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
// Implement $p \oplus$ using unbiased random bits only.

$x := p;$

\[ b := \text{true} \oplus \frac{1}{2} \oplus \text{false}; \]

\[
\text{do } b \rightarrow \\
\quad x := 2x; \\
\quad \text{if } x \geq 1 \text{ then } x := x - 1 \text{ fi;}
\]

\[ b := \text{true} \oplus \frac{1}{2} \oplus \text{false}; \]

\text{od;}

\[
\text{if } x \geq 1/2 \quad \text{// Variable } x \text{ at least } 1/2 \text{ with probability exactly } p.
\]

\text{then } ... \quad \text{else } ... \quad \text{fi;}

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

Proof rules for standard loops

\[ \begin{align*}
x, b, e &:= 1, B, E; \\
\textbf{do } & e \neq 0 \rightarrow \\
& \quad \textbf{if } \text{even } e \\
& \quad \quad \textbf{then } b, e := b^2, e/2 \\
& \quad \quad \textbf{else } e, x := e - 1, x \times b \\
& \quad \textbf{fi} \\
& \textbf{od} \\
\end{align*} \]

Set \( x \) to \( B^E \) in logarithmic time.


EW Dijkstra, 1975.
Gries, Backhouse, Kaldewaij, Cohen...
**Proof rules for standard loops**

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

RW Floyd. *Assigning meanings to programs.*
Proof rules for standard loops

\{ B > 0 \text{ and } E \geq 0 \}\n\begin{align*}
& x,b,e := 1,B,E; \\
& \{ b > 0 \text{ and } e \geq 0 \text{ and } B^E = x \times b^e \}
\end{align*}
do e \neq 0 \rightarrow
\begin{align*}
& \{ \ldots \text{ and } e > 0 \}
\end{align*}
if even e
\begin{align*}
& \text{then} \quad \{ e \geq 2 \text{ and even } e \ldots \ b,e := b^2, \ e \div 2 \quad \{ B^E = x \times b^e \}
\end{align*}
... and \( B^E = x \times b^e \)
\begin{align*}
& \text{else} \quad \{ B^E = x \times b^e \}
\end{align*}
\begin{align*}
& e,x := e-1, \ x \times b \quad \{ B^E = x \times b^e \}
\end{align*}
fi
\begin{align*}
& \{ B^E = x \times b^e \}
\end{align*}
od
\begin{align*}
& \{ B^E = x \times b^e \text{ and } e = 0 \}
\end{align*}
\begin{align*}
& \{ x = B^E \}
\end{align*}

Purely local reasoning.

Proof rules for standard loops

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

\[
\ldots \text{ and } B^E = x \times b^e
\]

\[
B^E = x \times b^e
\]

- \[\equiv \] \((B^E = x \times b^e) \langle b, e := b^2, e \div 2 \rangle\) \quad \text{wp.} (b, e := b^2, e \div 2)
- \[\equiv B^E = x \times (b^2)^{\left(\frac{e}{2}\right)}\] \quad \text{substitution}
- \[\iff B^E = x \times b^e \land \text{even } e\] \quad \text{arithmetic}


... *A Discipline of Programming.* Prentice-Hall, 1976.
Proof rules for standard loops

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 “bananas”.

{ pre }
init;
{ inv }
\[ \text{do } G \rightarrow \{ \text{G } \wedge \text{inv} \} \]
body
{ inv }
\[ \text{od } \{ ! G \wedge \text{inv} \} \]
Proof rules for standard loops

\{ pre \}
init;
\{ inv \}
do \ G \to\n\{ G \land inv \}
body
\{ inv \}
od\n\{ ! G \land inv \}
Proof rules for **probabilistic loops**

\[
\begin{align*}
\{ \text{pre} \} \\
\text{init;} \\
\{ \text{inv} \} \\
\textbf{do} \ G \rightarrow \\
\quad \{ [G] \times \text{inv} \} \\
\quad \text{body} \\
\quad \{ \text{inv} \} \\
\textbf{od} \\
\quad \{ \neg G \times \text{inv} \}
\end{align*}
\]

\[
\begin{align*}
\{ \text{pre} \} \\
\text{init;} \\
\{ \text{inv} \} \\
\textbf{do} \ G \rightarrow \\
\quad \{ G \land \text{inv} \} \\
\quad \text{body} \\
\quad \{ \text{inv} \} \\
\textbf{od} \\
\quad \{ \neg G \land \text{inv} \}
\end{align*}
\]
Proof rules for probabilistic loops: example

\{
{ \text{?} } \\
x := p; \\
\}

b := \text{true} \oplus \frac{1}{2} \oplus \text{false};

\textbf{do } b \rightarrow \\
\quad x := 2x;

\quad \textbf{if } x \geq 1 \textbf{ then } x := x - 1 \textbf{ fi;}

\quad b := \text{true} \oplus \frac{1}{2} \oplus \text{false};

