Programming Research Group

PARTIAL CORRECTNESS FOR PROBABILISTIC DEMONIC PROGRAMS

Annabelle McIver and Carroll Morgan

PRG-TR-35-97

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD

Partial correctness
for probabilistic demonic programs

Annabelle McIver and Carroll Morgan*

Draft of 12 November 1997

Abstract

Recent work in sequential program semantics has produced both an operational [4] and an axiomatic [13, 17] treatment of total correctness for probabilistic demonic programs, extending Kozen's original work [9, 10] that did not include demonic nondeterminism.

For practical applications however (*eg.* combining loop invariants with termination constraints) it is important to retain the traditional distinction between partial and total correctness. Jones [6] defines probabilistic partial correctness for probabilistic, but again not demonic programs.

In this paper we combine all the above, giving an operational and axiomatic framework for both partial and total correctness of probabilistic and demonic sequential programs; among other things that provides the theory to support practical reasoning about probabilistic demonic loops [11].

1 Introduction

Deterministic computation over a state space S can be modelled as functions of type $S \to S$, from initial to final states. A 'powerdomain' construction extends that to nondeterminism, and although the traditional powerdomains — Smyth, Hoare and Plotkin — differ in their treatment of non-termination, they all agree that nondeterminism is 'demonic', resolved in some arbitrary way.

The *probabilistic* powerdomain [7, 6] instead resolves nondeterminism according to some specified distribution over final states: demonic choice is removed, and replaced by probabilistic choice.

*Both authors are at the Programming Research Group, Wolfson Building, Parks Road, Oxford OX1 3QD, UK; {*anabel,carroll*}@*comlab.ox.ac.uk*.

He *et al.* [4] do not remove demonic choice; rather they model demonic and probabilistic nondeterminism jointly by combining a special case of the probabilistic powerdomain, for imperative programs [9], with the Smyth (demonic) construction. Morgan *et al.* [13] then complement that with a programming logic of 'greatest pre-expectations' (extending Kozen's work [10]), resulting overall in a treatment of total correctness for probabilistic and demonic sequential programs.

In this paper we extend the constructions of He and Morgan to partial correctness also. One important application for both forms of correctness is the justification of invariant/variant principles for probabilistic demonic loops: although published [11], those principles are based on (only) postulated connections between *wp* and *wlp* for probabilistic programs — here we provide the theory for their proof.

Another application is in the abstraction from probabilistic to demonic choice; we discuss both issues in the conclusion.

To model partial (Hoare-style) and total (Smyth-style) correctness within the same framework, we choose the Egli-Milner construction for demonic choice (rather than the Smyth); and its being based on distributions rather than simple states introduces some novel considerations (in particular the need for linear interpolation between distributions). For the logic we are then able to formulate both greatest- and greatest liberal pre-expectations, reflecting the same total/partial distinction as in standard programming logic [2].

In Sec. 2 we construct the probabilistic Plotkin-style powerdomain, the 'convex' powerdomain (which is a subset of the general representation in Abramsky and Jung [1]), and we show how to extract partial and total information from it; in Sec. 3 we link that to greatest-precondition-based probabilistic programming logic; and in Sec. 4 and Sec. 5 we specialise to both a liberal and 'ordinary' logic for a sequential programming language with both probabilistic and demonic nondeterminism.

An extensive discussion of examples, and the general treatment of loops, is given elsewhere [11].

Throughout we use infix dot '.' for function application, associating to the left so that $f.x.y$ means $(f(x))(y)$; and we write ':=' for 'is defined to be equal to'. Quantifications (including set comprehensions) are written in the order *quantifier* (or '{' for sets), *bound variable* with optional type, *range* and finally *term* — thus for example

$$\{n{:}\,\mathbb{N} \mid 0 \leq n < 10 \cdot n^2\}$$

is the set of the first ten squares.

2 A convex powerdomain of distributions

In program semantics, powerdomains are used to study nondeterminism, a phenomenon arising when a program outputs an arbitrary result (drawn from a set of possible results) rather than a single, determined function of its input. Here we consider powerdomains over a domain of probability distributions rather than of 'definite' single states.

There are several ways of ordering sets of distributions (each resulting in a different powerdomain): the choice depends on criteria which can be explained in terms of the desired treatment of programs' possible non-terminating behaviour. The Smyth order[1] (Def. B2) treats non-termination as the worst behaviour and thus the Smyth powerdomain models total correctness. Similarly the Hoare order (Def. B3) models partial correctness: non-termination is treated as the best outcome in that order. The Plotkin powerdomain uses the Egli-Milner order (Def. B4) which combines both views, and is useful when both partial and total correctness are to be modelled within a single framework.

In general the Plotkin powerdomain is not decomposable into the Smyth and Hoare powerdomains, but in some special cases it is: Abramsky and Jung [1] show that one such case is when the underlying domain is ω-continuous (Def. B9). In this section we show how Abramsky and Jung's results apply to an appropriately-defined powerdomain of distributions, thus providing a single semantic space for both partial and total correctness of programs exhibiting probabilistic and demonic nondeterminism.

We write S for the state space, and assume it is finite. The space of (discrete) probability distributions over S is defined as follows.

Definition 2.1 For state space S, the space of *distributions*[2] $(\overline{S}, \sqsubseteq)$ over S is defined

$$\overline{S} \quad := \quad \{F\colon S \to [0,1] \mid \sum_{s:S} F.s \leq 1\} \ ,$$

and for F, F' in $\overline{S}$ we define

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

□

[1]For this and other facts and definitions from domain theory, we follow the conventions set out in [1]. We summarise the details for this paper (often specialising them to our particular application) in Appendix B.

[2]These special distributions are more precisely called *discrete sub-probability measures* [9]; they do not necessarily sum to 1, and the deficit gives the probability of nontermination. The 'everywhere zero' distribution for example, that assigns zero probability to all states, models nowhere-terminating behaviour. (An alternative though less convenient treatment would assign probability 1 to some special state $\perp$.)

We show first that $(\overline{S}, \sqsubseteq)$ is an ω-continuous complete partial order.

Lemma 2.2 For S a finite state space, its distributions $(\overline{S}, \sqsubseteq)$ form an ω-continuous complete partial order.
Proof: The completeness of $(\overline{S}, \sqsubseteq)$ is trivial, given the completeness of the interval $[0,1]$ under $\leq$ over the reals.

To show that $\overline{S}$ is ω-continuous we only need exhibit a countable basis (Def. B8). Since S is finite, we use the set of distributions contained in

$$S \to ([0,1] \cap \mathbb{Q}) \ ,$$

since any real is the least upper bound of the rationals way-below it (Def. B7). $\Box$

We now define a Plotkin-style powerdomain over $\overline{S}$. For subset A of $\overline{S}$ we write $\uparrow\! A$ for its up-closure and $\downarrow\! A$ for its down-closure (Def. B1). We say that A is *up-closed* (*Smyth-closed*) if $\uparrow\! A = A$, that it is *down-closed* (*Hoare-closed*) if $\downarrow\! A = A$ and that it is *Egli-Milner-closed*[3] if $A = \uparrow\! A \cap \downarrow\! A$.

A further closure condition is related to continuity.

Definition 2.3 For subset A of $\overline{S}$ we define its *limit closure* $lc.A$ to be the smallest set containing A itself, together with $\sqcup A'$ and $\sqcap A''$ for all up-directed (Def. B5) subsets A' and down-directed (Def. B6) subsets A'' of A.[4] We say that A is *limit-closed* if $lc.A = A$. $\Box$

Before defining our powerdomain we must introduce one further closure condition, specific to distributions. For distributions F, F' in $\overline{S}$ and p in $[0,1]$ we can form $F\,{}_p\!\oplus F'$, the weighted average, defined pointwise over S as $p \times F + (1-p) \times F'$ (with usual scalar multiplication and addition). For sets of distributions we define p-averaging as follows.

