VPHL: A Verified Partial-Correctness Logic for Probabilistic Programs

Robert Rand, Steve Zdancewic

University of Pennsylvania

Mathematical Foundations of Programming Semantics XXXI
Verified Probabilistic Hoare Logic
Verified Probabilistic Hoare Logic
Verified Probabilistic Hoare Logic
Let’s Take a Random Walk...
Rabbit Hunting
Rabbit Hunting
Rabbit Hunting
Rabbit Hunting
Rabbit Hunting
A Program to Analyze

Rabbit Hunting

\[ i := 0 \]
\[ caught := F \]
\[ \textbf{while } i < n \textbf{ do} \]
\[ \quad \textit{rabbit} := \text{UNIFORM}(k) \]
\[ \quad \textit{hunter} := \text{UNIFORM}(k) \]
\[ \quad caught := caught \lor (\textit{hunter} = \textit{rabbit}) \]
\[ \quad i := i + 1 \]
\[ \textbf{end while} \]

\[ \{ \text{Pr}(caught) = ? \} \]
\[ 4/32 \]
A Program to Analyze

Rabbit Hunting

\{Pr(True) = 1\}
i := 0
caught := F

while \(i < n\) do
\begin{align*}
rabbit &:= \text{UNIFORM}(k) \\
hunter &:= \text{UNIFORM}(k) \\
caught &:= caught \lor (\text{hunter} = \text{rabbit}) \\
i &:= i + 1
\end{align*}
end while

\{Pr(caught) = ?\}
## Comparison

<table>
<thead>
<tr>
<th>Paper</th>
<th>Full Distributions</th>
<th>While Loops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ramshaw, 1979</td>
<td>No</td>
<td>Partial</td>
</tr>
<tr>
<td>Den Hartog &amp; De Vink, 2002</td>
<td>No</td>
<td>Partial</td>
</tr>
<tr>
<td>Chadha et. al., 2007</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>VPHL</td>
<td>Yes</td>
<td>Partial</td>
</tr>
</tbody>
</table>
Principles

- Simple
  - Full Distributions
  - Truth-functional propositions
  - Resembles standard Hoare-logic

- Reliable
  - Rigorously verified deductive system
  - Can be safely extended

- Powerful
  - Support for non-termination
  - Capable of analyzing standard randomized algorithms
A Probabilistic Language

Classic Imperative Language \textit{Imp}:

\[ \theta : id \rightarrow value \]

Probabilistic Imperative Language \textit{PrImp}:

\[ \Theta : \theta \rightarrow [0, 1] \]
Representing Distributions

Full Distributions with Finite Support

\[ \sum_{\theta} \Theta(\theta) = 1 \]

Requiring finite support it allows us to represent distributions using a simple inductive structure.
Representing Distributions

\[ \Theta(\theta_1) = \frac{1}{6} \]
\[ \Theta(\theta_2) = \frac{1}{6} \]
\[ \Theta(\theta_3) = \frac{2}{3} \]
Representing Distributions

\[ \Theta(\theta_1) = \frac{1}{6} \quad \Theta(\theta_2) = \frac{1}{6} \]

\[ \Theta(\theta_3) = \frac{2}{3} \]
Representing Distributions

\[
\Theta(\theta_1) = \frac{1}{6} \\
\Theta(\theta_2) = \frac{1}{5} \\
\Theta(\theta_3) = \frac{2}{3}
\]
Representing Distributions

\[ \Theta(\theta_1) = \frac{1}{6} \]

\[ \Theta(\theta_2) = \frac{1}{6} \quad \Theta(\theta_3) = \frac{2}{3} \]
Probability

For a boolean expression $b$ and distribution $\Theta$:

$$Pr_\Theta(b) = \sum_{\theta} \{ \Theta(\theta) \mid b \text{ is true in } \theta \}$$
For a boolean expression $b$ and distribution $\Theta$:

$$Pr_\Theta(b) = \sum_{\theta} \{\Theta(\theta) \mid b \text{ is true in } \theta\}$$

$$\Theta(\theta_1) = 1/6 \quad \Theta(\theta_2) = 1/6 \quad \Theta(\theta_3) = 2/3$$
Probability

