VPHL: A Verified Partial-Correctness Logic for Probabilistic Programs

Robert Rand, Steve Zdancewic

University of Pennsylvania

Mathematical Foundations of Programming Semantics XXXI

VPHL

Verified
Probabilistic
Hoare
Logic

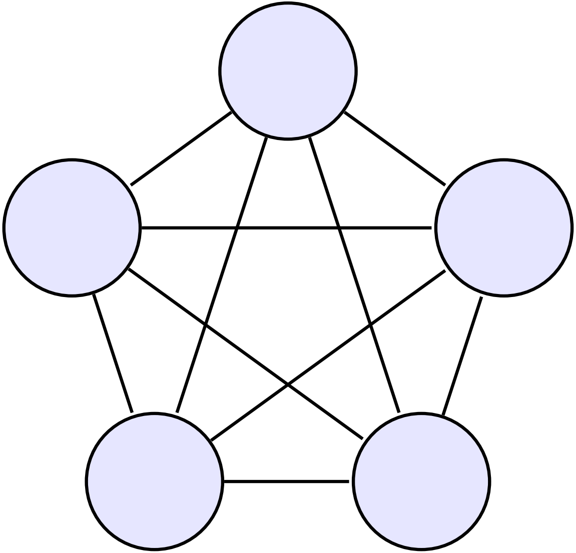
VPHL

Verified
Probabilistic
Hoare
Logic

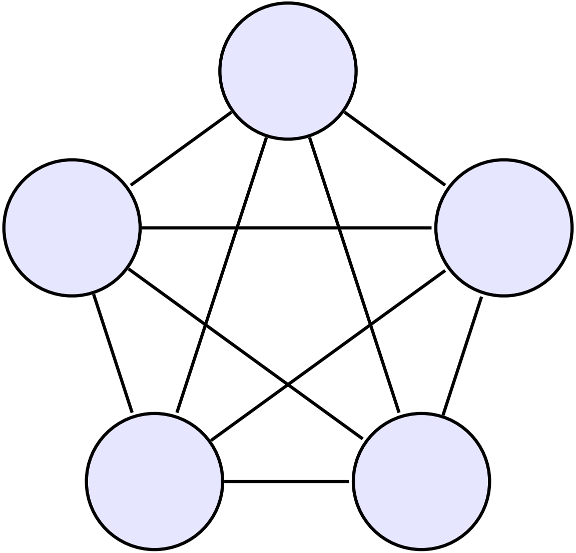
VPHL

Verified
Probabilistic
Hoare
Logic

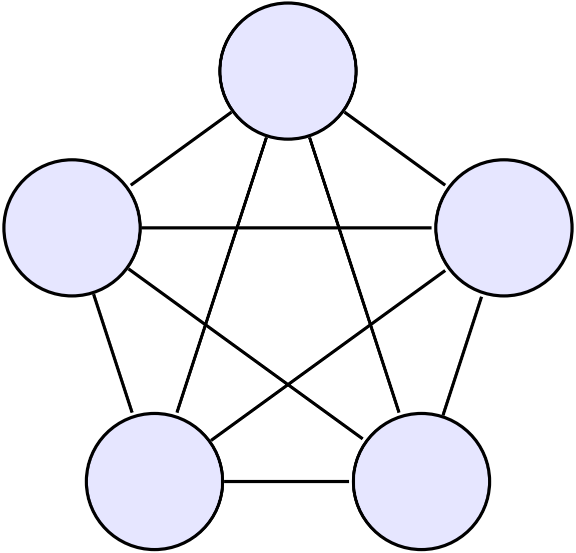
Let's Take a Random Walk...



