Probabilistic Predicate Transformers

CARROLL MORGAN, ANNABELLE McIVER, and KAREN SEIDEL
Oxford University

Probabilistic predicates generalize standard predicates over a state space; with probabilistic predicate transformers one thus reasons about imperative programs in terms of probabilistic pre- and postconditions. Probabilistic healthiness conditions generalize the standard ones, characterizing "real" probabilistic programs, and are based on a connection with an underlying relational model for probabilistic execution; in both contexts demonic nondeterminism coexists with probabilistic choice. With the healthiness conditions, the associated weakest-precondition calculus seems suitable for exploring the rigorous derivation of small probabilistic programs.

Categories and Subject Descriptors: D.2.4 [**Program Verification**]: Correctness Proofs; D.3.1 [**Programming Languages**]: Formal Definitions and Theory; F.1.2 [**Modes of Computation**]: Probabilistic Computation; F.3.1 [**Semantics of Programming Languages**]: Specifying and Verifying and Reasoning about Programs; G.3 [**Probability and Statistics**]: Probabilistic Algorithms

General Terms: Verification, Theory, Algorithms, Languages

Additional Key Words and Phrases: Galois connection, nondeterminism, predicate transformers, probability, program derivation, refinement, weakest preconditions

1. MOTIVATION AND INTRODUCTION

The weakest-precondition approach of Dijkstra [1976] has been reasonably successful for rigorous derivation of small imperative programs, even (and especially) those involving demonic nondeterminism. The associated space of predicate transformers has been studied as a semantic model in its own right.

We adapt that standard (nonprobabilistic) approach to include programs containing probabilistic choice, aiming to extend rigorous reasoning to situations such as the following.

Distributed Systems. Explicit probability can be used to construct distributed algorithms for which no standard counterpart exists (symmetry breaking).

Randomized Algorithms. Some algorithms achieve high efficiency at the expense of only probable correctness (primality testing).

Morgan was partially supported during this work by the Dutch *Specification and Transformation of Programs* (STOP) project. McIver and Seidel were supported by the EPSRC.
Authors' address: Programming Research Group, Wolfson Building, Parks Rd, Oxford OX1 3QD, UK; email: {carroll; anabel; karen}@comlab.ox.ac.uk.

Quantifiable Unreliability. Probability may be unavoidably present in an implementation due to inherent unreliability of components (communication protocols over unreliable media).

Success will depend in part on achieving a low cost for the extra reasoning required.

Probabilistic predicates can be defined in a way first proposed by Kozen [1983], which we extend to allow nondeterminism as well. As an introduction, consider deterministic, nondeterministic, and probabilistic programs in turn.

To express that the deterministic program $n{:=}\ 1$ is guaranteed to establish $n = 1$, using ordinary predicate transformers, we write

$$wp.(n{:=}\ 1).(n = 1) \quad = \quad true\ .$$

Deterministic programs take each initial state to at most one terminating final state.

Nondeterministic programs can have more than one terminating final state for each initial state: for example, the program $n{:=}\ 1 \sqcap n{:=}\ 2$ chooses demonically between the two alternatives $n{:=}\ 1$ and $n{:=}\ 2$. Although it is guaranteed to establish $n = 1 \vee n = 2$, we can say nothing about whether it will establish $n = 1$ or $n = 2$ separately. The weakest-precondition formulation

$$wp.(n{:=}\ 1 \sqcap n{:=}\ 2).(n = 1) \quad = \quad false \quad = \quad wp.(n{:=}\ 1 \sqcap n{:=}\ 2).(n = 2)$$

expresses that pessimistic view (the usual): there are no conditions under which termination establishing $n = 1$ (or $n = 2$) is guaranteed.

Finally, the probabilistic program

$$n{:=}\ 1 \ \ {}_{\frac{1}{2}}\!\oplus\ n{:=}\ 2$$

also chooses between two alternatives, and clearly it too is guaranteed to establish $n = 1 \vee n = 2$. Here however we can say something about whether it will establish $n = 1$ or $n = 2$ separately: with a suitable generalization of wp we will find that

$$wp.(n{:=}\ 1 \ {}_{\frac{1}{2}}\!\oplus\ n{:=}\ 2).(n = 1) \quad = \quad \frac{1}{2} \quad = \quad wp.(n{:=}\ 1 \ {}_{\frac{1}{2}}\!\oplus\ n{:=}\ 2).(n = 2)\ ,$$

expressing that termination establishing $n = 1$ (or $n = 2$) is guaranteed with probability at least $1/2$.

Our source for the probabilistic predicate transformers is Claire Jones' thesis [1990], where it is shown how Jones and Plotkin's general technique for introducing probability into a semantic domain, described in Jones and Plotkin [1989], can be used to reproduce the probabilistic predicate transformer model proposed earlier by Kozen [1983].

Since Kozen and Jones consider only deterministic programs, their underlying operational model is one of functions from initial to final state. Jifeng He [1996] proposed a relational (rather than functional) model of probabilistic execution, adding nondeterminism by exploiting the "convex and up-closed" sets of distributions proposed in Morgan et al. [1994]. (Nondeterminism as sets of distributions is suggested in Lowe [1993] also.)

The technical contribution of this article is to link Kozen's and He's models—we show that the relations of He provide a more general operational semantics

underlying the predicate transformers of Kozen, leading in particular to a simple treatment of nondeterminism and probability together. In the standard case such a link generates "healthiness" (well-formedness) conditions for predicate transformers, characterizing collectively the images of relational programs in the richer predicate-transformer space: strictness, monotonicity, positive conjunctivity, and continuity.

In the probabilistic case, we are able to formulate and prove probabilistic healthiness conditions that generalize the standard ones.

The presentation is as follows: in Section 6 we introduce the correspondence between He's and Kozen's models; in Section 7 we propose probabilistic healthiness conditions; in Section 8 we show that those conditions characterize "real" programs exactly; and in Section 9 we extend Kozen's work to include nondeterminism. Section 10 discusses probabilistic termination and recursion.

Earlier sections summarize He's and Kozen's approaches, on which our work is based; and example programs are given in Sections 2, 4, 9, 10, and 11.

Applications of our results to practical program derivation appear in Morgan [1995].

We use infix dot for function application (thus $f.x$, not $f(x)$), and it associates to the left; we use the syntax

$$(\mathcal{Q}s{:}\, S \mid \mathit{range} \cdot \mathit{expr})$$

for $\mathcal{Q}$-quantification (for example, $\mathcal{Q}$ as $\forall$, $\exists$, λ, or set comprehension) of the values *expr* formed when s ranges over those elements of S satisfying *range*. A glossary is given in Appendix B.

2. DETERMINISTIC PROBABILITY

Standard programs are "deterministic" when their final states (or outputs) are predictable from their initial states (or inputs). We call probabilistic programs deterministic when the *distribution* of their final states is predictable from their initial states. Thus flipping a fair coin (repeatedly) is a deterministic activity, since the distribution of head/tail results is known in advance.[1]

We now consider a simple operational model for deterministic probabilistic behavior.

An imperative program begins execution in an *initial* state and terminates (if it does) in a *final* state. Without probability, we take *deterministic* to mean that for every initial state from which termination is guaranteed there is exactly one corresponding final state; from an initial state that can lead to nontermination, however, any final state is possible as well.[2]

That operational view is expressed as follows: given a set S of states, a (nonprobabilistic) deterministic program denotes a total function of type $S \to S_\perp$, where we write $S_\perp$ for the set $S \cup \{\perp\}$ so that nontermination can be indicated when

[1]Making an arbitrary choice from a collection of biased coins, and then flipping that, is nondeterministic: although the distribution is fixed once the choice is made, it cannot be predicted beforehand.

[2]Some writers call that *predeterministic*, others *liberally deterministic* [Hesselink 1992]: it means "deterministic if terminating."

appropriate by an (improper) final state $\perp$. We call nonprobabilistic programs *standard*.

In this article we require the *state space* S to be finite.[3]

A probabilistic program takes an (initial) state to a (final) probability "distribution" over S rather than to some element of $S_\perp$:

Definition 2.1. For state space S, the set of *distributions* over S is

$$\overline{S} \quad := \quad \{D\colon S \to [0,1] \mid \textstyle\sum D \leq 1\} \ ,$$

the set of functions from S into the closed interval of reals $[0,1]$ that sum to no more than 1. (We write $\sum D$ to abbreviate $\sum_{s:S} D.s$.)

Thus a distribution over S assigns a probability to every element individually of S, and those probabilities must sum to no more than 1.[4]

Definition 2.1 differs slightly from the usual "probability function over an event space" of elementary probability (e.g., Grimmett and Welsh [1986]). It is more specific in that by assigning probabilities to individual elements of S we are taking the (discrete) event space comprising all subsets of S, which means for example that we cannot describe a "uniform" distribution if S is infinite: we are therefore unable to treat programs like

$$x{:=}\ \text{"some value chosen uniformly from } [0,1]\text{"} \ .$$

It is more general in that the restriction $\sum D \leq 1$ is not an equality, so that for D in $\overline{S}$ the difference

$$1 - \textstyle\sum D$$

may be regarded as the probability of "no state at all"—a convenient treatment of nontermination that allows us to forgo $\perp$.

There is a partial order over $\overline{S}$, inherited pointwise from $[0,1]$:

Definition 2.2. For $D, D'\colon \overline{S}$ we define

$$D \sqsubseteq D' \quad := \quad (\forall s\colon S \cdot D.s \leq D'.s) \ .$$

The least element of $\overline{S}$ is $(\lambda s\colon S \cdot 0)$, which we write $\mathbf{0}$; and D in $\overline{S}$ is maximal when $\sum D = 1$, in which case D indicates certain termination.

Thus we define a space of probabilistic deterministic programs together with a refinement order:

Definition 2.3. For state space S the space of deterministic probabilistic programs over S is defined

$$\mathcal{D}S \quad := \quad (S \to \overline{S}, \sqsubseteq) \ ,$$

where for programs d, d' in $S \to \overline{S}$ we define $\sqsubseteq$ pointwise (again):

$$d \sqsubseteq d' \quad := \quad (\forall s\colon S \cdot d.s \sqsubseteq d'.s) \ .$$

[3]Infinite state spaces interfere not so much with definitions as with proofs: Lemma 8.4 would then be argued in an infinite dimensional Euclidean space, where Lemma A.2 does not apply. In McIver and Morgan [1996] the results of this article are shown to apply in the infinite case provided continuity is imposed on the predicate transformers.

[4]In the terminology of Kozen [1981, p.331] such distributions would be called *discrete subprobability measures*.

The order $\sqsubseteq$ of $\mathcal{D}S$ is called the *refinement* order.

It is easy to see that $\mathcal{D}S$ is a complete partial order; and its least element is the nowhere-terminating program

$$\mathbf{abort}.s := \mathbf{0} \quad \text{or equivalently} \quad \mathbf{abort}.s.s' := 0 \ , \tag{1}$$

for any initial state s and final state s'. (Recall that **abort** is deterministic in our sense, because it is nowhere-terminating.) We see thus that for every program d in $\mathcal{D}S$ we have $\mathbf{abort} \sqsubseteq d$.