\textbf{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

\[ x := \frac{1}{2} \cdot (2x) \]

\[
\begin{align*}
  &x := p; \\
  &b := \text{true}_{1/2} \oplus \text{false}; \\
  &\{ 0 \leq x \leq 1 \} \\
  &\textbf{do} \quad b \rightarrow \\
  &\quad \{ b \land 0 \leq x \leq 1 \} \\
  &\quad x := 2x; \\
  &\quad \textbf{if} \ x \geq 1 \quad \textbf{then} \quad x := x - 1 \quad \textbf{fi}; \\
  &b := \text{true}_{1/2} \oplus \text{false}; \\
  &\{ 0 \leq x \leq 1 \} \\
  &\textbf{od} \\
  &\{ [x \geq 1/2] \}
\end{align*}
\]
Proof rules for probabilistic loops: example

\[
\begin{align*}
[x \geq 1/2] & \iff \left[\neg b\right] \times \left( \frac{2x}{\lfloor 2x \rfloor} \triangleleft b \triangleright \text{int}(2x) \right) \\
& \quad \frac{2x}{\lfloor 2x \rfloor} \text{ if } b \text{ else } \text{int}(2x)
\end{align*}
\]

\[
\begin{align*}
x := p; \\
b := \text{true}_{1/2} \oplus \text{false};
\end{align*}
\]

\[
\begin{align*}
\text{do } b & \rightarrow \\
& \quad \left\{ \frac{2x}{\lfloor 2x \rfloor} \triangleleft b \triangleright \text{int}(2x) \right\}
\end{align*}
\]

Proof rules for probabilistic loops: example

\[
\begin{align*}
\text{frac}(2x) &< b \Rightarrow \text{int}(2x) \\
\equiv & \quad (\text{frac}(2x) + \text{int}(2x))/2 \\
\equiv & \quad 2x/2 \\
\equiv & \quad x
\end{align*}
\]

\[
x := p;
\]

\[
b := \text{true}_{1/2} \oplus \text{false};
\]

\[
\textbf{do} \quad b \rightarrow \\
x := \text{frac}(2x);
\quad \{ x \} \\
\text{b} := \text{true}_{1/2} \oplus \text{false};
\quad \{ \text{frac}(2x) < b \Rightarrow \text{int}(2x) \} \\
\textbf{od} \\
\quad \{ [x \geq 1/2] \}
Proof rules for probabilistic loops: example

Assignment;
loop initialisation;
and then we repeat the earlier step.
Proof rules for probabilistic loops: example

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

is just \( p \).
Proof rules for probabilistic loops: example

The probability that the program establishes \( x \geq 1/2 \) is just \( p \).

The loop invariant was

\[
\{ \frac{2x}{2} < b \triangleright \text{int.}(2x) \},
\]

“established” by the initialisation and “maintained” by the body.
Proof rules for probabilistic loops: termination

\{ \text{inv} \} \quad \text{do} \quad G \to \quad \{ [G] \times \text{inv} \}

body
\{ \text{inv} \}
\text{od}
\{ [! G] \times \text{inv} \}

\{ \text{inv} \} \quad \text{do} \quad G \to \quad \{ G \wedge \text{inv} \}

body
\{ \text{inv} \}
\text{od}
\{ [! G] \times \text{inv} \}

In addition, show that \( \text{inv} \Rightarrow \text{term} \), where \( \text{term} \) is the probability of termination ...

... in which case the conclusion \( \{ \text{inv} \} \text{ do } \cdots \text{ od } \{ [! G] \times \text{inv} \} \)
expresses total — rather than just partial — correctness.
The \textit{inv} \Rightarrow \textit{term} rule: a paradox?

Suppose \textit{term} is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now \textit{true} is an invariant for the loop, which is just the everywhere-1 random variable. By \textit{scaling} we therefore have also that \textit{p} is invariant, for any non-negative constant \textit{p}.
The \textit{inv} \Rightarrow \textit{term} rule: a paradox?

Suppose \textit{term} is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

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

\begin{center}
\begin{verbatim}
\textbf{do} \ G \rightarrow \{ [G] \times p \} \\
\textbf{body} \ \
\{ p \} \\
\textbf{od}
\end{verbatim}
\end{center}
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 ⇒ term, and conclude

\[
\{ p \} \textbf{do} \cdots \textbf{od} \{ [!] G \times p \},
\]
The $\mathit{inv} \Rightarrow \mathit{term}$ rule: a paradox?

Suppose $\mathit{term}$ is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now $[\mathit{true}]$ 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 $p \Rightarrow \mathit{term}$, and conclude 

\[
\{ p \} \mathbf{do} \cdots \mathbf{od} \{ [G] \times p \} ,
\]

whence — by scaling back again — we have 

\[
\{ [\mathit{true}] \} \mathbf{do} \cdots \mathbf{od} \{ ![\! G] \} .
\]

Thus in fact the loop terminates with probability one everywhere.
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 ⇒ term, and conclude
\[
\{ p \} \;\text{do} \; \cdots \; \text{od} \; \{ [\neg G] \times p \} ,
\]
whence — by scaling back again — we have
\[
\{ [true] \} \;\text{do} \; \cdots \; \text{od} \; \{ [! G] \} .
\]

Thus in fact the loop terminates with probability one everywhere.
Suppose \textit{term} is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

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

Choose nonzero \texttt{p} \Rightarrow \textit{term}, and conclude
\[
\{ \texttt{p} \} \; \texttt{do} \; \cdots \; \texttt{od} \; \{ \; \texttt{[!] G} \; \times \; \texttt{p} \; \}
\]

whence — by scaling back again — we have
\[
\{ \; \texttt{[true]} \; \} \; \texttt{do} \; \cdots \; \texttt{od} \; \{ \; \texttt{[!] G} \; \}
\]

Thus in fact the loop terminates with probability one everywhere.
Suppose \( \text{term} \) is everywhere nonzero — but not necessarily one — and that the loop body is itself terminating.

Now \([\text{true}]\) 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 \( p \Rightarrow \text{term} \), and conclude
\[
\{ p \} \text{ do } \cdots \text{ od } \{ [! G] \times p \}, \text{whence — by scaling back again — we have}
\]
\[
\{ [\text{true}] \} \text{ do } \cdots \text{ od } \{ [! G] \}.
\]
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.

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.