Formal Methods for Probabilistic Systems

Annabelle McIver
Carroll Morgan

- Source-level program logic
- Meta-theorems for loops
  - Introduction and example
  - Review of rules for standard loops
  - Rules for probabilistic loops
  - Analysis of an example
  - Probability-one termination
  - The Zero-One Law

Introduction and example

// Implement $_p\oplus$ using $\boxed{\text{unbiased random bits}}$ only.
$x := p$;
$\boxed{b := \text{true}\ {}_{1/2}\oplus\ \text{false};}$
do $b \rightarrow$
  $x := 2x$;
  if $x \geq 1$ **then** $x := x-1$ **fi**;
  $\boxed{b := \text{true}\ {}_{1/2}\oplus\ \text{false};}$
od;

e.g. /dev/random

if $x \geq 1/2$ // Variable x at least 1/2 with probability exactly p.
  then ...
  else ...
fi;

Example due to Joe Hurd (Cambridge, now Oxford).

CC Morgan. *Proof rules for probabilistic loops*. Proc. BCS-FACS 7th Refinement Workshop. Springer, 1996. `ewic.bcs.org/conferences/1996/refinement/papers/paper10.htm`

$\boxed{\text{Proof rules}}$ for standard loops

$x,b,e := 1,B,E$;
do $e \neq 0 \rightarrow$
  if even e
    then $b,e := b^2,\ e \div 2$
    else $e,x := e-1,\ x \times b$
  fi
od

Set x to B^E in logarithmic time.

RW Floyd. *Assigning meanings to programs*. Proc. Symp. Appl. Math. Mathematical Aspects of Computer Science 19:19-32, JT Schwartz (ed.). American Math. Soc, 1967.

CAR Hoare, 1969.
EW Dijkstra, 1975.
Gries, Backhouse, Kaldewaij, Cohen...

Proof rules for standard loops

$\{ B > 0 \text{ and } E \geq 0 \}$
$x,b,e := 1,B,E$;
$\{ b > 0 \text{ and } e \geq 0 \text{ and } B^E = x \times b^e \}$
do $e \neq 0 \rightarrow$
  $\{ \ldots \text{ and } e > 0 \}$
  if even e
    then $\{ e \geq 2 \text{ and even } e \ldots$ $\ldots \text{ and } B^E = x \times b^e \}$ $b,e := b^2,\ e \div 2$ $\{ B^E = x \times b^e \}$
    else $\{ B^E = x \times b^e \}$ $e,x := e-1,\ x \times b$ $\{ B^E = x \times b^e \}$
  fi
  $\{ B^E = x \times b^e \}$
od
$\{ B^E = x \times b^e \text{ and } e = 0 \}$
$\{ x = B^E \}$

RW Floyd. *Assigning meanings to programs*. Proc. Symp. Appl. Math. Mathematical Aspects of Computer Science 19:19-32, JT Schwartz (ed.). American Math. Soc, 1967.

Proof rules for standard loops

$\{\, B > 0 \text{ and } E \geq 0 \,\}$
$x,b,e := 1,B,E;$
$\{\, b > 0 \text{ and } e \geq 0 \text{ and } B^E = x \times b^e \,\}$
do $e \neq 0 \rightarrow$
$\{\, \ldots \text{ and } e > 0 \,\}$
if even e

then $\boxed{\{\, e \geq 2 \text{ and even } e \ldots \text{ and } B^E = x \times b^e \,\} \quad b,e := b^2,\ e{\div}2 \quad \{\, B^E = x \times b^e \,\}}$

else $\{\, B^E = x \times b^e \,\} \quad e,x := e{-}1,\ x \times b \quad \{\, B^E = x \times b^e \,\}$

fi
$\{\, B^E = x \times b^e \,\}$
od
$\{\, B^E = x \times b^e \text{ and } e = 0 \,\}$
$\{\, x = B^E \,\}$

Purely local *reasoning*.

CAR Hoare. *An axiomatic basis for computer programming*. Comm. ACM 12(10); 576-580,583, 1969.