For a boolean expression $b$ and distribution $\Theta$:

$$Pr_{\Theta}(b) = \sum_{\theta} \{ \Theta(\theta) \mid b \text{ is true in } \theta \}$$

$\Theta(\theta_1) = 1/6 \quad \Theta(\theta_2) = 1/6 \quad \Theta(\theta_3) = 2/3$

$\theta_1(x) = 1 \quad \theta_2(x) = 2 \quad \theta_3(x) = 3$
Probability

For a boolean expression \( b \) and distribution \( \Theta \):

\[
Pr_{\Theta}(b) = \sum_{\theta} \{ \Theta(\theta) \mid b \text{ is true in } \theta \}
\]

\( \Theta(\theta_1) = \frac{1}{6} \quad \Theta(\theta_2) = \frac{1}{6} \quad \Theta(\theta_3) = \frac{2}{3} \)

\( \theta_1(x) = 1 \quad \theta_2(x) = 2 \quad \theta_3(x) = 3 \)

\( Pr_{\Theta}(x \text{ odd}) \)
Probability

For a boolean expression $b$ and distribution $\Theta$:

$$Pr_{\Theta}(b) = \sum_{\theta} \{\Theta(\theta) \mid b \text{ is true in } \theta\}$$

$\Theta(\theta_1) = 1/6$  \hspace{1cm} $\Theta(\theta_2) = 1/6$  \hspace{1cm} $\Theta(\theta_3) = 2/3$

$\theta_1(x) = 1$  \hspace{1cm} $\theta_2(x) = 2$  \hspace{1cm} $\theta_3(x) = 3$

$$Pr_{\Theta}(x \text{ odd}) = 1/6 + 2/3 = 5/6$$
Tautology

For any distribution $\Theta$ and tautology $T$:

$$Pr_{\Theta}(T) = 1$$
Complement

For any distribution $\Theta$ and boolean $b$:

$$Pr_\Theta(\neg b) = 1 - Pr_\Theta(b)$$
**Probability**

**Marginalization**

For any distribution $\Theta$ and booleans $a, b$:

$$Pr_\Theta(a) = Pr_\Theta(a \land b) + Pr_\Theta(a \land \neg b)$$
Primp Commands
Primp Commands

\[
\begin{align*}
[c] \oplus_{1/2} & \quad \oplus_{1/3} & \quad [c] \theta_3 \\
\theta_1 & \quad \theta_2
\end{align*}
\]
Primp Commands

\[
\begin{align*}
[c] \theta_1 & \quad \oplus_{1/2} \quad [c] \theta_2 \\
& \quad \oplus_{1/3} \quad [c] \theta_3
\end{align*}
\]
Primp Commands

\[ c \equiv y := \text{toss}(\frac{1}{5}) \]
**Primp Commands**

\[ c \equiv y := \text{toss}(\frac{1}{5}) \]
VPHL: Hoare Logic

Definition: \( \{ P \} c \{ Q \} \)

\[
P(\Theta) \quad c / \Theta \downarrow \Theta' \quad \frac{Q(\Theta')}{Q(\Theta')}\]
VPHL: Hoare Logic

Truth-functional assertions over full distributions

\( P, Q ::= Pr(B) = p \mid Pr(B) < p \mid Pr(B) > p \)

\( \mid P \land P \mid P \lor P \)
Basic Rules

\[
P' \rightarrow P \quad \{P\} \ c \ \{Q\} \quad Q \rightarrow Q' \quad \text{Consequence}
\]

\[
\{P'\} \ c \ \{Q'\}
\]

Skip

\[
\{P\} \ \text{skip} \ \{P\}
\]

Assign

\[
\{P[z \rightarrow e]\} \ z := e \ \{P\}
\]