Definition 2.4 For p in $[0,1]$ and subsets A, A' of $\overline{S}$ we define

$$A\,{}_p\!\oplus A' \quad := \quad \{F\colon A, F'\colon A' \cdot F\,{}_p\!\oplus F'\} \ .$$

We say that A is *convex* if $A\,{}_p\!\oplus A \quad = \quad A$ for all p in $[0,1]$. $\Box$

We now can define our powerdomain over $\overline{S}$; it is a subset of the Plotkin powerdomain.

Definition 2.5 The *convex powerdomain* $(\mathcal{C}S, \sqsubseteq_{EM})$ over the space of distributions $\overline{S}$ comprises those subsets of $\overline{S}$ that are non-empty, Egli-Milner closed, limit closed and convex. Its order $\sqsubseteq_{EM}$ is the usual Egli-Milner order (Def. B4). $\Box$

[3] This is often called *convex closed*; but we will need that term for another purpose.
[4] Note that both $\sqcup A$ and $\sqcap A$ exist for all subsets A of $\overline{S}$ in particular, directed or not.

The convex powerdomain is a subset of the Plotkin powerdomain because it includes only the convex subsets of the latter. Our aim for this section is to show that $\mathcal{C}S$ is itself limit complete, and that from its limits the Smyth and Hoare limits can be extracted — for that is what makes it suitable for our application of it to probabilistic program semantics. Thus we show that the $\sqsubseteq_{EM}$- least upper bound of an Egli-Milner-directed subset of $\mathcal{C}S$ lies within $\mathcal{C}S$, and that it is a combination of the Hoare least upper bound ($\sqcup_H$) and the Smyth least upper bound ($\sqcup_S$).

The next lemma is the specialisation of a general decomposition result of Plotkin powerdomains to $(\overline{S}, \sqsubseteq)$. We write $Lens(\overline{S})$ (Def. B10) for the set of non-empty, Egli-Milner closed, limit closed subsets of $\overline{S}$.[5]

Lemma 2.6 For any Egli-Milner-directed subset $\mathcal{A}$ of $Lens(\overline{S})$ the limit $\sqcup_{EM}\mathcal{A}$ exists, and satisfies

$$\sqcup_{EM}\mathcal{A} \quad = \quad \sqcup_S\, \mathcal{A} \;\cap\; lc.(\sqcup_H \mathcal{A})\ . \tag{1}$$

(Insisting on limit closure after $\sqcup_H$ can be seen here as a continuity[6] condition.)

Proof: The decomposition will be a consequence of the isomorphim between the abstract Plotkin powerdomain (Def. B12) and the space of lenses of an ω-continuous domain with the topological Egli-Milner ordering (Def. B13). The isomorphism is given by Abramsky and Jung [1] and is reproduced here in Thm. B.1. The decomposition (1) holds for abstract Plotkin powerdomains in general [8].

To establish the isomorphism, note first that the closure conditions on $Lens(\overline{S})$ imply that $\sqsubseteq_{EM}$ between lenses reduces to the topological Egli-Milner ordering; then Lem. 2.2 ensures the conditions of Thm. B.1, namely that $(\overline{S}, \sqsubseteq)$ is an ω-continuous complete partial order. □

Since $\mathcal{C}S$ is a subset of $Lens(\overline{S})$, we have in the following corollary our closure under limits.

Corollary 2.7 For any Egli-Milner-directed subset $\mathcal{A}$ of $\mathcal{C}S$ the limit $\sqcup_{EM}$ exists in $\mathcal{C}S$, and satisfies

$$\sqcup_{EM}\mathcal{A} \quad = \quad \sqcup_S\, \mathcal{A} \;\cap\; lc.(\sqcup_H \mathcal{A})\ .$$

Proof: Given Lem. 2.6, we need only show that $\sqcup_S\mathcal{A} \;\cap\; lc.(\sqcup_H\mathcal{A})$ is convex if all the elements of $\mathcal{A}$ are, and that follows from these elementary facts: up-closing preserves convexity ($\sqsubseteq$-monotonicity of ${}_p\oplus$); the intersection of convex sets

[5] Thus $\mathcal{C}S$ comprises just the convex lenses of $\overline{S}$.

[6] Consider the Egli-Milner chain on sets of real intervals in $[0, 1]$,

$$\{0\} \sqsubseteq_{EM} \{1/2\} \sqsubseteq_{EM} \cdots \sqsubseteq_{EM} \{(n-1)/n\} \sqsubseteq_{EM} \cdots\ ,$$

which has limit $\{1\}$ (the limit point of the underlying series). The union of the down sets is the half-closed interval $[0, 1)$ however, and the intersection of the up sets is $\{1\}$. Failing to limit-close the Hoare limit would produce an empty result.

is convex; down-closing preserves convexity (similar to up-closing); the union of a $\subseteq$-directed set of convex sets is convex; and limit-closing preserves convexity ($\sqsubseteq$-continuity of ${}_p\oplus$). $\Box$

Cor. 2.7 gives us our main result, that the Egli-Milner limit determines the Smyth limit (in the Smyth ordering) and the Hoare limit (in the Hoare ordering). We write $\equiv_S$, $\equiv_H$ respectively (Def. B14) for Smyth equivalence and Hoare equivalence between elements of $\mathcal{C}S$: our theorem below shows in addition that the limits are indistinguishable relative to the appropriate equivalences.

Theorem 2.8 For any $\sqsubseteq_{EM}$-directed subset $\mathcal{A}$ of $\mathcal{C}S$, the following equivalences hold:

$$\begin{array}{lll} \sqcup_{EM}\mathcal{A} & \equiv_S & \sqcup_S \mathcal{A} \\ \sqcup_{EM}\mathcal{A} & \equiv_H & lc.(\sqcup_H \mathcal{A}) . \end{array}$$

Proof: This too is a property of abstract Plotkin powerdomains [8], and so follows from the isomorphism (Thm. B.1) used in the proof of Thm. 2.7. $\Box$

This section has defined the convex powerdomain, whose use for modelling probabilistic imperative programs now follows from the constructions for the Smyth-style domain [4]: for example sequential composition is a generalised functional composition; nondeterministic choice is union (then convex closure); and probabilistic choice is weighted average as defined above. In Sec. 4 we give further details.

For recursion one takes limits of chains, and here is the significance of Thm. 2.8: we must be sure that taking the limit in the convex domain agrees with the more specialised limit in the Smyth domain *and* with the more specialised limit in the Hoare domain — for that is what allows us to use the more general convex domain for either. It is known that the equivalence holds for standard (nonprobabilistic) domains[7]; Thm. 2.8 confirms the preservation of the property when probability is included.

In the next section we link the convex powerdomain with its partial/total probabilistic programming logic.

3 Probabilistic programs and logic

In this section we use the convex powerdomain $\mathcal{C}S$ of Sec. 2 to construct a model for both total and partial correctness of probabilistic demonic programs.[8] We begin

[7] In fact the same general argument of this section establishes that, since the standard approach takes a flat domain S for the state space, trivially ω-continuous if S is finite, or even countable.

[8] Kozen [9] has modelled total correctness of probabilistic (but not demonic) programs, and Jones [6] extended that to partial correctness. He [4] models total (but not partial) correctness of probabilistic demonic programs.

with a review of methods that treat the two aspects separately.

