = S(x^+) - S(x^-).$$

As addition and scalar multiplication are order continuous, S is linear on the positive cone **P** of **C**, thus **S** is linear on all **C** (Birkhoff, 1967, Lemma 2, p. 365). S is positive and therefore order bounded, and easily shown to be the least upper bound of the S_α, thus (5.3) is satisfied. In addition,

$$\begin{aligned}
\|S\| = \|S\|_P &= \sup_{x>0} \|S(x)\|/\|x\|, \\
&= \sup_{x>0} \|\sup_\alpha S_\alpha(x)\|/\|x\|, \\
&= \sup_{x>0} \sup_\alpha \|S_\alpha(x)\|/\|x\| \qquad \text{since } \mathbf{D} \text{ satisfies (5.4)}, \\
&= \sup_\alpha \sup_{x>0} \|S_\alpha(x)\|/\|x\| \\
&= \sup_\alpha \|S_\alpha\|,
\end{aligned}$$

thus (5.4) is satisfied. We have shown that $\mathbf{C} \to \mathbf{D}$ is a normed vector lattice satisfying (5.1)–(5.4). It remains to show that $\mathbf{C} \to \mathbf{D}$ is a Banach space, i.e., that $\mathbf{C} \to \mathbf{D}$ is complete in the $\| \|_+$ metric. This follows immediately from

LEMMA. *Any normed vector lattice satisfying* (5.1)–(5.4) *is a Banach space.*

Proof. Suppose $(\mathbf{C}, \mathbf{P}, \| \|)$ is a normed vector lattice satisfying (5.1)–(5.4), and let x_n be a Cauchy sequence. Assume a subsequence has been chosen so that for all n, $\|x_m - x_k\| \leqslant 2^{-n}$ for all $m, k \geqslant n$. Define

$$a_n = \sum_{m \geqslant n} |x_{m+1} - x_m|.$$

a_n exists by (5.3), since the set of all

$$\sum_{n \leqslant m \leqslant N} |x_{m+1} - x_m|, \qquad N \geqslant n,$$

is a $\|\ \|$-bounded directed set:

$$\left\| \sum_{n \leqslant m \leqslant N} |x_{m+1} - x_m| \right\| \leqslant \sum_{n \leqslant m \leqslant N} \|x_{m+1} - x_m\| \leqslant 2^{-n+1}.$$

Moreover, by (5.4), $\|a_n\| \leqslant 2^{-n+1}$. Now

$$\begin{aligned} x_{n+1} + a_{n+1} &= x_n + x_{n+1} - x_n + a_{n+1} \\ &\leqslant x_n + |x_{n+1} - x_n| + a_{n+1} \\ &= x_n + a_n, \end{aligned}$$

and similarly $x_{n+1} - a_{n+1} \geqslant x_n - a_n$, so the sequence of intervals $[x_n - a_n, x_n + a_n]$ is a descending chain with respect to set inclusion. By conditional completeness, there is an x contained in the intersection of the $[x_n - a_n, x_n + a_n]$, thus $|x_n - x| \leqslant 2a_n$. By (5.1) and (5.2), $\|x_n - x\| \leqslant 2\|a_n\| \downarrow 0$, thus x_n converges to x. ∎

6. A Result on Probabilistic Programs

In Section 3 we showed that all probabilistic programs denote elements of $\mathbf{S}' \cap \mathbf{P}'$. However, the reverse inclusion does not hold, so it is natural to seek a characterization of those elements of $\mathbf{S}' \cap \mathbf{P}'$ which are denoted by programs. Theorem 6.1 below sheds some light on this question. It says that *all programs are completely determined by their behavior on fixed inputs*. That is, if S and T are two programs such that $S(\mu) = T(\mu)$ whenever μ is a point mass, then $S = T$ under Semantics 2.

The closure of the linear span of the point masses is the set of discrete measures, i.e., those μ such that $\mu = \mu_B$ for some countable set B. Since programs are linear and continuous, it is immediate that programs which agree on all point masses also agree on all discrete measures. However, this argument does not say anything about the behavior of the programs on nondiscrete measures.