Sequence

\[
\{P\} \ c_1 \ \{Q\} \quad \{Q\} \ c_2 \ \{R\}
\]

\[
\{P\} \ c_1; \ c_2 \ \{R\}
\]
The Toss Rule

\[
\text{y free in } P \\
\{P\} \ y := \text{toss}(p) \ \{P \triangleleft^y_p\}
\]
The Toss Rule

\[
y \text{ free in } P \\
\{P\} \ y := \text{toss}(p) \ \{P \triangleleft^y_p\} \quad \text{Toss}
\]

\[
[Pr(b) = a] \triangleleft^y_p \equiv \quad Pr(b \land y) = pa \quad \land \\
Pr(b \land \neg y) = (1 - p)a
\]
The If Rule

c ≡ \text{if } y \text{ then } c_1 \text{ else } c_2
The If Rule

\[ c \equiv \text{if } y \text{ then } c_1 \text{ else } c_2 \]
The If Rule

\[ c \equiv \text{if } y \text{ then } c_1 \text{ else } c_2 \]

where \( \theta_1(y) = T \), \( \theta_2(y) = F \) and \( \theta_3(y) = F \)
The If Rule

c ≡ if \( y \) then \( c_1 \) else \( c_2 \)

where \( \theta_1(y) = T \), \( \theta_2(y) = F \) and \( \theta_3(y) = F \)
The If Rule

\( c \equiv \text{if } y \text{ then } c_1 \text{ else } c_2 \)
The If Rule

\[ c \equiv \text{if } y \text{ then } c_1 \text{ else } c_2 \]
The If Rule

c \equiv \text{if } y \text{ then } c_1 \text{ else } c_2

$$\text{\{P}_1 \land y\} \ c_1 \ {\text{\{Q}_1\}} \ \text{\oplus_{1/6}} \ \text{\{P}_2 \land \neg y\} \ c_2 \ \text{\{Q}_2\}$$

\dots

\dots

\dots

22/32
The If Rule

c \equiv \text{if } y \text{ then } c_1 \text{ else } c_2

\{Pr(y) = p \land P'_1 \land P'_2\} \ c \ \{Q'_1 \land Q'_2\}

\{P_1 \land y\} \ c_1 \ \{Q_1\}

\{P_2 \land \neg y\} \ c_2 \ \{Q_2\}
The If Rule

Why $P'_1$?

- Scaling – we have to normalize the probabilities in each branch

- Conditioning on the guard – we need to avoid conflict
The If Rule

Why $P'_1$?

- **Scaling** – we have to normalize the probabilities in each branch

- Conditioning on the guard – we need to avoid conflict
The If Rule

Why $P'_1$?

- **Scaling** – we have to normalize the probabilities in each branch
  \[ Pr(b) = a \Rightarrow Pr(b) = p \times a \]

- Conditioning on the guard – we need to avoid conflict
The If Rule

Why $P'_1$?

- Scaling – we have to normalize the probabilities in each branch
  \[ \Pr(b) = a \Rightarrow \Pr(b) = p \times a \]
- **Conditioning on the guard** – we need to avoid conflict
The If Rule

Why $P'_1$?

- Scaling – we have to normalize the probabilities in each branch
  \[ Pr(b) = a \Rightarrow Pr(b) = p \times a \]

- **Conditioning on the guard** – we need to avoid conflict
  \[ Pr(b) = p \times a \Rightarrow Pr(b \land y) = p \times a \]
Applying the IF Rule

$UNIFORM(3)$

$u_1 := \text{toss}(\frac{1}{3})$;
if $u_1$ then
  $x := 3$
else
  $u_2 := \text{toss}(\frac{1}{2})$;
  if $u_2$ then
    $x := 2$
  else
    $x := 1$
  end if
end if

end if
Applying the IF Rule

\textit{UNIFORM(3)}

\begin{align*}
  u_1 & := \text{toss}(\frac{1}{3}); \\
  \text{if } u_1 & \text{ then} \\
  & \{Pr(3 = 3) = 1\} \ x := 3 \ \{Pr(x = 3) = 1\} \\
  \text{else} & \\
  u_2 & := \text{toss}(\frac{1}{2}); \\
  \text{if } u_2 & \text{ then} \\
  & \{Pr(2 = 2) = 1\} \ x := 2 \ \{Pr(x = 2) = 1\} \\
  \text{else} & \\
  & \{Pr(1 = 1) = 1\} \ x := 1 \ \{Pr(x = 1) = 1\} \\
  \text{end if}
\end{align*}

\text{end if}
Applying the IF Rule

UNIFORM(3)

