Formal Methods for Probabilistic Systems

Annabelle McIver
Carroll Morgan

- Source-level program logic
- Meta-theorems for loops
- Examples
- Relational operational model
  - Standard, deterministic, terminating
  - Standard, deterministic, non-terminating
  - Standard, demonic, non-terminating
  - Standard powerdomains
  - Probabilistic powerdomains
  - Demonic, probabilistic powerdomains
  - Examples

Standard, deterministic, terminating relational semantics

A program \( f \) is a function of type state-to-state.

Function \( f \) applied to initial state \( s \) gives final state \( s' \).

\[ f : S \rightarrow S \]

\[ f.s = s' \]

This is function application, that is \( f(s) \).

Standard, deterministic, terminating relational semantics

No probabilistic choice.

Output predictable from input.

No infinite loops or "run-time errors".

"Relates" initial states to final states, and so gives an "operational" view.

Initial states

\[ s : = (s - 1) \mod 4 \]

Syntax

Final states

Function \( f \), where \( f.s = (s - 1) \mod 4 \).

of type \( \{0..3\} \rightarrow \{0..3\} \).
A program $f$ is now a function of type state to state-or-bottom.

$$f : S \rightarrow S \perp$$

Function $f$ applied to initial state $s$ gives final state $s'$... or the special nontermination state $\perp$.

$$f \cdot s = s' \quad \text{if } f \text{ terminates, from } s, \text{ at } s'$$

$$= \perp \quad \text{otherwise}$$

We suppose it is a "run-time error" to attempt to set $s$ to $-1$.

A program $r$ is now a relation of type state to state-or-bottom.

$$r : S \leftrightarrow S \perp$$

Usually $r$ is total
image-finite
and up-closed.

"Miracles" are excluded.
Continuity is required.
If $r$ can fail to terminate, then all (other) behaviours are deemed possible as well.

$$[r \cdot s \cdot s'] \text{ holds just when } r \text{ can reach } s' \text{ from } s.$$
Standard relational semantics

Initial states | Final states
---|---
0 | 0
1 | 1
2 | 2
3 | 3

$s: = s \pm 1$

Remember that it is a "run-time error" to attempt to set $s$ outside its type $\{0..3\}$.

Standard relational semantics

Initial states | Final states
---|---
0 | 0
1 | 1
2 | 2
3 | 3

$s: = s \pm 1$

A program $r$ is a relation of type state to state-or-bottom... or equivalently a set-valued function.
The significance of up-closure

A flat domain, based on the integers; for example \( \bot \subsetneq 3 \).

a proper outcome

nontermination

Informally, we regard 3 as a “better” outcome than \( \bot \). And we regard a program \( f_2 \) that delivers consistently better results than some other program \( f_1 \) as a “better” program overall:

\[
\text{If } (\forall s \cdot f_1 s \subseteq f_2 s) \text{ then we say } f_1 \sqsubseteq f_2.
\]

Program \( f_1 \) is refined by \( f_2 \) if some of \( f_1 \)'s nontermination is replaced by proper outcomes in \( f_2 \).

This in effect “promotes” the order \( \sqsubseteq \) from an order on individual values to an order on functions resulting in those values.

For nondeterminism we seek a similar promotion, but this time to the sets of values that represent the demonic choice.

The significance of up-closure: equivalence classes

The Smyth pre-order.

For subsets \( S_1, S_2 \) of the state space \( S \), we say that \( S_1 \sqsubseteq S_2 \) iff

\[
(\forall s_2 \mid s_2 \in S_2 \cdot (\exists s_1 \mid s_1 \in S_1 \cdot s_1 \sqsubseteq s_2)).
\]

Notice that both \( \{ \bot, 2 \} \subsetneq \{ \bot, 3 \} \) and \( \{ \bot, 3 \} \subsetneq \{ \bot, 2 \} \). This means that \( \subseteq \) is not a partial order — rather it is a pre-order, satisfying reflexivity and transitivity but not anti-symmetry. We say that \( \{ \bot, 2 \} \cong \{ \bot, 3 \} \), that they are equivalent.

This is a nuisance, but a well known one with a well known solution: we form equivalence classes, and take a distinguished representative from each class. That representative happens to be the up-closure — that is, for subset \( S_1 \) of \( S \), the set

\[
S_1^+ \text{ defined } \{ s \mid s \in S \cdot (\exists s_1 \mid s_1 \in S_1 \cdot s \sqsubseteq s_1) \}
\]

Not only does this have the property \( S_1 \cong S_2 \) iff \( S_1^+ = S_2^+ \), as we would expect from the equivalence-class-representative construction, but we have also that

\[
S_1 \sqsubseteq S_2 \text{ iff } S_1^+ \supseteq S_2^+.
\]

The significance of up-closure: powerdomains

Given two relational programs \( r_1 \) and \( r_2 \) we say that \( r_1 \sqsubseteq r_2 \) iff, for all initial states \( s \), any outcome in the set \( r_1 s \) can be justified by some outcome in the set \( r_2 s \), that is if every behaviour of the implementation \( r_2 s \) is justified by the specification \( r_1 s \):

\[
\text{for all } s \text{ and } s_2 \subseteq r_2 s \text{ there is an } s_1 \subseteq r_1 s \text{ such that } s_1 \sqsubseteq s_2.
\]

That is, \( r_2 s \succeq_s \) holds, or equivalently \( (s, s_2) \in r_2 \).

This is known as the Smyth order.

