Reasoning about Probabilistic Sequential Programs

Luís Cruz-Filipe
(joint work with R. Chadha, P. Mateus and A. Sernadas)

Security and Quantum Information Group
Instituto de Telecomunicações
Lisbon, Portugal

Logic and Computation Seminar
November 3, 2006

Motivation

- reasoning about non-deterministic programs
- new approach: truth values for formulas

Motivation

- reasoning about non-deterministic programs
- new approach: truth values for formulas

Motivation

- reasoning about non-deterministic programs
- new approach: truth values for formulas

1. The State Logic: EPPL
   - Language
   - Semantics
   - Calculus
   - Properties
2. The Programming Language
   - Syntax
   - Semantics
3. The Hoare Calculus
   - The calculus
   - Soundness
   - Completeness
4. Conclusions

1. The State Logic: EPPL
   - Language
   - Semantics
   - Calculus
   - Properties

2. The Programming Language
   - Syntax
   - Semantics

3. The Hoare Calculus
   - The calculus
   - Soundness
   - Completeness

4. Conclusions

1. The State Logic: EPPL
   - Language
   - Semantics
   - Calculus
   - Properties
2. The Programming Language
   - Syntax
   - Semantics
3. The Hoare Calculus
   - The calculus
   - Soundness
   - Completeness
4. Conclusions

1. The State Logic: EPPL
   - Language
   - Semantics
   - Calculus
   - Properties
2. The Programming Language
   - Syntax
   - Semantics
3. The Hoare Calculus
   - The calculus
   - Soundness
   - Completeness
4. Conclusions

Why EPPL

- two-layered design (exogenous approach)
- classical propositional logic at the lower level
- probabilistic logic built at the higher level

Why EPPL

- two-layered design (exogenous approach)
- classical propositional logic at the lower level
- probabilistic logic built at the higher level

Why EPPL

- two-layered design (exogenous approach)
- classical propositional logic at the lower level
- probabilistic logic built at the higher level

Real-closed fields

Definition

A *real closed field* is an ordered field $\mathcal{K}$ where:

- every non-negative element of the K has a square root in K;
- every polynomial of odd degree with coefficients in K has at least one solution in K.

Example

- the set of real numbers with the usual multiplication, addition and order relation;
- the set of computable real numbers with the same operations.

Real-closed fields

Definition

A *real closed field* is an ordered field $\mathcal{K}$ where:

- every non-negative element of the K has a square root in K;
- every polynomial of odd degree with coefficients in K has at least one solution in K.

Example

- the set of real numbers with the usual multiplication, addition and order relation;
- the set of computable real numbers with the same operations.

Real-closed fields

Definition

A *real closed field* is an ordered field $\mathcal{K}$ where:

- every non-negative element of the K has a square root in K;
- every polynomial of odd degree with coefficients in K has at least one solution in K.

Example

- the set of real numbers with the usual multiplication, addition and order relation;
- the set of computable real numbers with the same operations.

Real-closed fields

Definition

A *real closed field* is an ordered field $\mathcal{K}$ where:

- every non-negative element of the K has a square root in K;
- every polynomial of odd degree with coefficients in K has at least one solution in K.

Example

- the set of real numbers with the usual multiplication, addition and order relation;
- the set of computable real numbers with the same operations.

Real-closed fields

Definition

A *real closed field* is an ordered field $\mathcal{K}$ where:

- every non-negative element of the K has a square root in K;
- every polynomial of odd degree with coefficients in K has at least one solution in K.

Example

- the set of real numbers with the usual multiplication, addition and order relation;
- the set of computable real numbers with the same operations.

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \ldots, m-1\}$ of indices
- registers $\mathbf{xM} = \{\mathbf{xm}_k \mid k \in \mathbf{m}\}$ containing real values
- registers $\mathbf{bM} = \{\mathbf{bm}_k \mid k \in \mathbf{m}\}$ containing booleans
- variables $B = \{B_k : k \in \mathbb{N}\}$ ranging over truth values
- variables $X = \{X_k : k \in \mathbb{N}\}$ ranging over D
- real-closed field $\mathcal{K}$ with set of algebraic numbers $\mathcal{A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over $\mathcal{K}$

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \ldots, m-1\}$ of indices
- registers $\mathbf{xM} = \{\mathbf{xm}_k \mid k \in \mathbf{m}\}$ containing real values
- registers $\mathbf{bM} = \{\mathbf{bm}_k \mid k \in \mathbf{m}\}$ containing booleans
- variables $B = \{B_k : k \in \mathbb{N}\}$ ranging over truth values
- variables $X = \{X_k : k \in \mathbb{N}\}$ ranging over D
- real-closed field $\mathcal{K}$ with set of algebraic numbers $\mathcal{A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over $\mathcal{K}$

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \ldots, m-1\}$ of indices
- registers $\mathbf{xM} = \{\mathbf{xm}_k \mid k \in \mathbf{m}\}$ containing real values
- registers $\mathbf{bM} = \{\mathbf{bm}_k \mid k \in \mathbf{m}\}$ containing booleans
- variables $B = \{B_k : k \in \mathbb{N}\}$ ranging over truth values
- variables $X = \{X_k : k \in \mathbb{N}\}$ ranging over D
- real-closed field $\mathcal{K}$ with set of algebraic numbers $\mathcal{A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over $\mathcal{K}$

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \ldots, m-1\}$ of indices
- registers $\mathbf{xM} = \{\mathbf{xm}_k \mid k \in \mathbf{m}\}$ containing real values
- registers $\mathbf{bM} = \{\mathbf{bm}_k \mid k \in \mathbf{m}\}$ containing booleans
- variables $B = \{B_k : k \in \mathbb{N}\}$ ranging over truth values
- variables $X = \{X_k : k \in \mathbb{N}\}$ ranging over D
- real-closed field $\mathcal{K}$ with set of algebraic numbers $\mathcal{A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over $\mathcal{K}$

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \ldots, m-1\}$ of indices
- registers $\mathbf{xM} = \{\mathbf{xm}_k \mid k \in \mathbf{m}\}$ containing real values
- registers $\mathbf{bM} = \{\mathbf{bm}_k \mid k \in \mathbf{m}\}$ containing booleans
- variables $B = \{B_k : k \in \mathbb{N}\}$ ranging over truth values
- variables $X = \{X_k : k \in \mathbb{N}\}$ ranging over D
- real-closed field $\mathcal{K}$ with set of algebraic numbers $\mathcal{A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over $\mathcal{K}$

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \ldots, m-1\}$ of indices
- registers $\mathbf{xM} = \{\mathbf{xm}_k \mid k \in \mathbf{m}\}$ containing real values
- registers $\mathbf{bM} = \{\mathbf{bm}_k \mid k \in \mathbf{m}\}$ containing booleans
- variables $B = \{B_k : k \in \mathbb{N}\}$ ranging over truth values
- variables $X = \{X_k : k \in \mathbb{N}\}$ ranging over D
- real-closed field $\mathcal{K}$ with set of algebraic numbers $\mathcal{A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over $\mathcal{K}$

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \ldots, m-1\}$ of indices
- registers $\mathbf{xM} = \{\mathbf{xm}_k \mid k \in \mathbf{m}\}$ containing real values
- registers $\mathbf{bM} = \{\mathbf{bm}_k \mid k \in \mathbf{m}\}$ containing booleans
- variables $B = \{B_k : k \in \mathbb{N}\}$ ranging over truth values
- variables $X = \{X_k : k \in \mathbb{N}\}$ ranging over D
- real-closed field $\mathcal{K}$ with set of algebraic numbers $\mathcal{A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over $\mathcal{K}$

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \ldots, m-1\}$ of indices
- registers $\mathbf{xM} = \{\mathbf{xm}_k \mid k \in \mathbf{m}\}$ containing real values
- registers $\mathbf{bM} = \{\mathbf{bm}_k \mid k \in \mathbf{m}\}$ containing booleans
- variables $B = \{B_k : k \in \mathbb{N}\}$ ranging over truth values
- variables $X = \{X_k : k \in \mathbb{N}\}$ ranging over D
- real-closed field $\mathcal{K}$ with set of algebraic numbers $\mathcal{A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over $\mathcal{K}$

Language

Real terms (with $c \in D$)

$$t ::= c \mid \mathbf{xm} \mid X \mid (t + t) \mid (t\, t)$$

Classical state formulas

$$\gamma ::= \mathbf{bm} \mid B \mid (t \leq t) \mid \mathrm{ff} \mid (\gamma \Rightarrow \gamma)$$

Probability terms (with $r \in \mathcal{A}$)

$$p ::= r \mid y \mid \widetilde{r} \mid (\textstyle\int \gamma) \mid (p + p) \mid (p\, p)$$

Probabilistic state formulas

$$\eta ::= (p \leq p) \mid \mathrm{fff} \mid (\eta \supset \eta)$$

Language

Real terms (with $c \in D$)

$$t ::= c \mid \mathbf{xm} \mid X \mid (t + t) \mid (t\, t)$$

Classical state formulas

$$\gamma ::= \mathbf{bm} \mid B \mid (t \leq t) \mid \mathrm{ff} \mid (\gamma \Rightarrow \gamma)$$

Probability terms (with $r \in \mathcal{A}$)

$$p ::= r \mid y \mid \tilde{r} \mid (\textstyle\int \gamma) \mid (p + p) \mid (p\, p)$$

Probabilistic state formulas

$$\eta ::= (p \leq p) \mid \mathrm{fff} \mid (\eta \supset \eta)$$

Language

Real terms (with $c \in D$)

$$t ::= c \mid \mathbf{xm} \mid X \mid (t + t) \mid (t\, t)$$

Classical state formulas

$$\gamma ::= \mathbf{bm} \mid B \mid (t \leq t) \mid \mathrm{ff} \mid (\gamma \Rightarrow \gamma)$$

Probability terms (with $r \in \mathcal{A}$)

$$p ::= r \mid y \mid \widetilde{r} \mid (\textstyle\int \gamma) \mid (p + p) \mid (p\, p)$$

Probabilistic state formulas

$$\eta ::= (p \leq p) \mid \mathrm{fff} \mid (\eta \supset \eta)$$

Language

Real terms (with $c \in D$)

$$t ::= c \mid \mathbf{xm} \mid X \mid (t + t) \mid (t\, t)$$

Classical state formulas

$$\gamma ::= \mathbf{bm} \mid B \mid (t \leq t) \mid \mathrm{ff} \mid (\gamma \Rightarrow \gamma)$$

Probability terms (with $r \in \mathcal{A}$)

$$p ::= r \mid y \mid \widetilde{r} \mid (\textstyle\int \gamma) \mid (p + p) \mid (p\, p)$$

Probabilistic state formulas

$$\eta ::= (p \leq p) \mid \mathrm{fff} \mid (\eta \supset \eta)$$

Language

Real terms (with $c \in D$)

$$t ::= c \mid \mathbf{xm} \mid X \mid (t + t) \mid (t\, t)$$

Classical state formulas

$$\gamma ::= \mathbf{bm} \mid B \mid (t \leq t) \mid \mathrm{ff} \mid (\gamma \Rightarrow \gamma)$$

Probability terms (with $r \in \mathcal{A}$)

$$p ::= r \mid y \mid \widetilde{r} \mid (\textstyle\int \gamma) \mid (p + p) \mid (p\, p)$$

Probabilistic state formulas

$$\eta ::= (p \leq p) \mid \mathrm{fff} \mid (\eta \supset \eta)$$

Language

Real terms (with $c \in D$)

$$t ::= c \mid \mathbf{xm} \mid X \mid (t + t) \mid (t\, t)$$

Classical state formulas

$$\gamma ::= \mathbf{bm} \mid B \mid (t \leq t) \mid \mathrm{ff} \mid (\gamma \Rightarrow \gamma)$$

Probability terms (with $r \in \mathcal{A}$)

$$p ::= r \mid y \mid \widetilde{r} \mid (\textstyle\int \gamma) \mid (p + p) \mid (p\, p)$$

Probabilistic state formulas

$$\eta ::= (p \leq p) \mid \mathrm{fff} \mid (\eta \supset \eta)$$

Language

Real terms (with $c \in D$)

$$t ::= c \mid \mathbf{xm} \mid X \mid (t + t) \mid (t\, t)$$

Classical state formulas

$$\gamma ::= \mathbf{bm} \mid B \mid (t \leq t) \mid \mathrm{ff} \mid (\gamma \Rightarrow \gamma)$$

Probability terms (with $r \in \mathcal{A}$)

$$p ::= r \mid y \mid \widetilde{r} \mid (\textstyle\int \gamma) \mid (p + p) \mid (p\, p)$$

Probabilistic state formulas

$$\eta ::= (p \leq p) \mid \mathrm{fff} \mid (\eta \supset \eta)$$

Language

Real terms (with $c \in D$)

$$t ::= c \mid \mathbf{xm} \mid X \mid (t + t) \mid (t\, t)$$

Classical state formulas

$$\gamma ::= \mathbf{bm} \mid B \mid (t \leq t) \mid \mathrm{ff} \mid (\gamma \Rightarrow \gamma)$$

Probability terms (with $r \in \mathcal{A}$)

$$p ::= r \mid y \mid \widetilde{r} \mid (\textstyle\int \gamma) \mid (p + p) \mid (p\, p)$$

Probabilistic state formulas

$$\eta ::= (p \leq p) \mid \mathrm{fff} \mid (\eta \supset \eta)$$

Useful notions

Definition

An *analytical term* is a term without occurrences of probability terms.

$$a ::= r \mid y \mid \widetilde{r} \mid (a + a) \mid (aa)$$

Definition

An *analytical formula* is a formula without occurrences of probability terms.

$$\kappa ::= (a \leq a) \mid \mathrm{fff} \mid (\kappa \supset \kappa)$$

$(\Box\gamma)$ stands for the formula $((\int\gamma) = (\int \mathrm{tt}))$
$(\Diamond\gamma)$ stands for the formula $(\ominus(\Box(\neg\gamma)))$

Useful notions

Definition

An *analytical term* is a term without occurrences of probability terms.

$$a ::= r \mid y \mid \widetilde{r} \mid (a + a) \mid (aa)$$

Definition

An *analytical formula* is a formula without occurrences of probability terms.

$$\kappa ::= (a \leq a) \mid \mathrm{fff} \mid (\kappa \supset \kappa)$$

$(\Box\gamma)$ stands for the formula $((\int\gamma) = (\int \mathrm{tt}))$
$(\Diamond\gamma)$ stands for the formula $(\ominus(\Box(\neg\gamma)))$

Useful notions

Definition

An *analytical term* is a term without occurrences of probability terms.

$$a ::= r \mid y \mid \widetilde{r} \mid (a + a) \mid (aa)$$

Definition

An *analytical formula* is a formula without occurrences of probability terms.

$$\kappa ::= (a \leq a) \mid \mathrm{fff} \mid (\kappa \supset \kappa)$$

$(\Box\gamma)$ stands for the formula $((\int\gamma) = (\int \mathrm{tt}))$
$(\Diamond\gamma)$ stands for the formula $(\ominus(\Box(\neg\gamma)))$

Valuations

Definition

A *valuation* is a map that provides values to the memory variables and corresponding logical variables. The set of all valuations is denoted by $\mathcal{V}$.

The denotation $[\![t]\!]_v$ of a real term t given a valuation v is defined inductively as expected.
Satisfaction $v \Vdash_c \gamma$ of a classical state formula γ by a valuation v is also defined inductively as usual.

Definition

The *extent* of a classical state formula γ in a set V of valuations is

$$|\gamma|_V = \{v \in V \mid v \Vdash_c \gamma\}.$$

Valuations

Definition

A *valuation* is a map that provides values to the memory variables and corresponding logical variables. The set of all valuations is denoted by $\mathcal{V}$.

The denotation $[\![t]\!]_v$ of a real term t given a valuation v is defined inductively as expected.

Satisfaction $v \Vdash_c \gamma$ of a classical state formula γ by a valuation v is also defined inductively as usual.

Definition

The *extent* of a classical state formula γ in a set V of valuations is

$$|\gamma|_V = \{v \in V \mid v \Vdash_c \gamma\}.$$

Valuations

Definition

A *valuation* is a map that provides values to the memory variables and corresponding logical variables. The set of all valuations is denoted by $\mathcal{V}$.

The denotation $[\![t]\!]_v$ of a real term t given a valuation v is defined inductively as expected.
Satisfaction $v \Vdash_c \gamma$ of a classical state formula γ by a valuation v is also defined inductively as usual.

Definition

The *extent* of a classical state formula γ in a set V of valuations is

$$|\gamma|_V = \{v \in V \mid v \Vdash_c \gamma\}.$$

Valuations

Definition

A *valuation* is a map that provides values to the memory variables and corresponding logical variables. The set of all valuations is denoted by $\mathcal{V}$.

The denotation $[\![t]\!]_v$ of a real term t given a valuation v is defined inductively as expected.
Satisfaction $v \Vdash_c \gamma$ of a classical state formula γ by a valuation v is also defined inductively as usual.

Definition

The *extent* of a classical state formula γ in a set V of valuations is

$$|\gamma|_V = \{v \in V \mid v \Vdash_c \gamma\}.$$

Measure functions

Definition

A finitely additive, discrete and bounded $\mathcal{K}$-measure μ on a set X is a map from X to $\mathcal{K}^+$ such that:

- $\mu(\emptyset) = 0$;
- if $U_1 \cap U_2 = \emptyset$, then $\mu(U_1 \cup U_2) = \mu(U_1) + \mu(U_2)$.