Any measure μ can be decomposed uniquely into its discrete and continuous parts $e_{\text{disc}}(\mu)$ and $e_{\text{cont}}(\mu) = \mu - e_{\text{disc}}(\mu)$. The projection e_{disc} which takes a measure into its discrete part is a continuous linear transformation in $\mathbf{S}' \cap \mathbf{P}'$, given by $\sup e_B$, where the supremum is taken over all countable measurable sets B. This supremum exists and is in $\mathbf{S}' \cap \mathbf{P}'$ by Theorem 5.6. The projection $e_{\text{cont}} = I - e_{\text{disc}}$, also in $\mathbf{S}' \cap \mathbf{P}'$, takes a measure into its continuous part. There are certainly distinct elements of $\mathbf{S}' \cap \mathbf{P}'$ which agree on the discrete measures; I and e_{disc}, for example. Since I is given by a program, Theorem 6.1 says that there is no program to compute e_{disc}.

It is relatively easy to see why Theorem 6.1 holds 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 described a program S_i consisting of the finite sequence of simple assignments that occur along that path, with no conditional tests or **while** loops. Moreover, the set of inputs that follow this

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, and

$$S = \sum_i S_i \circ e_{B_i}.$$

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 much 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, $0 \leqslant x \leqslant 1$, halts with probability exactly x. In this example, there are uncountably many behavior patterns, one for each $0 \leqslant x \leqslant 1$.

In general, the situation is even more complicated than this. The random number generator may satisfy an arbitrary distribution, discrete or continuous. 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, we have

THEOREM 6.1. *If S, T are programs such that $S(\mu) = T(\mu)$ for all point masses $\mu \in \mathbf{B}(X^n, M^{(n)})$, then $S = T$.*

Proof. Suppose S and T agree on all point masses. Then they agree on all 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 Semantics 1 of Section 3.2, S and T denote partial measurable functions f_S, $f_T: X^{n+\omega} \to X^{n+\omega}$, 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, 1950, Theorem A, p. 141), and

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

(Halmos, 1950, Theorem B, p. 144). Let $\varepsilon \geqslant 0$ be arbitrarily small. By definition of integral, there is a simple function s_A with

$$0 \leqslant s_A(x) \leqslant \rho^\omega(A_x) \qquad \text{for all } x, \tag{6.2}$$

such that

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

or in other words

$$(\mu \times \rho^\omega)(A) - \int_{X^n} s_A(x)\, d\mu \leqslant \varepsilon. \tag{6.3}$$

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 $0 \leqslant a_{A,C}$.

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 $\mu(C)$. Then

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

by 6.2, so

$$\begin{aligned} \int_{X^n} s_A(x)\, d\mu &= \sum_{C \in \sigma} a_{A,C} \mu(C) \\ &\leqslant \sum_{C \in \sigma} \rho^\omega(A_{x_C})\, \mu(C) \\ &= (\nu \times \rho^\omega)(A). \end{aligned} \tag{6.4}$$

Also,

$$\begin{aligned} (\nu \times \rho^\omega)(X^{n+\omega}) &= (\nu \times \rho^\omega)\left(\bigcup_{A \in \pi} A\right) \\ &= \sum_{A \in \pi} \sum_{C \in \sigma} \rho^\omega(A_{x_C})\, \mu(C) \\ &= \sum_{C \in \sigma} \mu(C) \sum_{A \in \pi} \rho^\omega(A_{x_C}) \\ &= \mu(X^n)\, \rho^\omega(X^\omega) \\ &= (\mu \times \rho^\omega)(X^{n+\omega}). \end{aligned} \tag{6.5}$$

By 6.3, 6.4, and 6.5, for any $A \in \pi$, $(\mu \times \rho^\omega)(A)$ and $(\nu \times \rho^\omega)(A)$ differ by no more than $|\pi|\varepsilon$, 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, $(\mu \times \rho^\omega)(f_S^{-1}(B \times X^\omega))$ differs from $(\nu \times \rho^\omega)(f_S^{-1}(B \times X^\omega))$ by no more than $2|\pi|\varepsilon$, 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 Theorem 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 $4|\pi|\varepsilon$. As ε was arbitrary,

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

thus by Theorem 3.3.9, $S(\mu) = T(\mu)$. ∎

REFERENCES

ADLEMAN, L. (1978), Two theorems on random polynomial time, *in* "Proceedings, 19th Symposium on Foundation of Computer Science," Ann Arbor, pp. 75–83.