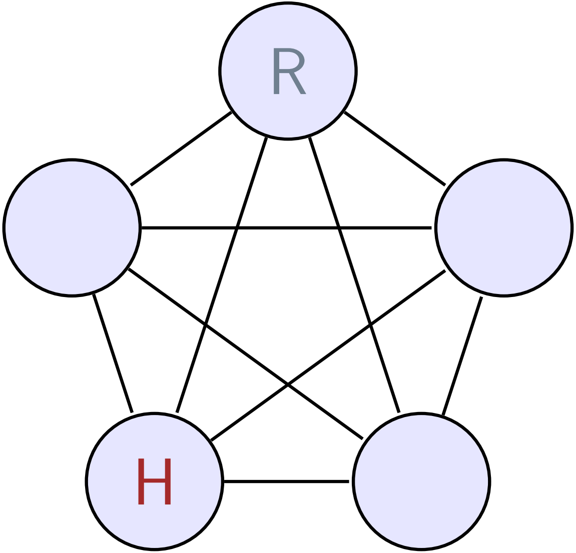
Rabbit Hunting



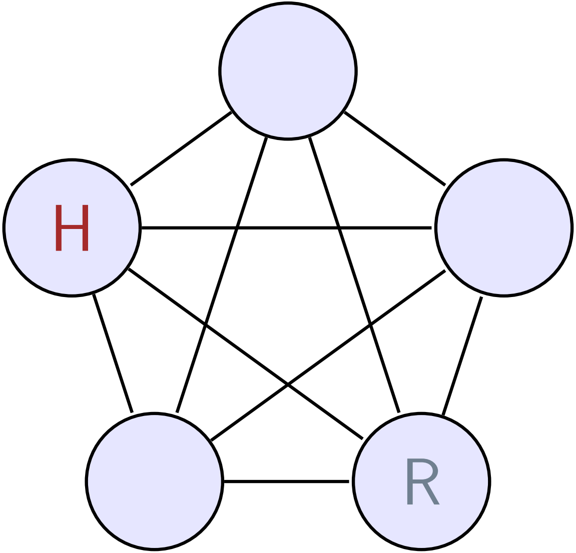
Rabbit Hunting



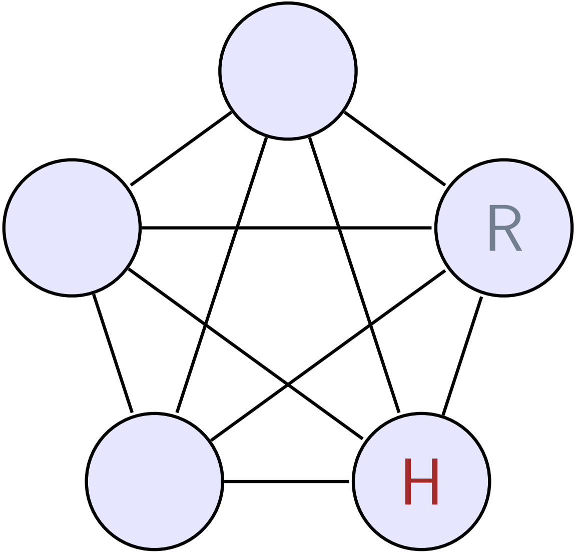
Rabbit Hunting



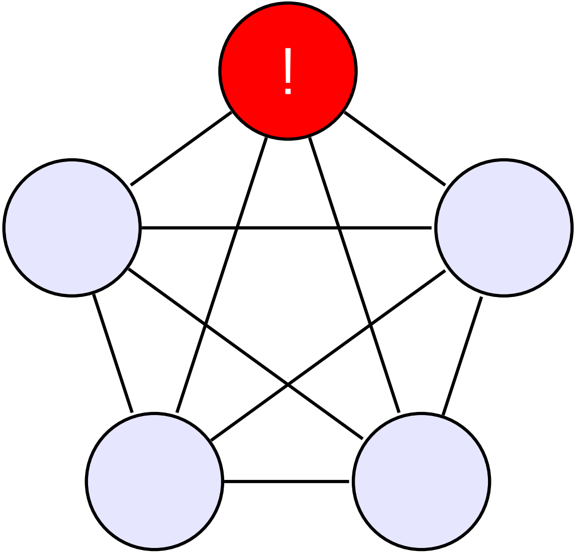
Rabbit Hunting



Rabbit Hunting



Rabbit Hunting



A Program to Analyze

Rabbit Hunting

i := 0
caught := F
while $i < n$ **do**
    rabbit := UNIFORM(k)
    hunter := UNIFORM(k)
    caught := $caught \vee (hunter = rabbit)$
    i := $i + 1$
end while

A Program to Analyze

Rabbit Hunting

$\{Pr(True) = 1\}$
$i := 0$
$caught := \texttt{F}$
while $i < n$ **do**
  $rabbit := \texttt{UNIFORM}(k)$
  $hunter := \texttt{UNIFORM}(k)$
  $caught := caught \vee (hunter = rabbit)$
  $i := i + 1$
end while
$\{Pr(caught) = ?\}$

Comparison

Paper	Full Distributions	While Loops
Ramshaw, 1979	No	Partial
Den Hartog & De Vink, 2002	No	Partial
Chadha et. al., 2007	No	No
VPHL	Yes	Partial

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 *Imp*:

$$\theta : id \rightarrow value$$

Probabilistic Imperative Language *PrImp*:

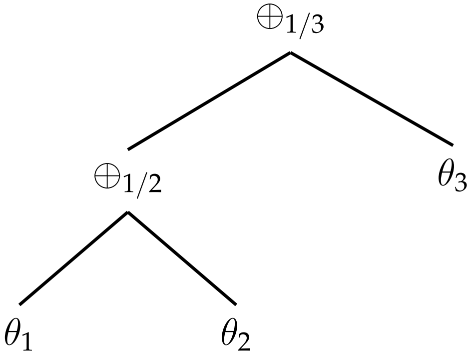
$$\Theta : \theta \rightarrow [0, 1]$$

Full Distributions with Finite Support

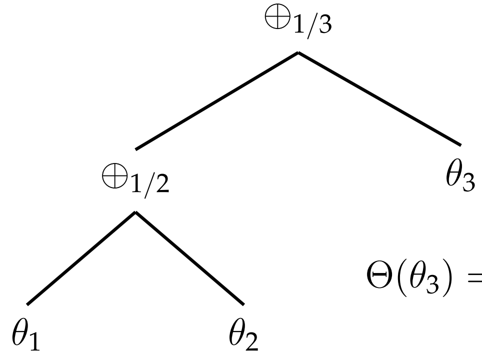
$$\sum_{\theta} \Theta(\theta) = 1$$

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

Representing Distributions



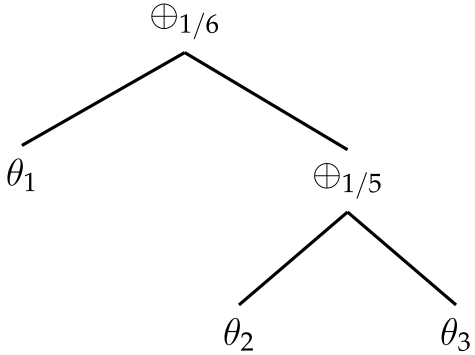
Representing Distributions



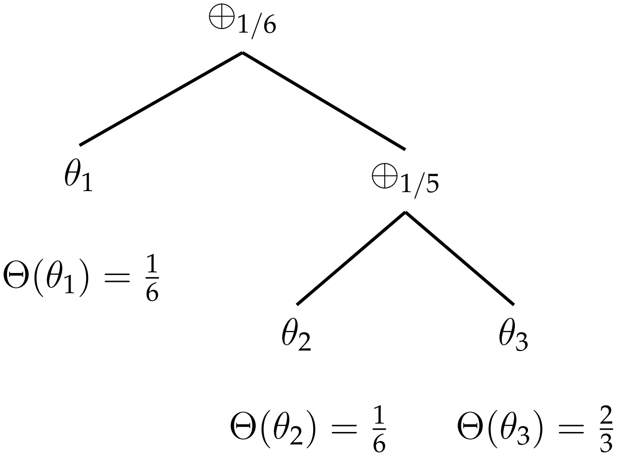
$\Theta(\theta_3) = \frac{2}{3}$

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

Representing Distributions



Representing Distributions



Probability

For a boolean expression b and distribution Θ:

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

Probability

For a boolean expression b and distribution Θ:

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

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

Probability

For a boolean expression b and distribution Θ:

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

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

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

Probability

For a boolean expression b and distribution Θ:

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

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

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

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

Probability

For a boolean expression b and distribution Θ:

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

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

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

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

Tautology

For any distribution Θ and tautology T:

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

Complement

For any distribution Θ and boolean b:

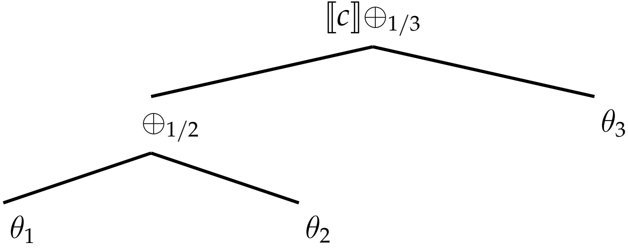
$$Pr_\Theta(\neg b) = 1 - Pr_\Theta(b)$$

Marginalization

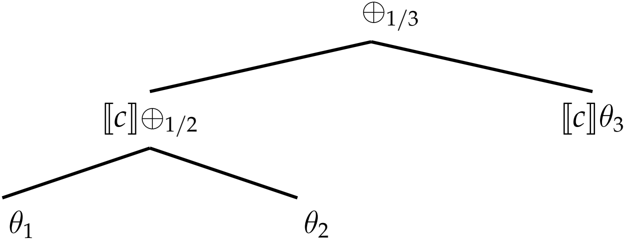
For any distribution Θ and booleans a, b:

$$Pr_{\Theta}(a) = Pr_{\Theta}(a \wedge b) + Pr_{\Theta}(a \wedge \neg b)$$

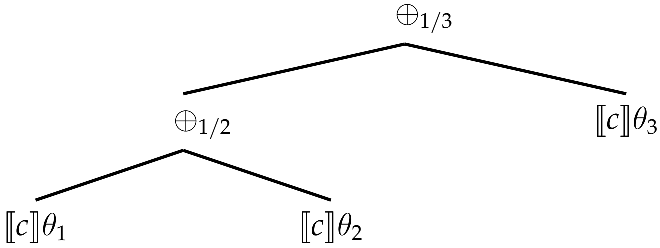
Primp Commands



Primp Commands

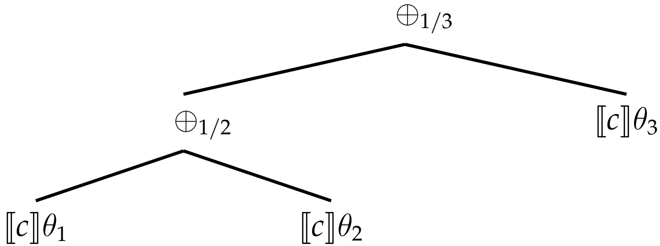


Primp Commands



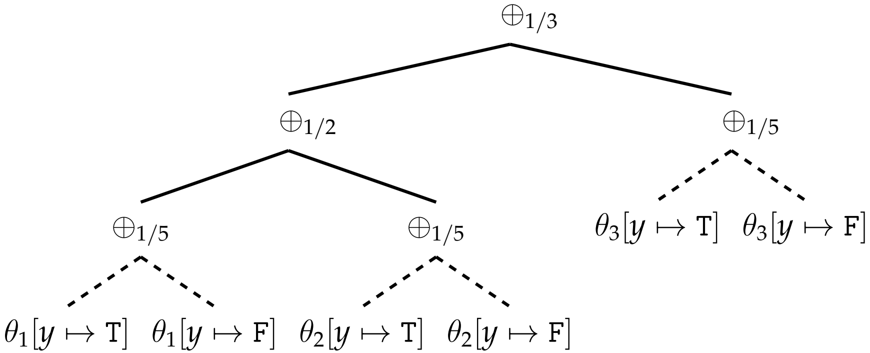
Primp Commands

$c \equiv y$ `:=` $\texttt{toss}(\frac{1}{5})$



Primp Commands

$$c \equiv y \;\texttt{:=}\; \texttt{toss}(\tfrac{1}{5})$$



VPHL: Hoare Logic

Definition: $\{P\}\ c\ \{Q\}$

$$\frac{P(\Theta) \qquad c \,/\, \Theta \Downarrow \Theta'}{Q(\Theta')}$$

Truth-functional assertions over full distributions

$$\mathcal{P}, \mathcal{Q} ::= Pr(\mathcal{B}) = p \mid Pr(\mathcal{B}) < p \mid Pr(\mathcal{B}) > p \\ \mid \mathcal{P} \wedge \mathcal{P} \mid \mathcal{P} \vee \mathcal{P}$$

Basic Rules

$$\frac{P' \to P \qquad \{P\}\ c\ \{Q\} \qquad Q \to Q'}{\{P'\}\ c\ \{Q'\}}\ \text{Consequence}$$

$$\text{Skip}\ \frac{}{\{P\}\ \texttt{skip}\ \{P\}} \qquad \frac{}{\{P[z \mapsto e]\}\ z\ \texttt{:=}\ e\ \{P\}}\ \text{Assign}$$

$$\frac{\{P\}\ c_1\ \{Q\} \qquad \{Q\}\ c_2\ \{R\}}{\{P\}\ c_1;\ c_2\ \{R\}}\ \text{Sequence}$$

The Toss Rule

$$\frac{y \text{ free in } P}{\{P\}\ y\ \texttt{:=}\ \texttt{toss}(p)\ \{P \triangleleft_p^y\}}\ \text{Toss}$$

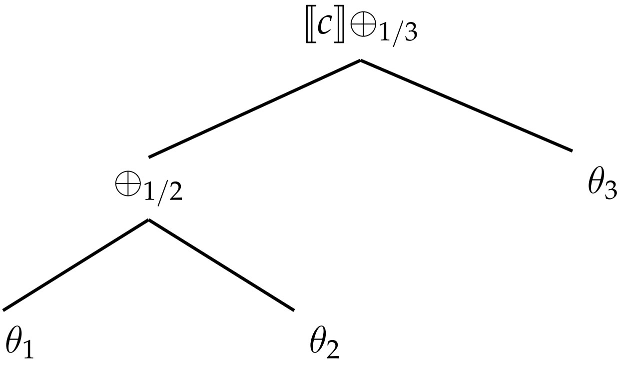
The Toss Rule

$$\frac{y \text{ free in } P}{\{P\}\ y\ \texttt{:=}\ \texttt{toss}(p)\ \{P \lhd_p^y\}}\ \text{Toss}$$

$$[Pr(b) = a]\lhd_p^y \equiv \qquad \begin{array}{l} Pr(b \wedge y) = pa \quad \wedge \\ Pr(b \wedge \neg y) = (1-p)a \end{array}$$