A $\mathcal{K}$-measure μ over X is a *probability measure* if $\mu(X) = 1$.

Measure functions

Definition

A finitely additive, discrete and bounded $\mathcal{K}$-measure μ on a set X is a map from X to $\mathcal{K}^+$ such that:

- $\mu(\emptyset) = 0$;
- if $U_1 \cap U_2 = \emptyset$, then $\mu(U_1 \cup U_2) = \mu(U_1) + \mu(U_2)$.

A $\mathcal{K}$-measure μ over X is a *probability measure* if $\mu(X) = 1$.

Measure functions

Definition

A finitely additive, discrete and bounded $\mathcal{K}$-measure μ on a set X is a map from X to $\mathcal{K}^+$ such that:

- $\mu(\emptyset) = 0$;
- if $U_1 \cap U_2 = \emptyset$, then $\mu(U_1 \cup U_2) = \mu(U_1) + \mu(U_2)$.

A $\mathcal{K}$-measure μ over X is a *probability measure* if $\mu(X) = 1$.

Measure functions

Definition

A finitely additive, discrete and bounded $\mathcal{K}$-measure μ on a set X is a map from X to $\mathcal{K}^+$ such that:

- $\mu(\emptyset) = 0$;
- if $U_1 \cap U_2 = \emptyset$, then $\mu(U_1 \cup U_2) = \mu(U_1) + \mu(U_2)$.

A $\mathcal{K}$-measure μ over X is a *probability measure* if $\mu(X) = 1$.

Interpretation

Definition

A *generalized probabilistic state* consists of a real closed field $\mathcal{K}$ and a finitely additive, discrete and finite $\mathcal{K}$-measure over $\wp\mathcal{V}$.

Given a classical formula γ we define

$$\mu_\gamma = \lambda V.\mu(|\gamma|_V).$$

Definition

Given a real closed field $\mathcal{K}$, a $\mathcal{K}$*-assignment* is a map $\rho : Y \to \mathcal{K}$.

Interpretation

Definition

A *generalized probabilistic state* consists of a real closed field $\mathcal{K}$ and a finitely additive, discrete and finite $\mathcal{K}$-measure over $\wp\mathcal{V}$.

Given a classical formula γ we define

$$\mu_\gamma = \lambda V.\mu(|\gamma|_V).$$

Definition

Given a real closed field $\mathcal{K}$, a $\mathcal{K}$*-assignment* is a map $\rho : Y \to \mathcal{K}$.

Interpretation

Definition

A *generalized probabilistic state* consists of a real closed field $\mathcal{K}$ and a finitely additive, discrete and finite $\mathcal{K}$-measure over $\wp\mathcal{V}$.

Given a classical formula γ we define

$$\mu_\gamma = \lambda V.\mu(|\gamma|_V).$$

Definition

Given a real closed field $\mathcal{K}$, a $\mathcal{K}$-*assignment* is a map $\rho : Y \to \mathcal{K}$.

Interpretation

Denotation of probability terms

$$\begin{aligned}
[\![r]\!]^{\rho}_{K,\mu} &= r \\
[\![y]\!]^{\rho}_{K,\mu} &= \rho(y) \\
[\![(\textstyle\int\gamma)]\!]^{\rho}_{K,\mu} &= \mu(|\gamma|\nu) \\
[\![p_1 + p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} + [\![p_2]\!]^{\rho}_{K,\mu} \\
[\![p_1 p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} \times [\![p_2]\!]^{\rho}_{K,\mu}
\end{aligned}$$

Satisfaction of probabilistic formulas

$$(K,\mu)\rho \Vdash (p_1 \leq p_2) \quad \text{iff} \quad [\![p_1]\!]^{\rho}_{K,\mu} \leq [\![p_2]\!]^{\rho}_{K,\mu}$$

$$(K,\mu)\rho \nVdash \text{fff}$$

$$(K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad \text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho \nVdash \eta_1$$

Interpretation

Denotation of probability terms

$$
\begin{aligned}
[\![r]\!]^{\rho}_{K,\mu} &= r \\
[\![y]\!]^{\rho}_{K,\mu} &= \rho(y) \\
[\![(\textstyle\int\gamma)]\!]^{\rho}_{K,\mu} &= \mu(|\gamma|_{\mathcal{V}}) \\
[\![p_1 + p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} + [\![p_2]\!]^{\rho}_{K,\mu} \\
[\![p_1 p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} \times [\![p_2]\!]^{\rho}_{K,\mu}
\end{aligned}
$$

Satisfaction of probabilistic formulas

$$
\begin{aligned}
(K,\mu)\rho \Vdash (p_1 \leq p_2) \quad &\text{iff} \quad [\![p_1]\!]^{\rho}_{K,\mu} \leq [\![p_2]\!]^{\rho}_{K,\mu} \\
(K,\mu)\rho \nVdash \mathrm{fff} \quad & \\
(K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad &\text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho \nVdash \eta_1
\end{aligned}
$$

Interpretation

Denotation of probability terms

$$
\begin{aligned}
[\![r]\!]^{\rho}_{K,\mu} &= r \\
[\![y]\!]^{\rho}_{K,\mu} &= \rho(y) \\
[\![(\textstyle\int\gamma)]\!]^{\rho}_{K,\mu} &= \mu(|\gamma|_{\mathcal{V}}) \\
[\![p_1 + p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} + [\![p_2]\!]^{\rho}_{K,\mu} \\
[\![p_1 p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} \times [\![p_2]\!]^{\rho}_{K,\mu}
\end{aligned}
$$

Satisfaction of probabilistic formulas

$$
\begin{aligned}
(K,\mu)\rho \Vdash (p_1 \leq p_2) \quad &\text{iff} \quad [\![p_1]\!]^{\rho}_{K,\mu} \leq [\![p_2]\!]^{\rho}_{K,\mu} \\
(K,\mu)\rho \not\Vdash \mathrm{fff}& \\
(K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad &\text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho \not\Vdash \eta_1
\end{aligned}
$$

Interpretation

Denotation of probability terms

$$\begin{aligned}
[\![r]\!]^{\rho}_{K,\mu} &= r \\
[\![y]\!]^{\rho}_{K,\mu} &= \rho(y) \\
[\![(\textstyle\int\gamma)]\!]^{\rho}_{K,\mu} &= \mu(|\gamma|_{\mathcal{V}}) \\
[\![p_1 + p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} + [\![p_2]\!]^{\rho}_{K,\mu} \\
[\![p_1 p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} \times [\![p_2]\!]^{\rho}_{K,\mu}
\end{aligned}$$

Satisfaction of probabilistic formulas

$$\begin{aligned}
(K,\mu)\rho \Vdash (p_1 \leq p_2) \quad &\text{iff} \quad [\![p_1]\!]^{\rho}_{K,\mu} \leq [\![p_2]\!]^{\rho}_{K,\mu} \\
(K,\mu)\rho \nVdash \text{fff} \quad & \\
(K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad &\text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho \nVdash \eta_1
\end{aligned}$$

Interpretation

Denotation of probability terms

$$\begin{aligned}
[\![r]\!]^{\rho}_{K,\mu} &= r \\
[\![y]\!]^{\rho}_{K,\mu} &= \rho(y) \\
[\![(\textstyle\int \gamma)]\!]^{\rho}_{K,\mu} &= \mu(|\gamma|_{\mathcal{V}}) \\
[\![p_1 + p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} + [\![p_2]\!]^{\rho}_{K,\mu} \\
[\![p_1 p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} \times [\![p_2]\!]^{\rho}_{K,\mu}
\end{aligned}$$

Satisfaction of probabilistic formulas

$$\begin{aligned}
(K,\mu)\rho \Vdash (p_1 \leq p_2) \quad &\text{iff} \quad [\![p_1]\!]^{\rho}_{K,\mu} \leq [\![p_2]\!]^{\rho}_{K,\mu} \\
(K,\mu)\rho \nVdash \text{fff} & \\
(K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad &\text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho \nVdash \eta_1
\end{aligned}$$

Interpretation

Denotation of probability terms

$$
\begin{aligned}
[\![r]\!]^{\rho}_{K,\mu} &= r \\
[\![y]\!]^{\rho}_{K,\mu} &= \rho(y) \\
[\![(\textstyle\int\gamma)]\!]^{\rho}_{K,\mu} &= \mu(|\gamma|_{\mathcal{V}}) \\
[\![p_1 + p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} + [\![p_2]\!]^{\rho}_{K,\mu} \\
[\![p_1 p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} \times [\![p_2]\!]^{\rho}_{K,\mu}
\end{aligned}
$$

Satisfaction of probabilistic formulas

$$(K,\mu)\rho \Vdash (p_1 \leq p_2) \quad \text{iff} \quad [\![p_1]\!]^{\rho}_{K,\mu} \leq [\![p_2]\!]^{\rho}_{K,\mu}$$

$$(K,\mu)\rho \not\Vdash \mathrm{fff}$$

$$(K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad \text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho \not\Vdash \eta_1$$

Interpretation

Denotation of probability terms

$$\begin{aligned}
[\![r]\!]^{\rho}_{K,\mu} &= r \\
[\![y]\!]^{\rho}_{K,\mu} &= \rho(y) \\
[\![(\textstyle\int\gamma)]\!]^{\rho}_{K,\mu} &= \mu(|\gamma|_{\mathcal{V}}) \\
[\![p_1 + p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} + [\![p_2]\!]^{\rho}_{K,\mu} \\
[\![p_1 p_2]\!]^{\rho}_{K,\mu} &= [\![p_1]\!]^{\rho}_{K,\mu} \times [\![p_2]\!]^{\rho}_{K,\mu}
\end{aligned}$$

Satisfaction of probabilistic formulas

$$\begin{aligned}
(K,\mu)\rho \Vdash (p_1 \leq p_2) \quad &\text{iff} \quad [\![p_1]\!]^{\rho}_{K,\mu} \leq [\![p_2]\!]^{\rho}_{K,\mu} \\
(K,\mu)\rho \nVdash \text{fff}& \\
(K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad &\text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho \nVdash \eta_1
\end{aligned}$$

Auxiliary notions

Definition

A classical state formula γ is said to be *valid* if it holds for all valuations $v \in \mathcal{V}$.

Example

$((\mathbf{x1} \leq \mathbf{x2}) \wedge (\mathbf{x1} > 0)) \Rightarrow (\mathbf{x1}^2 \leq \mathbf{x2}^2)$

Since D is finite, the set of valid classical state formulas is recursive.

Auxiliary notions

Definition

A classical state formula γ is said to be *valid* if it holds for all valuations $v \in \mathcal{V}$.

Example

$$((\mathbf{x1} \leq \mathbf{x2}) \wedge (\mathbf{x1} > 0)) \Rightarrow (\mathbf{x1}^2 \leq \mathbf{x2}^2)$$

Since D is finite, the set of valid classical state formulas is recursive.

Auxiliary notions

Definition

A classical state formula γ is said to be *valid* if it holds for all valuations $v \in \mathcal{V}$.

Example

$$((\mathbf{x1} \leq \mathbf{x2}) \wedge (\mathbf{x1} > 0)) \Rightarrow (\mathbf{x1}^2 \leq \mathbf{x2}^2)$$

Since D is finite, the set of valid classical state formulas is recursive.

Auxiliary notions

Definition

A probabilistic formula η is said to be a *probabilistic tautology* if there exists a propositional tautology β such that η is obtained from β by replacing all occurrences of $\perp$ by fff, $\rightarrow$ by $\supset$ and each propositional symbol (uniformly) by a probabilistic state formula.

Example

$((\int(x_1 \leq x_2)) < 1) \supset (((\int(x_1 \leq x_2)) < 1) \cap \mathrm{tt})$

Auxiliary notions

Definition

A probabilistic formula η is said to be a *probabilistic tautology* if there exists a propositional tautology β such that η is obtained from β by replacing all occurrences of $\perp$ by fff, $\rightarrow$ by $\supset$ and each propositional symbol (uniformly) by a probabilistic state formula.

Example

$$((\int(x_1 \leq x_2)) < 1) \supset (((\int(x_1 \leq x_2)) < 1) \cap \mathrm{tt})$$

Auxiliary notions

Definition

An analytical formula κ is a *valid analytical formula* if κ is satisfied by ρ for any real closed field $\mathcal{K}$ and any $\mathcal{K}$-assignment ρ.

Example

$((y_1 \leq y_2) \wedge (y_1 > 0)) \supset (y_1^2 \leq y_2^2)$

The set of valid analytical formulas is decidable.

Auxiliary notions

Definition

An analytical formula κ is a *valid analytical formula* if κ is satisfied by ρ for any real closed field $\mathcal{K}$ and any $\mathcal{K}$-assignment ρ.

Example

$$((y_1 \leq y_2) \wedge (y_1 > 0)) \supset (y_1^2 \leq y_2^2)$$

The set of valid analytical formulas is decidable.

Auxiliary notions

Definition

An analytical formula κ is a *valid analytical formula* if κ is satisfied by ρ for any real closed field $\mathcal{K}$ and any $\mathcal{K}$-assignment ρ.

Example

$((y_1 \leq y_2) \wedge (y_1 > 0)) \supset (y_1^2 \leq y_2^2)$

The set of valid analytical formulas is decidable.

Calculus

Axioms

[CTaut] $\vdash$ $(\Box\gamma)$ for each valid state formula γ

[PTaut] $\vdash$ η for each probabilistic tautology η

[RCF] $\vdash$ $\kappa^{\vec{y}}_{\vec{p}}$ for any valid analytical formula κ

[Meas∅] $\vdash$ $((\int \mathrm{ff}) = 0)$

[FAdd] $\vdash$ $(((\int(\gamma_1 \wedge \gamma_2)) = 0) \supset ((\int(\gamma_1 \vee \gamma_2)) = (\int \gamma_1) + (\int \gamma_2)))$

[Mon] $\vdash$ $((\Box(\gamma_1 \Rightarrow \gamma_2)) \supset ((\int \gamma_1) \leq (\int \gamma_2)))$

Inference rule

[PMP] $\eta_1, (\eta_1 \supset \eta_2) \vdash \eta_2$

Calculus

Axioms

$$[\mathbf{CTaut}] \quad \vdash \quad (\Box\gamma) \text{ for each valid state formula } \gamma$$

$$[\mathbf{PTaut}] \quad \vdash \quad \eta \text{ for each probabilistic tautology } \eta$$

$$[\mathbf{RCF}] \quad \vdash \quad \kappa^{\vec{y}}_{\vec{p}} \text{ for any valid analytical formula } \kappa$$

$$[\mathbf{Meas}\emptyset] \quad \vdash \quad ((\textstyle\int \mathrm{ff}) = 0)$$

$$[\mathbf{FAdd}] \quad \vdash \quad (((\textstyle\int(\gamma_1 \wedge \gamma_2)) = 0) \supset ((\textstyle\int(\gamma_1 \vee \gamma_2)) = (\textstyle\int\gamma_1) + (\textstyle\int\gamma_2)))$$

$$[\mathbf{Mon}] \quad \vdash \quad ((\Box(\gamma_1 \Rightarrow \gamma_2)) \supset ((\textstyle\int\gamma_1) \leq (\textstyle\int\gamma_2)))$$

Inference rule

$$[\mathbf{PMP}] \quad \eta_1, (\eta_1 \supset \eta_2) \vdash \eta_2$$

Calculus

Axioms

[CTaut] $\vdash$ $(\Box\gamma)$ for each valid state formula γ

[PTaut] $\vdash$ η for each probabilistic tautology η

[RCF] $\vdash$ $\kappa^{\vec{y}}_{\vec{p}}$ for any valid analytical formula κ

[Meas∅] $\vdash$ $((\int \mathrm{ff}) = 0)$

[FAdd] $\vdash$ $(((\int(\gamma_1 \wedge \gamma_2)) = 0) \supset ((\int(\gamma_1 \vee \gamma_2)) = (\int \gamma_1) + (\int \gamma_2)))$

[Mon] $\vdash$ $((\Box(\gamma_1 \Rightarrow \gamma_2)) \supset ((\int \gamma_1) \leq (\int \gamma_2)))$

Inference rule

[PMP] $\eta_1, (\eta_1 \supset \eta_2) \vdash \eta_2$

Calculus

Axioms

[CTaut] $\vdash$ $(\Box\gamma)$ for each valid state formula γ

[PTaut] $\vdash$ η for each probabilistic tautology η

[RCF] $\vdash$ $\kappa_{\vec{p}}^{\vec{y}}$ for any valid analytical formula κ

[Meas∅] $\vdash$ $((\int \mathrm{ff}) = 0)$

[FAdd] $\vdash$ $(((\int(\gamma_1 \wedge \gamma_2)) = 0) \supset ((\int(\gamma_1 \vee \gamma_2)) = (\int \gamma_1) + (\int \gamma_2)))$

[Mon] $\vdash$ $((\Box(\gamma_1 \Rightarrow \gamma_2)) \supset ((\int \gamma_1) \leq (\int \gamma_2)))$

Inference rule

[PMP] $\eta_1, (\eta_1 \supset \eta_2) \vdash \eta_2$

Soundness

Theorem

The axiom system of EPPL is sound: if $\vdash \eta$, *then* $\vDash \eta$.

Proof.

Straightforward from the definition of the semantics. □

Soundness

Theorem

The axiom system of EPPL is sound: if $\vdash \eta$, *then* $\models \eta$.