Over standard (non-probabilistic) demonic programs, a popular model for total correctness is $S \to \mathcal{S}S_\perp$, where $S_\perp$ is the flat domain extending state space S with $\perp$ for non-termination, and $\mathcal{S}$ forms the Smyth powerdomain over that; Dijkstra's weakest 'ordinary' preconditions $\mathbb{P}S \to \mathbb{P}S$ [2] support a programming logic suitable for total correctness. For partial correctness one can use $S \to \mathcal{H}S_\perp$ (Hoare) for the model and weakest 'liberal' preconditions for the logic. Finally, although partial and total correctness are available simultaneously via $S \to \mathcal{K}S_\perp$ (Plotkin), for r in $S \to \mathcal{K}S_\perp$ and postcondition Q in $\mathbb{P}S$ still it is more convenient to define separately

$$\begin{array}{lcll} \mathit{wp}.r.Q & := & \{s \mid r.s \subseteq Q\} & \text{weakest precondition} \\ \mathit{wlp}.r.Q & := & \{s \mid r.s \subseteq Q_\perp\} & \text{weakest liberal precondition} \end{array} \tag{2}$$

to give the total (wp) and partial (wlp) programming logics. Note that the definitions (2) work together only over $\mathcal{K}S$ (the intersection of $\mathcal{H}S_\perp$ and $\mathcal{S}S_\perp$) — wp does not work over $\mathcal{H}S_\perp$ and wlp does not work over $\mathcal{S}S_\perp$. (Nelson [14] gives a nice treatment of the issues.)

For probabilistic programs, He *et al.* [4] propose $S \to \mathcal{C}_S S$ for total correctness, where $\mathcal{C}_S S$ is like $\mathcal{C}S$ of the previous section, but based on the Smyth order. Morgan *et al.* [13] provide a probabilistic 'greatest pre-expectation' logic for that, where *expectations* are non-negative real-valued functions over the state space (extending Kozen's treatment [10] for non-demonic programs).

To access total and partial correctness simultaneously, by analogy with the standard case we simply replace He's Smyth-based $\mathcal{C}_S$ by our Egli-Milner-based $\mathcal{C}$. Yet rather than define two forms of logic (as at (2) above) we generalise as do Morgan and McIver [12] by allowing expectations to take *negative* values: roughly speaking, for total correctness one uses non-negative post-expectations and for partial correctness one uses non-positive.

We begin the details with the construction of the probabilistic, demonic model of programs.

Definition 3.1 For (finite) state space S the space of *probabilistic, demonic programs* $(\mathcal{M}S, \sqsubseteq_{EM})$ is given by

$$\mathcal{M}S \quad := \quad S \to \mathcal{C}S \ ,$$

with the order induced pointwise from $\mathcal{C}S$, so that for r, r' in $\mathcal{M}S$ we have

$$r \sqsubseteq_{EM} r' \quad := \quad (\forall s\colon S \cdot r.s \sqsubseteq_{EM} r'.s) \ .$$

We occasionally use $\sqsubseteq_S$ and $\sqsubseteq_H$ over $\mathcal{M}S$, analogously lifted from $\mathcal{C}S$. $\square$

Thus our programs take initial states to sets of final distributions: the plurality of the sets represents *demonic* nondeterminism; the distributions they contain each represent *probabilistic* nondeterminism.

The next task is to investigate the dual representation of programs as expectation transformers. We extend the expectations found in Morgan *et al.* [13, 11], where the topic was total correctness (the Smyth order and up-closed sets) and expectations were of type $S \to [0,1]$, by using $[-1,1]$ instead: we write $\mathcal{E}S$ for $S \to [-1,1]$, and use lower-case Greek letters for typical elements.

Expectation transformers $\mathcal{T}S$ are thus functions of type $\mathcal{E}S \to \mathcal{E}S$. We write $\int_F \alpha$ for the expected value of α in $\mathcal{E}S$ averaged over distribution F in $\overline{S}$. As a special case of expectations, we interpret predicates as $\{0,1\}$-valued functions of the state space, and for predicate A holding at state s we write either $s \in A$ or $A.s = 1$ as convenient. For a scalar c we write $\underline{c}$ for the constant expectation evaluating to c over all of S. With those conventions the predicates *true* and *false* correspond to the expectations $\underline{1}$ and $\underline{0}$ respectively. Finally, for relations between expectations we write

$\Rrightarrow$ — everywhere no more than
$\equiv$ — everywhere equal to
$\Lleftarrow$ — everywhere no less than ,

so that we generalise respectively implication, equivalence and reverse implication on predicates.

Our logic is based on the 'extended greatest pre-expectation transformer', defined as follows.

Definition 3.2 Let r be a program in $\mathcal{M}S$, taking initial states in S to sets of final distributions over S. Then the *greatest* pre-expectation at state s of program r, with respect to post-expectation α in $\mathcal{E}S$, is defined:[9]

$$ewp.r.\alpha.s \quad := \quad (\sqcap F\colon r.s \cdot \int_F \alpha) .$$

□

The effect of the definition is to consider all possible post-distributions F in $r.s$, and then demonically to choose the one that gives the least (the 'worst') expectation for α: thus nondeterminism is demonic in that it minimises the pre-expectation at each initial state, and Def. 3.2 is then the greatest expectation everywhere no more than those pointwise minima.

[9]This reduces to Kozen's definition [9] for deterministic programs. There programs are *functions* from initial state to distributions, so that the minimisation ranges over a singleton set and is thus superfluous.

For standard programs, if executing a program r from a state s is *certain* to establish a postcondition A then that state is contained in the associated weakest precondition; with our definition we would have $ewp.r.A.s = 1$. For probabilistic programs, if the standard postcondition A is established with only a probability at least p say, then the greatest pre-expectation on executing r from s initially is at least p and we have $ewp.r.A.s = p$.[10] Thus as a special case, when A is a predicate we can interpret $ewp.r.A.s$ as the greatest assured probability that A holds after execution of r from s.

Now we discover the various refinement orders over $\mathcal{T}S$ that correspond via *ewp* with orders over $\mathcal{M}S$. First, we generalise the observation from standard programming (*eg.* [14]) that the Smyth order on programs corresponds to the implication order lifted to predicate transformers and that the Hoare order similarly corresponds to (lifted) reverse implication. We use $\mathcal{P}S$ (typical element π) to denote the set of non-negative valued expectations and $\mathcal{N}S$ (typical element ν) for the non-positive valued expectations. They are both subsets of $\mathcal{E}S$.

Lemma 3.3 For r, r' in $\mathcal{M}S$,

$$\begin{array}{lll} r \sqsubseteq_S r' & \text{implies} & (\forall \pi \colon \mathcal{P}S \cdot ewp.r.\pi \Rrightarrow ewp.r'.\pi) \\ r \sqsubseteq_H r' & \text{implies} & (\forall \nu \colon \mathcal{N}S \cdot ewp.r.\nu \Lleftarrow ewp.r'.\nu) \ . \end{array}$$

Proof: For r, r' in $\mathcal{M}S$, any s in S and π in $\mathcal{P}S$ we reason as follows

$$\begin{array}{lll} & r \sqsubseteq_S r' & \\ \text{implies} & \uparrow(r.s) \supseteq \uparrow(r'.s) & \text{definition } \sqsubseteq_S \\ \text{implies} & (\sqcap F \colon \uparrow(r.s) \cdot \int_F \pi) \ \le \ (\sqcap F \colon \uparrow(r'.s) \cdot \int_F \pi) & \\ \text{iff} & (\sqcap F \colon r.s \cdot \int_F \pi) \ \le \ (\sqcap F \colon r'.s \cdot \int_F \pi) & \underline{0} \Rrightarrow \pi\text{; see below} \\ \text{iff} & ewp.r.\pi.s \le ewp.r'.\pi.s \ . & \text{Def. 3.2} \end{array}$$

For the deferred justification we appeal to the monotonicity of the arithmetic over non-negative arguments without subtraction: $r.s$ differs from $\uparrow(r.s)$ only by the addition of 'larger elements' according to Def. 2.1, and so the minimum selection made in Def. 3.2 cannot be *increased* by making the selection over the up-closure instead.