Proof rules for standard loops

then $\{\, e \geq 2 \text{ and even } e \ldots \text{ and } B^E = x \times b^e \,\} \quad b,e := b^2, e{\div}2 \quad \{\, B^E = x \times b^e \,\}$

$$
\begin{array}{lll}
 & B^E = x \times b^e & \\
\equiv & (B^E = x \times b^e)\langle b,e := b^2, e{\div}2 \rangle & wp.(b,e := b^2, e{\div}2) \\
\equiv & B^E = x \times (b^2)^{(e \div 2)} & \text{substitution} \\
\Leftarrow & B^E = x \times b^e \quad \wedge \quad \text{even } e & \text{arithmetic}
\end{array}
$$

EW Dijkstra. *Guarded commands, nondeterminacy, and formal derivation of programs*. Comm. ACM 18(8):453-457, 1975.

... *A Discipline of Programming*. Prentice-Hall, 1976.

Proof rules for standard loops

$\{\, pre \,\}$
$init;$
$\{\boxed{inv}\}$
do $G \rightarrow$
$\{\, G \wedge \boxed{inv} \}$
body
$\{\boxed{inv}\}$
od
$\{\, \neg G \wedge \boxed{inv} \}$

The *loop invariant* makes it unnecessary to reason about "last time" or "next time" or "how many times" in the loop.

No "fence-post problem"...
No "banananas".

Proof rules for standard loops

$\{\, pre \,\}$
$init;$
$\{\, inv \,\}$
do $G \rightarrow$
$\{\, G \wedge inv \,\}$
body
$\{\, inv \,\}$
od
$\{\, \neg G \wedge inv \,\}$

Proof rules for probabilistic loops

$\{ pre \}$
init;
$\{ inv \}$
do $G \rightarrow$
  $\{ [G] \times inv \}$
  body
  $\{ inv \}$
od
$\{ [\neg G] \times inv \}$

$\{ pre \}$
init;
$\{ inv \}$
do $G \rightarrow$
$\{ G \wedge inv \}$
  body
  $\{ inv \}$
od
$\{ \neg G \wedge inv \}$

Proof rules for probabilistic loops: example

$\{ ? \}$
$x := p$;
$b :=$ true ${}_{1/2}\oplus$ false;
do $b \rightarrow$
  $x := 2x$;
  if $x \geq 1$ **then** $x := x\text{-}1$ **fi**;
  $b :=$ true ${}_{1/2}\oplus$ false;
od
$\{ [x \geq 1/2] \}$

What is the probability that x exceeds 1/2 on termination?

Proof rules for probabilistic loops: example

If we assume $0 \leq p \leq 1$, then it's clear that $0 \leq x \leq 1$ is a loop invariant...

...and we can therefore write the assignments to x in the loop body in the more convenient form

$\boxed{x := \text{frac}.(2x)}$.

$x := p$;
$b :=$ true ${}_{1/2}\oplus$ false;
$\{ 0 \leq x \leq 1 \}$
do $b \rightarrow$
  $\{ b \wedge 0 \leq x \leq 1 \}$
  $x := 2x$;
  if $x \geq 1$ **then** $x := x\text{-}1$ **fi**;
  $b :=$ true ${}_{1/2}\oplus$ false;
  $\{ 0 \leq x \leq 1 \}$
od
$\{ [x \geq 1/2] \}$

Proof rules for probabilistic loops: example

$[x \geq 1/2]$
$\Leftarrow \quad [\neg b] \times (\text{frac}.(2x) \boxed{\triangleleft}\ b\ \boxed{\triangleright} \text{int}.(2x))$

frac.(2x) **if** b **else** int.(2x)

$x := p$;
$b :=$ true ${}_{1/2}\oplus$ false;
do $b \rightarrow$
  $\boxed{x := \text{frac}.(2x);}$
  $b :=$ true ${}_{1/2}\oplus$ false;
  $\{ \text{frac}.(2x) \triangleleft b \triangleright \text{int}.(2x) \}$
od
$\{ [x \geq 1/2] \}$

CAR Hoare. *A couple of novelties in the Propositional Calculus.* Zeitschr. für Math. Logik und Grundlagen der Math. 31(2):173-178, 1985.