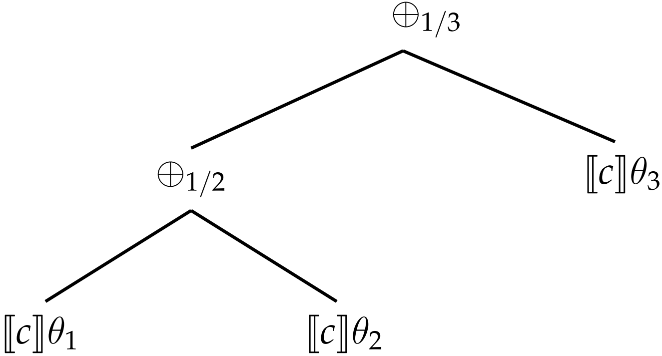
The If Rule

$c \equiv \texttt{if}\ y\ \texttt{then}\ c_1\ \texttt{else}\ c_2$



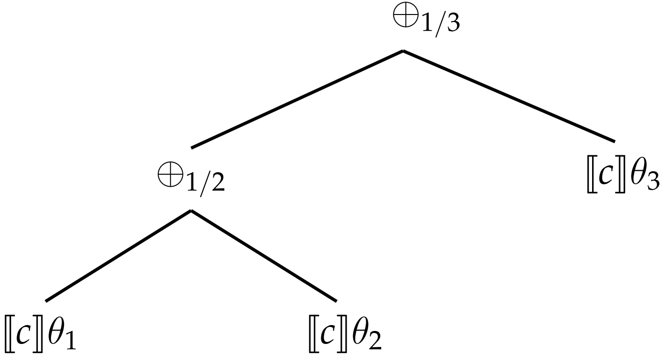
The If Rule

$$c \equiv \texttt{if}\; y \;\texttt{then}\; c_1 \;\texttt{else}\; c_2$$



The If Rule

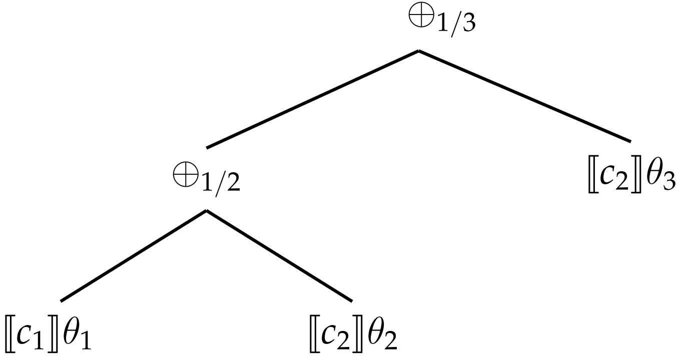
$c \equiv \texttt{if}\ y\ \texttt{then}\ c_1\ \texttt{else}\ c_2$



where $\theta_1(y) = \texttt{T}$, $\theta_2(y) = \texttt{F}$ and $\theta_3(y) = \texttt{F}$

The If Rule

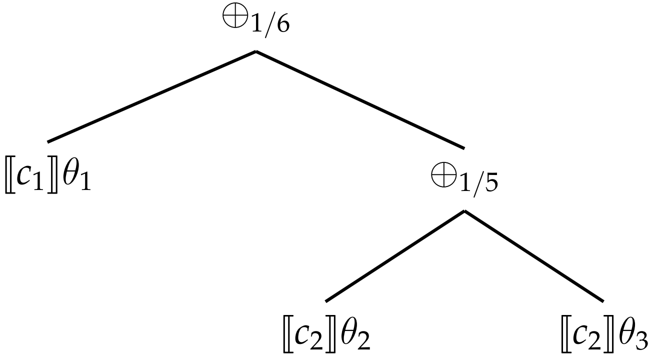
$c \equiv \texttt{if}\, y\, \texttt{then}\, c_1\, \texttt{else}\, c_2$



where $\theta_1(y) = \texttt{T}$, $\theta_2(y) = \texttt{F}$ and $\theta_3(y) = \texttt{F}$