Proof.

Straightforward from the definition of the semantics. □

Completeness and Decidability

Theorem

The proof system of EPPL is weakly complete: if $\vDash \eta$, then $\vdash \eta$. Moreover, the set of theorems of EPPL is recursive.

Proof.

The central result is to show that if η is consistent then there is a model $(\mathcal{K}, \mu)\rho$ such that $(\mathcal{K}, \mu)\rho \Vdash \eta$. The decidability follows by showing that the consistency of a formula is decidable. □

Completeness and Decidability

Theorem

The proof system of EPPL is weakly complete: if $\vDash \eta$, then $\vdash \eta$. Moreover, the set of theorems of EPPL is recursive.

Proof.

The central result is to show that if η is consistent then there is a model $(\mathcal{K}, \mu)\rho$ such that $(\mathcal{K}, \mu)\rho \Vdash \eta$. The decidability follows by showing that the consistency of a formula is decidable. $\square$

Construction of the model

1. compute the (finite) set of valuations over the memory cells and the logical variables in the sets B and X occurring in η and let this set of valuations be V;
2. let κ_1 be the analytical formula obtained from η by effectively replacing measure terms $(\int \gamma)$ by sums $\sum_{v \Vdash_c \gamma, v \in V} y_v$ where y_v represents the probability of the valuation v;
3. let κ be the analytical formula $\kappa_1 \cap \bigcap_{y_v | v \in V} (0 \leq y_v)$;
4. η is consistent iff κ is;
5. finally, consistency of κ is decided by the axiom **RCF** and the model is constructed for a consistent κ by solving for y_v in real closed fields.

Construction of the model

1. compute the (finite) set of valuations over the memory cells and the logical variables in the sets B and X occurring in η and let this set of valuations be V;
2. let κ_1 be the analytical formula obtained from η by effectively replacing measure terms $(\int \gamma)$ by sums $\sum_{v \Vdash_c \gamma, v \in V} y_v$ where y_v represents the probability of the valuation v;
3. let κ be the analytical formula $\kappa_1 \cap \bigcap_{y_v | v \in V} (0 \leq y_v)$;
4. η is consistent iff κ is;
5. finally, consistency of κ is decided by the axiom **RCF** and the model is constructed for a consistent κ by solving for y_v in real closed fields.

Construction of the model

1. compute the (finite) set of valuations over the memory cells and the logical variables in the sets B and X occurring in η and let this set of valuations be V;
2. let κ_1 be the analytical formula obtained from η by effectively replacing measure terms $(\int\gamma)$ by sums $\sum_{v \Vdash_c \gamma, v \in V} y_v$ where y_v represents the probability of the valuation v;
3. let κ be the analytical formula $\kappa_1 \cap \bigcap_{y_v | v \in V} (0 \leq y_v)$;
4. η is consistent iff κ is;
5. finally, consistency of κ is decided by the axiom **RCF** and the model is constructed for a consistent κ by solving for y_v in real closed fields.

Construction of the model

1. compute the (finite) set of valuations over the memory cells and the logical variables in the sets B and X occurring in η and let this set of valuations be V;
2. let κ_1 be the analytical formula obtained from η by effectively replacing measure terms $(\int \gamma)$ by sums $\sum_{v \Vdash_c \gamma, v \in V} y_v$ where y_v represents the probability of the valuation v;
3. let κ be the analytical formula $\kappa_1 \cap \bigcap_{y_v | v \in V} (0 \leq y_v)$;
4. η is consistent iff κ is;
5. finally, consistency of κ is decided by the axiom **RCF** and the model is constructed for a consistent κ by solving for y_v in real closed fields.

Construction of the model

1. compute the (finite) set of valuations over the memory cells and the logical variables in the sets B and X occurring in η and let this set of valuations be V;
2. let κ_1 be the analytical formula obtained from η by effectively replacing measure terms $(\int \gamma)$ by sums $\sum_{v \Vdash_c \gamma, v \in V} y_v$ where y_v represents the probability of the valuation v;
3. let κ be the analytical formula $\kappa_1 \cap \bigcap_{y_v | v \in V} (0 \leq y_v)$;
4. η is consistent iff κ is;
5. finally, consistency of κ is decided by the axiom **RCF** and the model is constructed for a consistent κ by solving for y_v in real closed fields.

Syntax

$$s ::= \mathrm{skip} \mid \mathbf{xm} \leftarrow t \mid \mathbf{bm} \leftarrow \gamma \mid \mathrm{toss}(\mathbf{bm}, r) \mid s; s \mid \mathrm{if}\ \gamma\ \mathrm{then}\ s\ \mathrm{else}\ s$$

Definition

An *expression* is either a term t or a classical state formula γ.

Expressions may contain variables in the set X (input to the program).

Syntax

$$s ::= \text{skip} \mid \mathbf{xm} \leftarrow t \mid \mathbf{bm} \leftarrow \gamma \mid \text{toss}(\mathbf{bm}, r) \mid s; s \mid \text{if } \gamma \text{ then } s \text{ else } s$$

Definition

An *expression* is either a term t or a classical state formula γ.

Expressions may contain variables in the set X (input to the program).

Syntax

$$s ::= \mathsf{skip} \mid \mathbf{xm} \leftarrow t \mid \mathbf{bm} \leftarrow \gamma \mid \mathsf{toss}(\mathbf{bm}, r) \mid s; s \mid \mathsf{if}\ \gamma\ \mathsf{then}\ s\ \mathsf{else}\ s$$

Definition

An *expression* is either a term t or a classical state formula γ.

Expressions may contain variables in the set X (input to the program).

Notation

$[\![\gamma]\!]_v = \mathrm{tt}$ if $v \Vdash_c \gamma$ and $[\![\gamma]\!]_v = \mathrm{ff}$ otherwise

if m is a memory cell and e is an expression of the same type, then $\delta_e^{\mathrm{m}}(v)$ assigns the value $[\![e]\!]_v$ to the cell m and coincides with v elsewhere

$(\mathcal{K}, \mu_1) + (\mathcal{K}, \mu_2) = (\mathcal{K}, \mu_1 + \mu_2)$

$r(\mathcal{K}, \mu) = (\mathcal{K}, r\mu)$

Notation

$[\![\gamma]\!]_v = \mathsf{tt}$ if $v \Vdash_c \gamma$ and $[\![\gamma]\!]_v = \mathsf{ff}$ otherwise

if m is a memory cell and e is an expression of the same type, then $\delta_e^{\mathrm{m}}(v)$ assigns the value $[\![e]\!]_v$ to the cell m and coincides with v elsewhere

$(\mathcal{K}, \mu_1) + (\mathcal{K}, \mu_2) = (\mathcal{K}, \mu_1 + \mu_2)$

$r(\mathcal{K}, \mu) = (\mathcal{K}, r\mu)$

Notation

$[\![\gamma]\!]_v = \mathtt{tt}$ if $v \Vdash_c \gamma$ and $[\![\gamma]\!]_v = \mathtt{ff}$ otherwise

if m is a memory cell and e is an expression of the same type, then $\delta_e^{\mathrm{m}}(v)$ assigns the value $[\![e]\!]_v$ to the cell m and coincides with v elsewhere

$(\mathcal{K}, \mu_1) + (\mathcal{K}, \mu_2) = (\mathcal{K}, \mu_1 + \mu_2)$

$r(\mathcal{K}, \mu) = (\mathcal{K}, r\mu)$

Notation

$[\![\gamma]\!]_v = \mathsf{tt}$ if $v \Vdash_c \gamma$ and $[\![\gamma]\!]_v = \mathsf{ff}$ otherwise

if m is a memory cell and e is an expression of the same type, then $\delta_e^{\mathsf{m}}(v)$ assigns the value $[\![e]\!]_v$ to the cell m and coincides with v elsewhere

$(\mathcal{K}, \mu_1) + (\mathcal{K}, \mu_2) = (\mathcal{K}, \mu_1 + \mu_2)$

$r(\mathcal{K}, \mu) = (\mathcal{K}, r\mu)$

Denotation of programs

The denotation of a program s is a function on generalized probabilistic states.

$$
\begin{aligned}
[\![\mathsf{skip}]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu) \\
[\![\mathbf{xm} \leftarrow t]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_t^{\mathbf{xm}})^{-1}) \\
[\![\mathbf{bm} \leftarrow \gamma]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_\gamma^{\mathbf{bm}})^{-1}) \\
[\![\mathsf{toss}(\mathbf{bm}, r)]\!] &= \lambda(\mathcal{K},\mu).(\widetilde{r}([\![\mathbf{bm} \leftarrow \mathsf{tt}]\!](\mathcal{K},\mu)) + \\
&\qquad (1-\widetilde{r})([\![\mathbf{bm} \leftarrow \mathsf{ff}]\!](\mathcal{K},\mu))) \\
[\![s_1 ; s_2]\!] &= \lambda(\mathcal{K},\mu).[\![s_2]\!]([\![s_1]\!](\mathcal{K},\mu)) \\
[\![\mathsf{if}\ \gamma\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2]\!] &= \lambda(\mathcal{K},\mu).([\![s_1]\!](\mathcal{K},\mu_\gamma) + [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma)}))
\end{aligned}
$$

Denotation of programs

The denotation of a program s is a function on generalized probabilistic states.

$$\begin{aligned}
[\![\mathsf{skip}]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu)\\
[\![\mathbf{xm} \leftarrow t]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu\circ(\delta_t^{\mathbf{xm}})^{-1})\\
[\![\mathbf{bm} \leftarrow \gamma]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu\circ(\delta_\gamma^{\mathbf{bm}})^{-1})\\
[\![\mathsf{toss}(\mathbf{bm},r)]\!] &= \lambda(\mathcal{K},\mu).(\widetilde{r}([\![\mathbf{bm}\leftarrow \mathtt{tt}]\!](\mathcal{K},\mu)) +\\
&\qquad (1-\widetilde{r})([\![\mathbf{bm}\leftarrow \mathtt{ff}]\!](\mathcal{K},\mu)))\\
[\![s_1;s_2]\!] &= \lambda(\mathcal{K},\mu).[\![s_2]\!]([\![s_1]\!](\mathcal{K},\mu))\\
[\![\mathsf{if}\ \gamma\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2]\!] &= \lambda(\mathcal{K},\mu).([\![s_1]\!](\mathcal{K},\mu_\gamma)+[\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma)}))
\end{aligned}$$

Denotation of programs

The denotation of a program s is a function on generalized probabilistic states.

$$
\begin{aligned}
[\![\mathsf{skip}]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu)\\
[\![\mathbf{xm} \leftarrow t]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu\circ(\delta_t^{\mathbf{xm}})^{-1})\\
[\![\mathbf{bm} \leftarrow \gamma]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu\circ(\delta_\gamma^{\mathbf{bm}})^{-1})\\
[\![\mathsf{toss}(\mathbf{bm},r)]\!] &= \lambda(\mathcal{K},\mu).(\widetilde{r}([\![\mathbf{bm} \leftarrow \mathtt{tt}]\!](\mathcal{K},\mu)) +\\
&\qquad (1-\widetilde{r})([\![\mathbf{bm} \leftarrow \mathtt{ff}]\!](\mathcal{K},\mu)))\\
[\![s_1 ; s_2]\!] &= \lambda(\mathcal{K},\mu).[\![s_2]\!]([\![s_1]\!](\mathcal{K},\mu))\\
[\![\mathsf{if}\ \gamma\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2]\!] &= \lambda(\mathcal{K},\mu).([\![s_1]\!](\mathcal{K},\mu_\gamma) + [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma)}))
\end{aligned}
$$

Denotation of programs

The denotation of a program s is a function on generalized probabilistic states.

$$
\begin{aligned}
[\![\mathsf{skip}]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu) \\
[\![\mathbf{xm} \leftarrow t]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_t^{\mathbf{xm}})^{-1}) \\
[\![\mathbf{bm} \leftarrow \gamma]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_\gamma^{\mathbf{bm}})^{-1}) \\
[\![\mathsf{toss}(\mathbf{bm},r)]\!] &= \lambda(\mathcal{K},\mu).(\widetilde{r}([\![\mathbf{bm} \leftarrow \mathtt{tt}]\!](\mathcal{K},\mu)) + \\
&\qquad (1-\widetilde{r})([\![\mathbf{bm} \leftarrow \mathtt{ff}]\!](\mathcal{K},\mu))) \\
[\![s_1; s_2]\!] &= \lambda(\mathcal{K},\mu).[\![s_2]\!]([\![s_1]\!](\mathcal{K},\mu)) \\
[\![\mathsf{if}\ \gamma\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2]\!] &= \lambda(\mathcal{K},\mu).([\![s_1]\!](\mathcal{K},\mu_\gamma) + [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma)}))
\end{aligned}
$$

Denotation of programs

The denotation of a program s is a function on generalized probabilistic states.

$$
\begin{aligned}
[\![\mathsf{skip}]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu)\\
[\![\mathbf{xm} \leftarrow t]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_t^{\mathbf{xm}})^{-1})\\
[\![\mathbf{bm} \leftarrow \gamma]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_\gamma^{\mathbf{bm}})^{-1})\\
[\![\mathsf{toss}(\mathbf{bm},r)]\!] &= \lambda(\mathcal{K},\mu).(\widetilde{r}([\![\mathbf{bm} \leftarrow \mathtt{tt}]\!](\mathcal{K},\mu)) +\\
&\qquad (1-\widetilde{r})([\![\mathbf{bm} \leftarrow \mathtt{ff}]\!](\mathcal{K},\mu)))\\
[\![s_1 ; s_2]\!] &= \lambda(\mathcal{K},\mu).[\![s_2]\!]([\![s_1]\!](\mathcal{K},\mu))\\
[\![\mathsf{if}\ \gamma\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2]\!] &= \lambda(\mathcal{K},\mu).([\![s_1]\!](\mathcal{K},\mu_\gamma) + [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma)}))
\end{aligned}
$$

Denotation of programs

The denotation of a program s is a function on generalized probabilistic states.

$$
\begin{aligned}
[\![\mathsf{skip}]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu)\\
[\![\mathbf{xm} \leftarrow t]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_t^{\mathbf{xm}})^{-1})\\
[\![\mathbf{bm} \leftarrow \gamma]\!] &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_\gamma^{\mathbf{bm}})^{-1})\\
[\![\mathsf{toss}(\mathbf{bm},r)]\!] &= \lambda(\mathcal{K},\mu).(\widetilde{r}([\![\mathbf{bm} \leftarrow \mathtt{tt}]\!](\mathcal{K},\mu)) +\\
&\qquad (1-\widetilde{r})([\![\mathbf{bm} \leftarrow \mathtt{ff}]\!](\mathcal{K},\mu)))\\
[\![s_1; s_2]\!] &= \lambda(\mathcal{K},\mu).[\![s_2]\!]([\![s_1]\!](\mathcal{K},\mu))\\
[\![\mathsf{if}\ \gamma\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2]\!] &= \lambda(\mathcal{K},\mu).([\![s_1]\!](\mathcal{K},\mu_\gamma) + [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma)}))
\end{aligned}
$$

Hoare assertions

$$\Psi ::= \eta \mid \{\eta\}\, s\, \{\eta\}$$

$$(\mathcal{K},\mu)\rho \Vdash_h \eta \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta$$
$$(\mathcal{K},\mu)\rho \Vdash_h \{\eta_1\}\, s\, \{\eta_2\} \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta_2 \text{ whenever } [\![s]\!](\mathcal{K},\mu)\rho \Vdash \eta_1$$

Definition

A Hoare assertion Ψ is *semantically valid* ($\vDash_h \Psi$) if $(\mathcal{K},\mu)\rho \Vdash_h \Psi$ for every generalized probabilistic state $(\mathcal{K},\mu)$ and any $\mathcal{K}$-assignment ρ.

Hoare assertions

$$\Psi ::= \eta \mid \{\eta\}\, s\, \{\eta\}$$

$$(\mathcal{K},\mu)\rho \Vdash_h \eta \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta$$

$$(\mathcal{K},\mu)\rho \Vdash_h \{\eta_1\}\, s\, \{\eta_2\} \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta_2 \text{ whenever } [\![s]\!](\mathcal{K},\mu)\rho \Vdash \eta_1$$

Definition

A Hoare assertion Ψ is *semantically valid* ($\vDash_h \Psi$) if $(\mathcal{K},\mu)\rho \Vdash_h \Psi$ for every generalized probabilistic state $(\mathcal{K},\mu)$ and any $\mathcal{K}$-assignment ρ.

Hoare assertions

$$\Psi ::= \eta \mid \{\eta\}\, s\, \{\eta\}$$

$$(\mathcal{K},\mu)\rho \Vdash_h \eta \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta$$

$$(\mathcal{K},\mu)\rho \Vdash_h \{\eta_1\}\, s\, \{\eta_2\} \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta_2 \text{ whenever } [\![s]\!](\mathcal{K},\mu)\rho \Vdash \eta_1$$

Definition

A Hoare assertion Ψ is *semantically valid* ($\vDash_h \Psi$) if $(\mathcal{K},\mu)\rho \Vdash_h \Psi$ for every generalized probabilistic state $(\mathcal{K},\mu)$ and any $\mathcal{K}$-assignment ρ.