In general, if $\sum d.s = 1$ then $d.s.s'$ is the probability that d takes s to s'; but if $\sum d.s < 1$ then $d.s.s'$ is only a lower bound for that probability, since the aborting component of weight $1 - \sum d.s$ is unpredictable. Thus for **abort** itself, where that lower bound is 0, we are told nothing about the probability of reaching any particular final s'.

For state s in S we write $\overline{s}$ for the distribution "certainly s":

Definition 2.4. For state s the *point distribution* at s is defined

$$\begin{array}{rcl} \overline{s}.s' & := & \begin{array}{l} 1 \text{ if } s = s' \\ 0 \text{ otherwise.} \end{array} \end{array}$$

Thus a standard deterministic program can be lifted into the probabilistic space as follows:

Definition 2.5. For every standard deterministic program t in $S \to S_\perp$ there is a corresponding probabilistic program in $\mathcal{D}S$, defined

$$\begin{array}{rcl} \overline{t}.s & := & \begin{array}{ll} \overline{t.s} & \text{if } t.s \neq \perp \\ \mathbf{0} & \text{otherwise,} \end{array} \end{array}$$

where s is an arbitrary (initial) state. Equivalently we could define (for arbitrary final state s')

$$\begin{array}{rcl} \overline{t}.s.s' & := & \begin{array}{l} 1 \text{ if } t.s = s' \\ 0 \text{ otherwise,} \end{array} \end{array}$$

noting that s, s' are restricted to proper values (not $\perp$).

A properly probabilistic program over state space (some finite subset of) N is for example

$$n{:=}\ 1 \ {}_{\frac{1}{3}}\!\oplus\, n{:=}\ 2 \ , \tag{2}$$

where in general ${}_p\!\oplus$ chooses its left, right operand with probability $p, 1-p$ respectively. The meaning of (2) is d, where

$$\begin{array}{rcl} d.n.n' & := & \begin{array}{ll} 1/3 & \text{if } n' = 1 \\ 2/3 & \text{if } n' = 2 \\ 0 & \text{otherwise.} \end{array} \end{array}$$

(The absence of n on the right reflects only that (2) is insensitive to its initial state.)

We do not give here the definitions of program operators for $\mathcal{D}S$, since we treat them later in a more general setting (Definitions 5.5–5.7). But note for example that if we define

$$n{:=}\ 1 \ {}_{\frac{1}{3}}\!\oplus\, \mathbf{abort} \tag{3}$$

by d' where

$$d'.n.n' \quad := \quad \begin{array}{ll} 1/3, & \text{if } n' = 1 \\ 0, & \text{if } n' = 2 \\ 0, & \text{otherwise,} \end{array}$$

then we have $\mathbf{abort} \sqsubset d' \sqsubset d$, thus illustrating both nonstrictness of ${}_{1/3}\oplus$ and proper (nontrivial) refinement. Note also that terminating programs in $\mathcal{D}S$ are maximal in the refinement order, which justifies further their being called deterministic.

3. EXPECTED VALUES

Expected values of random variables are the basis for probabilistic predicate transformers.

A *random variable* α is a real-valued function over the sample space[5] (for us, S), and given a probability distribution D (from $\overline{S}$) we can define the expected value of α as follows [Grimmett and Welsh 1986]:

Definition 3.1. For random variable $\alpha\colon S \to R$ and distribution $D\colon \overline{S}$ the *expected value of* α *over* D is

$$\int_D \alpha \quad := \quad \sum_{s:S} (\alpha.s \times D.s) \ .$$

(We use the integral notation for compatibility with expectations in general.[6])

With Definition 3.1 we can write the expectation of the square of the final value delivered by a program d, if executed in initial state n, as

$$\int_{d.n} n^2 \,\mathrm{d}n \ , \tag{4}$$

where we read $(n^2\,\mathrm{d}n)$ as the function $(\lambda n \cdot n^2)$ over (that finite subset of) the natural numbers forming the state space. Note that n in the subscript $d.n$ is free (the initial state), but n in the body $(n^2\,\mathrm{d}n)$ is bound. Applying (4) to Program (2) gives us $(1/3)1^2 + (2/3)2^2$, or expectation 3 for the square of the final values delivered.[7]

More significantly, Definition 3.1 can be used to apply a probabilistic program to a distribution of initial states (rather than only to a "definite" single state),

[5]That simple definition is possible because of the discrete nature of our distributions $\mathcal{D}S$; the functions are "measurable," satisfying for example Grimmett and Welsh [1986, p.22, Cond. 2] trivially.

[6]For example, the expectation (or average) of the function $f\colon R \to R$ over the interval $[a, b]$ is given in analysis by

$$\frac{1}{b-a} \times \int_a^b f(x)\,\mathrm{d}x \ ,$$

which in the above style one could write $\int_{\langle a,b\rangle} f$, inventing the notation $\langle a, b\rangle$ to denote a uniform distribution on $[a, b]$. We are in fact dealing with integration over measures; but for ease of calculation we write $\int_\mu f$ rather than the $\int f\mathrm{d}\mu$ of Jones [1990].

[7]In practice we consider only nonnegative random variables, so that even in the presence of nondeterminism we get a "minimum" expectation.

for example to the (intermediate) distribution yielded by the program immediately before it in a sequential composition. Following Jones [1990, Sec. 4.2] we have

Definition 3.2. For $d{:}\,\mathcal{D}S$, (initial) distribution $I{:}\,\overline{S}$ and (final) state $s'{:}\,S$ we define $d^\dagger$, an element of $\overline{S} \to \overline{S}$, as follows:

$$d^\dagger.I.s' \quad := \quad \int_I d.s.s' \,\mathrm{d}s \ .$$

Thus we may if we wish regard probabilistic deterministic programs as homogeneous functions (from $\overline{S}$ to itself), taking initial to final distributions—the distributions then represent "probabilistic states," as for example in Kozen [1981], and in execution the program moves from one probabilistic state to the next.[8]

The specialization to our earlier view ($S \to \overline{S}$) is obtained for initial state s by taking the point distribution $\overline{s}$ concentrated there; we then observe that from Definition 3.2 (with change of bound variable) and Definition 3.1 we have

$$d^\dagger.\overline{s}.s' \quad = \quad \int_{\overline{s}} d.s''.s' \,\mathrm{d}s'' \quad = \quad d.s.s' \ ,$$

so that $d.s$ is recovered as $d^\dagger.\overline{s}$.

The homogeneous view allows for example a simple definition for sequential composition of the deterministic programs d, d'; but we postpone consideration of program operators until Section 5, where we treat nondeterminism as well.

4. PROBABILISTIC PROGRAM LOGIC

The logic of Jones [1990, Ch. 7] treats deterministic probabilistic programs at the level of Hoare triples [Hoare 1969] and weakest preconditions [Dijkstra 1976]. We summarize it below and then generalize to include nondeterminism.

A standard program t is said to satisfy

$$\begin{array}{rll} & \{pre\}\ t\ \{post\} & \text{[Hoare 1969] but total} \\ \text{or equivalently} & pre \Rightarrow wp.t.post & \text{[Dijkstra 1976],} \end{array}$$

for predicates $pre, post$ over the state space S, if whenever execution of t begins in a state satisfying pre it will terminate in a state satisfying $post$.

In the probabilistic case we use random variables α, β rather than predicates $pre, post$, and expected values rather than membership. For a probabilistic and deterministic program d, the formulae

$$\{\alpha\}\ d\ \{\beta\} \quad \text{or equivalently} \quad \alpha \le wp.d.\beta$$

mean that for every initial state s we have

$$\alpha.s \quad \le \quad \int_{d.s} \beta \ .$$

A more symmetric form comes from taking the homogeneous approach of the last section: we have for any initial distribution I that

$$\int_I \alpha \quad \le \quad \int_{d^\dagger.I} \beta \ .$$

[8] But our programs $d^\dagger$ satisfy Kozen [1981, Thm. 6.1] by construction: they are completely determined by their behavior on the point distributions $\overline{s}$ of $\overline{S}$.

standard	*expression*	*probabilistic*
true iff x and y are equal	$x = y$	1 if x equals y, 0 if they differ
true iff α is *false*	$\neg\alpha$	$1 - \alpha$
true iff α or β is *true*	$\alpha \vee \beta$	$\alpha \sqcup \beta$, the maximum of α and β .

Fig. 1. Idioms for probabilistic predicates.

Finally we generalize to nondeterministic programs, which can take an initial distribution I to many final distributions F; that gives

> for every pair of distributions $I, F{:}\,\overline{S}$ such that (initial) I can be taken to (final) F, we have

$$\int_I \alpha \quad \leq \quad \int_F \beta \ . \tag{5}$$

Thus whereas in the standard case "the truth value of the assertions must not decrease as execution proceeds," in the probabilistic case it is the expectations that must not decrease.

It is instructive to lift the standard logic into the probabilistic. We consider *pre* and *post* as subsets of S, and we form random variables

$$\begin{aligned} \alpha &:= \chi_{pre} \\ \beta &:= \chi_{post} \end{aligned}$$

where for general subset T of S we write χ_T for the *characteristic function* of T. The standard $\{pre\}\ t\ \{post\}$ then says about t just what the probabilistic

$$\{\chi_{pre}\}\ h\ \{\chi_{post}\}$$

says about h; if we take for I, F the point distributions $\overline{s}, \overline{s'}$ corresponding to standard initial and final states s, s' then (5) becomes

> for every pair of states $s, s'{:}\,S$, such that s can be taken to s', we have

$$\int_{\overline{s}} \chi_{pre} \quad \leq \quad \int_{\overline{s'}} \chi_{post} \ .$$

That simplifies (using Definition 3.1) first to $\chi_{pre}.s \leq \chi_{post}.s'$ and finally to

$$s \in pre \quad \Rightarrow \quad s' \in post \ .$$

We call random variables over the state (probabilistic) predicates, when used like α, β above for reasoning about programs: they are our preconditions and postconditions. And we call such a predicate *standard* if it takes values in $\{0, 1\}$ only (equivalently, if it is χ_T for some T a subset of S).

In practice, probabilistic predicates are written as real-valued expressions over the program variables, punning as indicated in Figure 1 between Boolean values and $\{0, 1\}$ where convenient. With those conventions we give in Figure 2 the weakest-precondition semantics of a simple (deterministic) probabilistic language [Jones 1990, p. 146 simplified].

Returning to Program (2), for example, we have from Figure 2 that

$$
\begin{array}{rcl}
wp.\mathbf{abort}.\beta & := & 0 \\
wp.\mathbf{skip}.\beta & := & \beta \\
wp.(x := E).\beta & := & \beta[x := E] \\
wp.(h; h').\beta & := & wp.h.(wp.h'.\beta) \\
wp.(\mathbf{if}\ B\ \mathbf{then}\ h\ \mathbf{else}\ h'\ \mathbf{fi}).\beta & := & B \times wp.h.\beta + (\neg B) \times wp.h'.\beta \\
wp.(h\ {}_p\!\oplus\ h').\beta & := & p \times wp.h.\beta + (1-p) \times wp.h'.\beta
\end{array}
$$

The notation $\beta[x := E]$ denotes syntactic replacement of all free occurrences of variable x in predicate β by expression E.