The result now follows by generalising on s, and a similar argument justifies the second statement (but note the reversal $\Lleftarrow$). □

Lem. 3.3 is the key to defining the expectation-transformer equivalents to the Smyth, Hoare and Egli-Milner orders where, as usual, the Egli-Milner order is the intersection of the Smyth and Hoare orders.

[10] The apparent confusion between expectations and probabilities is deliberate and harmless: the probability of an event A over a distribution is equal to the expected value of (the characteristic function of) A over that same distribution.

Definition 3.4 For t, t' in $\mathcal{T}S$ we define

$$\begin{array}{lcl} t \sqsubseteq_S t' & := & (\forall \pi{:}\, \mathcal{P}S \cdot t.\pi \Rrightarrow t'.\pi)\ , \\ t \sqsubseteq_H t' & := & (\forall \nu{:}\, \mathcal{N}S \cdot t.\nu \Lleftarrow t'.\nu)\ , \\ t \sqsubseteq_{EM} t' & := & t \sqsubseteq_S t' \ \wedge\ t \sqsubseteq_H t'\ . \end{array}$$

□

That the Egli-Milner order between programs is preserved under *ewp* now follows directly.

Corollary 3.5 For r, r' in $\mathcal{M}S$,

$$r \sqsubseteq_{EM} r' \quad \text{implies} \quad ewp.r \sqsubseteq_{EM} ewp.r'\ .$$

Proof: Lem. 3.3 and Def. 3.4. □

The corollary shows only that *ewp* is an order-preserving mapping between $(\mathcal{M}S, \sqsubseteq_{EM})$ and $(\mathcal{T}S, \sqsubseteq_{EM})$. The next result of this section is to show that it is also an injection, and therefore that programs can be modelled equivalently either as relations or as expectation transformers. To establish that *ewp* is an embedding we prove the converse to Cor. 3.5.

Lemma 3.6 For r, r' in $\mathcal{M}S$, if $ewp.r \sqsubseteq_{EM} ewp.r'$ then $r \sqsubseteq_{EM} r'$.
Proof: Suppose for a contradiction that $ewp.r \sqsubseteq_{EM} ewp.r'$ but $r \not\sqsubseteq_{EM} r'$, for some r, r' in $\mathcal{M}S$. Without loss of generality assume $r \not\sqsubseteq_S r'$, so that for some distribution F and state s we have both

$$F \ \notin\ \uparrow(r.s) \tag{3}$$

$$\text{and}\quad F \ \in\ \uparrow(r'.s)\ . \tag{4}$$

From (3), with the aid of the separating hyperplane lemma A.1, we have for some expectation π in $\mathcal{P}S$ that

$$\int_F \pi \quad < \quad (\sqcap F'{:}\uparrow(r.s) \cdot \int_{F'} \pi)\ ,$$

and thus that $\int_F \pi\ <\ ewp.r.\pi.s$. From (4) however we have $ewp.r'.\pi.s\ \leq\ \int_F \pi$ directly, giving together

$$ewp.r'.\pi.s \quad \leq \quad \int_F \pi \quad < \quad ewp.r.\pi.s$$

and contradicting the hypothesis (at the state s). □

Since in the proof of Lem. 3.6 we have actually proved the converse to Lem. 3.3, we can now state the correspondence between the relational model and program logic for all three orders.

Theorem 3.7 The following equivalences hold for all r, r' in $\mathcal{M}S$:

$$\begin{array}{lll} r \sqsubseteq_S r' & \text{iff} & ewp.r \sqsubseteq_S ewp.r' \ , \\ r \sqsubseteq_H r' & \text{iff} & ewp.r \sqsubseteq_H ewp.r' \ , \\ r \sqsubseteq_{EM} r' & \text{iff} & ewp.r \sqsubseteq_{EM} ewp.r' \ . \end{array}$$

□

We have shown that *ewp* injects $\mathcal{M}S$ into $\mathcal{T}S$ — but there are many $\sqsubseteq_{EM}$-monotonic expectation transformers that are not *ewp*-images of $\mathcal{M}S$. The final result of this section identifies the exact sub-space of expectation transformers that form the image of $\mathcal{M}S$ through *ewp*. We identify 'healthiness conditions' over $\mathcal{T}S$[11] in the style of Dijkstra [2] — for standard programs — and of Morgan *et al.* [13] — for probabilistic programs — that identify the (images of) programs of $\mathcal{M}S$ within it. The importance of that result is that theorems proved within $\mathcal{T}S$ about healthy expectation transformers correspond to theorems about programs in $\mathcal{M}S$.

The first healthiness condition is a slight generalisation of the sublinearity of Morgan [13]. To state it we define, for expectations α, β in $\mathcal{E}S$ and real non-negative scalar c, the expectations $\alpha + \beta$ and $c\alpha$, where (as for p-averaging of distributions) we mean a pointwise lifting of standard addition and scalar multiplication.

Definition 3.8 An expectation transformer, $t\colon \mathcal{T}S$ is *sublinear* iff for all α, β in $\mathcal{E}S$, and a, b, c non-negative reals,

$$t.(a\alpha + b\beta - \underline{c}) \quad \Leftarrow \quad a(t.\alpha) + b(t.\beta) - \underline{c} \ .$$

□

We note first that sublinearity is satisfied by all images of $\mathcal{M}S$ under *ewp*.

Lemma 3.9 Any expectation transformer *ewp.r*, for r in $\mathcal{M}S$, is sublinear.
Proof: Def. 3.2 and properties of arithmetic. (Morgan [13] gives a more detailed proof.) □

For total correctness (for the Smyth $\mathcal{C}_S$), sublinearity tells the whole story [13, Thm. 8.7]; in our more general $\mathcal{C}S$ however, there are sublinear elements of $\mathcal{M}S$

[11] The relevance of the healthiness conditions applied to the programming logic are treated in Sec. 5.

that are not *ewp*-images: take S to be the two-element state space $\{x, y\}$, and consider the result set

$$\{F: \overline{S} \mid F.x = F.y\} \ .$$

It is convex, but not Egli-Milner closed[12]; its associated expectation transformer formed by *ewp* is sublinear, but it is not the *ewp*-image of any element of $\mathcal{M}S$.

The characterisation of Egli-Milner closure is captured by a second healthiness condition — 'partial linearity' — which states that $t.\alpha$ depends only on the pre-expectations of t applied to expectations in $\mathcal{P}S \cup \mathcal{N}S$.

Definition 3.10 An expectation transformer, t in $\mathcal{T}S$ is said to be *partially linear* if for all states s in S, and all expectations α in $\mathcal{E}S$, there are expectations π in $\mathcal{P}S$ and ν in $\mathcal{N}S$ such that $\alpha = \pi + \nu$ and

$$t.\alpha.s \quad = \quad t.\pi.s + t.\nu.s \ .$$

□

Note that the implicit existential quantification in Def. 3.10 means there may be many decompositions of α as a sum $\pi + \nu$.[13]

We complete the correspondence between healthy expectation transformers and $\mathcal{M}S$ with the next theorem, which we state only. The proof is omitted as it is overly technical and not necessary for the rest of the paper.