Hoare assertions

$$\Psi ::= \eta \mid \{\eta\}\, s\, \{\eta\}$$

$$(\mathcal{K},\mu)\rho \Vdash_h \eta \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta$$

$$(\mathcal{K},\mu)\rho \Vdash_h \{\eta_1\}\, s\, \{\eta_2\} \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta_2 \text{ whenever } [\![s]\!](\mathcal{K},\mu)\rho \Vdash \eta_1$$

Definition

A Hoare assertion Ψ is *semantically valid* ($\vDash_h \Psi$) if $(\mathcal{K},\mu)\rho \Vdash_h \Psi$ for every generalized probabilistic state $(\mathcal{K},\mu)$ and any $\mathcal{K}$-assignment ρ.

Tossed terms

Let **bm** be a memory cell, $r \in \mathcal{A}$ be a constant and p be a probabilistic term.

The term $\mathsf{toss}(\mathbf{bm}, r; p)$ is the term obtained from p by replacing every occurrence of each measure term $(\int \gamma)$ by $\widetilde{r}(\int \gamma_{\mathsf{tt}}^{\mathbf{bm}}) + (1 - \widetilde{r})(\int \gamma_{\mathsf{ff}}^{\mathbf{bm}})$.

$$\begin{aligned}
\mathsf{toss}(\mathbf{bm}, r; r') &= r' \\
\mathsf{toss}(\mathbf{bm}, r; y) &= y \\
\mathsf{toss}(\mathbf{bm}, r; (\textstyle\int \gamma)) &= (\widetilde{r}(\textstyle\int \gamma_{\mathsf{tt}}^{\mathbf{bm}}) + (1 - \widetilde{r})(\textstyle\int \gamma_{\mathsf{ff}}^{\mathbf{bm}})) \\
\mathsf{toss}(\mathbf{bm}, r; (p + p')) &= (\mathsf{toss}(\mathbf{bm}, r; p) + \mathsf{toss}(\mathbf{bm}, r; p')) \\
\mathsf{toss}(\mathbf{bm}, r; (pp')) &= (\mathsf{toss}(\mathbf{bm}, r; p)\,\mathsf{toss}(\mathbf{bm}, r; p'))
\end{aligned}$$

Tossed terms

Let **bm** be a memory cell, $r \in \mathcal{A}$ be a constant and p be a probabilistic term.
The term $\mathrm{toss}(\mathbf{bm}, r; p)$ is the term obtained from p by replacing every occurrence of each measure term $(\int \gamma)$ by $\widetilde{r}(\int \gamma^{\mathbf{bm}}_{\mathrm{tt}}) + (1 - \widetilde{r})(\int \gamma^{\mathbf{bm}}_{\mathrm{ff}})$.

$$
\begin{aligned}
\mathrm{toss}(\mathbf{bm}, r; r') &= r' \\
\mathrm{toss}(\mathbf{bm}, r; y) &= y \\
\mathrm{toss}(\mathbf{bm}, r; (\textstyle\int \gamma)) &= (\widetilde{r}(\textstyle\int \gamma^{\mathbf{bm}}_{\mathrm{tt}}) + (1 - \widetilde{r})(\textstyle\int \gamma^{\mathbf{bm}}_{\mathrm{ff}})) \\
\mathrm{toss}(\mathbf{bm}, r; (p + p')) &= (\mathrm{toss}(\mathbf{bm}, r; p) + \mathrm{toss}(\mathbf{bm}, r; p')) \\
\mathrm{toss}(\mathbf{bm}, r; (pp')) &= (\mathrm{toss}(\mathbf{bm}, r; p)\,\mathrm{toss}(\mathbf{bm}, r; p'))
\end{aligned}
$$

Tossed terms

Let **bm** be a memory cell, $r \in \mathcal{A}$ be a constant and p be a probabilistic term.

The term $\mathsf{toss}(\mathbf{bm}, r; p)$ is the term obtained from p by replacing every occurrence of each measure term $(\int \gamma)$ by $\widetilde{r}(\int \gamma^{\mathbf{bm}}_{\mathsf{tt}}) + (1 - \widetilde{r})(\int \gamma^{\mathbf{bm}}_{\mathsf{ff}})$.

$$
\begin{aligned}
\mathsf{toss}(\mathbf{bm}, r; r') &= r' \\
\mathsf{toss}(\mathbf{bm}, r; y) &= y \\
\mathsf{toss}(\mathbf{bm}, r; (\textstyle\int \gamma)) &= (\widetilde{r}(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{tt}}) + (1 - \widetilde{r})(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{ff}})) \\
\mathsf{toss}(\mathbf{bm}, r; (p + p')) &= (\mathsf{toss}(\mathbf{bm}, r; p) + \mathsf{toss}(\mathbf{bm}, r; p')) \\
\mathsf{toss}(\mathbf{bm}, r; (pp')) &= (\mathsf{toss}(\mathbf{bm}, r; p)\, \mathsf{toss}(\mathbf{bm}, r; p'))
\end{aligned}
$$

Tossed formulas

Let **bm** be a memory cell, $r \in \mathcal{A}$ be a constant and p be a probabilistic term.

The formula $\mathsf{toss}(\mathbf{bm}, r; \eta)$ is the formula obtained from η by replacing every occurrence of each measure term $(\int \gamma)$ by $\widetilde{r}(\int \gamma^{\mathbf{bm}}_{\mathrm{tt}}) + (1 - \widetilde{r})(\int \gamma^{\mathbf{bm}}_{\mathrm{ff}})$.

$$
\begin{aligned}
\mathsf{toss}(\mathbf{bm}, r; \mathrm{fff}) &= \mathrm{fff} \\
\mathsf{toss}(\mathbf{bm}, r; (p \leq p')) &= (\mathsf{toss}(\mathbf{bm}, r; p) \leq \mathsf{toss}(\mathbf{bm}, r; p')) \\
\mathsf{toss}(\mathbf{bm}, r; (\eta \supset \eta')) &= (\mathsf{toss}(\mathbf{bm}, r; \eta) \supset \mathsf{toss}(\mathbf{bm}, r; \eta'))
\end{aligned}
$$

Tossed formulas

Let **bm** be a memory cell, $r \in \mathcal{A}$ be a constant and p be a probabilistic term.
The formula $\mathrm{toss}(\mathbf{bm}, r; \eta)$ is the formula obtained from η by replacing every occurrence of each measure term $(\int \gamma)$ by $\widetilde{r}(\int \gamma^{\mathbf{bm}}_{\mathrm{tt}}) + (1 - \widetilde{r})(\int \gamma^{\mathbf{bm}}_{\mathrm{ff}})$.

$$\begin{aligned}
\mathrm{toss}(\mathbf{bm}, r; \mathrm{fff}) &= \mathrm{fff} \\
\mathrm{toss}(\mathbf{bm}, r; (p \leq p')) &= (\mathrm{toss}(\mathbf{bm}, r; p) \leq \mathrm{toss}(\mathbf{bm}, r; p')) \\
\mathrm{toss}(\mathbf{bm}, r; (\eta \supset \eta')) &= (\mathrm{toss}(\mathbf{bm}, r; \eta) \supset \mathrm{toss}(\mathbf{bm}, r; \eta'))
\end{aligned}$$

Tossed formulas

Let **bm** be a memory cell, $r \in \mathcal{A}$ be a constant and p be a probabilistic term.
The formula $\mathsf{toss}(\mathbf{bm}, r; \eta)$ is the formula obtained from η by replacing every occurrence of each measure term $(\int \gamma)$ by $\widetilde{r}(\int \gamma^{\mathbf{bm}}_{\mathrm{tt}}) + (1 - \widetilde{r})(\int \gamma^{\mathbf{bm}}_{\mathrm{ff}})$.

$$
\begin{aligned}
\mathsf{toss}(\mathbf{bm}, r; \mathrm{fff}) &= \mathrm{fff} \\
\mathsf{toss}(\mathbf{bm}, r; (p \leq p')) &= (\mathsf{toss}(\mathbf{bm}, r; p) \leq \mathsf{toss}(\mathbf{bm}, r; p')) \\
\mathsf{toss}(\mathbf{bm}, r; (\eta \supset \eta')) &= (\mathsf{toss}(\mathbf{bm}, r; \eta) \supset \mathsf{toss}(\mathbf{bm}, r; \eta'))
\end{aligned}
$$

Conditioned terms

Let γ be classical state formula and p be a probabilistic term.

The term (p/γ) is the term obtained from p by replacing every occurrence of each measure term $(\int \gamma')$ by $(\int (\gamma' \wedge \gamma))$.

$$
\begin{aligned}
r/\gamma &= r \\
y/\gamma &= y \\
(\textstyle\int \gamma')/\gamma &= (\textstyle\int (\gamma \wedge \gamma')) \\
(p + p')/\gamma &= (p/\gamma + p'/\gamma) \\
(pp')/\gamma &= ((p/\gamma)\,(p'/\gamma))
\end{aligned}
$$

Conditioned terms

Let γ be classical state formula and p be a probabilistic term. The term (p/γ) is the term obtained from p by replacing every occurrence of each measure term $(\int\gamma')$ by $(\int(\gamma' \wedge \gamma))$.

$$
\begin{aligned}
r/\gamma &= r \\
y/\gamma &= y \\
(\textstyle\int\gamma')/\gamma &= (\textstyle\int(\gamma \wedge \gamma')) \\
(p + p')/\gamma &= (p/\gamma + p'/\gamma) \\
(pp')/\gamma &= ((p/\gamma)\,(p'/\gamma))
\end{aligned}
$$

Conditioned terms

Let γ be classical state formula and p be a probabilistic term. The term (p/γ) is the term obtained from p by replacing every occurrence of each measure term $(\int \gamma')$ by $(\int (\gamma' \wedge \gamma))$.

$$
\begin{aligned}
r/\gamma &= r \\
y/\gamma &= y \\
(\int \gamma')/\gamma &= (\int (\gamma \wedge \gamma')) \\
(p + p')/\gamma &= (p/\gamma + p'/\gamma) \\
(pp')/\gamma &= ((p/\gamma)\,(p'/\gamma))
\end{aligned}
$$

Conditioned formulas

Let γ be classical state formula and p be a probabilistic term.
The formula η/γ is the formula obtained from η by replacing every occurrence of each measure term $(\int \gamma')$ by $(\int(\gamma' \wedge \gamma))$.

$$\begin{aligned} \mathrm{fff}/\gamma &= \mathrm{fff} \\ (p \leq p')/\gamma &= (p/\gamma \leq p'/\gamma) \\ (\eta \supset \eta')/\gamma &= (\eta/\gamma \supset \eta'/\gamma) \end{aligned}$$

$(\eta_1 \curlyvee_\gamma \eta_2)$ stands for $((\eta_1/\gamma) \cap (\eta_2/(\neg\gamma)))$.

Conditioned formulas

Let γ be classical state formula and p be a probabilistic term. The formula η/γ is the formula obtained from η by replacing every occurrence of each measure term $(\int\gamma')$ by $(\int(\gamma' \wedge \gamma))$.

$$\begin{aligned} \text{fff}/\gamma &= \text{fff} \\ (p \leq p')/\gamma &= (p/\gamma \leq p'/\gamma) \\ (\eta \supset \eta')/\gamma &= (\eta/\gamma \supset \eta'/\gamma) \end{aligned}$$

$(\eta_1 \curlyvee_\gamma \eta_2)$ stands for $((\eta_1/\gamma) \cap (\eta_2/(\neg\gamma)))$.

Conditioned formulas

Let γ be classical state formula and p be a probabilistic term.
The formula η/γ is the formula obtained from η by replacing every occurrence of each measure term $(\int\gamma')$ by $(\int(\gamma' \wedge \gamma))$.

$$
\begin{aligned}
\text{fff}/\gamma &= \text{fff} \\
(p \leq p')/\gamma &= (p/\gamma \leq p'/\gamma) \\
(\eta \supset \eta')/\gamma &= (\eta/\gamma \supset \eta'/\gamma)
\end{aligned}
$$

$(\eta_1 \curlyvee_\gamma \eta_2)$ stands for $((\eta_1/\gamma) \cap (\eta_2/(\neg\gamma)))$.

Conditioned formulas

Let γ be classical state formula and p be a probabilistic term. The formula η/γ is the formula obtained from η by replacing every occurrence of each measure term $(\int\gamma')$ by $(\int(\gamma' \wedge \gamma))$.

$$
\begin{array}{rcl}
\mathrm{fff}/\gamma & = & \mathrm{fff} \\
(p \leq p')/\gamma & = & (p/\gamma \leq p'/\gamma) \\
(\eta \supset \eta')/\gamma & = & (\eta/\gamma \supset \eta'/\gamma)
\end{array}
$$

$(\eta_1 \curlyvee_\gamma \eta_2)$ stands for $((\eta_1/\gamma) \cap (\eta_2/(\neg\gamma)))$.

Axioms

[TAUT]	$\vdash \eta$	if η is an EPPL theorem
[$\int$ **FREE**]	$\vdash \{\kappa\}\, s\, \{\kappa\}$	if κ is an analytical formula

[SKIP]	$\vdash \{\eta\}$ skip $\{\eta\}$
[ASGR]	$\vdash \{\eta_t^{\mathbf{xm}}\}\ \mathbf{xm} \leftarrow t\ \{\eta\}$
[ASGB]	$\vdash \{\eta_\gamma^{\mathbf{bm}}\}\ \mathbf{bm} \leftarrow \gamma\ \{\eta\}$
[TOSS]	$\vdash \{\text{toss}(\mathbf{bm}, \eta; r)\}\ \text{toss}(\mathbf{bm}, r)\ \{\eta\}$

Axioms

[TAUT]	$\vdash \eta$	if η is an EPPL theorem
[$\int$ FREE]	$\vdash \{\kappa\}\, s\, \{\kappa\}$	if κ is an analytical formula
[SKIP]	$\vdash \{\eta\}\ \mathsf{skip}\ \{\eta\}$	
[ASGR]	$\vdash \{\eta^{\mathbf{xm}}_{t}\}\ \mathbf{xm} \leftarrow t\ \{\eta\}$	
[ASGB]	$\vdash \{\eta^{\mathbf{bm}}_{\gamma}\}\ \mathbf{bm} \leftarrow \gamma\ \{\eta\}$	
[TOSS]	$\vdash \{\mathsf{toss}(\mathbf{bm}, \eta; r)\}\ \mathsf{toss}(\mathbf{bm}, r)\ \{\eta\}$	

Axioms

[TAUT]	$\vdash \eta$	if η is an EPPL theorem
[$\int$ FREE]	$\vdash \{\kappa\}\, s\, \{\kappa\}$	if κ is an analytical formula

[SKIP]	$\vdash \{\eta\}\, \mathsf{skip}\, \{\eta\}$
[ASGR]	$\vdash \{\eta_t^{\mathbf{xm}}\}\, \mathbf{xm} \leftarrow t\, \{\eta\}$
[ASGB]	$\vdash \{\eta_\gamma^{\mathbf{bm}}\}\, \mathbf{bm} \leftarrow \gamma\, \{\eta\}$
[TOSS]	$\vdash \{\mathsf{toss}(\mathbf{bm}, \eta; r)\}\, \mathsf{toss}(\mathbf{bm}, r)\, \{\eta\}$

Axioms

[TAUT]	$\vdash \eta$	if η is an EPPL theorem
[$\int$ FREE]	$\vdash \{\kappa\}\, s\, \{\kappa\}$	if κ is an analytical formula
[SKIP]	$\vdash \{\eta\}\, \mathsf{skip}\, \{\eta\}$	
[ASGR]	$\vdash \{\eta^{\mathbf{xm}}_{t}\}\, \mathbf{xm} \leftarrow t\, \{\eta\}$	
[ASGB]	$\vdash \{\eta^{\mathbf{bm}}_{\gamma}\}\, \mathbf{bm} \leftarrow \gamma\, \{\eta\}$	
[TOSS]	$\vdash \{\mathsf{toss}(\mathbf{bm}, \eta; r)\}\, \mathsf{toss}(\mathbf{bm}, r)\, \{\eta\}$	

Axioms

[TAUT]	$\vdash \eta$	if η is an EPPL theorem
[$\int$ FREE]	$\vdash \{\kappa\}\, s\, \{\kappa\}$	if κ is an analytical formula
[SKIP]	$\vdash \{\eta\}\,\mathsf{skip}\,\{\eta\}$	
[ASGR]	$\vdash \{\eta^{\mathbf{xm}}_{t}\}\,\mathbf{xm} \leftarrow t\,\{\eta\}$	
[ASGB]	$\vdash \{\eta^{\mathbf{bm}}_{\gamma}\}\,\mathbf{bm} \leftarrow \gamma\,\{\eta\}$	
[TOSS]	$\vdash \{\mathsf{toss}(\mathbf{bm}, \eta; r)\}\,\mathsf{toss}(\mathbf{bm}, r)\,\{\eta\}$	

Axioms

[TAUT]	$\vdash \eta$	if η is an EPPL theorem
[$\int$ FREE]	$\vdash \{\kappa\}\, s\, \{\kappa\}$	if κ is an analytical formula

$$\textbf{[SKIP]} \quad \vdash \{\eta\}\ \mathsf{skip}\ \{\eta\}$$

$$\textbf{[ASGR]} \quad \vdash \{\eta^{\mathbf{xm}}_{t}\}\ \mathbf{xm} \leftarrow t\ \{\eta\}$$

$$\textbf{[ASGB]} \quad \vdash \{\eta^{\mathbf{bm}}_{\gamma}\}\ \mathbf{bm} \leftarrow \gamma\ \{\eta\}$$

$$\textbf{[TOSS]} \quad \vdash \{\mathsf{toss}(\mathbf{bm}, \eta; r)\}\ \mathsf{toss}(\mathbf{bm}, r)\ \{\eta\}$$