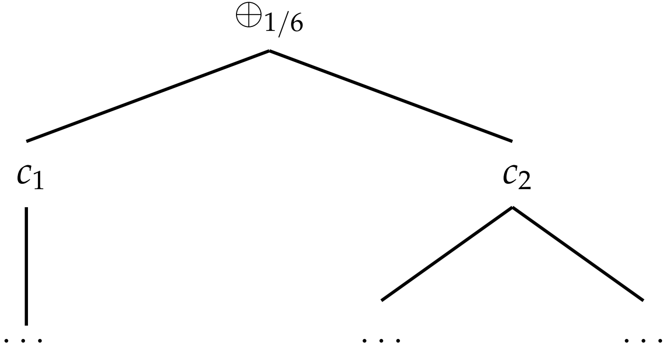
The If Rule

$$c \equiv \texttt{if}\, y\, \texttt{then}\, c_1\, \texttt{else}\, c_2$$



The If Rule

$$c \equiv \texttt{if}\ y\ \texttt{then}\ c_1\ \texttt{else}\ c_2$$



The If Rule

$c \equiv \texttt{if}\, y\, \texttt{then}\, c_1\, \texttt{else}\, c_2$

$$\oplus_{1/6}$$

$\{P_1 \wedge y\}\ c_1\ \{Q_1\}$ $\qquad$ $\{P_2 \wedge \neg y\}\ c_2\ \{Q_2\}$

$\dots$ $\qquad$ $\dots$ $\qquad$ $\dots$

The If Rule

$$c \equiv \texttt{if}\, y\, \texttt{then}\, c_1\, \texttt{else}\, c_2$$

$$\{Pr(y) = p \wedge P_1' \wedge P_2'\}\ c\ \{Q_1' \wedge Q_2'\}$$

$$\{P_1 \wedge y\}\ c_1\ \{Q_1\} \qquad\qquad \{P_2 \wedge \neg y\}\ c_2\ \{Q_2\}$$

$$\ldots \qquad\qquad \ldots \qquad\qquad \ldots$$

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
$$Pr(b) = a \Rightarrow Pr(b) = p * a$$
- 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
  $$Pr(b) = a \Rightarrow Pr(b) = p * a$$
- 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
  $$Pr(b) = a \Rightarrow Pr(b) = p * a$$
- Conditioning on the guard – we need to avoid conflict
  $$Pr(b) = p * a \Rightarrow Pr(b \wedge y) = p * a$$

Applying the IF Rule

UNIFORM(3)

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

end if

Applying the IF Rule

UNIFORM(3)

u_1 `:=` $\texttt{toss}(\frac{1}{3})$;
if u_1 **then**
    $\{Pr(3 = 3) = 1\}\ x \mathrel{:=} 3\ \{Pr(x = 3) = 1\}$
else
    u_2 `:=` $\texttt{toss}(\frac{1}{2})$;
    if u_2 **then**
        $\{Pr(2 = 2) = 1\}\ x \mathrel{:=} 2\ \{Pr(x = 2) = 1\}$
    else
        $\{Pr(1 = 1) = 1\}\ x \mathrel{:=} 1\ \{Pr(x = 1) = 1\}$
    end if

end if

Applying the IF Rule

UNIFORM(3)

$\{Pr(True) = 1\}\ u_1$ `:=` `toss`$(\frac{1}{3})$; $\{Pr(True \wedge u_1) = \frac{1}{3}\}$
if u_1 **then**
  $\{Pr(3 = 3) = 1\}\ x$ `:=` $3\ \{Pr(x = 3) = 1\}$
else
  $\{Pr(True) = 1\}\ u_2$ `:=` `toss`$(\frac{1}{2})$; $\{Pr(True \wedge u_2) = \frac{1}{2}\}$
  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

Applying the IF Rule

UNIFORM(3)

$\{Pr(True) = 1\}\ u_1$ `:=` $\texttt{toss}(\frac{1}{3})$; $\{Pr(u_1) = \frac{1}{3}\}$
if u_1 **then**
    $\{Pr(3 = 3) = 1\}\ x$ `:=` $3\ \{Pr(x = 3) = 1\}$
else
    $\{Pr(True) = 1\}\ u_2$ `:=` $\texttt{toss}(\frac{1}{2})$; $\{Pr(u_2) = \frac{1}{2}\}$
    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

Applying the IF Rule

UNIFORM(3)

$\{Pr(True) = 1\}\ u_1 := \texttt{toss}(\frac{1}{3});\ \{Pr(u_1) = \frac{1}{3}\}$
if u_1 **then**
  $\{Pr(3 = 3) = 1\}\ x := 3\ \{Pr(x = 3) = 1\}$
else
  $\{Pr(True) = 1\}\ u_2 := \texttt{toss}(\frac{1}{2});\ \{Pr(u_2) = \frac{1}{2}\}$
  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

Applying the IF Rule

UNIFORM(3)

$\{Pr(True) = 1\}\ u_1\ \texttt{:=}\ \texttt{toss}(\frac{1}{3});\ \{Pr(u_1) = \frac{1}{3}\}$
if u_1 **then**
  $\{Pr(3 = 3) = 1\}\ x\ \texttt{:=}\ 3\ \{Pr(x = 3) = 1\}$
else
  $\{Pr(True) = 1\}\ u_2\ \texttt{:=}\ \texttt{toss}(\frac{1}{2});\ \{Pr(u_2) = \frac{1}{2}\}$
  if u_2 **then**
    $\{Pr(2 = 2) = 1\}\ x\ \texttt{:=}\ 2\ \{Pr(x = 2) = 1\}$
  else
    $\{Pr(1 = 1) = 1\}\ x\ \texttt{:=}\ 1\ \{Pr(x = 1) = 1\}$
  end if
  $\{Pr(x = 2 \wedge u_2) = \frac{1}{2} \wedge Pr(x = 1 \wedge \neg u_2) = \frac{1}{2}\}$
end if