Fig. 2. Probabilistic predicate semantics.

$$
\begin{array}{rll}
 & wp.(n := 1\ {}_{\frac{1}{3}}\!\oplus\ n := 2).(n = 1) & \\
= & 1/3 \times wp.(n := 1).(n = 1) + 2/3 \times wp.(n := 2).(n = 1) & \\
= & 1/3 \times (1 = 1) + 2/3 \times (2 = 1) & \\
= & 1/3 \times 1 + 2/3 \times 0 & \text{punning between } \mathit{true} \text{ and } 1, \mathit{false} \text{ and } 0 \\
= & 1/3\ , & (6)
\end{array}
$$

and similarly $wp.(n := 1\ {}_{\frac{1}{3}}\!\oplus\ n := 2).(n = 2)$ is $2/3$. Finally, we reason

$$
\begin{array}{rll}
 & wp.(n := 1\ {}_{\frac{1}{3}}\!\oplus\ n := 2).(n = 1 \vee n = 2) & \\
= & 1/3 \times (1 \sqcup 0) + 2/3 \times (0 \sqcup 1) & \\
= & 1\ . & (7)
\end{array}
$$

Note that the final values $1/3$, $2/3$, and 1 are (constant) expressions over the state variables and should be regarded when necessary as constant functions $(\lambda n \cdot 1/3)$ *etc.*

An interpretation of (6) and (7) is that Program (2) has (at least) probability $1/3$ of establishing $n = 1$, and has probability 1 (thus the certainty) of establishing $n = 1 \vee n = 2$. The general reasoning leading to those conclusions is as follows.

Let h be a probabilistic program, *post* a set of (desired final) states, α a probabilistic precondition, and s an (initial) state. Then

$$\{\alpha\}\ h\ \{\chi_{post}\} \tag{8}$$

means that for any initial distribution I, and final distribution F produced by h from it, we have

$$\int_I \alpha \quad \leq \quad \int_F \chi_{post}\ .$$

Taking I to be $\overline{s}$ in particular, and referring to Definition 3.1, we have then

$$\alpha.s \quad \leq \quad \sum_{s' : post} F.s'\ , \tag{9}$$

and thus that the value of α at the initial state s bounds below the probability that the final state s' lies in *post*, no matter what final F may be chosen by h executed from s. Since $wp.h.\beta$ is in general the weakest (largest) α such that $\{\alpha\}\ h\ \{\beta\}$, we conclude that

$$wp.h.\chi_{post}.s \tag{10}$$

is the largest $\alpha.s$ satisfying (9) for all such F, and so is the minimum probability, over all possible final distributions, that h takes initial s into final *post*. (That

"minimum" view will prove to be convenient later, when we consider nondeterminism explicitly: demonic choice strives to make the probability of reaching *post* as low as possible.)

More generally than (8), however, the postconditions will not be standard, and then the (properly) probabilistic predicates will work backward through sequential composition as in this example:

$$\begin{array}{ll} & wp.(\,(x{:=}\ 1\ {}_{\frac{1}{2}}\!\oplus x{:=}\ 2);\ (y{:=}\ 1\ {}_{\frac{1}{2}}\!\oplus y{:=}\ 2)\,).(x \le y) \\ = & wp.(x{:=}\ 1\ {}_{\frac{1}{2}}\!\oplus x{:=}\ 2).(wp.(y{:=}\ 1\ {}_{\frac{1}{2}}\!\oplus y{:=}\ 2).(x \le y)) \\ = & wp.(x{:=}\ 1\ {}_{\frac{1}{2}}\!\oplus x{:=}\ 2).((x \le 1)/2 + (x \le 2)/2) \qquad (11) \\ = & (1 \le 1)/4 + (1 \le 2)/4 + (2 \le 1)/4 + (2 \le 2)/4 \\ = & 1/4 + 1/4 + 0/4 + 1/4 \\ = & 3/4\ . \end{array}$$

Note in particular that at line (11) the "intermediate" postcondition

$$(x \le 1)/2 + (x \le 2)/2$$

is not standard, yet the overall result may still be interpreted as at (8) and (10) above: the program establishes $x \le y$ with probability at least $3/4$. (In fact it is exactly $3/4$, because the program is deterministic and terminating.) Jones [1990, Sec. 7.8] relates the operational view of Section 2 to the axiomatic view (this section), showing soundness and completeness. For deterministic probabilistic program h and initial state s, the connection she gives is

$$wp.h.\beta.s = \int_{h.s} \beta \quad \text{and} \quad h.s.s' = wp.h.\chi_{\{s'\}}.s\ .$$

Our function *wp* of Section 6 generalizes that.

5. NONDETERMINISM AND RELATIONS

He et al. [1996] propose a relational model for imperative probability more general than in Section 2 (from Kozen [1981] and Jones [1990, Ch. 7]): a program executed in an initial state produces not a (single) final distribution but rather a set of them, and the plurality of the set represents nondeterminism. We show that He's construction, with a slight modification, provides a model for the program logic and indeed motivates its extension to include nondeterminism. In this section we explain He's model, and in the following we give the connection between it and the probabilistic program logic.

Not every set of distributions is appropriate as the result of a probabilistic nondeterministic program; we consider only nonempty, up-closed, convex, and Cauchy-closed sets, constraints which we now discuss.

Nonemptiness is imposed to avoid "miracles," or "infeasible programs" [Morgan 1988; Morris 1987; Nelson 1989]. Although they simplify the space structurally (see footnote following Definition 8.1) the presentation here would be complicated by ∞ values in the arithmetic over R.

For up-closure, we recall from Definition 2.2 that $\overline{S}$ is a *cpo*; thus we have the following definition.

Definition 5.1. A subset $\mathcal{F}$ of $\overline{S}$, a set of distributions, is *up-closed* if it is closed under refinement of its elements—if for all $F, F' : \overline{S}$ we have

$$F \in \mathcal{F} \wedge F \sqsubseteq F' \quad \Rightarrow \quad F' \in \mathcal{F} \ .$$

As in Morgan et al. [1994, Sec. 13.3] we insist on up-closure so that refinement (of nondeterministic programs) is expressed by reverse subset inclusion of their result sets: program h is refined by h' if for every initial state s we have $h'.s \subseteq h.s$.

Recall for example the program **abort**. The standard **abort** takes every s to $\bot$; in the deterministic probabilistic model of Section 2 it takes every s to the constant distribution $\mathbf{0}$; and so in our current model it takes every s to the up-closure of $\{\mathbf{0}\}$. But $\mathbf{0}$ is the least element of $\overline{S}$—and so the up-closure is $\overline{S}$ itself, allowing every possible behavior just as one expects from **abort**.

Convexity of a set of distributions is defined as follows:

Definition 5.2. A set $\mathcal{F}$ of distributions is *convex* if for every $F, F' : \mathcal{F}$ and $p : [0, 1]$ we have $F \,{}_p\!\oplus F' \in \mathcal{F}$ also, where

$$(F \,{}_p\!\oplus F').s \quad := \quad p \times F.s \ + \ (1 - p) \times F'.s \ .$$

We insist on convexity so that nondeterminism is refined by any probabilistic choice. Consider for example the program

$$n := 1 \ \sqcap \ n := 2 \ , \tag{12}$$

where we use $\sqcap$ to indicate nondeterministic (demonic) choice.[9] The result set of distributions will contain the two point-distributions $\overline{1}, \overline{2}$, and by convexity will contain also the distributions $\overline{1} \,{}_p\!\oplus \overline{2}$ for any $p : [0, 1]$. Thus our notion of refinement allows Program (12) to be refined to

$$n := 1 \ \,{}_p\!\oplus \ n := 2 \ ,$$

for any probability p.

For Cauchy-closure, we note that our distributions F in $\overline{S}$ are points in Euclidean space, where each axis corresponds to an element s of S and the s-coordinate of F is $F.s$. We have then

Definition 5.3. A set of distributions over S is *Cauchy-closed* if as a subset of R^N it is closed in the usual (Euclidean) sense, where N is the (finite) cardinality of S.

We now give the relational model for nondeterministic probabilistic programs.

Definition 5.4. Given a (finite) state space S, the set of nonempty, up-closed, convex, and Cauchy-closed subsets of $\overline{S}$ is written $\mathcal{C}S$. The *cpo* of nondeterministic probabilistic programs over S is then

$$\mathcal{H}S \quad := \quad (S \to \mathcal{C}S, \sqsubseteq) \ ,$$

where for $h, h' : \mathcal{H}S$ we have

$$h \sqsubseteq h' \quad := \quad (\forall s : S \cdot h.s \supseteq h'.s) \ .$$

[9] The standard definition is $wp.(t \sqcap t').post := wp.t.post \wedge wp.t'.post$.

Note that Cauchy-closure in $\mathcal{C}S$ ensures that $\mathcal{H}S$ is indeed a *cpo*, for the infinite intersection of a $\supseteq$-directed collection of nonempty closed subsets of $[0,1]^N$ cannot be empty. The other conditions on $\mathcal{C}S$ are preserved trivially by arbitrary intersection.

For a full account of the structure of $\mathcal{H}S$ we refer to He et al. [1996]; here we discuss only probabilistic choice ${}_p\oplus$, nondeterministic choice $\sqcap$, and sequential composition. For well-definedness (preservation of the closure conditions) we refer to He et al. [1996] for the first two; we have added Cauchy-closure ourselves, and it is discussed following the definitions.

Probabilistic choice between programs is formed by taking the appropriate combinations of the elements of the result sets.

Definition 5.5. For two programs $h, h' : \mathcal{H}S$, their p-probabilistic combination is defined as

$$(h \,{}_p\oplus\, h').s \quad := \quad \{F : h.s; F' : h'.s \cdot F \,{}_p\oplus\, F'\} \ .$$

Nondeterministic choice is formed by taking all possible probabilistic combinations. Thus

Definition 5.6. For two programs $h, h' : \mathcal{H}S$, their nondeterministic combination is defined as

$$(h \sqcap h').s \quad := \quad (\cup p : [0,1] \cdot (h \,{}_p\oplus\, h').s) \ .$$

Note that the extreme values 1,0 for p ensure that $h \sqcap h' \sqsubseteq h$ and $h \sqcap h' \sqsubseteq h'$.

Finally, for sequential composition we have

Definition 5.7. For two programs $h, h' : \mathcal{H}S$, their sequential composition is defined as

$$(h; h').s \quad := \quad \{F : h.s; d : \mathcal{D}S \mid h' \sqsubseteq d \cdot d^\dagger.F\} \ ,$$

where by $h' \sqsubseteq d$ we mean $(\forall s : S \cdot d.s \in h'.s)$, agreeing with the notion of refinement that would result were $\mathcal{D}S$ to be embedded in $\mathcal{H}S$ in the obvious way.