Inference rules

[SEQ] $\{\eta_0\}\, s_1\, \{\eta_1\}, \{\eta_1\}\, s_2\, \{\eta_2\} \vdash \{\eta_0\}\, s_1 ; s_2\, \{\eta_2\}$

[IF] $\{\eta_1\}\, s_1\, \{y_1 = (\int \gamma_0)\}, \{\eta_2\}\, s_2\, \{y_2 = (\int \gamma_0)\}$
$\vdash \{\eta_1 \curlyvee_\gamma \eta_2\} \text{if } \gamma \text{ then } s_1 \text{ else } s_2 \{y_1 + y_2 = (\int \gamma_0)\}$

[ELIMV] $\{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \vdash \{\eta_1{}^y_p\}\, s\, \{\eta_2\}$
y does not occur in p or η_2

[CONS] $\eta_0 \supset \eta_1, \{\eta_1\}\, s\, \{\eta_2\}, \eta_2 \supset \eta_3 \vdash \{\eta_0\}\, s\, \{\eta_3\}$

[OR] $\{\eta_0\}\, s\, \{\eta_2\}, \{\eta_1\}\, s\, \{\eta_2\} \vdash \{\eta_0 \cup \eta_1\}\, s\, \{\eta_2\}$

[AND] $\{\eta_0\}\, s\, \{\eta_1\}, \{\eta_0\}\, s\, \{\eta_2\} \vdash \{\eta_0\}\, s\, \{\eta_1 \cap \eta_2\}$

Inference rules

[SEQ] $\{\eta_0\}\, s_1\, \{\eta_1\}, \{\eta_1\}\, s_2\, \{\eta_2\} \vdash \{\eta_0\}\, s_1; s_2\, \{\eta_2\}$

[IF] $\{\eta_1\}\, s_1\, \{y_1 = (\int \gamma_0)\}, \{\eta_2\}\, s_2\, \{y_2 = (\int \gamma_0)\}$
$\vdash \{\eta_1 \curlyvee_\gamma \eta_2\} \text{if } \gamma \text{ then } s_1 \text{ else } s_2 \{y_1 + y_2 = (\int \gamma_0)\}$

[ELIMV] $\{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \vdash \{\eta_1{}^y_p\}\, s\, \{\eta_2\}$
y does not occur in p or η_2

[CONS] $\eta_0 \supset \eta_1, \{\eta_1\}\, s\, \{\eta_2\}, \eta_2 \supset \eta_3 \vdash \{\eta_0\}\, s\, \{\eta_3\}$

[OR] $\{\eta_0\}\, s\, \{\eta_2\}, \{\eta_1\}\, s\, \{\eta_2\} \vdash \{\eta_0 \cup \eta_1\}\, s\, \{\eta_2\}$

[AND] $\{\eta_0\}\, s\, \{\eta_1\}, \{\eta_0\}\, s\, \{\eta_2\} \vdash \{\eta_0\}\, s\, \{\eta_1 \cap \eta_2\}$

Inference rules

[SEQ] $\{\eta_0\}\, s_1\, \{\eta_1\}, \{\eta_1\}\, s_2\, \{\eta_2\} \vdash \{\eta_0\}\, s_1; s_2\, \{\eta_2\}$

[IF] $\{\eta_1\}\, s_1\, \{y_1 = (\int \gamma_0)\}, \{\eta_2\}\, s_2\, \{y_2 = (\int \gamma_0)\}$
$\vdash \{\eta_1 \curlyvee_\gamma \eta_2\}\text{if } \gamma \text{ then } s_1 \text{ else } s_2\{y_1 + y_2 = (\int \gamma_0)\}$

[ELIMV] $\{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \vdash \{\eta_1{}_p^y\}\, s\, \{\eta_2\}$
y does not occur in p or η_2

[CONS] $\eta_0 \supset \eta_1, \{\eta_1\}\, s\, \{\eta_2\}, \eta_2 \supset \eta_3 \vdash \{\eta_0\}\, s\, \{\eta_3\}$

[OR] $\{\eta_0\}\, s\, \{\eta_2\}, \{\eta_1\}\, s\, \{\eta_2\} \vdash \{\eta_0 \cup \eta_1\}\, s\, \{\eta_2\}$

[AND] $\{\eta_0\}\, s\, \{\eta_1\}, \{\eta_0\}\, s\, \{\eta_2\} \vdash \{\eta_0\}\, s\, \{\eta_1 \cap \eta_2\}$

Inference rules

[SEQ] $\{\eta_0\}\, s_1\, \{\eta_1\}, \{\eta_1\}\, s_2\, \{\eta_2\} \vdash \{\eta_0\}\, s_1 ; s_2\, \{\eta_2\}$

[IF] $\{\eta_1\}\, s_1\, \{y_1 = (\int \gamma_0)\}, \{\eta_2\}\, s_2\, \{y_2 = (\int \gamma_0)\}$
$\vdash \{\eta_1 \curlyvee_\gamma \eta_2\} \text{if } \gamma \text{ then } s_1 \text{ else } s_2 \{y_1 + y_2 = (\int \gamma_0)\}$

[ELIMV] $\{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \vdash \{\eta_1{}^y_p\}\, s\, \{\eta_2\}$
y does not occur in p or η_2

[CONS] $\eta_0 \supset \eta_1, \{\eta_1\}\, s\, \{\eta_2\}, \eta_2 \supset \eta_3 \vdash \{\eta_0\}\, s\, \{\eta_3\}$

[OR] $\{\eta_0\}\, s\, \{\eta_2\}, \{\eta_1\}\, s\, \{\eta_2\} \vdash \{\eta_0 \cup \eta_1\}\, s\, \{\eta_2\}$

[AND] $\{\eta_0\}\, s\, \{\eta_1\}, \{\eta_0\}\, s\, \{\eta_2\} \vdash \{\eta_0\}\, s\, \{\eta_1 \cap \eta_2\}$

Substitution Lemma for classical valuations

Lemma

*For any valuation $v \in \mathcal{V}$, any classical state formula γ, any memory cell m (**xm** or **bm**) and term e of the same type,*

$$v^{m}_{[\![e]\!]_v} \Vdash_c \gamma \textit{ iff } v \Vdash_c \gamma^m_e .$$

Proof.

Induction on the structure of γ. □

Substitution Lemma for classical valuations

Lemma

*For any valuation $v \in \mathcal{V}$, any classical state formula γ, any memory cell m (**xm** or **bm**) and term e of the same type,*

$$v^{m}_{[\![e]\!]_v} \Vdash_c \gamma \text{ iff } v \Vdash_c \gamma^m_e .$$

Proof.

Induction on the structure of γ. □

Substitution Lemma for classical valuations

Lemma

*For any valuation $v \in \mathcal{V}$, any classical state formula γ, any memory cell m (**xm** or **bm**) and term e of the same type,*

$$v^{m}_{[\![e]\!]_v} \Vdash_c \gamma \text{ iff } v \Vdash_c \gamma^{m}_{e}.$$

Proof.

Induction on the structure of γ. □

Substitution Lemma for assignment

Lemma

Let $(\mathcal{K}, \mu)$ be a generalized probabilistic structure and ρ be a $\mathcal{K}$-assignment. Given a memory cell m and a term e of the same type, let $\mu' = \mu \circ (\delta_e^m)^{-1}$. Then

$$[\![(\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![(\int \gamma_e^m)]\!]^{\rho}_{(\mathcal{K},\mu)}$$

for any classical state formula γ.
Furthermore, for any probabilistic term p,

$$[\![p]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![p_e^m]\!]^{\rho}_{(\mathcal{K},\mu)},$$

and, for any probabilistic formula η,

$$(\mathcal{K}, \mu')\rho \Vdash \eta \text{ iff } (\mathcal{K}, \mu)\rho \Vdash \eta_e^m.$$

Substitution Lemma for assignment

Lemma

Let $(\mathcal{K}, \mu)$ be a generalized probabilistic structure and ρ be a $\mathcal{K}$-assignment. Given a memory cell m and a term e of the same type, let $\mu' = \mu \circ (\delta_e^m)^{-1}$. Then

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![(\textstyle\int \gamma^m_e)]\!]^{\rho}_{(\mathcal{K},\mu)}$$

for any classical state formula γ.

Furthermore, for any probabilistic term p,

$$[\![p]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![p^m_e]\!]^{\rho}_{(\mathcal{K},\mu)},$$

and, for any probabilistic formula η,

$$(\mathcal{K}, \mu')\rho \Vdash \eta \text{ iff } (\mathcal{K}, \mu)\rho \Vdash \eta^m_e.$$

Substitution Lemma for assignment

Lemma

Let $(\mathcal{K}, \mu)$ be a generalized probabilistic structure and ρ be a $\mathcal{K}$-assignment. Given a memory cell m and a term e of the same type, let $\mu' = \mu \circ (\delta_e^m)^{-1}$. Then

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![(\textstyle\int \gamma^m_e)]\!]^{\rho}_{(\mathcal{K},\mu)}$$

for any classical state formula γ.
Furthermore, for any probabilistic term p,

$$[\![p]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![p^m_e]\!]^{\rho}_{(\mathcal{K},\mu)},$$

and, for any probabilistic formula η,

$$(\mathcal{K}, \mu')\rho \Vdash \eta \text{ iff } (\mathcal{K}, \mu)\rho \Vdash \eta^m_e.$$

Substitution Lemma for assignment

Proof.

$(\delta_e^m)^{-1}(|\gamma|_\nu) = |\gamma_e^m|_\nu$ and hence $\mu((\delta_e^m)^{-1}(|\gamma|_\nu)) = \mu(|\gamma_e^m|_\nu)$.

Therefore, by definition,

$$[\![(\int\gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \mu \circ (\delta_e^m)^{-1}(|\gamma|_\nu) = \mu(|\gamma_e^m|_\nu) = [\![(\int\gamma_e^m)]\!]^{\rho}_{(\mathcal{K},\mu)}.$$

The result is extended to probabilistic terms and formulas by induction. ∎

Corollary

Axioms **ASGB** *and* **ASGR** *are sound.*

Substitution Lemma for assignment

Proof.

$$(\delta^m_e)^{-1}(|\gamma|_{\mathcal{V}}) = |\gamma^m_e|_{\mathcal{V}} \text{ and hence } \mu((\delta^m_e)^{-1}(|\gamma|_{\mathcal{V}})) = \mu(|\gamma^m_e|_{\mathcal{V}}).$$

Therefore, by definition,

$$[\![(\textstyle\int\gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \mu\circ(\delta^m_e)^{-1}(|\gamma|_{\mathcal{V}}) = \mu(|\gamma^m_e|_{\mathcal{V}}) = [\![(\textstyle\int\gamma^m_e)]\!]^{\rho}_{(\mathcal{K},\mu)}.$$

The result is extended to probabilistic terms and formulas by induction. ∎

Corollary

Axioms **ASGB** *and* **ASGR** *are sound.*

Substitution Lemma for assignment

Proof.

$$(\delta_e^m)^{-1}(|\gamma|_\mathcal{V}) = |\gamma_e^m|_\mathcal{V} \text{ and hence } \mu((\delta_e^m)^{-1}(|\gamma|_\mathcal{V})) = \mu(|\gamma_e^m|_\mathcal{V}).$$

Therefore, by definition,

$$[\![(\textstyle\int\gamma)]\!]^\rho_{(\mathcal{K},\mu')} = \mu \circ (\delta_e^m)^{-1}(|\gamma|_\mathcal{V}) = \mu(|\gamma_e^m|_\mathcal{V}) = [\![(\textstyle\int\gamma_e^m)]\!]^\rho_{(\mathcal{K},\mu)}.$$

The result is extended to probabilistic terms and formulas by induction. □

Corollary

Axioms **ASGB** *and* **ASGR** *are sound.*

Substitution Lemma for assignment

Proof.

$$(\delta_e^m)^{-1}(|\gamma|_{\mathcal{V}}) = |\gamma_e^m|_{\mathcal{V}} \text{ and hence } \mu((\delta_e^m)^{-1}(|\gamma|_{\mathcal{V}})) = \mu(|\gamma_e^m|_{\mathcal{V}}).$$

Therefore, by definition,

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \mu \circ (\delta_e^m)^{-1}(|\gamma|_{\mathcal{V}}) = \mu(|\gamma_e^m|_{\mathcal{V}}) = [\![(\textstyle\int \gamma_e^m)]\!]^{\rho}_{(\mathcal{K},\mu)}.$$

The result is extended to probabilistic terms and formulas by induction. □

Corollary

Axioms **ASGB** *and* **ASGR** *are sound.*

Substitution Lemma for assignment

Proof.

$$(\delta_e^m)^{-1}(|\gamma|_{\mathcal{V}}) = |\gamma_e^m|_{\mathcal{V}} \text{ and hence } \mu((\delta_e^m)^{-1}(|\gamma|_{\mathcal{V}})) = \mu(|\gamma_e^m|_{\mathcal{V}}).$$

Therefore, by definition,

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \mu \circ (\delta_e^m)^{-1}(|\gamma|_{\mathcal{V}}) = \mu(|\gamma_e^m|_{\mathcal{V}}) = [\![(\textstyle\int \gamma_e^m)]\!]^{\rho}_{(\mathcal{K},\mu)}.$$

The result is extended to probabilistic terms and formulas by induction. □

Corollary

Axioms **ASGB** *and* **ASGR** *are sound.*

Substitution Lemma for assignment

Proof.

$$(\delta_e^m)^{-1}(|\gamma|_\mathcal{V}) = |\gamma_e^m|_\mathcal{V} \text{ and hence } \mu((\delta_e^m)^{-1}(|\gamma|_\mathcal{V})) = \mu(|\gamma_e^m|_\mathcal{V}).$$

Therefore, by definition,

$$[\![(\textstyle\int\gamma)]\!]^\rho_{(\mathcal{K},\mu')} = \mu\circ(\delta_e^m)^{-1}(|\gamma|_\mathcal{V}) = \mu(|\gamma_e^m|_\mathcal{V}) = [\![(\textstyle\int\gamma_e^m)]\!]^\rho_{(\mathcal{K},\mu)}.$$

The result is extended to probabilistic terms and formulas by induction. □

Corollary

Axioms **ASGB** *and* **ASGR** *are sound.*

Substitution Lemma for assignment

Proof.

$$(\delta_e^m)^{-1}(|\gamma|_\mathcal{V}) = |\gamma_e^m|_\mathcal{V} \text{ and hence } \mu((\delta_e^m)^{-1}(|\gamma|_\mathcal{V})) = \mu(|\gamma_e^m|_\mathcal{V}).$$

Therefore, by definition,

$$[\![(\textstyle\int\gamma)]\!]^\rho_{(\mathcal{K},\mu')} = \mu \circ (\delta_e^m)^{-1}(|\gamma|_\mathcal{V}) = \mu(|\gamma_e^m|_\mathcal{V}) = [\![(\textstyle\int\gamma_e^m)]\!]^\rho_{(\mathcal{K},\mu)}.$$

The result is extended to probabilistic terms and formulas by induction. □

Corollary

Axioms **ASGB** *and* **ASGR** *are sound.*

Substitution Lemma for probabilistic tosses

Lemma

Let (K, μ) and ρ be as before, $r \in \mathcal{A}$ be a constant and $\mu' = \widetilde{r}\mu \circ (\delta_{\mathrm{tt}}^{\mathbf{bm}})^{-1} + (1 - \widetilde{r})\mu \circ (\delta_{\mathrm{ff}}^{\mathbf{bm}})^{-1}$.

For any classical state formula γ,

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r}[\![(\textstyle\int \gamma_{\mathrm{tt}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)} + (1 - \widetilde{r})[\![(\textstyle\int \gamma_{\mathrm{ff}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)}.$$

Furthermore, for any probabilistic term p,

$$[\![p]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![\mathrm{toss}(\mathbf{bm}, r; p)]\!]^{\rho}_{(K,\mu)},$$

and, for any probabilistic formula η,

$$(\mathcal{K}, \mu')\rho \Vdash \eta \text{ iff } (K, \mu)\rho \Vdash \mathrm{toss}(\mathbf{bm}, r; \eta).$$

Substitution Lemma for probabilistic tosses

Lemma

Let (K,μ) *and* ρ *be as before,* $r \in \mathcal{A}$ *be a constant and* $\mu' = \widetilde{r}\mu \circ (\delta_{\mathsf{tt}}^{\mathbf{bm}})^{-1} + (1-\widetilde{r})\mu \circ (\delta_{\mathsf{ff}}^{\mathbf{bm}})^{-1}$.

For any classical state formula γ,

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r}[\![(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{tt}})]\!]^{\rho}_{(K,\mu)} + (1-\widetilde{r})[\![(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{ff}})]\!]^{\rho}_{(K,\mu)}.$$

Furthermore, for any probabilistic term p,

$$[\![p]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![\mathrm{toss}(\mathbf{bm}, r; p)]\!]^{\rho}_{(K,\mu)},$$

and, for any probabilistic formula η,

$$(\mathcal{K},\mu')\rho \Vdash \eta \textit{ iff } (K,\mu)\rho \Vdash \mathrm{toss}(\mathbf{bm}, r; \eta).$$

Substitution Lemma for probabilistic tosses

Lemma

Let (K, μ) and ρ be as before, $r \in \mathcal{A}$ be a constant and $\mu' = \widetilde{r}\mu \circ (\delta_{\mathsf{tt}}^{\mathbf{bm}})^{-1} + (1 - \widetilde{r})\mu \circ (\delta_{\mathsf{ff}}^{\mathbf{bm}})^{-1}$.