\{ Pr(\text{True}) = 1 \} \quad u_1 := \text{toss}(\frac{1}{3}) \quad \{ Pr(\text{True} \land u_1) = \frac{1}{3} \}
if \quad u_1 \quad \text{then}
\{ Pr(3 = 3) = 1 \} \quad x := 3 \quad \{ Pr(x = 3) = 1 \}
else
\{ Pr(\text{True}) = 1 \} \quad u_2 := \text{toss}(\frac{1}{2}) \quad \{ Pr(\text{True} \land u_2) = \frac{1}{2} \}
if \quad u_2 \quad \text{then}
\{ Pr(2 = 2) = 1 \} \quad x := 2 \quad \{ Pr(x = 2) = 1 \}
else
\{ Pr(1 = 1) = 1 \} \quad x := 1 \quad \{ Pr(x = 1) = 1 \}
end if
end if
Applying the IF Rule

**UNIFORM(3)**

\[
\{Pr(True) = 1\} \ u_1 := \text{toss}(\frac{1}{3}); \ \{Pr(u_1) = \frac{1}{3}\}
\]

if \( u_1 \) then

\[
\{Pr(3 = 3) = 1\} \ x := 3 \ \{Pr(x = 3) = 1\}
\]

else

\[
\{Pr(True) = 1\} \ u_2 := \text{toss}(\frac{1}{2}); \ \{Pr(u_2) = \frac{1}{2}\}
\]

if \( u_2 \) then

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

else

\[
\{Pr(1 = 1) = 1\} \ x := 1 \ \{Pr(x = 1) = 1\}
\]

end if

end if

\[
\{Pr(x = 2) = 1\} \land \{Pr(x = 3) = 1\} \land \{Pr(x = 1) = 1\}
\]
Applying the IF Rule

\textbf{UNIFORM}(3)

\{Pr(True) = 1\} u_1 := \text{toss}(\frac{1}{3}); \{Pr(u_1) = \frac{1}{3}\}

\textbf{if} u_1 \textbf{then}

\{Pr(3 = 3) = 1\} x := 3 \{Pr(x = 3) = 1\}

\textbf{else}

\{Pr(True) = 1\} u_2 := \text{toss}(\frac{1}{2}); \{Pr(u_2) = \frac{1}{2}\}

\textbf{if} u_2 \textbf{then}

\{Pr(2 = 2) = 1\} x := 2 \{Pr(x = 2) = 1\}

\textbf{else}

\{Pr(1 = 1) = 1\} x := 1 \{Pr(x = 1) = 1\}

\textbf{end if}

\textbf{end if}
Applying the IF Rule

UNIFORM(3)

\{Pr(True) = 1\} \ u_1 := \text{toss}(\frac{1}{3}); \ \{Pr(u_1) = \frac{1}{3}\}

\text{if } u_1 \text{ then}

\{Pr(3 = 3) = 1\} \ x := 3 \ \{Pr(x = 3) = 1\}

\text{else}

\{Pr(True) = 1\} \ u_2 := \text{toss}(\frac{1}{2}); \ \{Pr(u_2) = \frac{1}{2}\}

\text{if } u_2 \text{ then}

\{Pr(2 = 2) = 1\} \ x := 2 \ \{Pr(x = 2) = 1\}

\text{else}

\{Pr(1 = 1) = 1\} \ x := 1 \ \{Pr(x = 1) = 1\}

\text{end if}

\{Pr(x = 2 \land u_2) = \frac{1}{2} \land Pr(x = 1 \land \neg u_2) = \frac{1}{2}\}

\text{end if}
Applying the IF Rule

UNIFORM(3)

\{Pr(True) = 1\} \ u_1 := \text{toss}(\frac{1}{3}); \ \{Pr(u_1) = \frac{1}{3}\}
if \ u_1 \ then
\{Pr(3 = 3) = 1\} \ x := 3 \ \{Pr(x = 3) = 1\}
else
\{Pr(True) = 1\} \ u_2 := \text{toss}(\frac{1}{2}); \ \{Pr(u_2) = \frac{1}{2}\}
if \ u_2 \ then
\{Pr(2 = 2) = 1\} \ x := 2 \ \{Pr(x = 2) = 1\}
else
\{Pr(1 = 1) = 1\} \ x := 1 \ \{Pr(x = 1) = 1\}
end if
\{Pr(x = 2) \geq \frac{1}{2} \land Pr(x = 1) \geq \frac{1}{2}\}
end if
Applying the IF Rule