Proof rules for probabilistic loops: example

$$
\begin{aligned}
& \mathsf{frac}.(2x) \ \triangleleft\ b \triangleright \mathsf{int}.(2x) \\
\boxed{\cdot}\ \equiv\ & (\,\mathsf{frac}.(2x) + \mathsf{int}.(2x)\,)/2 \\
\equiv\ & 2x/2 \\
\equiv\ & x
\end{aligned}
$$

$x{:=}\ p;$

$b{:=}\ \mathsf{true}\ {}_{1/2}{\oplus}\ \mathsf{false};$

do $b \rightarrow$

$\quad x{:=}\ \mathsf{frac}.(2x);$
$\quad \{\, x \,\}$
$\quad \boxed{b{:=}\ \mathsf{true}\ {}_{1/2}{\oplus}\ \mathsf{false};}$
$\quad \{\, \mathsf{frac}.(2x) \ \triangleleft\ b\ \triangleright\ \mathsf{int}.(2x) \,\}$

od
$\{\, [x \ge 1/2] \,\}$

Proof rules for probabilistic loops: example

Assignment;
loop initialisation;
and then we repeat the earlier step.

$x{:=}\ p;$
$\{\, x \,\}$
$b{:=}\ \mathsf{true}\ {}_{1/2}{\oplus}\ \mathsf{false};$
$\{\, \mathsf{frac}.(2x) \ \triangleleft\ b \triangleright\ \mathsf{int}.(2x) \,\}$
do $b \rightarrow$
$\quad \{\, \mathsf{frac}.(2x) \,\}$
$\quad x{:=}\ \mathsf{frac}.(2x);$
$\quad \{\, x \,\}$
$\quad b{:=}\ \mathsf{true}\ {}_{1/2}{\oplus}\ \mathsf{false};$
$\quad \{\, \mathsf{frac}.(2x) \ \triangleleft\ b\ \triangleright\ \mathsf{int}.(2x) \,\}$
od
$\{\, [x \ge 1/2] \,\}$

Proof rules for probabilistic loops: example

And finally we see that the
pre-expectation overall...

is just p.

$\{\, p \,\}$
$x{:=}\ p;$
$\{\, x \,\}$
$b{:=}\ \mathsf{true}\ {}_{1/2}{\oplus}\ \mathsf{false};$
$\{\, \mathsf{frac}.(2x) \ \triangleleft\ b \triangleright\ \mathsf{int}.(2x) \,\}$
do $b \rightarrow$
$\quad \{\, \mathsf{frac}.(2x) \,\}$
$\quad x{:=}\ \mathsf{frac}.(2x);$
$\quad \{\, x \,\}$
$\quad b{:=}\ \mathsf{true}\ {}_{1/2}{\oplus}\ \mathsf{false};$
$\quad \{\, \mathsf{frac}.(2x) \ \triangleleft\ b\ \triangleright\ \mathsf{int}.(2x) \,\}$
od
$\{\, [x \ge 1/2] \,\}$

Proof rules for probabilistic loops: example

The probability that the
program establishes $x \ge 1/2$
is just p.

The loop invariant was

$\{\, \mathsf{frac}.(2x) \ \triangleleft\ b\ \triangleright\ \mathsf{int}.(2x) \,\}$,

"established" by the
initialisation and "maintained"
by the body.

$\{\, p \,\}$
$x{:=}\ p;$
$\{\, x \,\}$
$b{:=}\ \mathsf{true}\ {}_{1/2}{\oplus}\ \mathsf{false};$
$\{\, \mathsf{frac}.(2x) \ \triangleleft\ b \triangleright\ \mathsf{int}.(2x) \,\}$
do $b \rightarrow$
$\quad \{\, \mathsf{frac}.(2x) \,\}$
$\quad x{:=}\ \mathsf{frac}.(2x);$
$\quad \{\, x \,\}$
$\quad b{:=}\ \mathsf{true}\ {}_{1/2}{\oplus}\ \mathsf{false};$
$\quad \{\, \mathsf{frac}.(2x) \ \triangleleft\ b\ \triangleright\ \mathsf{int}.(2x) \,\}$
od
$\{\, [x \ge 1/2] \,\}$

Proof rules for probabilistic loops: termination

{ *inv* }
do G →
  { $[G] \times inv$ }
  body
  { *inv* }
od
{ $[\neg G] \times inv$ }

{ *inv* }
do G →
{ $G \wedge inv$ }
  body
  { *inv* }
od
{ $\neg G \wedge inv$ }

In addition, show that $inv \Rrightarrow term$, where *term* is the probability of termination ...

... in which case the conclusion { *inv* } **do ••• od** { $[\neg G] \times inv$ } expresses total — rather than just partial — correctness.