Thus the sequential composition is formed by taking all possible "intermediate" distributions F that h can deliver from s and for each continuing with all possible deterministic refinements d of h'.

For well-definedness we are obliged to show that for Cauchy-closed arguments each of the above three operators produces a Cauchy-closed result. The reason is the same in each case: a continuous function between Euclidean spaces is applied to a closed and bounded (hence compact) set, and so its image is closed. We consider Definition 5.6 as an example:

LEMMA 5.8. *If subsets $h.s, h'.s$ of $\overline{S}$ are Cauchy-closed then so is*

$$(\cup p : [0,1] \cdot (h \,{}_p\oplus\, h').s) \ .$$

PROOF. By Definition 5.5 the set concerned may be written

$$\{F : h.s; F' : h'.s; p : [0,1] \cdot F \,{}_p\oplus\, F'\} \ ,$$

which is the image in R^N through a continuous function of the closed and bounded subset

$$h.s \times h'.s \times [0,1]$$

of R^{2N+1}, where N is the cardinality of S. □

	standard	probabilistic
relational	$[S \leftrightarrow S_\perp]$	$\mathcal{H}S$
predicate	$PS \leftarrow PS$	$\mathcal{J}S$

Fig. 3. Nomenclature for the four models.

6. REGULAR PREDICATE TRANSFORMERS

6.1 The Relational and Predicate Models

A *relational* model for standard nondeterministic programs over S is

$$S \leftrightarrow S_\perp \ ,$$

where a standard program t relates s to s' just when executing t from initial state s can yield final state s'. A convenient constraint on elements t of $S \leftrightarrow S_\perp$ is that whenever s is taken by t to $\perp$ then it is taken to all of $S_\perp$ (thus allowing simple $\supseteq$ to be the refinement order).

If an element of $S \leftrightarrow S_\perp$ relates every initial state to at least one final state, we say that it is *total*; and if it relates every initial state to at most finitely many final states, we say that it is *image finite*.[10] We write $[S \leftrightarrow S_\perp]$ for the total and image-finite elements of $S \leftrightarrow S_\perp$.

A probabilistic analogue for $[S \leftrightarrow S_\perp]$ is $\mathcal{H}S$—totality corresponds to nonemptiness of result sets, and image-finiteness corresponds to their Cauchy closure.

The *predicate-transformer* model for standard nondeterministic programs is

$$PS \leftarrow PS \ ,$$

where we write the functional arrow backward to emphasize that such programs map sets of final states (postconditions) to sets of initial states (weakest preconditions). A probabilistic analogue for $PS \leftarrow PS$ is suggested by the program logic of Section 4:

Definition 6.1.1. For state space S, the set of probabilistic predicates over S is defined

$$\mathcal{P}S \quad := \quad (S \to R_{\geq}, \leq) \ ,$$

where $R_{\geq}$ is the nonnegative reals, and $\leq$ is inherited pointwise from $R_{\geq}$. The probabilistic predicate transformer model for programs is then

$$\mathcal{J}S \quad := \quad (\mathcal{P}S \leftarrow \mathcal{P}S, \sqsubseteq) \ ,$$

where the *refinement* order $\sqsubseteq$ over programs is derived pointwise from the order on $\mathcal{P}S$.

We refer to the four models as shown in Figure 3; thus $\mathcal{J}S$ for example is the probabilistic predicate model.

The standard relations can be embedded in the standard predicate transformers $PS \leftarrow PS$ such that the image of $S \leftrightarrow S_\perp$ is characterized by *positive conjunctivity* [Hesselink 1992]:

[10] That latter happens automatically under our present assumption that S is finite.

a predicate transformer $p: PS \leftarrow PS$ is the embedding of some relation in $S \leftrightarrow S_\perp$ iff for all postconditions $post, post'$ in PS we have

$$p.(post \cap post') \quad = \quad p.post \cap p.post' \ .$$

Note that it is the finiteness of S that allows us to consider only finite conjunctions.

Thus the standard predicate model is richer than the relational precisely because not all of its programs are positively conjunctive.[11]

In the remainder of this section we construct an embedding for the probabilistic models.

6.2 Embedding the Relational Model in the Predicate Model

Recall from Section 4 that for program $h: \mathcal{H}S$, initial s and every possible F that h may produce from s, we must have

$$wp.h.\beta.s \quad \leq \quad \int_F \beta \ . \tag{13}$$

(Specialize (5) to $I := \overline{s}$ and $\alpha := wp.h.\beta$.) Noting that $wp.h.\beta.s$ should be as large as possible while satisfying (13) for all F in $h.s$, we propose

Definition 6.2.1. The injection $wp: \mathcal{H}S \to \mathcal{J}S$ is defined

$$wp.h.\beta.s \quad := \quad (\sqcap F: h.s \cdot \int_F \beta) \ ,$$

for program $h: \mathcal{H}S$, predicate $\beta: \mathcal{P}S$, and state $s: S$.

It is a consequence of Lemma 8.2 below that (probabilistic) *wp* is indeed an injection.

That we take the minimum $\sqcap$ over result distributions F is related to our demonic view of nondeterminism: the demon resolves nondeterminism in a way that makes the weakest precondition as small (as strong) as possible. Note that well-definedness of the minimum is guaranteed by nonemptiness of $h.s$, which is why we impose it; a more general treatment is possible however if ∞ is allowed as a possible value for $wp.h.\beta.s$.

It is instructive to specialize Definition 6.2.1 to the standard case. Let β there be χ_{post} for arbitrary standard postcondition *post* and suppose h is standard, so that $h.s$ is the closure of some set of point distributions; then

$$\begin{array}{ll} & wp.h.\chi_{post}.s \\ = & (\sqcap F: h.s \cdot \int_F \chi_{post}) \\ = & \qquad\qquad\qquad\qquad\qquad\qquad h \text{ is a standard program: see below} \\ & (\sqcap s': \text{“standard results of } h.s\text{”} \cdot \int_{s'} \chi_{post}) \\ = & (\sqcap s': \text{“standard results of } h.s\text{”} \cdot \chi_{post}.s') \\ \\ = & \ 1 \text{ if “standard results of } h.s\text{”} \subseteq post \\ & \ 0 \text{ otherwise.} \end{array}$$

[11] The extra constraints on $[S \leftrightarrow S_\perp]$ induce strictness and continuity on the corresponding predicate transformers.

In considering only the standard results of $h.s$ (ignoring the closures), we note that the minimum over F of $\int_F \beta$, for any β, is unaffected by leaving out the points generated by closure—up-closure, for example, only increases that value, and convex closure generates only values between the extremes.

Thus $wp.h.\chi_{post}$ is the characteristic function of those initial states all of whose final states produced by h lie in *post*; and that is the standard embedding of $S \leftrightarrow S_\perp$ into $PS \leftarrow PS$.

With the function *wp* we define a subset of $\mathcal{J}S$, the image of $\mathcal{H}S$ there:

Definition 6.2.2. The set of *regular probabilistic predicate transformers* $\mathcal{J}_r S$ over S is the *wp*-image of $\mathcal{H}S$ in $\mathcal{J}S$, thus defined as

$$\mathcal{J}_r S \quad := \quad \{h{:}\,\mathcal{H}S \cdot wp.h\} \ .$$

In the next section we establish properties of $\mathcal{J}_r S$, and in the section following that we show they characterize it, thus identifying a probabilistic analogue of conjunctivity.

7. HEALTHINESS CONDITIONS FOR PROBABILISTIC PROGRAMS

Dijkstra [1976] imposes "healthiness conditions" on the standard predicate transformers $PS \leftarrow PS$: they are strictness,[12] monotonicity, positive conjunctivity, and continuity. The conditions characterize the images, under the standard embedding, of the total and image-finite relations $[S \leftrightarrow S_\perp]$.

In this section we consider probabilistic healthiness, and we see that the probabilistic analogues of strictness, monotonicity, positive conjunctivity, and continuity characterize $\mathcal{J}_r S$.

The analogue of standard positive conjunctivity is probabilistic sublinearity:

Definition 7.1. A predicate transformer $j{:}\,\mathcal{J}S$ is said to be *sublinear* iff for all $\beta_1, \beta_2{:}\,\mathcal{P}S$ and $c_0, c_1, c_2{:}\,R_\geq$ we have

$$c_1(j.\beta_1) + c_2(j.\beta_2) \ominus \boldsymbol{c_0} \quad \leq \quad j.(c_1\beta_1 + c_2\beta_2 \ominus \boldsymbol{c_0}) \ ,$$

where $\ominus$ denotes subtraction in $R_\geq$, so that $x \ominus y = (x - y) \sqcup 0$, and we write $c\beta$ for the pointwise multiplication of the predicate β by the scalar c.

Because $\ominus$ does not associate with $+$, we define its (lower) syntactic precedence: by $x + y \ominus z$ we mean $(x + y) \ominus z$.

We now show that all regular predicate transformers are sublinear.

LEMMA 7.2. (SUBLINEARITY).
All members of $\mathcal{J}_r S$ are sublinear.

PROOF. We write $j = wp.h$ for $h{:}\,\mathcal{H}S$, and then proceed using the notation of Definition 7.1. For arbitrary $s{:}\,S$, we calculate

$$\begin{array}{lll}
 & wp.h.(c_1\beta_1 + c_2\beta_2 \ominus \boldsymbol{c_0}).s & (14) \\
= & (\sqcap F{:}\, h.s \cdot \int_F (c_1\beta_1 + c_2\beta_2 \ominus \boldsymbol{c_0})) & \text{Definition 6.2.1} \\
\geq & (\sqcap F{:}\, h.s \cdot \int_F (c_1\beta_1 + c_2\beta_2 - \boldsymbol{c_0})) & \text{monotonicity of } \textstyle\int_F \text{ and } \sqcap \\
= & (\sqcap F{:}\, h.s \cdot c_1 \textstyle\int_F \beta_1 + c_2 \int_F \beta_2 - c_0 \int_F \mathbf{1}) & \text{distribute } +, -, \times \text{ through } \textstyle\int_F
\end{array}$$

[12] The *Law of the Excluded Miracle*.

$$
\begin{array}{lll}
\geq & (\sqcap F\colon h.s \cdot c_1 \int_F \beta_1 + c_2 \int_F \beta_2 - c_0) & F \in h.s \subseteq \overline{S} \text{ implies } \int_F \mathbf{1} \leq 1 \\
= & (\sqcap F\colon h.s \cdot c_1 \int_F \beta_1 + c_2 \int_F \beta_2) - c_0 & \\
\geq & & + \text{ and } (c\times) \text{ monotonic for } c \in R_{\geq} \\
 & c_1(\sqcap F\colon h.s \cdot \int_F \beta_1) + c_2(\sqcap F\colon h.s \cdot \int_F \beta_2) - c_0 & \\
= & c_1(wp.h.\beta_1.s) + c_2(wp.h.\beta_2.s) - c_0 \ , & \text{Definition 6.2.1}
\end{array}
$$

and since (14) is nonnegative we may replace the final $(-c_0)$ by $(\ominus\, c_0)$. □

That sublinearity is indeed the probabilistic analogue of conjunctivity may be seen by considering a special case: suppose sublinear $j\colon \mathcal{J}S$ takes standard postconditions to standard preconditions. We then have

$$
\begin{array}{lll}
 & j.(\chi_{post_1 \cap post_2}) & \\
= & j.(\chi_{post_1} \sqcap \chi_{post_2}) & \\
= & j.(\chi_{post_1} + \chi_{post_2} \ominus \mathbf{1}) & \\
\geq & j.\chi_{post_1} + j.\chi_{post_2} \ominus \mathbf{1} & \text{sublinearity} \\
= & j.\chi_{post_1} \sqcap j.\chi_{post_2} \ , & j.\chi_{post_{1,2}} \text{ standard}
\end{array}
$$

and equality follows from monotonicity of j (below).