For any classical state formula γ,

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r}[\![(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{tt}})]\!]^{\rho}_{(K,\mu)} + (1 - \widetilde{r})[\![(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{ff}})]\!]^{\rho}_{(K,\mu)}.$$

Furthermore, for any probabilistic term p,

$$[\![p]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![\mathrm{toss}(\mathbf{bm}, r; p)]\!]^{\rho}_{(K,\mu)},$$

and, for any probabilistic formula η,

$$(\mathcal{K}, \mu')\rho \Vdash \eta \textit{ iff } (K, \mu)\rho \Vdash \mathrm{toss}(\mathbf{bm}, r; \eta).$$

Substitution Lemma for probabilistic tosses

Lemma

Let (K, μ) *and* ρ *be as before,* $r \in \mathcal{A}$ *be a constant and* $\mu' = \widetilde{r}\mu \circ (\delta_{\mathsf{tt}}^{\mathbf{bm}})^{-1} + (1 - \widetilde{r})\mu \circ (\delta_{\mathsf{ff}}^{\mathbf{bm}})^{-1}$.

For any classical state formula γ*,*

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r}[\![(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{tt}})]\!]^{\rho}_{(K,\mu)} + (1 - \widetilde{r})[\![(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{ff}})]\!]^{\rho}_{(K,\mu)}.$$

Furthermore, for any probabilistic term p*,*

$$[\![p]\!]^{\rho}_{(\mathcal{K},\mu')} = [\![\mathrm{toss}(\mathbf{bm}, r; p)]\!]^{\rho}_{(K,\mu)},$$

and, for any probabilistic formula η*,*

$$(\mathcal{K}, \mu')\rho \Vdash \eta \text{ iff } (K, \mu)\rho \Vdash \mathrm{toss}(\mathbf{bm}, r; \eta).$$

Substitution Lemma for probabilistic tosses

Proof.

Let $\mu_1 = \mu \circ (\delta^{\mathbf{bm}}_{\mathsf{tt}})^{-1}$ and $\mu_2 = \mu \circ (\delta^{\mathbf{bm}}_{\mathsf{ff}})^{-1}$. Then

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \tilde{r}[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} + (1-\tilde{r})[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)}$$

by definition. Also

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} = [\![(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{tt}})]\!]^{\rho}_{(K,\mu)} \text{ and } [\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)} = [\![(\textstyle\int \gamma^{\mathbf{bm}}_{\mathsf{ff}})]\!]^{\rho}_{(K,\mu)}.$$

The claim for probabilistic terms and probabilistic formulas then follows by induction. □

Corollary

Axiom TOSS *is sound.*

Substitution Lemma for probabilistic tosses

Proof.

Let $\mu_1 = \mu \circ (\delta_{\mathrm{tt}}^{\mathbf{bm}})^{-1}$ and $\mu_2 = \mu \circ (\delta_{\mathrm{ff}}^{\mathbf{bm}})^{-1}$. Then

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r}[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} + (1-\widetilde{r})[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)}$$

by definition. Also

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} = [\![(\textstyle\int \gamma_{\mathrm{tt}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)} \text{ and } [\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)} = [\![(\textstyle\int \gamma_{\mathrm{ff}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)}.$$

The claim for probabilistic terms and probabilistic formulas then follows by induction. ∎

Corollary

Axiom **TOSS** *is sound.*

Substitution Lemma for probabilistic tosses

Proof.

Let $\mu_1 = \mu \circ (\delta_{\mathsf{tt}}^{\mathbf{bm}})^{-1}$ and $\mu_2 = \mu \circ (\delta_{\mathsf{ff}}^{\mathbf{bm}})^{-1}$. Then

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \tilde{r}[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} + (1-\tilde{r})[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)}$$

by definition. Also

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} = [\![(\textstyle\int \gamma_{\mathsf{tt}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)} \text{ and } [\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)} = [\![(\textstyle\int \gamma_{\mathsf{ff}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)}.$$

The claim for probabilistic terms and probabilistic formulas then follows by induction. □

Corollary

Axiom **TOSS** *is sound.*

Substitution Lemma for probabilistic tosses

Proof.

Let $\mu_1 = \mu \circ (\delta_{\mathsf{tt}}^{\mathbf{bm}})^{-1}$ and $\mu_2 = \mu \circ (\delta_{\mathsf{ff}}^{\mathbf{bm}})^{-1}$. Then

$$[\![(\textstyle\int\gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r}[\![(\textstyle\int\gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} + (1-\widetilde{r})[\![(\textstyle\int\gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)}$$

by definition. Also

$$[\![(\textstyle\int\gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} = [\![(\textstyle\int\gamma_{\mathsf{tt}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)} \text{ and } [\![(\textstyle\int\gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)} = [\![(\textstyle\int\gamma_{\mathsf{ff}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)}.$$

The claim for probabilistic terms and probabilistic formulas then follows by induction. □

Corollary

Axiom **TOSS** *is sound.*

Substitution Lemma for probabilistic tosses

Proof.

Let $\mu_1 = \mu \circ (\delta_{\mathsf{tt}}^{\mathbf{bm}})^{-1}$ and $\mu_2 = \mu \circ (\delta_{\mathsf{ff}}^{\mathbf{bm}})^{-1}$. Then

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r}[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} + (1-\widetilde{r})[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)}$$

by definition. Also

$$[\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_1)} = [\![(\textstyle\int \gamma_{\mathsf{tt}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)} \text{ and } [\![(\textstyle\int \gamma)]\!]^{\rho}_{(\mathcal{K},\mu_2)} = [\![(\textstyle\int \gamma_{\mathsf{ff}}^{\mathbf{bm}})]\!]^{\rho}_{(K,\mu)}.$$

The claim for probabilistic terms and probabilistic formulas then follows by induction. □

Corollary

Axiom **TOSS** *is sound.*

Soundness of $\int$ FREE

Lemma

For any statement s, any analytical formula κ, any generalized state $(\mathcal{K}, \mu)$ and $\mathcal{K}$ assignment ρ,

$$([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \kappa \text{ iff } (\mathcal{K}, \mu)\rho \Vdash \kappa.$$

Proof.

The interpretation of analytical formulas depends only on ρ. □

Soundness of $\int$ FREE

Lemma

For any statement s, any analytical formula κ, any generalized state $(\mathcal{K}, \mu)$ and $\mathcal{K}$ assignment ρ,

$$([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \kappa \textit{ iff } (\mathcal{K}, \mu)\rho \Vdash \kappa.$$

Proof.

The interpretation of analytical formulas depends only on ρ. □

Soundness of $\int$ FREE

Lemma

For any statement s, any analytical formula κ, any generalized state $(\mathcal{K}, \mu)$ and $\mathcal{K}$ assignment ρ,

$$([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \kappa \text{ iff } (\mathcal{K}, \mu)\rho \Vdash \kappa.$$

Proof.

The interpretation of analytical formulas depends only on ρ. □

Soundness of **IF**

Lemma

For any generalized state $(\mathcal{K}, \mu)$, $\mathcal{K}$-assignment ρ and classical state formulas γ and γ',

$$[\![(\textstyle\int \gamma')/\gamma]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![(\textstyle\int \gamma')]\!]^{\rho}_{(\mathcal{K},\mu_\gamma)}.$$

Furthermore, for any probability term p,

$$[\![p/\gamma]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{(\mathcal{K},\mu_\gamma)},$$

and, for any probabilistic formula η,

$$(\mathcal{K}, \mu)\rho \Vdash \eta/\gamma \text{ iff } (\mathcal{K}, \mu_\gamma)\rho \Vdash \eta.$$

Soundness of **IF**

Lemma

For any generalized state $(\mathcal{K}, \mu)$, $\mathcal{K}$*-assignment* ρ *and classical state formulas* γ *and* γ',

$$[\![(\textstyle\int \gamma')/\gamma]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![(\textstyle\int \gamma')]\!]^{\rho}_{(\mathcal{K},\mu_\gamma)}.$$

Furthermore, for any probability term p,

$$[\![p/\gamma]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{(\mathcal{K},\mu_\gamma)},$$

and, for any probabilistic formula η,

$$(\mathcal{K}, \mu)\rho \Vdash \eta/\gamma \textit{ iff } (\mathcal{K}, \mu_\gamma)\rho \Vdash \eta.$$

Soundness of **IF**

Lemma

For any generalized state $(\mathcal{K}, \mu)$, $\mathcal{K}$*-assignment* ρ *and classical state formulas* γ *and* γ',

$$[\![(\textstyle\int \gamma')/\gamma]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![(\textstyle\int \gamma')]\!]^{\rho}_{(\mathcal{K},\mu_\gamma)}.$$

Furthermore, for any probability term p,

$$[\![p/\gamma]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{(\mathcal{K},\mu_\gamma)},$$

and, for any probabilistic formula η,

$$(\mathcal{K}, \mu)\rho \Vdash \eta/\gamma \text{ iff } (\mathcal{K}, \mu_\gamma)\rho \Vdash \eta.$$

Soundness of **IF**

Proof.

By definition,

$$[\![(\textstyle\int \gamma')]\!]^{\rho}_{(\mathcal{K},\mu_\gamma)} = \mu_\gamma(|\gamma'|_\mathcal{V}) = \mu(|\gamma'|_\mathcal{V} \cap |\gamma|_\mathcal{V}) = \mu(|\gamma' \wedge \gamma|_\mathcal{V}) =$$

$$[\![(\textstyle\int \gamma')/\gamma]\!]^{\rho}_{(\mathcal{K},\mu)}.$$

The claims for probabilistic terms and formulas follow by induction. □

Soundness of **IF**

Proof.

By definition,

$$[\![(\int\gamma')]\!]^{\rho}_{(\mathcal{K},\mu_\gamma)} = \mu_\gamma(|\gamma'|_\mathcal{V}) = \mu(|\gamma'|_\mathcal{V} \cap |\gamma|_\mathcal{V}) = \mu(|\gamma' \wedge \gamma|_\mathcal{V}) =$$

$$[\![(\int\gamma')/\gamma]\!]^{\rho}_{(\mathcal{K},\mu)}.$$

The claims for probabilistic terms and formulas follow by induction. □

Soundness of IF

Corollary

Given probabilistic state formulas η_1 and η_2, programs s_1 and s_2, variables $y_1 \in Y$ and $y_2 \in Y$ and a classical state formula γ,

$$\vDash_h \{\eta_1\}\, s_1\, \{y_1 = (\textstyle\int \gamma)\} \text{ and } \vDash_h \{\eta_2\}\, s_2\, \{y_2 = (\textstyle\int \gamma)\}$$

iff, for any classical state formula γ_0,

$$\vDash_h \{\eta_1 \curlyvee_{\gamma_0} \eta_2\} \text{ if } \gamma_0 \text{ then } s_1 \text{ else } s_2\, \{y_1 + y_2 = (\textstyle\int \gamma)\}.$$

Soundness of **IF**

Corollary

Given probabilistic state formulas η_1 and η_2, programs s_1 and s_2, variables $y_1 \in Y$ and $y_2 \in Y$ and a classical state formula γ,

$$\vDash_h \{\eta_1\}\, s_1\, \{y_1 = (\textstyle\int \gamma)\} \text{ and } \vDash_h \{\eta_2\}\, s_2\, \{y_2 = (\textstyle\int \gamma)\}$$

iff, for any classical state formula γ_0,

$$\vDash_h \{\eta_1 \curlyvee_{\gamma_0} \eta_2\}\, \textit{if } \gamma_0 \textit{ then } s_1 \textit{ else } s_2\, \{y_1 + y_2 = (\textstyle\int \gamma)\}.$$

Soundness of **IF**

Proof.

Suppose that $(\mathcal{K},\mu)\rho \Vdash \eta_1 \curlyvee_{\gamma_0} \eta_2$. Then $(\mathcal{K},\mu)\rho \Vdash \eta_1/\gamma_0$ and $(\mathcal{K},\mu)\rho \Vdash \eta_2/(\neg\gamma_0)$. Thus, $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$ and $(\mathcal{K},\mu_{(\neg\gamma_0)})\rho \Vdash \eta_2$. Let $(\mathcal{K},\mu_1) = [\![s_1]\!](\mathcal{K},\mu_{\gamma_0})$, $(\mathcal{K},\mu_2) = [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma_0)})$ and $\mu' = \mu_1 + \mu_2$.

Since $\Vdash_h \{\eta_1\}\, s_1\, \{y_1 = (\int\gamma)\}$ and $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$, it follows that $(\mathcal{K},\mu_1) \Vdash_h y_1 = (\int\gamma)$. Thus, by definition $\rho(y_1) = \mu_1(|\gamma|\nu)$.

Similarly, $\rho(y_2) = \mu_2(|\gamma|\nu)$.

Hence,

$\mu'(|\gamma|\nu) = \mu_1(|\gamma|\nu) + \mu_2(|\gamma|\nu) = \rho(y_1) + \rho(y_2) = \rho(y_1 + y_2)$ and $(\mathcal{K},\mu')\rho \Vdash (y_1 + y_2 = (\int\gamma))$ as required. $\square$

Soundness of **IF**

Proof.

Suppose that $(\mathcal{K},\mu)\rho \Vdash \eta_1 \curlyvee_{\gamma_0} \eta_2$. Then $(\mathcal{K},\mu)\rho \Vdash \eta_1/\gamma_0$ and $(\mathcal{K},\mu)\rho \Vdash \eta_2/(\neg\gamma_0)$. Thus, $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$ and $(\mathcal{K},\mu_{(\neg\gamma_0)})\rho \Vdash \eta_2$. Let $(\mathcal{K},\mu_1) = [\![s_1]\!](\mathcal{K},\mu_{\gamma_0})$, $(\mathcal{K},\mu_2) = [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma_0)})$ and $\mu' = \mu_1 + \mu_2$.

Since $\Vdash_h \{\eta_1\}\, s_1\, \{y_1 = (\int\gamma)\}$ and $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$, it follows that $(\mathcal{K},\mu_1) \Vdash_h y_1 = (\int\gamma)$. Thus, by definition $\rho(y_1) = \mu_1(|\gamma|\nu)$. Similarly, $\rho(y_2) = \mu_2(|\gamma|\nu)$.

Hence,

$\mu'(|\gamma|\nu) = \mu_1(|\gamma|\nu) + \mu_2(|\gamma|\nu) = \rho(y_1) + \rho(y_2) = \rho(y_1 + y_2)$ and $(\mathcal{K},\mu')\rho \Vdash (y_1 + y_2 = (\int\gamma))$ as required. $\square$

Soundness of **IF**

Proof.

Suppose that $(\mathcal{K},\mu)\rho \Vdash \eta_1 \curlyvee_{\gamma_0} \eta_2$. Then $(\mathcal{K},\mu)\rho \Vdash \eta_1/\gamma_0$ and $(\mathcal{K},\mu)\rho \Vdash \eta_2/(\neg\gamma_0)$. Thus, $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$ and $(\mathcal{K},\mu_{(\neg\gamma_0)})\rho \Vdash \eta_2$.
Let $(\mathcal{K},\mu_1) = [\![s_1]\!](\mathcal{K},\mu_{\gamma_0})$, $(\mathcal{K},\mu_2) = [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma_0)})$ and $\mu' = \mu_1 + \mu_2$.
Since $\Vdash_h \{\eta_1\}\, s_1\, \{y_1 = (\int\gamma)\}$ and $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$, it follows that $(\mathcal{K},\mu_1) \Vdash_h y_1 = (\int\gamma)$. Thus, by definition $\rho(y_1) = \mu_1(|\gamma|\nu)$.
Similarly, $\rho(y_2) = \mu_2(|\gamma|\nu)$.
Hence,
$\mu'(|\gamma|\nu) = \mu_1(|\gamma|\nu) + \mu_2(|\gamma|\nu) = \rho(y_1) + \rho(y_2) = \rho(y_1 + y_2)$ and $(\mathcal{K},\mu')\rho \Vdash (y_1 + y_2 = (\int\gamma))$ as required. □

Soundness of **IF**

Proof.

Suppose that $(\mathcal{K},\mu)\rho \Vdash \eta_1 \curlyvee_{\gamma_0} \eta_2$. Then $(\mathcal{K},\mu)\rho \Vdash \eta_1/\gamma_0$ and $(\mathcal{K},\mu)\rho \Vdash \eta_2/(\neg\gamma_0)$. Thus, $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$ and $(\mathcal{K},\mu_{(\neg\gamma_0)})\rho \Vdash \eta_2$. Let $(\mathcal{K},\mu_1) = [\![s_1]\!](\mathcal{K},\mu_{\gamma_0})$, $(\mathcal{K},\mu_2) = [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma_0)})$ and $\mu' = \mu_1 + \mu_2$.

Since $\Vdash_h \{\eta_1\}\ s_1\ \{y_1 = (\int\gamma)\}$ and $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$, it follows that $(\mathcal{K},\mu_1) \Vdash_h y_1 = (\int\gamma)$. Thus, by definition $\rho(y_1) = \mu_1(|\gamma|_\nu)$. Similarly, $\rho(y_2) = \mu_2(|\gamma|_\nu)$.