UNIFORM(3)

\[ \{ Pr(True) = 1 \} \quad u_1 := \text{toss}(\frac{1}{3}) \quad \{ Pr(u_1) = \frac{1}{3} \} \]

if \( u_1 \) then

\[ \{ Pr(3 = 3) = 1 \} \quad x := 3 \quad \{ Pr(x = 3) = 1 \} \]

else

\[ \{ Pr(True) = 1 \} \quad u_2 := \text{toss}(\frac{1}{2}) \quad \{ Pr(u_2) = \frac{1}{2} \} \]

if \( u_2 \) then

\[ \{ Pr(2 = 2) = 1 \} \quad x := 2 \quad \{ Pr(x = 2) = 1 \} \]

else

\[ \{ Pr(1 = 1) = 1 \} \quad x := 1 \quad \{ Pr(x = 1) = 1 \} \]

end if

\[ \{ Pr(x = 2) = \frac{1}{2} \land Pr(x = 1) = \frac{1}{2} \} \]

end if
Applying the IF Rule

UNIFORM(3)

\{Pr(True) = 1\} \ u_1 := \text{toss}(\frac{1}{3}); \ \{Pr(u_1) = \frac{1}{3}\}

\text{if } u_1 \text{ then}

\{Pr(3 = 3) = 1\} \ x := 3 \ \{Pr(x = 3) = 1\}

\text{else}

\{Pr(True) = 1\} \ u_2 := \text{toss}(\frac{1}{2}); \ \{Pr(u_2) = \frac{1}{2}\}

\text{if } u_2 \text{ then}

\{Pr(2 = 2) = 1\} \ x := 2 \ \{Pr(x = 2) = 1\}

\text{else}

\{Pr(1 = 1) = 1\} \ x := 1 \ \{Pr(x = 1) = 1\}

\text{end if}

\{Pr(x = 2) = \frac{1}{2} \land Pr(x = 1) = \frac{1}{2}\}

\text{end if}
Applying the IF Rule

\text{UNIFORM}(3)

\{Pr(\text{True}) = 1\} \ u_1 := \text{toss}(\frac{1}{3}); \ \{Pr(u_1) = \frac{1}{3}\}

\text{if} \ u_1 \ \text{then}
\{Pr(3 = 3) = 1\} \ x := 3 \ \{Pr(x = 3) = 1\}

\text{else}
\{Pr(\text{True}) = 1\} \ u_2 := \text{toss}(\frac{1}{2}); \ \{Pr(u_2) = \frac{1}{2}\}

\text{if} \ u_2 \ \text{then}
\{Pr(2 = 2) = 1\} \ x := 2 \ \{Pr(x = 2) = 1\}

\text{else}
\{Pr(1 = 1) = 1\} \ x := 1 \ \{Pr(x = 1) = 1\}

\text{end if}
\{Pr(x = 2) = \frac{1}{2} \land Pr(x = 1) = \frac{1}{2}\}

\text{end if}
\{Pr(x = 3) = \frac{1}{3} \land Pr(x = 2) = \frac{1}{3} \land Pr(x = 1) = \frac{1}{3}\}
The While Rule

We want to guarantee that the program terminates in some number of steps $n$, assuming that it terminates.
The While Rule

The *Deterministic Invariant* guarantees that the guard takes on a deterministic value.

The *Probabilistic Invariant* preserves a set of probabilities throughout loop execution.
Deterministic Invariant

Rabbit Hunting

while $i < n$ do

\[ \text{rabbit} := \text{UNIFORM}(k) \]
\[ \text{hunter} := \text{UNIFORM}(k) \]
\[ \text{caught} := \text{caught} \lor (\text{hunter} = \text{rabbit}) \]
\[ i := i + 1 \]

end while
Rabbit Hunting

\[ \text{while } i < n \text{ do} \]

\[ \{ \exists m \leq n : Pr(i = m) = 1 \land Pr(i < n) = 1 \} \]

\[ \text{rabbit} := \text{UNIFORM}(k) \]

\[ \text{hunter} := \text{UNIFORM}(k) \]

\[ \text{caught} := \text{caught} \lor (\text{hunter} = \text{rabbit}) \]

\[ i := i + 1 \]