The other probabilistic healthiness conditions are all consequences of sublinearity;[13] we treat them in turn, beginning with monotonicity:

LEMMA 7.3. (MONOTONICITY). *If $j\colon \mathcal{J}S$ is sublinear then it is monotonic.*

PROOF. Take $\beta_1, \beta_2\colon \mathcal{P}S$ with $\beta_1 \geq \beta_2$. Then

$$j.\beta_1 \ = \ j.(\beta_1 - \beta_2 + \beta_2) \ \geq \ j.(\beta_1 - \beta_2) + j.\beta_2 \ \geq \ j.\beta_2 \ .$$

□

Probabilistic monotonicity generalizes standard monotonicity, just as $\leq$ generalizes implication.

Feasibility (exclusion of miracles) is expressed as follows:

LEMMA 7.4. (FEASIBILITY). *If $j\colon \mathcal{J}S$ is sublinear, then for all $\beta\colon \mathcal{P}S$ we have*

$$j.\beta \quad \leq \quad \sqcup\boldsymbol{\beta} \ ,$$

where $\sqcup\beta$ abbreviates $(\sqcup s\colon S \cdot \beta.s)$.

PROOF. We prove first the special case in which $\beta = \mathbf{0}$; for then

$$2 \times j.\mathbf{0} \ \leq \ j.(2 \times \mathbf{0}) \ = \ j.\mathbf{0} \ ,$$

so that $j.\mathbf{0} = \mathbf{0}$. Now for the general case we continue

$$\mathbf{0} \ = \ j.\mathbf{0} \ = \ j.(\beta \ominus \sqcup\boldsymbol{\beta}) \ \geq \ j.\beta \ominus \sqcup\boldsymbol{\beta} \ ,$$

so that $\sqcup\boldsymbol{\beta} \geq j.\beta$ as required. □

Note that the special case $j.\mathbf{0} = \mathbf{0}$ corresponds to standard strictness, since we identify $\mathbf{0}$ and *false*.

Another consequence of sublinearity is the following.

[13] They are also easily proved from Definition 6.2.1 directly, a useful exercise.

feasibility	$j.\beta \le \sqcup\boldsymbol{\beta}$	for $\beta\colon \mathcal{P}S$
monotonicity	$\beta_1 \ge \beta_2 \Rightarrow j.\beta_1 \ge j.\beta_2$	for $\beta_1, \beta_2\colon \mathcal{P}S$
scaling	$j.(c\beta) = c(j.\beta)$	for $\beta\colon \mathcal{P}S$ and $c\colon R_{\ge}$
sublinearity	$c_1(j.\beta_1) + c_2(j.\beta_2) \ominus \mathbf{c_0}$ $\le\ j.(c_1\beta_1 + c_2\beta_2 \ominus \mathbf{c_0})$	for $\beta_1, \beta_2\colon \mathcal{P}S$ and $c_0, c_1, c_2\colon R_{\ge}$
continuity	$j.(\sqcup B) = (\sqcup\beta\colon B \cdot j.\beta)$	for $\ge$-directed subset B of $\mathcal{P}S$

Note that in the presence of scaling we can decompose sublinearity into the two properties

subadditivity	$j.\beta_1 + j.\beta_2 \le j.(\beta_1 + \beta_2)$	for $\beta_1, \beta_2\colon \mathcal{P}S$
truncation	$j.\beta \ominus \mathbf{1} \le j.(\beta \ominus \mathbf{1})$	for $\beta\colon \mathcal{P}S$

The following shows that truncation is essential: let S be the two-element state space $\{a, b\}$, and define $j\colon \mathcal{J}S$ so that

$$j.\beta.s \quad := \quad \beta.a + \beta.b - \sqrt{\frac{(\beta.a)^2 + (\beta.b)^2}{2}}\ .$$

Then j is feasible, monotonic, scaling, and continuous, and moreover it is subadditive. But it does not satisfy truncation (and so is not sublinear): take $\beta.a, \beta.b := 2, 1$ and consider $j.(\beta \ominus \mathbf{1})$.

Fig. 4. Summary of properties characterizing $\mathcal{J}_r S$, the regular predicate transformers.

LEMMA 7.5. (SCALING). *If $j\colon \mathcal{J}S$ is sublinear, then for all $\beta\colon \mathcal{P}S$ and $c\colon R_{\ge}$*

$$j.(c\beta) \quad = \quad c(j.\beta)\ .$$

PROOF. Given sublinearity, we need only prove $\le$; and for $c \neq 0$ we have

$$j.(c\beta) \ =\ c(1/c)(j.(c\beta)) \ \le\ c(j.((1/c)c\beta)) \ =\ c(j.\beta)\ .$$

When $c = 0$ the result is immediate from feasibility. □

There seems to be no standard analogue for scaling.

Finally, for continuity we appeal to the finiteness of S.

LEMMA 7.6. (BOUNDED CONTINUITY). *If $j\colon \mathcal{J}S$ is sublinear and B is a $\ge$-directed and bounded subset of $\mathcal{P}S$, so that $\sqcup B$ exists, then*

$$j.(\sqcup B) \quad = \quad (\sqcup\beta\colon B \cdot j.\beta)\ .$$

PROOF. By monotonicity we need only show $j.(\sqcup B) \le (\sqcup\beta\colon B \cdot j.\beta)$.

Take any c with $c > 1$. For every state s there is a β_s in B such that $\sqcup B.s \le c\beta_s.s$; since B is directed and S is finite we can thus find a single β_c with $\beta_s \le \beta_c$ for all s, so that $\sqcup B \le c\beta_c$. For any such c we have

	$j.(\sqcup B)$	
$\le$	$j.(c\beta_c)$	monotonicity
$=$	$c(j.\beta_c)$	scaling
$\le$	$c(\sqcup\beta\colon B \cdot j.\beta)$,	$\beta_c \in B$

which suffices since c can be arbitrarily close to 1. □

We summarize the properties of $\mathcal{J}_r S$ in Figure 4. It remains to show that those properties characterize $\mathcal{J}_r S$ exactly, and for that we define an inverse to *wp*.

8. CHARACTERIZING REGULAR PROGRAMS

We now show that sublinearity characterizes the regular predicate transformers $\mathcal{J}_r S$: they are exactly the sublinear members of $\mathcal{J}S$.

We begin by defining a map rp from $\mathcal{J}S$ back to $\mathcal{H}S$, to be inverse to wp on $\mathcal{J}_r S$. Thus for $j: \mathcal{J}_r S$ in particular the corresponding relation $rp.j$ should produce from initial s exactly those final distributions F that satisfy (13) for all possible postconditions β:

Definition 8.1. The function $rp: \mathcal{J}S \to \mathcal{H}S$ is defined

$$rp.j.s \quad := \quad \{F: \overline{S} \mid (\forall \beta: \mathcal{P}S \cdot j.\beta.s \leq \int_F \beta)\} \ ,$$

for transformer $j: \mathcal{J}S$ and state $s: S$.

Concerning definedness of rp, we note that $rp.j.s$ satisfies the closure conditions for $\mathcal{C}S$: in fact for fixed j, β, s the constraint

$$j.\beta.s \leq \int_F \beta$$

on F represents a closed upper half-space in R^N which satisfies the three closure conditions trivially, and those conditions are preserved by intersection. (It is an upper half-space because β takes only nonnegative values.)

If $rp.j.s$ is empty for some s, we consider rp to be undefined at that j; but Lemma 8.5 below shows $rp.j$ to be defined whenever j is sublinear.[14]

Our first use of rp is to show that wp is indeed an injection:

LEMMA 8.2. *For any $h: \mathcal{H}S$ we have*

$$rp.(wp.h) \quad = \quad h \ .$$

PROOF. Direct from the definitions we have (taking the contrapositive), for arbitrary $F: \overline{S}$, we have

$$\begin{array}{lll}
 & F \notin rp.(wp.h).s & \\
\text{iff} & F \notin \{F: \overline{S} \mid (\forall \beta: \mathcal{P}S \cdot wp.h.\beta.s \leq \textstyle\int_F \beta)\} & \text{Definition 8.1} \\
\text{iff} & (\exists \beta: \mathcal{P}S \cdot wp.h.\beta.s > \textstyle\int_F \beta) & \\
\text{iff} & (\exists \beta: \mathcal{P}S \cdot (\sqcap F': h.s \cdot \textstyle\int_{F'} \beta) > \textstyle\int_F \beta) & \text{Definition 8.1} \\
\text{iff} & (\exists \beta: \mathcal{P}S; r: R \cdot (\forall F': h.s \cdot \textstyle\int_{F'} \beta > r) \wedge \textstyle\int_F \beta < r) & \\
\text{iff} & F \notin h.s \ . & \text{see below}
\end{array}$$

The deferred "only if" is trivial. For "if" we need the *separating-hyperplane lemma*, which states that any point not in a closed convex subset of R^N can be separated from it by a hyperplane. (See Appendix A.)

In our use of the lemma, the point concerned is F; and $h.s$ is the closed and convex subset of R^N that F is not in, as stated on the last line of the proof above.

[14] If $\mathcal{H}S$ did note impose nonemptiness (Definition 5.4), allowing miracles, then rp would be everywhere defined; in that case for definedness of wp we would require

$$\mathcal{P}S \quad := \quad S \to (R_{\geq} \cup \{\infty\}) \ .$$

The plane separating F from $h.s$ is given to us by Lemma A.1, and can be described by its coefficients β and a constant term r—that is, a point F' lies on that plane iff

$$r \quad = \quad \int_{F'} \beta \ .$$

We choose the signs of β, r so that "$h.s$ on one side" is expressed

$$(\forall F' \colon h.s \cdot \int_{F'} \beta > r) \ , \tag{15}$$

and "F on the other side" is expressed

$$\int_F \beta \quad < \quad r \ ,$$

which two conditions together with the existence of β, r give us our deferred "if" above. Note that the coefficients β are nonnegative since, by up-closure of $h.s$, Condition (15) could not be true otherwise. □

The Cauchy-closure condition on $h.s$ is essential for Lemma 8.2: if $h.s$ were for example

$$\{p \colon (0,1) \cdot \overline{0} \,{}_p\!\oplus \overline{1}\} \ ,$$

thus all probabilistic combinations of the point distributions $\overline{0}$ and $\overline{1}$ excluding the endpoints, then $rp.(wp.h).s$ would be

$$\{p \colon [0,1] \cdot \overline{0} \,{}_p\!\oplus \overline{1}\} \ ,$$

in which closure has occurred.