For subsets \( S_1, S_2 \) of the state space \( S \), we say that \( S_1 \sqsubseteq S_2 \) iff

\[
(\forall s_2 \mid s_2 \in S_2 \cdot (\exists s_1 \mid s_1 \in S_1 \cdot s_1 \sqsubseteq s_2)).
\]

The refinement order between relations is just the Smyth order on result-sets, lifted functionally as we saw before.


Up-closure for a flat domain

In the flat domain \( S_{\bot} \) the definition of up-closure is particularly simple; it is

\[
S_1^+ = S_1 \text{ if } \bot \notin S_1 = S_{\bot} \text{ otherwise.}
\]

Although we have gone a long way to justify this easy construction, it is reassuring to know that it fits in with the general theory — and that will make things work much more smoothly later on.

Morgan’s Rule: If you’re going to re-invent the wheel... at least make sure it’s round.
A probabilistic powerdomain

The powerdomain construction we have just seen took an underlying set of values, with a partial order representing "refinement", and from it constructed — in a very general way — a partial order over sets of those values, one which can be used to describe demonically nondeterministic programs.

A similar construction — though more complex — takes the underlying set of values, with its refinement order, to a partial order over distributions on those values.

That is what we shall use.


A probabilistic powerdomain over a flat structure

• Take our underlying space \( S \), with its "flat" information order, and generate the Scott topology on it.

• Carry on by generating the space of probabilistic evaluations over that topology.

• Then notice that the result is isomorphic to

  \[
  \{ \Delta \subseteq [0,1] \mid (\sum_{s \in S} \Delta.s) \leq 1 \}
  \]

  with the order

  \( \Delta_1 \sqsubseteq \Delta_2 \) if \( \Delta_1.s \leq \Delta_2.s \) for all \( s \in S \).

• These are called discrete sub-probability measures.


Discrete sub-probability measures

A discrete probability distribution assigns probabilities to individual points, e.g. the function \( \{ H \mapsto 1/2, T \mapsto 1/2 \} \) that describes flipping an unbiased coin.

It gains the prefix "sub-" if it is not required to sum to one, as in the distribution \( \{ H \mapsto 1/3, T \mapsto 1/3 \} \) for a coin that "might not terminate"— this implicitly includes a probability \( 1 - 1/3 - 1/3 = 1/3 \) of nontermination \( \perp \).

Such a coin is refined by another that terminates at least as often; and any extra termination can be assigned to either proper outcome; for example, we have

\[
\{ H \mapsto 1/3, T \mapsto 1/3 \} \sqsubseteq \{ H \mapsto 2/5, T \mapsto 1/2 \}
\]

in which the right-hand coin refines the left-hand coin, but still has probability \( 1/10 \) of failing to terminate.

Again we have used theoretical tools (information orders, Scott topology, Jones/Plotkin evaluations...) that in the end (via isomorphism) have produced something quite simple.

We have thus ensured that the wheel is "round" — and so it rolls very nicely!

A demonic powerdomain over discrete sub-probability measures

To have both demonic and probabilistic choice available to us, we take the probabilistic powerdomain we have just constructed — discrete sub-probability measures — and apply our earlier "up-closure of sets" construction; the latter will add demonic choice, as it did before, but this time to elements that already model probability.

The flat domain over state space \( S \).

The probabilistic powerdomain over \( S_\perp \).

Sets of discrete sub-probability measures, for demonic choice.

Closed sets of discrete sub-probability measures, for refinement.

The relational model of demonic, probabilistic programs.

\[
\begin{array}{ccc}
S \perp & S & \perp S \\
\downarrow & \downarrow & \downarrow \\
S & \mathcal{C}S & \mathcal{C}S \subseteq \perp S \\
\rightarrow & \rightarrow & \rightarrow \\
\end{array}
\]

Discrete sub-probability measures.

Then up closure, convex closure, Cauchy closure?
A brief tour of \( C \) when \( S \) is the two-element space \( \{ H, T \} \) of coin-flip results

An unbiased coin.


A biased coin, up to 1/6 either way...

... one refinement of which is an unbiased coin.

Demonic choice between these
A brief tour of \( \mathcal{CS} \) when \( S \) is the two-element space \( \{ H, T \} \)
of coin-flip results

A possibly nonterminating coin whose refinements include
all three coins before.

Demonically, either of two possibly nonterminating coins.

\( \mathbb{S} \) for Jifeng He
He, McIver and Seidel. Probabilistic models for
the guarded command
language. Sci. Comp.,
Prog. 28:171-192, 1997.

\( \mathbb{S} \) is the relational model \( S \rightarrow \mathcal{CS} \)
A brief tour of C S concluded ... but what’s the connection with the programming logic?

Demonically, either of two possibly nonterminating coins.

What is the greatest guaranteed expected value of this program with respect to the post-expectation

$2[c=H] + [c=T]$ ?

It’s $5/6$.


A brief tour of $C\ S$ concluded... but what's the connection with the programming logic?

What is the greatest guaranteed expected value of this program with respect to the post-expectation $\forall s:S;\sigma:[prog].s \land [preE].s \leq \exp_{\sigma}.[postE]$?

It's $5/6$ again, because this time the demon goes to the other extreme.

Ex. 1: Experiment with the geometric presentation of the transformer semantics: what happens to the demonic coin with post-expectation $[c=H]$, $[c=T]$, $[true]$?

Ex. 2: Why isn't the answer to the third item above just $1$ again?

The two semantics are congruent

The two semantics are congruent