\[ \text{end while} \]
Deterministic Invariant

Rabbit Hunting

\[ \text{while } i < n \text{ do} \]
\[ \{ \exists m \leq n : Pr(i = m) = 1 \land Pr(i < n) = 1 \} \rightarrow \]
\[ \{ \exists m \leq n : Pr(i + 1 = m) = 1 \} \]
\[ \text{rabbit} := \text{UNIFORM}(k) \]
\[ \text{hunter} := \text{UNIFORM}(k) \]
\[ \text{caught} := \text{caught} \lor (\text{hunter} = \text{rabbit}) \]
\[ i := i + 1 \]

end while
Deterministic Invariant

Rabbit Hunting

while $i < n$ do

\[ \{ \exists m \leq n : Pr(i = m) = 1 \land Pr(i < n) = 1 \} \rightarrow \{ \exists m \leq n : Pr(i + 1 = m) = 1 \} \]

rabbit := UNIFORM($k$)

hunter := UNIFORM($k$)

caught := caught $\lor$ (hunter = rabbit)

$i := i + 1$

\[ \{ \exists m \leq n : Pr(i = m) = 1 \} \]

end while
Rabbit Hunting

while $i < n$ do

$rabbit := \text{UNIFORM}(k)$
$\text{hunter} := \text{UNIFORM}(k)$

$caught := caught \lor (\text{hunter} = rabbit)$
$i := i + 1$

end while
Probabilistic Invariant

Rabbit Hunting

while $i < n$ do

$\{Pr(\neg\text{caught}) = \left(\frac{k-1}{k}\right)^i\}$

$rabbit := \text{UNIFORM}(k)$

$hunter := \text{UNIFORM}(k)$

caught := caught $\lor$ ($hunter = rabbit$)

$i := i + 1$

end while
Probabilistic Invariant

Rabbit Hunting

while $i < n$ do

$\{ Pr(\neg \text{caught}) = \left( \frac{k-1}{k} \right)^i \}$

$rabbit := \text{UNIFORM}(k)$

$hunter := \text{UNIFORM}(k)$

$\{ Pr(\neg \text{caught} \land hunter \neq rabbit) = \left( \frac{k-1}{k} \right) \left( \frac{k-1}{k} \right)^i \}$

$caught := caught \lor (hunter = rabbit)$

$i := i + 1$

end while
Probabilistic Invariant

Rabbit Hunting

while $i < n$ do

$\{ Pr(\neg \text{caught}) = \left( \frac{k-1}{k} \right)^i \}$

$rabbit := \text{UNIFORM}(k)$

$\text{hunter} := \text{UNIFORM}(k)$

$\{ Pr(\neg \text{caught} \land \text{hunter} \neq \text{rabbit}) = \left( \frac{k-1}{k} \right)^{i+1} \}$

$\text{caught} := \text{caught} \lor (\text{hunter} = \text{rabbit})$

$i := i + 1$

end while
Probabilistic Invariant

Rabbit Hunting

while $i < n$ do

\{ $Pr(\neg \text{caught}) = \left( \frac{k-1}{k} \right)^i$ \}

$rabbit := \text{UNIFORM}(k)$

$hunter := \text{UNIFORM}(k)$

\{ $Pr(\neg \text{caught} \land hunter \neq rabbit) = \left( \frac{k-1}{k} \right)^{i+1}$ \}

$caught := caught \lor (hunter = rabbit)$

$i := i + 1$

\{ $Pr(\neg \text{caught}) = \left( \frac{k-1}{k} \right)^i$ \}

end while
**Catching Rabbits**

Rabbit Hunting

\[
\{ \Pr(\text{True}) = 1 \} \\
i := 0 \\
\text{caught} := F
\]

**while** \(i < n\) **do**

\[
rabbit := \text{UNIFORM}(k) \\
hunter := \text{UNIFORM}(k) \\
\text{caught} := (\text{hunter} = \text{rabbit}) \lor \text{caught} \\
i := i + 1
\]

**end while**
Catching Rabbits

Rabbit Hunting

\{Pr(True) = 1\}
\(i := 0\)
\(caught := F\)
\{Pr(\neg caught) = 1 \land Pr(i = 0) = 1\}