Theorem 3.11 An expectation transformer t in $\mathcal{T}S$ is both sublinear and partially linear if and only if there is r in $\mathcal{M}S$ such that $t = ewp.r$. □

4 Probabilistic Guarded Commands

In this section we illustrate the constructions of the previous two sections by giving equivalent relational ($\mathcal{M}S$) and expectation transformer ($\mathcal{T}S$) semantics for a simple sequential programming language that extends Dijkstra's guarded commands [2] with a binary probabilistic choice operator ${}_p\oplus$.[14]

[12] In fact its closure is $\{F: \overline{S} \mid F.x, F.y \leq \frac{1}{2}\}$, from which it is indistinguishable using *ewp* for any α in $\mathcal{P}S \cup \mathcal{N}S$.

[13] A perhaps more alluring healthiness condition would be that $t.\alpha$ is determined by its positive part $(\alpha \sqcup \underline{0})$ and its negative part $(\alpha \sqcap \underline{0})$; but

$$t.\alpha \quad = \quad t.(\alpha \sqcup \underline{0}) + t.(\alpha \sqcap \underline{0}) \ ,$$

does not hold for general probabilistic programs, although it does in the restricted set of standard programs and $\{0, 1, -1\}$-valued expectations [12].

[14] He *et al.* [4] and Morgan *et al.* [13] give similar presentations of semantics, but for total rather than partial correctness, and more detailed motivation can be found there. Note that demonic choice is retained, not replaced.

$$
\begin{array}{lcl}
\textbf{abort}.s & := & \{\overline{\bot}\} \\
\textbf{skip}.s & := & \{\overline{s}\} \\
(\textbf{assign } f).s & := & \{\overline{f.s}\} \quad \text{for function } f \text{ in } S \to S \\
(r \; {}_p\!\oplus r').s & := & r.s \; {}_p\!\oplus r'.s \\
(r \sqcap r').s & := & (\cup p{:}\, [0,1] \cdot r.s \; {}_p\!\oplus r'.s) \\
(r; r').s & := & \{F{:}\, r.s; f'{:}\, S \to \overline{S} \mid r' \ni f' \cdot \int_F f'\} \\
(r \textbf{ if } B \textbf{ else } r').s & := & r.s \text{ if } B.s \text{ else } r'.s \\
\textbf{rec } \mathcal{B} & := & \mu\mathcal{B} \quad \text{for } \sqsubseteq_{EM}\text{-monotonic } \mathcal{B} \text{ in } \mathcal{M}S \to \mathcal{M}S
\end{array}
$$

For ${}_p\!\oplus$, $\sqcap$ and sequential composition, the Egli-Milner closure should be taken of the right-hand side.

Figure 1: Probabilistic relational semantics

The relational semantics of Fig. 1 writes $\overline{s}$ for the point distribution[15] concentrated on a single state s, and $\sqcap$ for the demonic combination of programs. The symbol ${}_p\!\oplus$ is used both for probabilistic combination of programs and for the $p-$averaging explained in Sec. 2 between sets of distributions.

Because they contain no demonic nondeterminism, the three primitive commands all yield singleton result sets. Probabilistic choice ${}_p\!\oplus$ returns the weighted average of the result sets of its operands, a singleton set if its arguments are.[16] The result set of the demonic nondeterministic choice between two programs is the convex closure of the union of the results of the operands — the closure models operationally that a demon could resolve the choice by flipping a coin of arbitrary bias.

In sequential composition, both r and r' can be considered as sets of purely probabilistic programs 'compatible' with them: such programs, say f, f', are thus of type $S \to \overline{S}$, and we write $r \ni f$ for compatibility, meaning $(\forall s{:}\, S \cdot f.s \in r.s)$. Since the effect of two purely probabilistic programs in sequence is just a single distribution over s'' say, defined

$$
(f; f').s.s'' \quad := \quad \int_{f.s} f'.s'.s'' \, \mathrm{d}s' \tag{5}
$$

by 'averaging' the effects of f' over the intermediate distribution produced by $f.s$,

[15]For state s in S we define the point distribution $\overline{s}{:}\, S \to [0,1]$ to be

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

As a special case, we write $\overline{\bot}$ for $(\lambda s.0)$, the distribution that evaluates to 0 everywhere on S.

[16]In fact p can depend on the state; but for simplicity we assume here that it's constant.

$$
\begin{array}{lll}
ewp.\mathbf{abort}.\sigma & := & \underline{0} \\
ewp.\mathbf{skip}.\sigma & := & \sigma \\
ewp.(\mathbf{assign} f).\sigma.s & := & \sigma.(f.s) \quad \text{for function } f \text{ in } S \to S \\
ewp.(r\ {}_p\!\oplus r').\sigma & := & (r.\sigma)\ {}_p\!\oplus (r'.\sigma) \\
ewp.(r \sqcap r').\sigma & := & ewp.r.\sigma \sqcap ewp.r'.\sigma \\
ewp.(r; r').\sigma & := & ewp.r.(ewp.r'.\sigma) \\
ewp.(r \ \mathbf{if}\ B\ \mathbf{else}\ r').\sigma.s & := & ewp.r.\sigma.s \text{ if } B.s \text{ else } ewp.r'.\sigma.s \\
ewp.(\mathbf{rec}\ \mathcal{B}) & := & \mu\mathcal{F} \quad \text{where } \mathcal{F} \text{ is the } \sqsubseteq_{\mathcal{T}}\text{-monotonic} \\
 & & \text{function such that } \mathcal{F}.(ewp.r) = ewp.(\mathcal{B}.r)
\end{array}
$$

Figure 2: Probabilistic *ewp* semantics, where σ is in $\mathcal{P}S \cup \mathcal{N}S$ and s is in S.

for the general case r, r' we vary (5) over all f, f' to give

$$(r; r').s \quad = \quad \{f, f' \mid r \ni f \wedge r' \ni f' \cdot (f; f').s\} \ .$$

That simplifies to the definition in Fig. 1 if we regard f' in $\int_F f'$ as parametrised by s'', abbreviating $(\lambda s'' \cdot \int_F f'.s'.s'' \mathrm{d}s')$.

In the *ewp*-semantics of Fig. 2 the post-expectation σ varies only over $\mathcal{P}S \cup \mathcal{N}S$;[17] we use $\sqcap$ also for the pointwise minimum between expectations.

Here probabilistic choice ${}_p\!\oplus$ returns the weighted average of the results of its expectation operands; and $\sqcap$ takes the pointwise minimum, reflecting the demon's striving for the worst (least) result. Sequential composition becomes simple functional composition (as for standard predicate transformers [2]).

In both cases recursion is dealt with by least fixed points in the appropriate orders: Thm. 3.7 shows that the two orders correspond.

Our concern in the next section will be to recover total and partial semantics separately from Fig. 2: we will show how to define probabilistic *wp* and *wlp* that generalise Dijkstra's operational interpretations and that satisfy probabilistic versions of the standard laws for combining partial and total correctness.

5 Partial and total correctness

Athough *ewp* acts over all of $\mathcal{E}S$, we can extract two more-specialised logics from it: each acts conventionally over just the non-negative expectations $\mathcal{P}S$.

For a total correctness logic we merely restrict to $\mathcal{P}S$ directly, and use the order $\Rrightarrow$.