Applying the IF Rule

UNIFORM(3)

$\{Pr(True) = 1\}\ u_1$ `:=` $\texttt{toss}(\frac{1}{3});\ \{Pr(u_1) = \frac{1}{3}\}$
if u_1 **then**
  $\{Pr(3 = 3) = 1\}\ x$ `:=` $3\ \{Pr(x = 3) = 1\}$
else
  $\{Pr(True) = 1\}\ u_2$ `:=` $\texttt{toss}(\frac{1}{2});\ \{Pr(u_2) = \frac{1}{2}\}$
  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
  $\{Pr(x = 2) \geq \frac{1}{2} \wedge Pr(x = 1) \geq \frac{1}{2}\}$
end if

Applying the IF Rule

UNIFORM(3)

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

if u_1 **then**

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

else

$\quad \{Pr(True) = 1\}\ u_2 := \texttt{toss}(\frac{1}{2});\ \{Pr(u_2) = \frac{1}{2}\}$

$\quad$ **if** u_2 **then**

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

$\quad$ **else**

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

$\quad$ **end if**

$\quad \{Pr(x = 2) = \frac{1}{2} \wedge Pr(x = 1) = \frac{1}{2}\}$

end if

Applying the IF Rule

UNIFORM(3)

$\{Pr(True) = 1\}\ u_1 \texttt{ := } \texttt{toss}(\frac{1}{3});\ \{Pr(u_1) = \frac{1}{3}\}$
if u_1 **then**
  $\{Pr(3 = 3) = 1\}\ x \texttt{ := } 3\ \{Pr(x = 3) = 1\}$
else
  $\{Pr(True) = 1\}\ u_2 \texttt{ := } \texttt{toss}(\frac{1}{2});\ \{Pr(u_2) = \frac{1}{2}\}$
  if u_2 **then**
    $\{Pr(2 = 2) = 1\}\ x \texttt{ := } 2\ \{Pr(x = 2) = 1\}$
  else
    $\{Pr(1 = 1) = 1\}\ x \texttt{ := } 1\ \{Pr(x = 1) = 1\}$
  end if
  $\{Pr(x = 2) = \frac{1}{2} \wedge Pr(x = 1) = \frac{1}{2}\}$
end if

Applying the IF Rule

UNIFORM(3)

$\{Pr(True) = 1\}$ u_1 `:=` $\texttt{toss}(\frac{1}{3})$; $\{Pr(u_1) = \frac{1}{3}\}$
if u_1 **then**
    $\{Pr(3 = 3) = 1\}$ x `:=` 3 $\{Pr(x = 3) = 1\}$
else
    $\{Pr(True) = 1\}$ u_2 `:=` $\texttt{toss}(\frac{1}{2})$; $\{Pr(u_2) = \frac{1}{2}\}$
    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
    $\{Pr(x = 2) = \frac{1}{2} \wedge Pr(x = 1) = \frac{1}{2}\}$
end if
$\{Pr(x = 3) = \frac{1}{3} \wedge Pr(x = 2) = \frac{1}{3} \wedge Pr(x = 1) = \frac{1}{3}\}$

The While Rule

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

The *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**

$rabbit := \texttt{UNIFORM}(k)$
$hunter := \texttt{UNIFORM}(k)$
$caught := caught \vee (hunter = rabbit)$
$i := i + 1$

end while

Deterministic Invariant

Rabbit Hunting

while $i < n$ **do**

$\{\exists m \leq n : Pr(i = m) = 1 \wedge Pr(i < n) = 1\}$

$rabbit$:= $\texttt{UNIFORM}(k)$
$hunter$:= $\texttt{UNIFORM}(k)$
$caught$:= $caught \vee (hunter = rabbit)$
i := $i + 1$

end while

Deterministic Invariant

Rabbit Hunting

while $i < n$ **do**

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

$\{\exists m \leq n : Pr(i + 1 = m) = 1\}$

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

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

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

i := $i + 1$

end while

Deterministic Invariant

Rabbit Hunting

while $i < n$ **do**

$\{\exists m \leq n : Pr(i = m) = 1 \wedge Pr(i < n) = 1\} \rightarrow$

$\{\exists m \leq n : Pr(i + 1 = m) = 1\}$

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

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

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

i := $i + 1$

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

end while

Probabilistic Invariant

Rabbit Hunting

while $i < n$ **do**

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

$caught := caught \vee (hunter = rabbit)$
$i := i + 1$

end while

Probabilistic Invariant

Rabbit Hunting

while $i < n$ **do**

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

$rabbit$ `:=` `UNIFORM`(k)

$hunter$ `:=` `UNIFORM`(k)