BACKUS, J. *et al.* (1957), The Fortran automatic coding system, *in* "Proceedings, Western Joint Computer Conf." Los Angeles, pp. 188–198.

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

BIRKHOFF, G. (1967), "Lattice Theory," 3rd ed., Amer. Math. Soc. Colloquium Publications, Vol. 25, Providence, R. I.

CHUNG, K. L. (1974), "A Course in Probability Theory," 2nd ed. Academic Press, New York.

COLLATZ, L. (1966), "Functional Analysis and Numerical Mathematics," Academic Press, New York.

DUNFORD, N. AND SCHWARTZ, J. (1958), "Linear Operators," Vol. 1. Interscience, New York.

FELLER, W. (1968), "An Introduction to Probability Theory and Its Applications," Vol. 1, 3rd ed. Wiley, New York.

FLOYD, R. AND RIVEST, R. (1975), Expected time for selection, *Comm. Assoc. Comput. Mach.* **18**, 165–172.

GILL, J. (1974), Computational complexity of probabilistic Turing machines, *in* "Proceedings, 6th ACM Symposium on Theory of Computing," pp. 91–95.

GOUDA, M. AND MANNING, E. (1976), Probabilistic cost machines, *in* "Algorithms and Complexity" (J. F. Traub, Ed.), p. 462, Academic Press, New York.

HALMOS, P. (1950), "Measure Theory," Van Nostrand, New York.

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

KARP, R. (1976), Probabilistic analysis of combinatorial search, *in* "Algorithms and Complexity" (J. F. Traub, Ed.), pp. 1–20, Academic Press, New York.

KNUTH, D. (1973), "Art of Computer Programming: Sorting and Searching," Vol. 3, Addison–Wesley, Reading, Mass.

KURTZ, T. E. (1978), *SIGPLAN Notices* **13**:8, 103–118.

LEHMANN, D. (1976), Categories for fixpoint semantics, *in* "Proceedings, 17th IEEE Symposium on Foundations of Computer Science," pp. 122–126.

MANNA, Z. (1974), "Mathematical Theory of Computation," McGraw–Hill, New York.

MILLER, G. (1975), Riemann's hypothesis and tests for primality, *in* "Proceedings, 7th ACM Symposium on Theory of Computing," pp. 234–239.

PAZ, A. (1971), "Introduction to Probabilistic Automata," Academic Press, New York.

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

RABIN, M. O. (1976), Probabilistic algorithms, *in* "Algorithms and Complexity" (J. F. Traub, Ed.), pp. 21–40, Academic Press, New York.

RAMSHAW, L. H. (1979), "Formalizing the Analysis of Algorithms," Ph. D. Thesis, Computer Science, Stanford University.

SCOTT, D. (1970), Outline of a mathematical theory of computation, *in* "Proceedings, 4th Princeton Conf. on Info. Sci. and Sys.," Princeton, pp. 169–176.

SCOTT, D. AND STRACHEY, C. (1971), Towards a mathematical semantics for computer languages. Tech. mono. PRC6, Oxford Univ.

SOLOVAY, R. AND STRASSEN, V. (1977), Fast Monte Carlo tests for primality, *SIAM J. Comput.* 684–85.

VUILLEMIN, J. (1973), "Proof Techniques for Recursive Programs," Ph. D. thesis, Stanford University.

YAO, A. (1977), Probabilistic computations: toward a unified measure of complexity, *in* "Proceedings, 18th IEEE Symp. on foundations of Computer Science," Providence, pp. 222–227.

YAO, A. AND YAO, F. (1978), On the average case complexity of selecting the kth best, *in* "Proceedings, 19th IEEE Symp. on Foundations of Computer Science," Ann Arbor, pp. 180–289.

ZEIGER, H. P. (1969), Formal models for some features of programming languages, *in* "Proceedings, 1st ACM Symposium on Theory of Computing," Marina del Rey, Calif., pp. 211–215.