[17] Because an Egli-Milner-closed set is determined by its Smyth- and Hoare closures, that is sufficient.

Definition 5.1 Let r be a program in $\mathcal{M}S$; then the *greatest* pre-expectation of program r with respect to post-expectation π in $\mathcal{P}S$, associating 0 with non-termination, is defined:

$$wp.r.\pi \quad := \quad ewp.r.\pi \ .$$

$\square$

Well-definedness follows easily from sublinearity: if π is in $\mathcal{P}S$ then

$$\underline{0} \quad \Rrightarrow \quad ewp.r.\pi \quad \Rrightarrow \quad \underline{1} \ ,$$

so that $wp.r.\pi$ is in $\mathcal{P}S$ also. Moreover Lem. 3.3 shows that this wp semantics of programs corresponds to a relational model with the Smyth ordering [13] — non-termination is the worst outcome in both semantics.

For partial correctness we define a probabilistic wlp, again we restrict to the subspace $(\mathcal{P}S, \Rrightarrow)$.

Definition 5.2 [18] Let r be a program in $\mathcal{M}S$; then the greatest *liberal* pre-expectation of program r with respect to the post-expectation π in $\mathcal{P}S$, associating 1 with non-termination, is

$$wlp.r.\pi \quad := \quad \underline{1} + ewp.r.(\pi - \underline{1}) \ .$$

$\square$

Again it follows easily from sublinearity of $ewp.r$ that for ν in $\mathcal{N}S$,

$$\underline{-1} \quad \Rrightarrow \quad ewp.r.\nu \quad \Rrightarrow \quad \underline{0} \ ,$$

and thus since $\pi - \underline{1}$ lies in $\mathcal{N}S$, so does $ewp.r.(\pi - \underline{1})$ from which we deduce that $wlp.r$ is a well-defined expectation transformer in $\mathcal{P}S \to \mathcal{P}S$. Also Lem. 3.3 implies that the wlp semantics corresponds to a relational model with the Hoare ordering — accordingly non-termination is the best outcome.

For example, let S be some finite portion of $\mathbb{N}$, and for natural number N write $s := N$ for the assignment taking every initial state to the final state N. The program

$$(s := 0 \;{}_p\!\oplus\; s := 1) \;{}_q\!\oplus\; \mathbf{abort}$$

illustrates the difference between wp and wlp. Writing $[s = N]$ for the expectation that evaluates to 1 when s is N and to 0 otherwise, we have

[18] Via Def. 3.2 we can readily show Def. 5.2 to be identical to

$$wlp.r.\pi.s \quad := \quad (\sqcap F\!: r.s \cdot 1 - \int_F (\underline{1} - \pi)) \ , \tag{6}$$

which is a demonic generalisation of the probabilistic wlp defined only for *nondemonic* programs by Jones [6]. Morgan [12] shows that (6) also generalises standard wlp [2].

$$
\begin{array}{lll}
wlp.\mathbf{abort}.\pi & := & \underline{1} \\
\ldots & & \\
wlp.(\mathbf{rec}\ \mathcal{B}) & := & \nu\mathcal{F} \quad \text{where } \mathcal{F} \text{ is the } \sqsubseteq_{\mathcal{T}}\text{-monotonic} \\
 & & \text{function such that } \mathcal{F}.(ewp.r) = ewp.(\mathcal{B}.r)
\end{array}
$$

Figure 3: Probabilistic *wlp* semantics, where π is in $\mathcal{P}S$ and s is in S. The greatest fixed point ν is used for recursion.

$$
\begin{array}{lll}
 & wp.((s := 0\ {}_p\oplus\ s := 1)\ {}_q\oplus\ \mathbf{abort}).[s = 0] & \\
\equiv & \quad wp.(s := 0).[s = 0]\ \ {}_p\oplus\ \ wp.(s := 1).[s = 0] & \\
 & {}_q\oplus\ \ wp.\mathbf{abort}.[s = 0] & \\
\equiv & (\underline{1}\ {}_p\oplus\ \underline{0})\ {}_q\oplus\ \underline{0} & [s = 0].0 = 1\ \mathit{etc.} \\
\equiv & \underline{pq}\ ,
\end{array}
$$

indicating that the greatest expectation of termination in state 0 is pq, for all initial states.

The greatest expectation of either termination at 0 or nontermination is found with *wlp*; we have

$$
\begin{array}{ll}
 & wlp.((s := 0\ {}_p\oplus\ s := 1)\ {}_q\oplus\ \mathbf{abort}).[s = 0] \\
\equiv & (\underline{1}\ {}_p\oplus\ \underline{0})\ {}_q\oplus\ \underline{1} \\
\equiv & \underline{pq + 1 - q}\ .
\end{array}
$$

Thus the *wp* observation gives the greatest guaranteed probability[19] of termination at 0 — and non-termination guarantees nothing. The *wlp* observation on the other hand returns the probability that either 0 is reached or the program fails to terminate — the usual interpretation for partial correctness.

Similarly the *wlp* semantics of a looping program is calculated as the greatest fixed point[20] of a monotonic function over $\mathcal{P}S$ (rather than the least, as for *wp*). It is easily checked that specialising Fig. 2 to *wlp* produces only the changes shown in Fig. 3. (Note that $\mathcal{P}S \cup \mathcal{N}S$ is closed under $\pm\underline{1}$.)

An alternative view of *wlp* and *wp* often put forward is to regard them as functions on programs which satisfy certain properties — for example Nelson [14]

[19] This follows from the usual interpretation of the expected value of a $\{0, 1\}$-valued random variable A with respect to a probability distribution: it is the chance that the random outcome will establish the event "A evaluates to 1".

[20] This follows from Def. 5.2 since the least fixed point of a $\sqsubseteq_{\mathcal{T}}$-monotonic function becomes specialised first to $\Lleftarrow$ on $\mathcal{N}S \to \mathcal{N}S$ which order is then shifted to $\mathcal{P}S \to \mathcal{P}S$ by applying "$\underline{1}+$".

defines (standard) *wp* and *wlp* axiomatically by enforcing both the *coupling law* (at (7) below) and conjunctivity properties (both *wp* and *wlp* induce conjunctive predicate transformers). Thus the efficacy of the probabilistic definitions lies in the generalisation of those properties: Lem. 5.3 and Thm. 5.4 (also below) generalise respectively conjunctivity and coupling (7), and as such they form the main results of this section.

We state the coupling here for standard programs [5]:

$$wlp.r.A \wedge wp.r.true \quad \Rrightarrow \quad wp.r.A \ , \tag{7}$$

where r is a program and A a predicate and we are using (though only here) the original meanings for *wp* and *wlp* [2], with $\Rrightarrow$ for 'implies at all states'.

Law (7) implies that *wp.r* and *wlp.r* agree on initial states from which termination is guaranteed, and thus it underlies the practical treatment of looping programs — to prove total correctness of an iteration the work is divided between ensuring partial correctness (with a loop invariant), and an independent termination argument (with a variant). The probabilistic coupling Thm. 5.4 will allow a similar treatment for probabilistic looping programs.

We consider first the appropriate generalisation of conjunctivity: conjunction of predicates is replaced by *probabilistic conjunction* [17] defined for non-negative expectations π, π'

$$\pi \,\&\, \pi' \quad := \quad (\pi + \pi' - \underline{1}) \sqcup \underline{0} \ , \tag{8}$$

where $\sqcup$ is pointwise maximum between expectations.