$caught$ `:=` $caught \vee (hunter = rabbit)$

i `:=` $i + 1$

end while

Probabilistic Invariant

Rabbit Hunting

while $i < n$ **do**

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

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

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

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

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

$i := i + 1$

end while

Probabilistic Invariant

Rabbit Hunting

while $i < n$ **do**

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

$rabbit$:= `UNIFORM`(k)

$hunter$:= `UNIFORM`(k)

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

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

i := $i + 1$

end while

Probabilistic Invariant

Rabbit Hunting

while $i < n$ **do**

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

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

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

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

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

$i := i + 1$

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

end while

Catching Rabbits

Rabbit Hunting

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

while $i < n$ **do**
    $rabbit$ `:=` $\texttt{UNIFORM}(k)$
    $hunter$ `:=` $\texttt{UNIFORM}(k)$
    $caught$ `:=` $(hunter = rabbit) \vee caught$
    i `:=` $i + 1$
end while

Catching Rabbits

Rabbit Hunting

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

while $i < n$ **do**
  $rabbit$:= UNIFORM(k)
  $hunter$:= UNIFORM(k)
  $caught$:= $(hunter = rabbit) \vee caught$
  i := $i + 1$
end while

Catching Rabbits

Rabbit Hunting

$\{Pr(True) = 1\}$
$i := 0$
$caught := \texttt{F}$
$\{Pr(\neg caught) = 1 \land Pr(i = 0) = 1\} \rightarrow$
$\{Pr(\neg caught) = \left(\frac{k-1}{k}\right)^i \land \exists m \leq n : Pr(i = m) = 1\}$
while $i < n$ **do**
  $rabbit := \texttt{UNIFORM}(k)$
  $hunter := \texttt{UNIFORM}(k)$
  $caught := (hunter = rabbit) \lor caught$
  $i := i + 1$
end while

Catching Rabbits

Rabbit Hunting

$\{Pr(True) = 1\}$
i := 0
$caught$:= F
$\{Pr(\neg caught) = 1 \wedge Pr(i = 0) = 1\} \rightarrow$
$\{Pr(\neg caught) = \left(\frac{k-1}{k}\right)^i \wedge \exists m \leq n : Pr(i = m) = 1\}$
while $i < n$ **do**
    $rabbit$:= UNIFORM(k)
    $hunter$:= UNIFORM(k)
    $caught$:= $(hunter = rabbit) \vee caught$
    i := $i + 1$
end while
$\{Pr(\neg caught) = \left(\frac{k-1}{k}\right)^i \wedge \exists m \leq n : Pr(i = m) = 1 \wedge i \not< n\}$

Catching Rabbits

Rabbit Hunting

$\{Pr(True) = 1\}$
i := 0
$caught$:= F
$\{Pr(\neg caught) = 1 \wedge Pr(i = 0) = 1\} \rightarrow$
$\{Pr(\neg caught) = \left(\frac{k-1}{k}\right)^i \wedge \exists m \leq n : Pr(i = m) = 1\}$
while $i < n$ **do**
    $rabbit$:= UNIFORM(k)
    $hunter$:= UNIFORM(k)
    $caught$:= $(hunter = rabbit) \vee caught$
    i := $i + 1$
end while
$\{Pr(\neg caught) = \left(\frac{k-1}{k}\right)^i \wedge Pr(i = n) = 1\}$

Catching Rabbits

Rabbit Hunting

$\{Pr(True) = 1\}$
$i := 0$
$caught := \mathtt{F}$
$\{Pr(\neg caught) = 1 \wedge Pr(i = 0) = 1\} \rightarrow$
$\{Pr(\neg caught) = \left(\frac{k-1}{k}\right)^i \wedge \exists m \leq n : Pr(i = m) = 1\}$
while $i < n$ **do**
  $rabbit := \mathtt{UNIFORM}(k)$
  $hunter := \mathtt{UNIFORM}(k)$
  $caught := (hunter = rabbit) \vee caught$
  $i := i + 1$
end while
$\{Pr(\neg caught) = \left(\frac{k-1}{k}\right)^i \wedge Pr(i = n) = 1\} \rightarrow$
$\{Pr(caught) = 1 - \left(\frac{k-1}{k}\right)^n\}$

Probabilistic Termination

What about programs that terminate probabilistically?

What about programs that terminate probabilistically?

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

$$y \;\texttt{:=}\; \texttt{toss}\left(\tfrac{1}{2}\right);$$

$$\texttt{if}\ y\ \texttt{then}\ x \;\texttt{:=}\; 4\ \texttt{else}\ \texttt{loop}$$

$$\{\ \Pr(\text{x} = 4) = ?\ \}$$

Soundness

Theorem

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

Verified

https://github.com/rnrand/VPHL

$\mathcal{FIN}$

Thank You Questions?

https://github.com/rnrand/VPHL