\textbf{while} \(i < n\) \textbf{do}

\hspace{1em} \text{rabbit} := \text{UNIFORM}(k)
\hspace{1em} \text{hunter} := \text{UNIFORM}(k)
\hspace{1em} caught := (\text{hunter} = \text{rabbit}) \lor caught
\hspace{1em} i := i + 1

\textbf{end while}
CATCHING RABBITS

Rabbit Hunting

\{ Pr(\text{True}) = 1 \}
\[ i := 0 \]
\[ \text{caught} := \text{F} \]
\{ Pr(\neg \text{caught}) = 1 \land Pr(i = 0) = 1 \} \rightarrow
\{ Pr(\neg \text{caught}) = \left( \frac{k-1}{k} \right)^i \land \exists m \leq n : Pr(i = m) = 1 \}

\textbf{while} \ i < n \ \textbf{do}
\begin{align*}
\text{rabbit} & := \text{UNIFORM}(k) \\
\text{hunter} & := \text{UNIFORM}(k) \\
\text{caught} & := (\text{hunter} = \text{rabbit}) \lor \text{caught} \\
i & := i + 1
\end{align*}
\textbf{end while}
Catching Rabbits

Rabbit Hunting

\{Pr(\text{True}) = 1\}

\(i := 0\)

\(\text{caught} := \text{F}\)

\{Pr(\neg\text{caught}) = 1 \land Pr(i = 0) = 1\} \rightarrow

\{Pr(\neg\text{caught}) = \left(\frac{k-1}{k}\right)^i \land \exists m \leq n : Pr(i = m) = 1\}

\textbf{while} i < n \textbf{ do}

\quad \text{rabbit} := \text{UNIFORM}(k)

\quad \text{hunter} := \text{UNIFORM}(k)

\quad \text{caught} := (\text{hunter} = \text{rabbit}) \lor \text{caught}

\quad i := i + 1

\textbf{end while}

\{Pr(\neg\text{caught}) = \left(\frac{k-1}{k}\right)^i \land \exists m \leq n : Pr(i = m) = 1 \land i \neq n\}
CATCHING RABBITS

Rabbit Hunting

\{ Pr(True) = 1 \}
\ i := 0
caught := F
\{ Pr(\neg\text{caught}) = 1 \land Pr(i = 0) = 1 \} \rightarrow
\{ Pr(\neg\text{caught}) = \left( \frac{k-1}{k} \right)^i \land \exists m \leq n : Pr(i = m) = 1 \}

\textbf{while} \ i < n \ \textbf{do}
  rabbit := UNIFORM(k)
  hunter := UNIFORM(k)
  caught := (\text{hunter} = \text{rabbit}) \lor \text{caught}
  i := i + 1
\textbf{end while}

\{ Pr(\neg\text{caught}) = \left( \frac{k-1}{k} \right)^i \land Pr(i = n) = 1 \}
CATCHING RABBITS

Rabbit Hunting

\{ Pr(True) = 1 \}
i := 0
caught := \text{F}
\{ Pr(\neg caught) = 1 \land Pr(i = 0) = 1 \} \rightarrow
\{ Pr(\neg caught) = \left( \frac{k-1}{k} \right)^i \land \exists m \leq n : Pr(i = m) = 1 \}

\text{while } i < n \text{ do}
\quad rabbit := \text{UNIFORM}(k)
\quad hunter := \text{UNIFORM}(k)
\quad caught := (\text{hunter} = \text{rabbit}) \lor caught
\quad i := i + 1

\text{end while}
\{ Pr(\neg caught) = \left( \frac{k-1}{k} \right)^i \land Pr(i = n) = 1 \} \rightarrow
\{ Pr(caught) = 1 - \left( \frac{k-1}{k} \right)^n \}
Probabilistic Termination

What about programs that terminate probabilistically?
What about programs that terminate probabilistically?

\[
\{ \text{Pr(True)} = 1 \}
\]

\[
y := \text{toss} \left( \frac{1}{2} \right);
\]

if \( y \) then \( x := 4 \) else loop

\[
\{ \text{Pr}(x = 4) = ? \}
\]
Soundness

Theorem

All of the VPHL rules are sound with respect to the semantics of PrImp.
Verified

https://github.com/rnrand/VPHL
Thank You Questions?

https://github.com/rnrand/VPHL