Our next step is to prove an analogous result for the opposite direction; but since rp is not defined for all $\mathcal{J}S$ (for $j \notin \mathcal{J}_r S$ it may be that $rp.j.s$ is empty for some s), we state the lemmas conditionally.

Our first lemma concerns members of $\mathcal{J}S$ generally:

LEMMA 8.3. *For any $j \colon \mathcal{J}S$, if $rp.j$ is defined then*

$$wp.(rp.j) \quad \sqsupseteq \quad j.$$

PROOF. Take any $\beta \colon \mathcal{P}S$ and $s \colon S$; directly from the definitions we have:

$$\begin{array}{lll} & wp.(rp.j).\beta.s \geq j.\beta.s & \\ \text{iff} & (\sqcap F \colon rp.j.s \cdot \int_F \beta) \geq j.\beta.s & \text{Definition 6.2.1; } rp.j \text{ defined by assumption} \\ \text{iff} & (\forall F \colon rp.j.s \cdot \int_F \beta \geq j.\beta.s) & \\ \text{iff} & (\forall F \colon \overline{S} \cdot (\forall \beta' \colon \mathcal{P}S \cdot j.\beta'.s \leq \int_F \beta') \Rightarrow \int_F \beta \geq j.\beta.s) & \\ \text{iff} & \mathit{true} \ . & \end{array}$$

□

Our second lemma is restricted to sublinear elements of $\mathcal{J}S$:

LEMMA 8.4. *If $j \colon \mathcal{J}S$ is sublinear and $rp.j$ is defined, then*

$$wp.(rp.j) \quad \sqsubseteq \quad j \ .$$

PROOF. We proceed by contradiction: for arbitrary $s \colon S$ and $\beta \colon \mathcal{P}S$, we have

$$
\begin{array}{rll}
 & wp.(rp.j).\beta.s > j.\beta.s & \\
\text{iff} & (\sqcap F\colon rp.j.s \cdot \int_F \beta) > j.\beta.s & \text{Definition 6.2.1; } rp.j \text{ defined by assumption} \\
\text{implies} & (\forall F\colon rp.j.s \cdot \int_F \beta > j.\beta.s) & (16) \\
\text{iff} & (\forall F\colon \overline{S} \cdot (\forall \beta'\colon \mathcal{P}S \cdot j.\beta'.s \le \int_F \beta') \Rightarrow \int_F \beta > j.\beta.s) & \text{Definition 8.1} \\
\text{iff} & (\cap \beta'\colon \mathcal{P}S \cdot \{F\colon \overline{S} \mid j.\beta'.s \le \int_F \beta'\}) \subseteq \{F\colon \overline{S} \mid \int_F \beta > j.\beta.s\} & \\
 & & \\
\text{iff} & & A \subseteq B \text{ iff } A \cap B^C = \emptyset \\
 & \quad (\cap \beta'\colon \mathcal{P}S \cdot \{F\colon R^N \mid j.\beta'.s \le \int_F \beta'\}) & (17) \\
 & \cap\ \{F\colon R^N \mid -j.\beta.s \le \int_F (-\beta)\} & \\
 & \cap\ \{F\colon R^N \mid -1 \le \int_F (\mathbf{-1})\} & \\
 & = \ \emptyset\,, &
\end{array}
$$

where we have negated the second term to make the inequalities uniform. The third term, also negated, is added because of the retyping of the bound variables F: in (17) they are taken from all of R^N, rather than the more restrictive $\overline{S}$, and so we must compensate by requiring explicitly

$$\sum F \quad = \quad \int_F \mathbf{1} \quad \le \quad 1\,.$$

(The remaining constraints $F.s \ge 0$ applying to members F of $\overline{S}$, for every $s\colon S$, are included already in the first term above: take $\beta' := \chi_{\{s\}}$ and recall that $0 \le j.\beta'.s$.)

Now we argue that some finite subcollection of the sets (17) has empty intersection also. Take the sets determined by $\beta' := \chi_{\{s\}}$ for each $s\colon S$ (finitely many), and the final set $\{F\colon R^N \mid -1 \le \int_F \mathbf{-1}\}$: they determine a closed hyperpyramid in the positive hyperoctant of R^N (whose apex points downward toward the origin). Since the pyramid is bounded, it is compact—and so the finite-intersection lemma[15] applies within it. Thus we can restrict our attention to a collection of just M, say, of the sets (17).

We now appeal to another lemma in the style of linear programming (Appendix A). The finite M-collection from (17) with empty intersection may be regarded as a system of M equations

$$A \cdot x \quad \ge \quad r \tag{18}$$

that has no solution in x, where A is an $M \times N$ matrix (of coefficients, representing the β''s, $-\beta$, and $\mathbf{-1}$), x is an $N \times 1$ column vector (representing points F in R^N), and r is an $M \times 1$ column vector (of constant terms, representing the $j.\beta'.s$'s, $-j.\beta.s$, and -1). The expression $A \cdot x$ denotes matrix multiplication, and the inequality $\ge$ is taken rowwise.

Lemma A.2 then gives us a $1 \times M$ row-vector C of nonnegative reals such that

$$C \cdot A = 0 \quad \text{but} \quad C \cdot r > 0\,.$$

Returning to (17), we thus have a finite number of nonnegative coefficients c, $c_0 \cdots c_{M-2}$ such that

$$c\beta \quad = \quad c_1\beta'_1 + \cdots + c_{M-2}\beta'_{M-2} \ominus \mathbf{c_0}$$

[15]The *finite-intersection lemma* states that a collection of closed subsets of a compact set has empty intersection only if some finite subcollection of it does.

but

$$c(j.\beta.s) \quad < \quad c_1(j.\beta'_1.s) + \cdots + c_{M-2}(j.\beta'_{M-2}.s) \ominus c_0 \ ,$$

where in both cases we can use $\ominus$ on the right because the left-hand side is nonnegative.

We then finish with

$$\begin{array}{lll} & c(j.\beta.s) & \\ < & c_1(j.\beta'_1.s) + \cdots + c_{M-2}(j.\beta'_{M-2}.s) \ominus c_0 & \text{above} \\ \leq & j.(c_1\beta'_1 + \cdots + c_{M-2}\beta'_{M-2} \ominus \mathbf{c_0}).s \ , & \text{sublinearity of } j \\ = & j.(c\beta).s \ , & \text{above} \end{array}$$

which contradicts scaling (implied by sublinearity). □

Finally we deal with definedness of *rp*:

LEMMA 8.5. *If $j\colon \mathcal{J}S$ is sublinear, then $rp.j$ is defined.*

PROOF. The three closure conditions have already been dealt with. For nonemptiness, again we proceed by contradiction: suppose for some $s\colon S$ that $rp.j.s$ is empty. Then we have immediately

$$(\forall F\colon rp.j.s \cdot \int_F \beta > j.\beta.s) \ ,$$

by quantifying universally over the empty set (the body of the quantification is irrelevant). But that is identical to (16) in the proof of Lemma 8.4, and *rp* is not mentioned beyond that point. Thus we proceed to a contradiction as before. □

We thus have thus established the following:

LEMMA 8.6. *If $j\colon \mathcal{J}S$ is sublinear, then $rp.j$ is defined and*

$$wp.(rp.j) \quad = \quad j \ .$$

PROOF. The result is immediate from Lemmas 8.3–8.5. □

With Lemmas 7.2 and 8.6 we have finally our characterization of $\mathcal{J}_r S$:

THEOREM 8.7. *The regular predicate transformers $\mathcal{J}_r S$ are characterized by sublinearity: j in $\mathcal{J}S$ is sublinear iff it is in $\mathcal{J}_r S$.*

PROOF. "If" is given by Lemma 7.2; "only if" is given by Lemma 8.6. □

Note that Theorem 8.7 shows also that the conditions of Figure 4 characterize $\mathcal{J}_r S$ collectively, since by Lemmas 7.2–7.6 they are all implied by sublinearity.

9. NONDETERMINISM AND PREDICATE TRANSFORMERS

We have now seen that $\mathcal{H}S$ and $\mathcal{J}_r S$ are placed in 1-1 correspondence by the mutual inverses $wp\colon \mathcal{H}S \to \mathcal{J}_r S$ and $rp\colon \mathcal{J}_r S \to \mathcal{H}S$. But for the two spaces to give equivalent semantics for regular programs requires further that the program operators preserve the correspondence—that for example Definitions 5.5–5.7, for relations, correspond with the definitions of Figure 2 for predicate transformers.

In fact neither Kozen nor Jones treats nondeterminism, and thus in the case of Definition 5.6 rather than prove a correspondence we must generate a definition: for $h_1, h_2\colon \mathcal{H}S$, $\beta\colon \mathcal{P}S$ and $s\colon S$ we calculate

$$
\begin{array}{lll}
 & wp.(h_1 \sqcap h_2).\beta.s & \\
= & (\sqcap F\colon (h_1 \sqcap h_2).s \cdot \textstyle\int_F \beta) & \text{Definition 6.2.1} \\
= & (\sqcap F_1\colon h_1.s; F_2\colon h_2.s; p\colon [0,1] \cdot \textstyle\int_{F_1 {}_p\!\oplus F_2} \beta) & \text{Definition 5.6} \\
= & (\sqcap F_1\colon h_1.s; F_2\colon h_2.s; p\colon [0,1] \cdot p(\textstyle\int_{F_1} \beta) + (1-p)(\textstyle\int_{F_2} \beta)) & \text{Definition 3.1} \\
 & & \\
= & & \text{minimum occurs at extremes, thus at } p=0 \text{ or } p=1 \\
 & (\sqcap F_1\colon h_1.s; F_2\colon h_2.s \cdot \textstyle\int_{F_1} \beta \sqcap \textstyle\int_{F_2} \beta) & \\
= & (\sqcap F_1\colon h_1.s \cdot \textstyle\int_{F_1} \beta) \sqcap (\sqcap F_2\colon h_2.s \cdot \textstyle\int_{F_2} \beta) & \\
= & wp.h_1.\beta.s \sqcap wp.h_2.\beta.s\ . & \text{Definition 6.2.1}
\end{array}
$$

Thus we are given a definition of nondeterminism over $\mathcal{J}_r S$, which we extend to $\mathcal{J}S$:

Definition 9.1. For $j_1, j_2\colon \mathcal{J}S$, $\beta\colon S$, and $s\colon S$ we have

$$(j_1 \sqcap j_2).\beta.s \quad := \quad j_1.\beta.s \sqcap j_2.\beta.s\ .$$

Variations on Example (11) from Section 4 illustrate well the interaction of probabilistic choice and demonic nondeterminism. Suppose first that x is chosen nondeterministically (rather than probabilistically), and that we take postcondition $x = y$. Then

$$
\begin{array}{ll}
 & wp.(\,(x{:=}\ 1 \sqcap x{:=}\ 2);\ (y{:=}\ 1\ {}_{\frac{1}{2}}\!\oplus y{:=}\ 2)\,).(x=y) \\
= & wp.(x{:=}\ 1 \sqcap x{:=}\ 2).((x=1)/2 + (x=2)/2) \\
= & ((1=1)/2 + (1=2)/2) \sqcap ((2=1)/2 + (2=2)/2) \\
= & (1/2 + 0/2) \sqcap (0/2 + 1/2) \\
= & 1/2\ ,
\end{array}
$$

from which we see that program establishes $x = y$ with probability at least $1/2$: no matter which value is assigned to x, with probability $1/2$ the second command will assign the same to y.