Hence,

$\mu'(|\gamma|_\nu) = \mu_1(|\gamma|_\nu) + \mu_2(|\gamma|_\nu) = \rho(y_1) + \rho(y_2) = \rho(y_1 + y_2)$ and $(\mathcal{K},\mu')\rho \Vdash (y_1 + y_2 = (\int\gamma))$ as required. □

Soundness of **IF**

Proof.

Suppose that $(\mathcal{K},\mu)\rho \Vdash \eta_1 \curlyvee_{\gamma_0} \eta_2$. Then $(\mathcal{K},\mu)\rho \Vdash \eta_1/\gamma_0$ and $(\mathcal{K},\mu)\rho \Vdash \eta_2/(\neg\gamma_0)$. Thus, $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$ and $(\mathcal{K},\mu_{(\neg\gamma_0)})\rho \Vdash \eta_2$. Let $(\mathcal{K},\mu_1) = [\![s_1]\!](\mathcal{K},\mu_{\gamma_0})$, $(\mathcal{K},\mu_2) = [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma_0)})$ and $\mu' = \mu_1 + \mu_2$.

Since $\Vdash_h \{\eta_1\}\, s_1\, \{y_1 = (\int\gamma)\}$ and $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$, it follows that $(\mathcal{K},\mu_1) \Vdash_h y_1 = (\int\gamma)$. Thus, by definition $\rho(y_1) = \mu_1(|\gamma|_\mathcal{V})$. Similarly, $\rho(y_2) = \mu_2(|\gamma|_\mathcal{V})$.

Hence,

$\mu'(|\gamma|_\mathcal{V}) = \mu_1(|\gamma|_\mathcal{V}) + \mu_2(|\gamma|_\mathcal{V}) = \rho(y_1) + \rho(y_2) = \rho(y_1 + y_2)$ and $(\mathcal{K},\mu')\rho \Vdash (y_1 + y_2 = (\int\gamma))$ as required. □

Soundness of **IF**

Proof.

Suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1 \curlyvee_{\gamma_0} \eta_2$. Then $(\mathcal{K}, \mu)\rho \Vdash \eta_1/\gamma_0$ and $(\mathcal{K}, \mu)\rho \Vdash \eta_2/(\neg\gamma_0)$. Thus, $(\mathcal{K}, \mu_{\gamma_0})\rho \Vdash \eta_1$ and $(\mathcal{K}, \mu_{(\neg\gamma_0)})\rho \Vdash \eta_2$. Let $(\mathcal{K}, \mu_1) = [\![s_1]\!](\mathcal{K}, \mu_{\gamma_0})$, $(\mathcal{K}, \mu_2) = [\![s_2]\!](\mathcal{K}, \mu_{(\neg\gamma_0)})$ and $\mu' = \mu_1 + \mu_2$.

Since $\Vdash_h \{\eta_1\}\, s_1\, \{y_1 = (\int\gamma)\}$ and $(\mathcal{K}, \mu_{\gamma_0})\rho \Vdash \eta_1$, it follows that $(\mathcal{K}, \mu_1) \Vdash_h y_1 = (\int\gamma)$. Thus, by definition $\rho(y_1) = \mu_1(|\gamma|_\mathcal{V})$.

Similarly, $\rho(y_2) = \mu_2(|\gamma|_\mathcal{V})$.

Hence,

$\mu'(|\gamma|_\mathcal{V}) = \mu_1(|\gamma|_\mathcal{V}) + \mu_2(|\gamma|_\mathcal{V}) = \rho(y_1) + \rho(y_2) = \rho(y_1 + y_2)$ and $(\mathcal{K}, \mu')\rho \Vdash (y_1 + y_2 = (\int\gamma))$ as required. □

Soundness of IF

Proof.

Suppose that $(\mathcal{K},\mu)\rho \Vdash \eta_1 \curlyvee_{\gamma_0} \eta_2$. Then $(\mathcal{K},\mu)\rho \Vdash \eta_1/\gamma_0$ and $(\mathcal{K},\mu)\rho \Vdash \eta_2/(\neg\,\gamma_0)$. Thus, $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$ and $(\mathcal{K},\mu_{(\neg\,\gamma_0)})\rho \Vdash \eta_2$. Let $(\mathcal{K},\mu_1) = [\![s_1]\!](\mathcal{K},\mu_{\gamma_0})$, $(\mathcal{K},\mu_2) = [\![s_2]\!](\mathcal{K},\mu_{(\neg\,\gamma_0)})$ and $\mu' = \mu_1 + \mu_2$.

Since $\Vdash_h \{\eta_1\}\, s_1\, \{y_1 = (\int\gamma)\}$ and $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$, it follows that $(\mathcal{K},\mu_1) \Vdash_h y_1 = (\int\gamma)$. Thus, by definition $\rho(y_1) = \mu_1(|\gamma|_\mathcal{V})$. Similarly, $\rho(y_2) = \mu_2(|\gamma|_\mathcal{V})$.

Hence,
$\mu'(|\gamma|_\mathcal{V}) = \mu_1(|\gamma|_\mathcal{V}) + \mu_2(|\gamma|_\mathcal{V}) = \rho(y_1) + \rho(y_2) = \rho(y_1 + y_2)$ and $(\mathcal{K},\mu')\rho \Vdash (y_1 + y_2 = (\int\gamma))$ as required. □

Soundness of **IF**

Proof.

Suppose that $(\mathcal{K},\mu)\rho \Vdash \eta_1 \curlyvee_{\gamma_0} \eta_2$. Then $(\mathcal{K},\mu)\rho \Vdash \eta_1/\gamma_0$ and $(\mathcal{K},\mu)\rho \Vdash \eta_2/(\neg\gamma_0)$. Thus, $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$ and $(\mathcal{K},\mu_{(\neg\gamma_0)})\rho \Vdash \eta_2$.
Let $(\mathcal{K},\mu_1) = [\![s_1]\!](\mathcal{K},\mu_{\gamma_0})$, $(\mathcal{K},\mu_2) = [\![s_2]\!](\mathcal{K},\mu_{(\neg\gamma_0)})$ and $\mu' = \mu_1 + \mu_2$.
Since $\Vdash_h \{\eta_1\}\, s_1\, \{y_1 = (\int\gamma)\}$ and $(\mathcal{K},\mu_{\gamma_0})\rho \Vdash \eta_1$, it follows that $(\mathcal{K},\mu_1) \Vdash_h y_1 = (\int\gamma)$. Thus, by definition $\rho(y_1) = \mu_1(|\gamma|_{\mathcal{V}})$.
Similarly, $\rho(y_2) = \mu_2(|\gamma|_{\mathcal{V}})$.
Hence,
$\mu'(|\gamma|_{\mathcal{V}}) = \mu_1(|\gamma|_{\mathcal{V}}) + \mu_2(|\gamma|_{\mathcal{V}}) = \rho(y_1) + \rho(y_2) = \rho(y_1 + y_2)$ and $(\mathcal{K},\mu')\rho \Vdash (y_1 + y_2 = (\int\gamma))$ as required. □

Soundness of **ELIMV**

Lemma

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ *and* $\rho_1 = \rho^{y}_{k}$. *Then:*

- *for any probabilistic term* p_0, $[\![p_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![{p_0}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$;
- *for any probabilistic formula* η, $(\mathcal{K},\mu)\rho_1 \Vdash \eta$ *iff* $(\mathcal{K},\mu)\rho \Vdash \eta^{y}_{p}$.

Proof.

Let p_0 be a variable y_0.

If y_0 is y, then $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![y^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.

Otherwise, $[\![y_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = \rho_1(y_0) = \rho(y_0) = [\![y_0]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![{y_0}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.

The rest follows by induction. □

Soundness of **ELIMV**

Lemma

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ *and* $\rho_1 = \rho^{y}_{k}$. *Then:*

- *for any probabilistic term* p_0, $[\![p_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![p_0{}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$;
- *for any probabilistic formula* η, $(\mathcal{K},\mu)\rho_1 \Vdash \eta$ *iff* $(\mathcal{K},\mu)\rho \Vdash \eta^{y}_{p}$.

Proof.

Let p_0 be a variable y_0.
If y_0 is y, then $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![y^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.
Otherwise, $[\![y_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = \rho_1(y_0) = \rho(y_0) = [\![y_0]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![y_0{}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.
The rest follows by induction. □

Soundness of **ELIMV**

Lemma

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ *and* $\rho_1 = \rho^{y}_{k}$. *Then:*

- *for any probabilistic term* p_0, $[\![p_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![{p_0}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$;
- *for any probabilistic formula* η, $(\mathcal{K},\mu)\rho_1 \Vdash \eta$ *iff* $(\mathcal{K},\mu)\rho \Vdash \eta^{y}_{p}$.

Proof.

Let p_0 be a variable y_0.
If y_0 is y, then $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![y^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.
Otherwise, $[\![y_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = \rho_1(y_0) = \rho(y_0) = [\![y_0]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![{y_0}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.
The rest follows by induction. □

Soundness of **ELIMV**

Lemma

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ *and* $\rho_1 = \rho^{y}_{k}$. *Then:*

- *for any probabilistic term* p_0, $[\![p_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![{p_0}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$;
- *for any probabilistic formula* η, $(\mathcal{K},\mu)\rho_1 \Vdash \eta$ *iff* $(\mathcal{K},\mu)\rho \Vdash \eta^{y}_{p}$.

Proof.

Let p_0 be a variable y_0.

If y_0 is y, then $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![y^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.

Otherwise, $[\![y_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = \rho_1(y_0) = \rho(y_0) = [\![y_0]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![{y_0}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.

The rest follows by induction. □

Soundness of **ELIMV**

Lemma

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ *and* $\rho_1 = \rho^y_k$. *Then:*

- *for any probabilistic term* p_0, $[\![p_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![{p_0}^y_p]\!]^{\rho}_{(\mathcal{K},\mu)}$;
- *for any probabilistic formula* η, $(\mathcal{K},\mu)\rho_1 \Vdash \eta$ *iff* $(\mathcal{K},\mu)\rho \Vdash \eta^y_p$.

Proof.

Let p_0 be a variable y_0.

If y_0 is y, then $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![y^y_p]\!]^{\rho}_{(\mathcal{K},\mu)}$.

Otherwise, $[\![y_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = \rho_1(y_0) = \rho(y_0) = [\![y_0]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![{y_0}^y_p]\!]^{\rho}_{(\mathcal{K},\mu)}$.

The rest follows by induction. □

Soundness of **ELIMV**

Lemma

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ *and* $\rho_1 = \rho^y_k$. *Then:*

- *for any probabilistic term* p_0, $[\![p_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![{p_0}^y_p]\!]^{\rho}_{(\mathcal{K},\mu)}$;
- *for any probabilistic formula* η, $(\mathcal{K},\mu)\rho_1 \Vdash \eta$ *iff* $(\mathcal{K},\mu)\rho \Vdash \eta^y_p$.

Proof.

Let p_0 be a variable y_0.
If y_0 is y, then $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![y^y_p]\!]^{\rho}_{(\mathcal{K},\mu)}$.
Otherwise, $[\![y_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = \rho_1(y_0) = \rho(y_0) = [\![y_0]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![{y_0}^y_p]\!]^{\rho}_{(\mathcal{K},\mu)}$.
The rest follows by induction. □

Soundness of **ELIMV**

Lemma

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ *and* $\rho_1 = \rho^{y}_{k}$. *Then:*

- *for any probabilistic term* p_0, $[\![p_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![{p_0}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$;
- *for any probabilistic formula* η, $(\mathcal{K},\mu)\rho_1 \Vdash \eta$ *iff* $(\mathcal{K},\mu)\rho \Vdash \eta^{y}_{p}$.

Proof.

Let p_0 be a variable y_0.
If y_0 is y, then $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![y^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.
Otherwise, $[\![y_0]\!]^{\rho_1}_{(\mathcal{K},\mu)} = \rho_1(y_0) = \rho(y_0) = [\![y_0]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![{y_0}^{y}_{p}]\!]^{\rho}_{(\mathcal{K},\mu)}$.
The rest follows by induction. □

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η,

$$\text{if } \Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \text{ then } \Vdash_h \{\eta_1{}^y_p\}\, s\, \{\eta_2\}.$$

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1{}^y_p$.

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ and $\rho_1 = \rho^y_k$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k$. Also $[\![p]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![p^y_p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = k$.Therefore, $(\mathcal{K}, \mu)\rho_1 \Vdash (y = p)$.

Since $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and ρ_1 and ρ differ only in the value assigned to y, which does not occur in η_2, $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required. □

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η,

$$\textit{if } \Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \textit{ then } \Vdash_h \{\eta_1{}^y_p\}\, s\, \{\eta_2\}.$$

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1{}^y_p$.

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ and $\rho_1 = \rho^y_k$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k$. Also $[\![p]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![p^y_p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = k$.Therefore, $(\mathcal{K}, \mu)\rho_1 \Vdash (y = p)$.

Since $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and ρ_1 and ρ differ only in the value assigned to y, which does not occur in η_2, $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required. □

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η,

$$\text{if } \Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \text{ then } \Vdash_h \{{\eta_1}^y_p\}\, s\, \{\eta_2\}.$$

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash {\eta_1}^y_p$.

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ and $\rho_1 = \rho^y_k$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k$. Also $[\![p]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![p^y_p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = k$.Therefore, $(\mathcal{K}, \mu)\rho_1 \Vdash (y = p)$.

Since $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and ρ_1 and ρ differ only in the value assigned to y, which does not occur in η_2, $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required.

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η,

$$\text{if } \Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \text{ then } \Vdash_h \{\eta_1{}^y_p\}\, s\, \{\eta_2\}.$$

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1{}^y_p$.
Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ and $\rho_1 = \rho^y_k$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k$. Also $[\![p]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![p^y_p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = k$. Therefore, $(\mathcal{K}, \mu)\rho_1 \Vdash (y = p)$.
Since $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and ρ_1 and ρ differ only in the value assigned to y, which does not occur in η_2, $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required. □

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η,

$$\text{if } \Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \text{ then } \Vdash_h \{{\eta_1}^y_p\}\, s\, \{\eta_2\}.$$

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash {\eta_1}^y_p$.
Let $k = [\![p]\!]^\rho_{(\mathcal{K},\mu)}$ and $\rho_1 = \rho^y_k$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k$. Also $[\![p]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![p^y_p]\!]^\rho_{(\mathcal{K},\mu)} = [\![p]\!]^\rho_{(\mathcal{K},\mu)} = k$. Therefore, $(\mathcal{K}, \mu)\rho_1 \Vdash (y = p)$.
Since $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and ρ_1 and ρ differ only in the value assigned to y, which does not occur in η_2, $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required. □

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η,

$$\text{if } \Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \text{ then } \Vdash_h \{{\eta_1}^y_p\}\, s\, \{\eta_2\}.$$

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash {\eta_1}^y_p$.

Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ and $\rho_1 = \rho^y_k$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k$. Also $[\![p]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![p^y_p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = k$.Therefore, $(\mathcal{K}, \mu)\rho_1 \Vdash (y = p)$.

Since $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and ρ_1 and ρ differ only in the value assigned to y, which does not occur in η_2, $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required. □

Soundness of **ELIMV**

Lemma

Given y *not occurring in either* p *or in* η*,*

$$\textit{if } \Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\} \textit{ then } \Vdash_h \{{\eta_1}^y_p\}\, s\, \{\eta_2\}.$$

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash {\eta_1}^y_p$.
Let $k = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)}$ and $\rho_1 = \rho^y_k$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $[\![y]\!]^{\rho_1}_{(\mathcal{K},\mu)} = k$. Also $[\![p]\!]^{\rho_1}_{(\mathcal{K},\mu)} = [\![p^y_p]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{(\mathcal{K},\mu)} = k$.Therefore, $(\mathcal{K}, \mu)\rho_1 \Vdash (y = p)$.
Since $\Vdash_h \{\eta_1 \cap (y = p)\}\, s\, \{\eta_2\}$ and ρ_1 and ρ differ only in the value assigned to y, which does not occur in η_2, $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required. □

Soundness of the calculus

Theorem

If $\vdash \Psi$ *then* $\vDash_h \Psi$.

Proof.

By induction on the length of the derivation of $\vdash \Psi$ using the previous lemmas. □

Soundness of the calculus

Theorem

If $\vdash \Psi$ *then* $\vDash_h \Psi$.

Proof.