Probabilistic conjunction reduces to ordinary conjunction when specialised to predicates.[21] Its importance in probabilistic reasoning is that it subdistributes through both *wp* and *wlp* images of programs — another consequence of sublinearity.[22]

Lemma 5.3 For r in $\mathcal{M}S$ and $\pi,\ \pi'$ in $\mathcal{P}S$,

$$\begin{array}{lcl} wp.r.(\pi \,\&\, \pi') & \Lleftarrow & wp.r.\pi \,\&\, wp.r.\pi' \\ wlp.r.(\pi \,\&\, \pi') & \Lleftarrow & wlp.r.\pi \,\&\, wlp.r.\pi' \ . \end{array}$$

Proof: Sublinearity (with $a = b = c = 1$) and monotonicity of *ewp.r*, and Defs. 5.1, 5.2. □

[21]If $\pi,\ \pi'$ in $\mathcal{P}S$ take only the extreme values 0 and 1, then

$$\begin{array}{lcl} (\pi \,\&\, \pi').s & = & 1 \text{ , if } \pi.s = 1 \text{ and } \pi'.s = 1, \\ & = & 0 \text{ , otherwise .} \end{array}$$

[22]One might have guessed that $\sqcap$ is the appropriate generalisation of $\wedge$ — but $\sqcap$ does not (even sub-) distribute [6, 17].

Next we deal with coupling — Thm. 5.4, generalising (7), is the main result of this section.

Theorem 5.4 For r in $\mathcal{M}S$ and $\pi,\ \pi'$ in $\mathcal{P}S$,

$$wp.r.\pi \mathbin{\&} wlp.r.\pi' \quad \Rrightarrow \quad wp.r.(\pi \mathbin{\&} \pi') \ .$$

Proof:

$$\begin{array}{lll}
 & wp.r.(\pi \mathbin{\&} \pi') & \\
\equiv & ewp.r.(\pi \mathbin{\&} \pi') & \text{Def. 5.1} \\
\equiv & ewp.r.((\pi + \pi' - \underline{1}) \sqcup \underline{0}) & (8) \\
\Lleftarrow & ewp.r.(\pi + \pi' - \underline{1}) & \text{monotonicity} \\
\Lleftarrow & ewp.r.\pi \ + \ ewp.r.(\pi' - \underline{1}) & \text{sublinearity} \\
\equiv & ewp.r.\pi \ + \ \underline{1} \ + \ ewp.r.(\pi' - \underline{1}) \ - \ \underline{1} & \text{arithmetic} \\
\equiv & wp.r.\pi \ + \ wlp.r.\pi' \ - \ \underline{1} \ . & \text{Def. 5.2; Def. 5.1}
\end{array}$$

Having established $wp.r.(\pi \mathbin{\&} \pi') \Lleftarrow wp.r.\pi + wlp.r.\pi' - \underline{1}$, we conclude by taking $\sqcup\underline{0}$ on both sides: since on the left it has no effect, we achieve our result. □

As a corollary we recover the standard rule (generalised) for combining partial and total correctness.

Corollary 5.5 For r in $\mathcal{M}S$ and π in $\mathcal{P}S$,

$$wlp.r.\pi \mathbin{\&} wp.r.\underline{1} \quad \Rrightarrow \quad wp.r.\pi \ .$$

Proof: Thm. 5.4 and that $\pi \ \equiv \ \pi \mathbin{\&} \underline{1}$ for π in $\mathcal{P}S$. □

As a special case note that the *wlp* result implies the *wp* result at those states from which termination occurs with probability 1 — where $wp.r.\underline{1} \equiv \underline{1}$ — because $(\&\underline{1})$ is the identity.

6 Conclusion

The technical contribution of this paper is to extend the Plotkin construction, with its Egli-Milner order, to act over probability distributions rather than points, then showing that the decomposition [15] into respectively the convex Hoare (partial correcness) and convex Smyth (total correctness) powerdomains continues to hold — even under taking limits.

A key feature is the imposition of convexity (linear interpolation) between distributions.

Jones [6] defines a partial correctness logic based on expectations, but only for non-demonic programs, and she does not discuss the healthiness conditions on which the applicability of such logics (as calculational tools) depends. It was the realisation [13] that adding nondeterminism to Kozen's model corresponds to a weakening of the additive property of his logic to sublinearity that makes proofs such as in Thm. 5.4 and those in [11] reduce to simple arithmetic arguments. The use of general expectations (thus superseding purely non-negative expectations [13]) leads to an even simpler presentation of sublinearity — the more useful of the two healthiness conditions described here.

There are two immediate applications. The first is the discovery of proof rules for loops in which partial and total correctness are separated: with *wp* and *wlp* together it can be shown [11] that

$$I \sqcap G \quad \Rrightarrow \quad wp.body.I \qquad \text{probabilistic invariant preserved by loop body}$$

is sufficient for $I \Rrightarrow wp.(\mathbf{do}\ G \rightarrow body\ \mathbf{od}).(I \sqcap \neg G)$ provided $I \Rrightarrow T$, where T gives for each initial state the probability of the loop's termination.

The second application is abstraction. For some programs only termination is probabilistic, and (partial) correctness can be established informally by considering all probabilistic choices to be demonic: algorithms falling in to that category typically use randomization as a method for searching a large space of potential witnesses, and examples include finding perfect hash functions and finding irreducible polynomials (Mehlhorn, Rabin [3]).

Suppose the loop $\mathbf{do}\ G \rightarrow body\ \mathbf{od}$ contains probabilistic choices in *body*, and denote by $\underline{body}$ the result of replacing them all with demonic choice. If (the fairness of) probability was essential for the loop's termination, the resulting standard loop

$$\mathbf{do}\ G \rightarrow \underline{body}\ \mathbf{od}\ , \tag{9}$$

though probability-free, would not terminate and so a *wp* analysis of it would be useless.

With a theory of partial correctness however we can reason as follows: for a standard loop invariant I,

$$\begin{array}{lll} & G \sqcap I \Rrightarrow wlp.\underline{body}.I & \text{established by standard reasoning say} \\ \text{hence} & I \Rrightarrow wlp.\underline{loop}.(I \sqcap \neg G) & \text{standard loop rule} \\ \text{hence} & I \Rrightarrow wlp.loop.(I \sqcap \neg G)\ , & \text{see below} \end{array}$$

where $\underline{loop}$ and *loop* are the two loops containing $\underline{body}$ and *body* respectively. Thus from *standard* reasoning about $\underline{body}$ we reach *probabilistic* conlusions about *loop*.

The last step relies only on $\underline{loop} \sqsubseteq loop$, which by *wlp*-monotonicity of $\mathbf{do} \cdots \mathbf{od}$ follows from $\underline{body} \sqsubseteq body$; that in turn is guaranteed by $(\sqcap) \sqsubseteq ({}_p\oplus)$ for all p. Note that probabilistic *wlp* is essential — the last step would not hold if we had written *wp.body*.

One would conclude the argument by determining $T \equiv wp.loop.\underline{1}$ separately, then having

$$I \,\&\, T \quad \Rrightarrow \quad wp.loop.(I \sqcap \neg G)$$

from Thm. 5.4.

A Standard results from linear programming

Lemma A.1 *The separating hyperplane lemma.* Let C be a convex and (limit-) closed subset of $\mathbb{R}^N$, and F a point in $\mathbb{R}^N$ that does not lie in C. Then there is a separating hyperplane α with F on one side of it and and all of C on the other.
Proof: See for example Trustrum [18]. □

Our use in Lem. 3.6 is based on interpreting distribution F as a point in $\mathbb{R}^N$, and an expectation α as the collective normal of a family of parallel hyperplanes. The integral $\int_F \alpha$ then gives the constant term for the α-hyperplane that passes through the point F.