Now suppose instead that it is the second choice that is nondeterministic. Then we have

$$
\begin{array}{ll}
 & wp.(\,(x{:=}\ 1\ {}_{\frac{1}{2}}\!\oplus x{:=}\ 2);\ (y{:=}\ 1 \sqcap y{:=}\ 2)\,).(x=y) \\
= & wp.(x{:=}\ 1\ {}_{\frac{1}{2}}\!\oplus x{:=}\ 2).((x=1) \sqcap (x=2)) \\
= & ((1=1) \sqcap (1=2))/2 + ((2=1) \sqcap (2=2))/2 \\
= & (1 \sqcap 0)/2 + (0 \sqcap 1)/2 \\
= & 0\ ,
\end{array}
$$

reflecting that no matter what value is assigned probabilistically to x, the demon could choose subsequently to assign a different value to y.

Thus it is clear that the execution order of occurrence of the two choices plays a critical role in their interaction, and in particular that the demon in the first case cannot make the assignment "clairvoyantly" to x in order to avoid the value that later will be assigned to y.

It is straightforward, with calculations like that preceding Definition 9.1, to establish the correspondence of the remaining relational operators with their predicate-transformer counterparts. Usually the latter are simpler.

10. EXAMPLE: TERMINATION AND RECURSION

Definition 8.1 suggests that a program $j: \mathcal{J}_r S$ should be said to terminate from initial state $s: S$ just when $j.\mathbf{1}.s = 1$; for then we have $1 \leq \int_F \mathbf{1}$ for the final distributions F that $rp.j$ delivers, and the discussion preceding Definition 2.2 characterizes those as terminating. By monotonicity and scaling (both properties of $\mathcal{J}_r S$), that is equivalent to the following definition:

Definition 10.1. (*Termination*). A program $j: \mathcal{J}_r S$ is said to *terminate* at an (initial) state $s: S$ if for all $\beta: \mathcal{P}S$ we have

$$\sqcap\beta \quad \leq \quad j.\beta.s \ .$$

We have already a similar characterization of feasibility: at a state s, Lemma 7.4 requires $j.\beta.s \leq \sqcup\beta$, giving a duality between the two notions just as in the standard case;[16] and thus j is both feasible and terminating (at all states) just when for all $\beta: \mathcal{P}S$

$$\sqcap\boldsymbol{\beta} \quad \leq \quad j.\beta \quad \leq \quad \sqcup\boldsymbol{\beta} \ .$$

We now consider an example of probabilistic termination, the recursive program

$$G \quad := \quad \mathbf{skip} \ {}_{\frac{1}{2}}\!\oplus G \tag{19}$$

that chooses repeatedly with probability $1/2$ between termination and (fruitless) recursion. From Figure 2, for any $\beta: \mathcal{P}S$, we have

$$wp.G.\beta \quad = \quad \beta/2 + (wp.G.\beta)/2 \ ,$$

whose unique solution is given by $wp.G.\beta = \beta$. Thus by Definition 10.1 we see that G terminates at every state; indeed, from Figure 2 we have that $G = \mathbf{skip}$.

The above reasoning goes through for any ${}_p\!\oplus$ when $0 < p$ and illustrates well how the familiar concept of "termination with probability 1" appears for us here: the program G has probability $(1-p)^n$ of recursing at least n times, and thus for nonzero p has probability 0 of recursing forever (as $n \to \infty$).

Treatment of recursions in general, however, relies on the existence of fixed points in $\mathcal{J}_r S$. Since we consider only feasible programs, it is easy to see that $\sqsubseteq$-directed sets of programs have least upper bounds, in spite of the fact that limits in $\mathcal{P}S$ do not necessarily exist: for any directed set $\mathcal{G}$ of programs, the limit is defined pointwise—thus for any $\beta: \mathcal{P}S$

$$(\sqcup\mathcal{G}).\beta \quad := \quad (\sqcup G: \mathcal{G} \cdot G.\beta) \ ,$$

and the limit exists on the right because feasibility gives $G.\beta \leq \sqcup\beta$ for all elements G of $\mathcal{G}$. (We have the existence of $\sqcup\beta$ itself from our assumption that S is finite.) Since sublinearity is preserved by taking limits, by continuity of the arithmetic operators, we have $\sqcup\mathcal{G} \in \mathcal{J}_r S$ provided $\mathcal{G} \subseteq \mathcal{J}_r S$.

[16] The standard duality is that termination is indicated by $true \Rightarrow j.true$, and feasibility by $j.false \Rightarrow false$.

Thus, with continuity, recursions are given meaning via the usual ω-limit construction. Consider for example the program over N:[17]

$$\begin{array}{l} n{:=}\ 0; \\ (\mu G \cdot \mathbf{skip} \ \ {}_{\frac{1}{2}}\!\oplus \ \ (n{:=}\ n+1; G)\,)\ . \end{array} \tag{20}$$

Beginning with the second statement, we calculate $wp.G.(n = N)$ by taking the limit $(\sqcup i{:}\, N \cdot \gamma_i)$, where

$$\begin{array}{rcll} \gamma_0 & = & \mathbf{abort}.(n = N) & = 0 \\ \gamma_{i+1} & = & (n = N)/2 + (\gamma_i[n{:=}\ n+1])/2\ . & \end{array}$$

Induction shows that $\gamma_i = \sum_{i'<i}(n + i' = N)/2^{i'+1}$, and thus in the limit we find that

$$wp.G.(n = N) \quad = \quad \sum_i (n + i = N)/2^{i+1}\ .$$

Hence for the whole program we have

$$\begin{array}{lll} & wp.(n{:=}\ 0; G).(n = N) & \\ = & wp.(n{:=}\ 0).(\sum_i (n + i = N)/2^{i+1}) & \text{above} \\ = & \sum_i (i = N)/2^{i+1} & \text{Figure 2 for } n{:=}\ 0 \\ = & 1/2^{N+1}\ , & \end{array}$$

showing that Program (20) sets n finally to N with probability at least $1/2^{N+1}$. That it terminates can be seen as in our earlier example; alternatively, sublinearity and continuity can be used to sum the probabilities for all N, giving (at least) 1 as a total.

Note that we have found an "unboundedly probabilistic" program—every one of its infinitely many final states is accessible, however small the associated probability might be. But that differs from unbounded (demonic) nondeterminism; the program (as a predicate transformer) remains continuous and is in fact deterministic, since it is maximal with respect to $\sqsubseteq$.

11. COMPARISONS AND CONCLUSIONS

The key to a systematic exploration of probabilistic predicate transformers seems to have been the connection between Kozen's probabilistic PDL [Kozen 1983] and the more recent work of He et al. [1996]. (Our contribution to the latter directly is the introduction of Cauchy closure to characterize continuity.) Dijkstra's presentation of standard healthiness conditions, and their dependence on the embedding of standard relations in the standard predicate transformers, then suggested the approach reported here.

Rao [1992; 1994] extends UNITY [Chandy and Misra 1988] with a statement that chooses between its components with some unknown but nonzero probability for each one. Restrictions on the execution of the program as a whole ("unconditional,"

[17] This example comes from Jones [1990], and since it contains no nondeterminism it can be treated by the model there—even over its infinite state space. Our treatment for infinite S [McIver and Morgan 1996] would be the same. We thus relax our assumption that S is finite in order to make our later point about unbounded nondeterminism.

and "extreme" fairness) then allow conclusions to be drawn about the eventual execution of each probablistic alternative, independent of the actual probabilities concerned; and from that an extension of the UNITY logic allows properties (for example, termination) to be proved to hold with probability 1.

In our case, given components j and j' the program

$$j \;{}_p\!\oplus\; j' \quad \sqcap \quad j' \;{}_{p'}\!\oplus\; j$$

executes j, j' with probability at least p, p' respectively, provided $p + p' \leq 1$; but Cauchy closure (equivalently continuity, or bounded nondeterminism) prevents our abstracting from the specific values of p, p': we cannot write a program such that they are arbitrarily small but nonzero.

Rao's nondeterminism (between statements) is less demonic than ours: with the unconditional-fairness assumption of UNITY (applied by analogy to an explicit recursion) the program

$$(\mu G \cdot \mathbf{skip} \sqcap G)$$

would terminate because the body cannot be executed indefinitely without selecting its first component **skip**. However under the standard interpretation [Dijkstra 1976], thus ours as well, that program is equivalent to **abort**.

To approach the elegance of Rao's termination proofs (for example, Rao [1994, Sec. 15]) we use the probabilistic analogue of invariants and variants: proof rules for those turn out to be simple generalizations of their standard counterparts, and the rules are themselves proved using sublinearity [Morgan 1995]. For termination with probability 1, in particular, the *0–1 Law* [Hart et al. 1983; Morgan 1995] allows us very easily to ignore explicit (but nonzero) probabilities if we wish.

More generally, termination arguments may come to be based on simple probabilistic systems such as "random walk" or "gambler's ruin" [Grimmett and Welsh 1986]. For example, an approach as for Program (19) shows that the program

$$\begin{array}{l} x \neq 0 \;\to \\ \quad x{:=}\; x+1 \;\;{}_{\frac{1}{2}}\!\oplus\;\; x{:=}\; x-1 \\ , \end{array}$$

for $x{:}\, Z$, terminates with probability 1. (That cannot be done in Rao [1994] because the explicit probability $1/2$ is essential in the argument.)

That "symmetric random walk" pattern, or others, could be found spread out and "well buried" in more general programs—with x represented by an expression over many program variables. It could then be extracted using probabilistic *data refinement* (an analogue of the techniques of Gardiner and Morgan [1991] perhaps), in that way maintaining full rigor in the argument if desired. Once the pattern was revealed, the termination argument could be made separately.

Other investigations of the link between nondeterminism and probability address the situation of probabilistic programs running concurrently under a nondeterministic scheduler [Hart et al. 1983; Lehmann and Rabin 1981; Rabin 1982], often in the framework of temporal logic. It is shown that the "extent" of nondeterminism in the scheduler is crucial: can it take advantage of probabilistic choices already made, or even yet to be made? Here, like Lehmann and Rabin [1981], our nondeterminism is demonic rather than indifferent (it takes advantage of earlier probabilistic

choices), but not clairvoyant (it cannot take advantage of probabilistic choices still in the future).

Further work on the theory of probabilistic predicate transformers is reported in McIver and Morgan [1996], treating in particular the extension to infinite state spaces, the arithmetic characterization of standard programs (semilinearity) and deterministic programs (simple linearity), and exploring the more general transformers that do not correspond to relational programs (only *semi*-sublinearity). Those last include the "nonconjunctive" programs containing "angelic" choice; and for them there is a decomposition result in the probabilistic space, extending the standard result [Back and von Wright 1990].