The $inv \Rrightarrow term$ *rule: a paradox?*

Suppose *term* is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now [*true*] is an invariant for the loop, which is just the everywhere-1 random variable. By *scaling* we therefore have also that p is invariant, for any non-negative constant p.

do G →
  { $[G] \times 1$ }
  body
  { 1 }
od

The $inv \Rrightarrow term$ *rule: a paradox?*

Suppose *term* is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now [*true*] is an invariant for the loop, which is just the everywhere-1 random variable. By *scaling* we therefore have also that p is invariant, for any non-negative constant p.

do G →
  { $[G] \times p$ }
  body
  { p }
od

The $inv \Rrightarrow term$ *rule: a paradox?*

Suppose *term* is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now [*true*] is an invariant for the loop, which is just the everywhere-1 random variable. By *scaling* we therefore have also that p is invariant, for any non-negative constant p.

Choose nonzero $p \Rrightarrow term$, and conclude
  { p } **do ••• od** { $[\neg G] \times p$ },

{ p }
do G →
  { $[G] \times p$ }
  body
  { p }
od
{ $[\neg G] \times p$ }

The *inv* ⇛ *term* rule: a paradox?

Suppose *term* is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now $[true]$ is an invariant for the loop, which is just the everywhere-1 random variable. By *scaling* we therefore have also that p is invariant, for any non-negative constant p.

Choose nonzero $p \Rrightarrow term$, and conclude

$\{p\}$ **do ⋯ od** $\{[\neg G] \times p\}$,

whence — by scaling back again — we have

$\{[true]\}$ **do ⋯ od** $\{[\neg G]\}$.

Thus in fact the loop terminates with probability one everywhere.

$\{p\}$
do G →
  $\{[G] \times p\}$
  body
  $\{p\}$
od
$\{[\neg G] \times p\}$

The *inv* ⇛ *term* rule: a paradox?

Suppose *term* is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now $[true]$ is an invariant for the loop, which is just the everywhere-1 random variable. By *scaling* we therefore have also that p is invariant, for any non-negative constant p.

Choose nonzero $p \Rrightarrow term$, and conclude

$\{p\}$ **do ⋯ od** $\{[\neg G] \times p\}$,

whence — by scaling back again — we have

$\{[true]\}$ **do ⋯ od** $\{[\neg G]\}$.

Thus in fact the loop terminates with probability one everywhere.

$\{1\}$
do G →
  $\{[G] \times p\}$
  body
  $\{p\}$
od
$\{[\neg G] \times 1\}$

The *inv* ⇛ *term* rule: a paradox?

Suppose *term* is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now $[true]$ is an invariant for the loop, which is just the everywhere-1 random variable. By *scaling* we therefore have also that p is invariant, for any non-negative constant p.

Choose nonzero $p \Rrightarrow term$, and conclude

$\{p\}$ **do ⋯ od** $\{[\neg G] \times p\}$,

whence — by scaling back again — we have

$\{[true]\}$ **do ⋯ od** $\{[\neg G]\}$.

Thus in fact the loop terminates with probability one everywhere.

$\{1\}$
do G →
  $\{[G] \times p\}$
  body
  $\{p\}$
od
$\{[\neg G] \times 1\}$

The *inv* ⇛ *term* rule: a pa~~ra~~dox?

Suppose *term* is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now $[true]$ is an invariant for the loop, which is just the everywhere-1 random variable. By *scaling* we therefore have also that p is invariant, for any non-negative constant p.

Choose nonzero $p \Rrightarrow term$, and conclude

$\{p\}$ **do ⋯ od** $\{[\neg G] \times p\}$,

whence — by scaling back again — we have

$\{[true]\}$ **do ⋯ od** $\{[\neg G]\}$.

Thus in fact the loop terminates with probability one everywhere.

It's not a paradox: it's a zero-one law *proved entirely at the level of program logic.*

CC Morgan. *Proof rules for probabilistic loops*. Proc. BCS-FACS 7th Refinement Workshop. Springer, 1996.
`ewic.bcs.org/conferences/1996/refinement/papers/paper10.htm`

Exercises

Ex. 1: *Give an operational argument justifying the zero-one law.*

Ex. 2: *Find a version of the law that holds even in infinite state-spaces.*