More generally, for a convex region C in $\mathbb{R}^N$, the minimum

$$(\sqcap F{:}\, C \cdot \int_F \alpha)$$

gives the constant term for the α-hyperplane that touches C, with its normal pointing into C.

Thus when specialised for the applications in this paper, the lemma implies that if $F \notin C$ for some closed convex set of distributions C, then there is an expectation α with

$$\int_F \alpha \quad < \quad (\sqcap F'{:}\, C \cdot \int_{F'} \alpha) \; ;$$

Moreover if C is up-closed (down-closed) then the range of α above specialises to $\mathcal{P}S$ ($\mathcal{N}S$).

B Facts and definitions from domain theory

We summarise here Abramsky and Jung's presentation [1] of facts from domain theory, giving page numbers where appropriate. Assume $(D, \sqsubseteq)$ is a complete partial order — *i.e.* that every directed (defined below) set has a least upper bound.

1. **up-, down-closure** For subset A of D we define its *up-closure* $\uparrow A$ to be the set
$$\{d: D \mid (\exists a: A \cdot a \sqsubseteq d)\} \ .$$
Similarly we define its *down-closure* $\downarrow A$ to be the set
$$\{d: D \mid (\exists a: A \cdot a \sqsupseteq d)\} \ .$$

2. **Smyth order** (p.97) The *Smyth order* $\sqsubseteq_S$ on $\mathbb{P}D$, for subsets A, A' of D, is given by
$$A \sqsubseteq_S A' \quad \text{iff} \quad A' \subseteq \uparrow A \ .$$

3. **Hoare order** (p.97) The *Hoare order* $\sqsubseteq_H$ on $\mathbb{P}D$, for subsets A, A' of D, is given by
$$A \sqsubseteq_H A' \quad \text{iff} \quad A \subseteq \downarrow A' \ .$$

4. **Egli-Milner order** The Egli-Milner order $\sqsubseteq_{EM}$ between subsets combines the Smyth (Def. B2) and Hoare (Def. B3) order. For subsets A, A' of D we define
$$A \sqsubseteq_{EM} A' \quad \text{iff} \quad \uparrow A \supseteq A' \text{ and } A \subseteq \downarrow A' \ .$$

5. **up-directed** (Def. 2.1.8, p.10) A subset A of D is *up-directed* (or simply *directed*) iff for any u, v in A there is a w also in A such that $u \sqsubseteq w$ and $v \sqsubseteq w$. We write $\sqcup A$ for the least upper bound of A (if it exists).

6. **down-directed** A subset A of D is *down-directed* iff for any u, v in A there is a w also in A such that $w \sqsubseteq u$ and $w \sqsubseteq v$. We write $\sqcap A$ for the greatest lower bound of A (if it exists).

7. **way-below** (Def. 2.2.1, p.15) The *way-below* relation $\ll$ on D is defined as follows: for u, v in D we say $u \ll v$ iff for all up-directed subsets A of D with $v \sqsubseteq \sqcup A$ there is some w in A with $u \sqsubseteq w$. We also say u *approximates* v iff $u \ll v$.

8. **basis** (Def. 2.2.3, p.16) A *basis* for D is a subset B such that every element of D is the $\sqcup$-limit of the elements way below it in B.

9. ω-**continuity** (Def. 2.2.6, p.17) D is ω-*continuous* if it has a countable basis.

10. **lens** (Def. 6.2.15, p.100) We define the lenses of D, $\mathit{Lens}(D)$, to be the set of non-empty, Egli-Milner closed, limit-closed subsets of D.

11. **ideal** (p.10) A subset I is an *ideal* if it is directed and down-closed.

12. **Plotkin powerdomain** (Theorem 6.2.3, p.95) The *Plotkin powerdomain* of D with basis B is given by the ideal (Def. B11) completion of

$$(\mathcal{F}(B), \ll_{EM}) ,$$

where $\mathcal{F}(B)$ denotes the set of finite, nonempty subsets of B.

13. **topological Egli-Milner order** (Def. 6.2.16, p.101) Define the *topological Egli-Milner order* $\sqsubseteq_{TEM}$ on the set of Egli-Milner closed subsets, $\mathit{Lens}(D)$, of D as follows:

$$X \sqsubseteq_{TEM} Y \quad \text{iff} \quad Y \subseteq \uparrow X \ \wedge \ X \subseteq \mathit{lc.}(\downarrow Y) .$$

14. **Smyth-, Hoare-equivalence** Two subsets A, A' of D are *Smyth equivalent* if $\uparrow A = \uparrow A'$; they are *Hoare-equivalent* if $\downarrow A = \downarrow A'$.

Theorem B.1 (Theorem 6.2.19, p.101) If D is an ω-continuous complete partial order, its Plotkin powerdomain is isomorphic to $(\mathit{Lens}(D), \sqsubseteq_{TEM})$. □

References

[1] S. Abramsky and A. Jung. Domain theory. In Dov M. Gabbay S. Abramsky and T.S. Maibaum, editors, *Handbook of Logic and Computer Science*, volume 3, pages 1–168. Oxford Science Publications, 1994.

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

[3] S.A. Smolka G. Gupta and S. Bhaskar. On randomization in sequential and distributed systems. *ACM Computing Surveys*, 26(1):7–86, March 1994.

[4] Jifeng He, K.Seidel, and A. K. McIver. Probabilistic models for the guarded command language. *Science of Computer Programming*, 28(2,3):171–192, January 1997.

[5] Wim H Hesselink. *Programs, Recursion and Unbounded Choice.* Number 27 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, U.K., 1992.

[6] C. Jones. Probabilistic nondeterminism. Monograph ECS-LFCS-90-105, Edinburgh Univ. Edinburgh, U.K., 1990. (PhD Thesis).

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

[8] A. Jung. Private communication. 1997.

[9] D. Kozen. Semantics of probabilistic programs. *Journal of Computer and System Sciences*, 22:328–350, 1981.

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

[11] C. C. Morgan. Proof rules for probabilistic loops. In He Jifeng, John Cooke, and Peter Wallis, editors, *Proceedings of the BCS-FACS 7th Refinement Workshop*, Workshops in Computing. Springer Verlag, July 1996. *http://www.springer.co.uk/ewic/workshops/7RW*.

[12] C. C. Morgan and A. K. McIver. Unifying *wp* and *wlp*. *Information Processing Letters*, 20(3):159–164, 1996. Also available via *http* [16].

[13] C. C. Morgan, A. K. McIver, and K. Seidel. Probabilistic predicate transformers. *ACM Transactions on Programming Languages and Systems*, 18(3):325–353, May 1996.

[14] G. Nelson. A generalization of Dijkstra's calculus. *ACM Transactions on Programming Languages and Systems*, 11(4):517–561, October 1989.

[15] Gordon Plotkin. Post-graduate lecture notes in advanced domain theory (incorporating the "Pisa Notes"). Dept. of Computer Science, Univ. of Edinburgh, 1981.

[16] PSG. Probabilistic Systems Group: Collected reports. *http://www.comlab.ox.ac.uk/oucl/groups/probs/bibliography.html*.

[17] K. Seidel, C. C. Morgan, and A. K. McIver. An introduction to probabilistic predicate transformers. Technical Report PRG-TR-6-96, Programming Research Group, February 1996. Also available via *http* [16].

[18] K. Trustrum. *Linear Programming*. Library of Mathematics. Routledge and Kegan Paul, London, 1971.