More practical work is reported in Morgan [1995], where the results here have been used to reconstruct—for probabilistic programs—the standard apparatus for reasoning about loops. We believe the examples there show the cost of including explicit probability, at least in small programs, to be commensurate with the extra information gained about their behavior—thus that rigor in probabilistic imperative programming incurs a relative penalty no higher than in the standard case.

Another introduction to probabilistic predicate transformers in practice is given in Seidel et al. [1996], where informal explanations for some features of the model are provided, and the further use of probabilistic postconditions to reason about expected time to termination (probabilistic efficiency) is illustrated by example.

APPENDIX

A. LINEAR PROGRAMMING LEMMAS

Both of these lemmas are well known in linear programming.

LEMMA A.1. (THE SEPARATING-HYPERPLANE LEMMA). *Let $\mathcal{C}$ be a convex and Cauchy-closed subset of R^N, and p a point in R^N that does not lie in $\mathcal{C}$. Then there is a separating hyperplane S with p on one side of it and all of $\mathcal{C}$ on the other.*

PROOF. See for example Trustrum [1971, p.8]. □

LEMMA A.2. (FARKAS' LEMMA). *Let A be an $M \times N$ matrix, x an $N \times 1$ column-vector and r an $M \times 1$ column-vector, and suppose that A and r are so that the system of equations*

$$A \cdot x \quad \geq \quad r \tag{21}$$

has no solution in x, where $\cdot$ denotes matrix multiplication. Then there is a $1 \times M$ row-vector C of nonnegative values such that

$$C \cdot A = 0 \quad \textit{but} \quad C \cdot r > 0 \ . \tag{22}$$

PROOF. See for example Schrijver [1986, p.89], taking the contrapositive of Corollary 7.1e there. □

Lemma A.2 can be motivated by considering its converse, also true but trivially so: if there is a C satisfying (22) then inequation (21) can have no solution—for if it did, we could reason

$$0 \quad < \quad C \cdot r \quad \leq \quad C \cdot A \cdot x \quad = \quad 0 \cdot x \quad = \quad 0 \ ,$$

a contradiction. Thus the lemma can be read "if (21) has no solution in x then there is a witness C to that fact."

The connection between probability and linear programming is reported in Fagin et al. [1990] also.

B. GLOSSARY OF NOTATION

$wp.h.\beta$	The weakest precondition of (relational) program h with respect to postcondition β.
$j \sqcap j'$	The demonic nondeterministic choice between programs j and j'.
$j \ {}_p\!\oplus\ j'$	The p-probabilistic choice between programs j and j'.
$lhs := rhs$	Define *lhs* to be *rhs*.
$S_\perp$	State space S with a bottom element $\perp$ adjoined.
$f.x$	Function f applied to argument x.
$\overline{S}$	The set of distributions over state space S.
$\sqsubseteq$	The refinement order (over various domains).
$(\forall s{:}\, S \cdot body)$	"For all s in S we have *body*."
$(\lambda s{:}\, S \cdot expr)$	The function over S yielding *expr*, in which s denotes the argument.
	The constant function yielding *expr* everywhere.
$\mathcal{D}S$	The deterministic probabilistic programs over state space S.
$\overline{s}$	The point distribution at state s.
$f \circ g$	The functional composition of f and g.
$\int_D \alpha$	The expected value of random variable α over distribution D; the integral of function α over measure D.
$d^\dagger$	The distribution-to-distribution form of deterministic probabilistic program d.
χ_T	The characteristic function of set T.
$\beta[x := E]$	Syntactic substitution of E for free x in β; the equivalent operation on β regarded as a function of the state.
$\mathcal{H}S$	The relational nondeterministic programs.
$\{s{:}\, S \cdot expr\}$	The set of values *expr* formed as s ranges over S.
$\{s{:}\, S \mid range\}$	The set of values s in S that satisfy *range*.
$\{s{:}\, S \mid range \cdot expr\}$	The set of values *expr* formed as s ranges over those elements of S that satisfy *range*.
$R_\geq$	The nonnegative reals.
PS	The set of all subsets of S.
$\mathcal{P}S$	The probabilistic predicates over S.
$\mathcal{J}S$	The probabilistic predicate transformers over S.
$\mathcal{J}_r S$	The regular probabilististic predicate transformers over S.
$[S \leftrightarrow S_\perp]$	The total and image-finite relations between S and $S_\perp$.
$\sqcap$	Binary minimum; infinitary infimum.
$c\beta$	The pointwise multiplication of predicate β by scalar c.
$\ominus$	Natural number (truncated) subtraction.
$\sqcup$	Binary maximum; infinitary supremum.
$(\sqcap F{:}\, \mathcal{F} \cdot expr)$	The infimum of values *expr* as F ranges over $\mathcal{F}$.
$(\sqcup \beta{:}\, B \cdot expr)$	The supremum of values *expr* as β ranges over B.
$\cdot$	Matrix multiplication.

ACKNOWLEDGEMENTS

As we carried out most of the work reported here, Morgan was enjoying the hospitality of the Computer Science Department at Utrecht University.

We are grateful for the comments and contributions of Jifeng He, Tony Hoare, and the participants in the probabilistic predicate transformer seminars at the SVRC and Department of Computer Science of the University of Queensland: Mark Bofinger, David Carrington, Ian Hayes, Ray Nickson, and Mark Utting.

We thank the referees for their careful reading and detailed recommendations.

REFERENCES

Back, R.-J. R. and von Wright, J. 1990. Duality in specification languages: A lattice theoretical approach. *Acta Inf. 27*, 583–625.

Chandy, K. M. and Misra, J. 1988. *Parallel Program Design: A Foundation*. Addison-Wesley, Reading, Mass.

Dijkstra, E. W. 1976. *A Discipline of Programming*. Prentice Hall International, Englewood Cliffs, N.J.

Fagin, R., Halpern, J. Y., and Megiddo, N. 1990. A logic for reasoning about probabilities. *Inf. Comput. 87*, 78–128.

Gardiner, P. H. B. and Morgan, C. C. 1991. Data refinement of predicate transformers. *Theor. Comput. Sci. 87*, 143–162. Reprinted in *On the Refinement Calculus*, C.C. Morgan and T.N. Vickers, Eds. Springer-Verlag, Berlin, 1994.

Grimmett, G. and Welsh, D. 1986. *Probability: An Introduction*. Oxford Science Publications. Oxford University Press, Oxford.

Hart, S., Sharir, M., and Pnueli, A. 1983. Termination of probabilistic concurrent programs. *ACM Trans. Program. Lang. Syst. 5*, 356–380.

He, J., McIver, A. K., and Seidel, K. 1996. Probabilistic models for the guarded command language. *Sci. Comput. Program.* To appear in FMTA '95 special issue. Also available via *http* at http://www.comlab.ox.ac.uk/oucl/groups/probs/bibliography.html.

Hesselink, W. H. 1992. *Programs, Recursion and Unbounded Choice*. Cambridge Tracts in Theoretical Computer Science, vol. 27. Cambridge University Press, Cambridge, U.K.

Hoare, C. A. R. 1969. An axiomatic basis for computer programming. *Commun. ACM 12,* 10 (Oct.), 576–580, 583.

Jones, C. 1990. Probabilistic nondeterminism. Ph.D. thesis, Monograph ECS-LFCS-90-105, Edinburgh Univ. Edinburgh, U.K.

Jones, C. and Plotkin, G. 1989. A probabilistic powerdomain of evaluations. In *Proceedings of the IEEE 4th Annual Symposium on Logic in Computer Science*. IEEE Computer Society Press, Los Alamitos, Calif., 186–195.

Kozen, D. 1981. Semantics of probabilistic programs. *J. Comput. Syst. Sci. 22*, 328–350.

Kozen, D. 1983. A probabilistic PDL. In *Proceedings of the 15th ACM Symposium on Theory of Computing*. ACM, New York.

Lehmann, D. and Rabin, M. O. 1981. On the advantages of free choice: A symmetric and fully-distributed solution to the Dining Philosophers Problem. In *Proceedings of the 8th Annual ACM Symposium on Principles of Programming Languages*. ACM, New York, 133–138.

Lowe, G. 1993. Representing nondeterministic and probabilistic behaviour in reactive processes. Programming Research Group, Oxford.

McIver, A. K. and Morgan, C. C. 1996. Probabilistic predicate transformers: Part 2. Tech. Rep. PRG-TR-5-96, Programming Research Group, Oxford. Also available via *http* at http://www.comlab.ox.ac.uk/oucl/groups/probs/bibliography.html.

Morgan, C. C. 1988. The specification statement. *ACM Trans. Program. Lang. Syst. 10,* 3 (July). Reprinted in *On the Refinement Calculus*, C.C. Morgan and T.N. Vickers, Eds. Springer-Verlag, Berlin, 1994.

MORGAN, C. C. 1995. Proof rules for probabilistic loops. Tech. Rep. PRG-TR-25-95, Programming Research Group, Oxford. To appear in *Proceedings of the 7th BCS FACS Refinement Workshop* (July 1996), Springer-Verlag. Also available via *http* at http://www.comlab.ox.ac.uk/oucl/groups/probs/bibliography.html.

MORGAN, C. C., MCIVER, A. K., SEIDEL, K., AND SANDERS, J. W. 1994. Refinement-oriented probability for CSP. Tech. Rep. PRG-TR-12-94, Programming Research Group, Oxford. To appear in *Formal Aspects of Computing*. Also available via *http* at http://www.comlab.ox.ac.uk/oucl/groups/probs/bibliography.html.

MORRIS, J. M. 1987. A theoretical basis for stepwise refinement and the programming calculus. *Sci. Comput. Program. 9,* 3 (Dec.), 287–306.

NELSON, G. 1989. A generalization of Dijkstra's calculus. *ACM Trans. Program. Lang. Syst. 11,* 4 (Oct.), 517–561.

RABIN, M. O. 1982. The choice-coordination problem. *Acta Inf. 17,* 2 (June), 121–134.

RAO, J. R. 1992. Building on the UNITY experience: Compositionality, fairness and probability in parallelism. Ph.D. thesis, Univ. of Texas at Austin, Austin, Tex.

RAO, J. R. 1994. Reasoning about probabilistic parallel programs. *ACM Trans. Program. Lang. Syst. 16,* 3 (May).

SCHRIJVER, A. 1986. *Theory of Integer and Linear Programming*. Wiley, New York.

SEIDEL, K., MORGAN, C. C., AND MCIVER, A. K. 1996. An introduction to probabilistic predicate transformers. Tech. Rep. PRG-TR-6-96, Programming Research Group, Oxford. Also available via *http* at http://www.comlab.ox.ac.uk/oucl/groups/probs/bibliography.html.

TRUSTRUM, K. 1971. *Linear Programming*. Library of Mathematics. Routledge and Kegan Paul, London.

Received February 1995; revised February 1996; accepted March 1996