By induction on the length of the derivation of $\vdash \Psi$ using the previous lemmas. □

Preterms

$$\begin{aligned}
\mathrm{pt}(\mathrm{skip}, p) &= p \\
\mathrm{pt}(\mathbf{bm} \leftarrow \gamma, p) &= p_{\gamma}^{\mathbf{bm}} \\
\mathrm{pt}(\mathbf{xm} \leftarrow t, p) &= p_{t}^{\mathbf{xm}} \\
\mathrm{pt}(\mathrm{toss}(\mathbf{bm}, r), p) &= \mathrm{toss}(\mathbf{bm}, r; p) \\
\mathrm{pt}(s_1; s_2, p) &= \mathrm{pt}(s_1, \mathrm{pt}(s_2, p))
\end{aligned}$$

Preterms

$$
\begin{aligned}
\mathrm{pt}(\mathrm{skip},\ p) &= p \\
\mathrm{pt}(\mathbf{bm} \leftarrow \gamma,\ p) &= p_{\gamma}^{\mathbf{bm}} \\
\mathrm{pt}(\mathbf{xm} \leftarrow t,\ p) &= p_{t}^{\mathbf{xm}} \\
\mathrm{pt}(\mathrm{toss}(\mathbf{bm}, r),\ p) &= \mathrm{toss}(\mathbf{bm}, r; p) \\
\mathrm{pt}(s_1; s_2,\ p) &= \mathrm{pt}(s_1,\ \mathrm{pt}(s_2,\ p))
\end{aligned}
$$

Preterms

$$
\begin{aligned}
\mathrm{pt}(\mathrm{skip},\ p) &= p \\
\mathrm{pt}(\mathbf{bm} \leftarrow \gamma,\ p) &= p_{\gamma}^{\mathbf{bm}} \\
\mathrm{pt}(\mathbf{xm} \leftarrow t,\ p) &= p_{t}^{\mathbf{xm}} \\
\mathrm{pt}(\mathrm{toss}(\mathbf{bm}, r),\ p) &= \mathrm{toss}(\mathbf{bm}, r; p) \\
\mathrm{pt}(s_1; s_2,\ p) &= \mathrm{pt}(s_1,\ \mathrm{pt}(s_2,\ p))
\end{aligned}
$$

Preterms

$$
\begin{aligned}
\mathrm{pt}(\mathrm{skip},\ p) &= p \\
\mathrm{pt}(\mathbf{bm} \leftarrow \gamma,\ p) &= p_{\gamma}^{\mathbf{bm}} \\
\mathrm{pt}(\mathbf{xm} \leftarrow t,\ p) &= p_{t}^{\mathbf{xm}} \\
\mathrm{pt}(\mathrm{toss}(\mathbf{bm}, r),\ p) &= \mathrm{toss}(\mathbf{bm}, r; p) \\
\mathrm{pt}(s_1; s_2,\ p) &= \mathrm{pt}(s_1,\ \mathrm{pt}(s_2,\ p))
\end{aligned}
$$

Preterms

$$\begin{aligned}
\mathrm{pt}(\mathrm{skip},\ p) &= p \\
\mathrm{pt}(\mathbf{bm} \leftarrow \gamma,\ p) &= p_{\gamma}^{\mathbf{bm}} \\
\mathrm{pt}(\mathbf{xm} \leftarrow t,\ p) &= p_{t}^{\mathbf{xm}} \\
\mathrm{pt}(\mathrm{toss}(\mathbf{bm}, r),\ p) &= \mathrm{toss}(\mathbf{bm}, r; p) \\
\mathrm{pt}(s_1; s_2,\ p) &= \mathrm{pt}(s_1,\ \mathrm{pt}(s_2,\ p))
\end{aligned}$$

Preterms

$$
\begin{aligned}
\mathrm{pt}(\mathrm{if}\ \gamma\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2,\ r) &= r \\
\mathrm{pt}(\mathrm{if}\ \gamma\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2,\ y) &= y \\
\mathrm{pt}(\mathrm{if}\ \gamma\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2,\ (\textstyle\int\gamma_0)) &= (\mathrm{pt}(s_1,(\textstyle\int\gamma_0))/\gamma\ + \\
&\qquad \mathrm{pt}(s_2,(\textstyle\int\gamma_0))/(\neg\gamma)) \\
\mathrm{pt}(\mathrm{if}\ \gamma\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2,\ (p_1+p_2)) &= (\mathrm{pt}(\mathrm{if}\ \gamma\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2,\ p_1)\ + \\
&\qquad \mathrm{pt}(\mathrm{if}\ \gamma\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2,\ p_2)) \\
\mathrm{pt}(\mathrm{if}\ \gamma\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2,\ (p_1\,p_2)) &= (\mathrm{pt}(\mathrm{if}\ \gamma\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2,\ p_1)\ \times \\
&\qquad \mathrm{pt}(\mathrm{if}\ \gamma\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2,\ p_2))
\end{aligned}
$$

Preterms

$$
\begin{aligned}
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2, r) &= r \\
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2, y) &= y \\
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2, (\textstyle\int \gamma_0)) &= (\mathrm{pt}(s_1, (\textstyle\int \gamma_0))/\gamma + \\
&\qquad \mathrm{pt}(s_2, (\textstyle\int \gamma_0))/(\neg\gamma)) \\
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2, (p_1 + p_2)) &= (\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2, p_1) + \\
&\qquad \mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2, p_2)) \\
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2, (p_1\, p_2)) &= (\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2, p_1) \times \\
&\qquad \mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2, p_2))
\end{aligned}
$$

Preterms

$$
\begin{aligned}
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2,\, r) &= r \\
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2,\, y) &= y \\
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2,\, (\textstyle\int \gamma_0)) &= (\mathrm{pt}(s_1,\, (\textstyle\int \gamma_0))/\gamma + {} \\
&\qquad \mathrm{pt}(s_2,\, (\textstyle\int \gamma_0))/(\neg\gamma)) \\
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2,\, (p_1 + p_2)) &= (\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2,\, p_1) + {} \\
&\qquad \mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2,\, p_2)) \\
\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2,\, (p_1\, p_2)) &= (\mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2,\, p_1) \times {} \\
&\qquad \mathrm{pt}(\text{if } \gamma \text{ then } s_1 \text{ else } s_2,\, p_2))
\end{aligned}
$$

Properties of preterms

Lemma

$$[\![\mathrm{pt}(s,\ p)]\!]^{\rho}_{(\mathcal{K},\mu)} = [\![p]\!]^{\rho}_{[\![s]\!](\mathcal{K},\mu)}.$$

Weakest preconditions

$$\begin{aligned} \mathrm{wp}(s, \mathrm{fff}) &= \mathrm{fff} \\ \mathrm{wp}(s, (p_1 \leq p_2)) &= (\mathrm{pt}(s, p_1) \leq \mathrm{pt}(s, p_2)) \\ \mathrm{wp}(s, (\eta_1 \supset \eta_2)) &= (\mathrm{wp}(s, \eta_1) \supset \mathrm{wp}(s, \eta_2)) \end{aligned}$$

Theorem

$$(\mathcal{K}, \mu)\rho \Vdash_h \mathrm{wp}(s, \eta) \text{ iff } ([\![s]\!](\mathcal{K}, \mu))\rho \Vdash_h \eta.$$

Weakest preconditions

$$\begin{aligned} \mathrm{wp}(s, \mathrm{fff}) &= \mathrm{fff} \\ \mathrm{wp}(s, (p_1 \leq p_2)) &= (\mathrm{pt}(s,\ p_1) \leq \mathrm{pt}(s,\ p_2)) \\ \mathrm{wp}(s, (\eta_1 \supset \eta_2)) &= (\mathrm{wp}(s, \eta_1) \supset \mathrm{wp}(s, \eta_2)) \end{aligned}$$

Theorem

$(\mathcal{K}, \mu)\rho \Vdash_h \mathrm{wp}(s, \eta)$ *iff* $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash_h \eta$.

Weakest preconditions

$$\begin{aligned}
\mathrm{wp}(s, \mathrm{fff}) &= \mathrm{fff} \\
\mathrm{wp}(s, (p_1 \leq p_2)) &= (\mathrm{pt}(s, p_1) \leq \mathrm{pt}(s, p_2)) \\
\mathrm{wp}(s, (\eta_1 \supset \eta_2)) &= (\mathrm{wp}(s, \eta_1) \supset \mathrm{wp}(s, \eta_2))
\end{aligned}$$

Theorem

$$(\mathcal{K}, \mu)\rho \Vdash_h \mathrm{wp}(s, \eta) \ \textit{iff} \ ([\![s]\!](\mathcal{K}, \mu))\rho \Vdash_h \eta.$$

Weakest preconditions, semantically

Corollary

$$\vDash_h \{\eta'\}\, s\, \{\eta\} \textit{ iff } \vDash (\eta' \supset \mathrm{wp}(s, \eta)).$$

Proof.

($\Rightarrow$) Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$, hence $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$. Therefore $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$.

($\Leftarrow$) Suppose that $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$ and hence $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$. Therefore $\vDash_h \{\eta'\}\, s\, \{\eta\}$. $\square$

Weakest preconditions, semantically

Corollary

$$\vDash_h \{\eta'\}\, s\, \{\eta\} \textit{ iff } \vDash (\eta' \supset \mathrm{wp}(s, \eta)).$$

Proof.

$(\Rightarrow)$ Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$, hence $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$. Therefore $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$.

$(\Leftarrow)$ Suppose that $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$ and hence $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$. Therefore $\vDash_h \{\eta'\}\, s\, \{\eta\}$. □

Weakest preconditions, semantically

Corollary

$$\vDash_h \{\eta'\}\, s\, \{\eta\} \textit{ iff } \vDash (\eta' \supset \mathrm{wp}(s, \eta)).$$

Proof.

($\Rightarrow$) Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$, hence $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$. Therefore $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$.

($\Leftarrow$) Suppose that $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$ and hence $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$. Therefore $\vDash_h \{\eta'\}\, s\, \{\eta\}$. □

Weakest preconditions, semantically

Corollary

$$\vDash_h \{\eta'\}\, s\, \{\eta\} \textit{ iff } \vDash (\eta' \supset \mathrm{wp}(s,\eta)).$$

Proof.

($\Rightarrow$) Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$ and $(\mathcal{K},\mu)\rho \Vdash \eta'$.
Then $([\![s]\!](\mathcal{K},\mu))\rho \Vdash \eta$, hence $(\mathcal{K},\mu)\rho \Vdash \mathrm{wp}(s,\eta)$. Therefore $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$.

($\Leftarrow$) Suppose that $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$ and $(\mathcal{K},\mu)\rho \Vdash \eta'$.
Then $(\mathcal{K},\mu)\rho \Vdash \mathrm{wp}(s,\eta)$ and hence $([\![s]\!](\mathcal{K},\mu))\rho \Vdash \eta$. Therefore $\vDash_h \{\eta'\}\, s\, \{\eta\}$. □

Weakest preconditions, semantically

Corollary

$$\vDash_h \{\eta'\}\, s\, \{\eta\} \text{ iff } \vDash (\eta' \supset \mathsf{wp}(s,\eta)).$$

Proof.

($\Rightarrow$) Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$ and $(\mathcal{K},\mu)\rho \Vdash \eta'$.
Then $(\llbracket s \rrbracket(\mathcal{K},\mu))\rho \Vdash \eta$, hence $(\mathcal{K},\mu)\rho \Vdash \mathsf{wp}(s,\eta)$. Therefore $\vDash (\eta' \supset \mathsf{wp}(s,\eta))$.

($\Leftarrow$) Suppose that $\vDash (\eta' \supset \mathsf{wp}(s,\eta))$ and $(\mathcal{K},\mu)\rho \Vdash \eta'$.
Then $(\mathcal{K},\mu)\rho \Vdash \mathsf{wp}(s,\eta)$ and hence $(\llbracket s \rrbracket(\mathcal{K},\mu))\rho \Vdash \eta$. Therefore $\vDash_h \{\eta'\}\, s\, \{\eta\}$. □

Weakest preconditions, semantically

Corollary

$$\vDash_h \{\eta'\}\, s\, \{\eta\} \textit{ iff } \vDash (\eta' \supset \mathrm{wp}(s, \eta)).$$

Proof.

($\Rightarrow$) Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$, hence $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$. Therefore $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$.

($\Leftarrow$) Suppose that $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$ and hence $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$. Therefore $\vDash_h \{\eta'\}\, s\, \{\eta\}$. □

Weakest preconditions, semantically

Corollary

$$\vDash_h \{\eta'\}\, s\, \{\eta\} \textit{ iff } \vDash (\eta' \supset \mathrm{wp}(s,\eta)).$$

Proof.

($\Rightarrow$) Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$ and $(\mathcal{K},\mu)\rho \Vdash \eta'$.
Then $(\llbracket s \rrbracket(\mathcal{K},\mu))\rho \Vdash \eta$, hence $(\mathcal{K},\mu)\rho \Vdash \mathrm{wp}(s,\eta)$. Therefore $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$.

($\Leftarrow$) Suppose that $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$ and $(\mathcal{K},\mu)\rho \Vdash \eta'$.
Then $(\mathcal{K},\mu)\rho \Vdash \mathrm{wp}(s,\eta)$ and hence $(\llbracket s \rrbracket(\mathcal{K},\mu))\rho \Vdash \eta$. Therefore $\vDash_h \{\eta'\}\, s\, \{\eta\}$. □

Weakest preconditions, semantically

Corollary

$$\vDash_h \{\eta'\}\, s\, \{\eta\} \textit{ iff } \vDash (\eta' \supset \mathrm{wp}(s, \eta)).$$

Proof.

($\Rightarrow$) Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$, hence $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$. Therefore $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$.

($\Leftarrow$) Suppose that $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$ and $(\mathcal{K}, \mu)\rho \Vdash \eta'$.
Then $(\mathcal{K}, \mu)\rho \Vdash \mathrm{wp}(s, \eta)$ and hence $([\![s]\!](\mathcal{K}, \mu))\rho \Vdash \eta$. Therefore $\vDash_h \{\eta'\}\, s\, \{\eta\}$. □

Weakest preconditions, semantically

Corollary

$$\vDash_h \{\eta'\}\, s\, \{\eta\} \textit{ iff } \vDash (\eta' \supset \mathrm{wp}(s,\eta)).$$

Proof.

($\Rightarrow$) Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$ and $(\mathcal{K},\mu)\rho \Vdash \eta'$.
Then $([\![s]\!](\mathcal{K},\mu))\rho \Vdash \eta$, hence $(\mathcal{K},\mu)\rho \Vdash \mathrm{wp}(s,\eta)$. Therefore $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$.

($\Leftarrow$) Suppose that $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$ and $(\mathcal{K},\mu)\rho \Vdash \eta'$.
Then $(\mathcal{K},\mu)\rho \Vdash \mathrm{wp}(s,\eta)$ and hence $([\![s]\!](\mathcal{K},\mu))\rho \Vdash \eta$. Therefore $\vDash_h \{\eta'\}\, s\, \{\eta\}$. □

Weakest preconditions, sintactically

Lemma

For any probabilistic term p, statement s and variable y,

$$\vdash \{y = \mathrm{pt}(s, p)\}\, s\, \{y = p\}.$$

Theorem

For any statement s and any conditional-free formula η,

$$\vdash \{\mathrm{wp}(s, \eta)\}\, s\, \{\eta\}.$$

Weakest preconditions, sintactically

Lemma

For any probabilistic term p, statement s and variable y,

$$\vdash \{y = \mathrm{pt}(s, p)\}\, s\, \{y = p\}.$$

Theorem

For any statement s and any conditional-free formula η,

$$\vdash \{\mathrm{wp}(s, \eta)\}\, s\, \{\eta\}.$$

Completeness and decidability

Theorem

Let s be a probabilistic sequential program and η be an EPPL formula. If $\vDash_h \{\eta'\}\, s\, \{\eta\}$, then $\vdash \{\eta'\}\, s\, \{\eta\}$.

Moreover, the set of theorems of the Hoare calculus is recursive.

Completeness and decidability

Theorem

Let s be a probabilistic sequential program and η be an EPPL formula. If $\vDash_h \{\eta'\}\, s\, \{\eta\}$, then $\vdash \{\eta'\}\, s\, \{\eta\}$.

Moreover, the set of theorems of the Hoare calculus is recursive.

Completeness and decidability

Theorem

Let s be a probabilistic sequential program and η *be an EPPL formula. If* $\vDash_h \{\eta'\}\, s\, \{\eta\}$, *then* $\vdash \{\eta'\}\, s\, \{\eta\}$.

Moreover, the set of theorems of the Hoare calculus is recursive.

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically. □

Completeness and decidability

Proof.

Completeness. Suppose that $\models_h \{\eta'\}\, s\, \{\eta\}$. Then $\models (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\models_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically. □

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by CONS.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically. □

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically.

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically. □

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically.

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s, \eta))$. On the other hand, $\vdash \{\mathrm{wp}(s, \eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s, \eta)$ can be computed algorithmically.

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically.

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically. □

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically. □

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\}\, s\, \{\eta\}$. Then $\vDash (\eta' \supset \mathrm{wp}(s,\eta))$. By completeness of EPPL, $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. On the other hand, $\vdash \{\mathrm{wp}(s,\eta)\}\, s\, \{\eta\}$, whence $\vdash \{\eta'\}\, s\, \{\eta\}$ by **CONS**.

Decidability. By soundness and completeness, $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vDash_h \{\eta'\}\, s\, \{\eta\}$. By completeness of EPPL and the properties of weakest preconditions, it follows that $\vdash \{\eta'\}\, s\, \{\eta\}$ iff $\vdash (\eta' \supset \mathrm{wp}(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $\mathrm{wp}(s,\eta)$ can be computed algorithmically. □

Achievements

- logic for non-deterministic programs with truth-functional semantics
- sound, complete and decidable state logic
- sound, complete and decidable Hoare calculus

Achievements

- logic for non-deterministic programs with truth-functional semantics
- sound, complete and decidable state logic
- sound, complete and decidable Hoare calculus

Achievements

- logic for non-deterministic programs with truth-functional semantics
- sound, complete and decidable state logic
- sound, complete and decidable Hoare calculus

Future work

- unbounded iteration (`while`)
- quantum programming languages

Future work

- unbounded iteration (`while`)
- quantum programming languages