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

・ロト ・同ト ・ヨト ・ヨト

Motivation

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

< □ > < □ > < □ > < □ > < Ξ > < Ξ > □ Ξ



• reasoning about non-deterministic programs

new approach: truth values for formulas

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・



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

・ロン ・回 と ・ ヨ と ・ ヨ と



- Language
- Semantics
- Calculus
- Properties
- 2 The Programming Language
 - Syntax
 - Semantics
- 3 The Hoare Calculus
 - The calculus
 - Soundness
 - Completeness



・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・



- Language
- Semantics
- Calculus
- Properties
- 2 The Programming Language
 - Syntax
 - Semantics
- 3 The Hoare Calculus
 - The calculus
 - Soundness
 - Completeness

4 Conclusions

() < </p>

臣



- Language
- Semantics
- Calculus
- Properties
- 2 The Programming Language
 - Syntax
 - Semantics
- 3 The Hoare Calculus
 - The calculus
 - Soundness
 - Completeness

4 Conclusions

臣

伺 ト イヨト イヨト



- Language
- Semantics
- Calculus
- Properties
- 2 The Programming Language
 - Syntax
 - Semantics
- 3 The Hoare Calculus
 - The calculus
 - Soundness
 - Completeness



臣

- ∢ ⊒ ▶

Language Semantics Calculus Properties

Why EPPL

• two-layered design (exogenous approach)

- classical propositional logic at the lower level
- probabilistic logic built at the higher level

・ロト ・回ト ・ヨト ・ヨト

臣

Language Semantics Calculus Properties



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

・ロト ・回ト ・ヨト ・ヨト

크

Language Semantics Calculus Properties

Why EPPL

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

・ロト ・回ト ・ヨト ・ヨト

э

Language Semantics Calculus Properties

Real-closed fields

Definition

A real closed field is an ordered field ${\cal K}$ 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.

・ロト ・回ト ・ヨト ・ヨト

크

Language Semantics Calculus Properties

Real-closed fields

Definition

A real closed field is an ordered field ${\cal K}$ 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.

・ロト ・回ト ・ヨト ・ヨト

Language Semantics Calculus Properties

Real-closed fields

Definition

A real closed field is an ordered field ${\cal K}$ 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.

・ロト ・回ト ・ヨト ・ヨト

Language Semantics Calculus Properties

Real-closed fields

Definition

A real closed field is an ordered field ${\cal K}$ 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.

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

Language Semantics Calculus Properties

Real-closed fields

Definition

A real closed field is an ordered field ${\cal K}$ 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.

イロト イヨト イヨト イヨト

Language Semantics Calculus Properties

Setting

• finite range D of real numbers

- finite set $\mathbf{m} = \{0, \dots, m-1\}$ 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
- ullet real-closed field ${\cal K}$ with set of algebraic numbers ${\cal A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over \mathcal{K}

・ロン ・回 と ・ ヨ と ・ ヨ と

크

Language Semantics Calculus Properties

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \dots, m-1\}$ 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
- ullet real-closed field ${\cal K}$ with set of algebraic numbers ${\cal A}$
- logical variables $Y=\{y_k:k\in\mathbb{N}\}$ ranging over $\mathcal K$

・ロン ・回 と ・ヨン ・ ヨン

Language Semantics Calculus Properties

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \dots, m-1\}$ 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
- ullet real-closed field ${\cal K}$ with set of algebraic numbers ${\cal A}$
- logical variables $Y=\{y_k:k\in\mathbb{N}\}$ ranging over $\mathcal K$

・ロ・ ・ 日・ ・ ヨ・ ・ 日・

Language Semantics Calculus Properties

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \dots, m-1\}$ 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
- ullet real-closed field ${\mathcal K}$ with set of algebraic numbers ${\mathcal A}$
- logical variables $Y=\{y_k:k\in\mathbb{N}\}$ ranging over $\mathcal K$

(ロ) (同) (E) (E) (E)

Language Semantics Calculus Properties

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \dots, m-1\}$ 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
- ullet 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}

・ロ・ ・ 日・ ・ ヨ・ ・ 日・

Language Semantics Calculus Properties

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \dots, m-1\}$ 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
- ullet real-closed field ${\cal K}$ with set of algebraic numbers ${\cal A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over $\mathcal K$

(ロ) (同) (E) (E) (E)

Language Semantics Calculus Properties

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \dots, m-1\}$ 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
- \bullet real-closed field ${\cal K}$ with set of algebraic numbers ${\cal A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over \mathcal{K}

(ロ) (同) (E) (E) (E)

Language Semantics Calculus Properties

Setting

- finite range D of real numbers
- finite set $\mathbf{m} = \{0, \dots, m-1\}$ 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
- \bullet real-closed field ${\cal K}$ with set of algebraic numbers ${\cal A}$
- logical variables $Y = \{y_k : k \in \mathbb{N}\}$ ranging over \mathcal{K}

Language Semantics Calculus Properties

Language

Real terms (with $c \in D$)

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

Classical state formulas

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

Probability terms (with $r \in \mathcal{A}$) $p ::= r \mid y \mid \widetilde{r} \mid (\int \gamma) \mid (p + p) \mid (p p)$

Probabilistic state formulas

イロン イヨン イヨン イヨン

The State Logic: EPPL Lang The Programming Language Calculus The Hoare Calculus Conclusions Prop

Language Semantics Calculus Properties

Language

Real terms (with $c \in D$)

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

Classical state formulas

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

Probability terms (with $r \in A$) $p ::= r \mid y \mid \tilde{r} \mid (\int \gamma) \mid (p + p) \mid (p p)$

Probabilistic state formulas

イロト イヨト イヨト イヨト

 The State Logic: EPPL
 Language

 The Programming Language
 Semantics

 The Hoare Calculus
 Calculus

 Conclusions
 Properties

Language

Real terms (with $c \in D$)

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

Classical state formulas

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

Probability terms (with $r \in A$) $p ::= r \mid y \mid \tilde{r} \mid (\int \gamma) \mid (p + p) \mid (p p)$

Probabilistic state formulas

イロン イヨン イヨン イヨン

 The State Logic: EPPL
 Language

 The Programming Language
 Semantics

 The Hoare Calculus
 Conclusions

 Conclusions
 Properties

Language

Real terms (with $c \in D$)

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

Classical state formulas

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

Probability terms (with $r \in A$) $p ::= r \mid y \mid \tilde{r} \mid (\int \gamma) \mid (p + p) \mid (p p)$

Probabilistic state formulas

イロン イヨン イヨン イヨン

 The State Logic: EPPL
 Language

 The Programming Language
 Semantics

 The Hoare Calculus
 Conclusions

 Conclusions
 Properties

Language

Real terms (with $c \in D$)

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

Classical state formulas

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

Probability terms (with $r \in A$) $p ::= r \mid y \mid \tilde{r} \mid (\int \gamma) \mid (p + p) \mid (p p)$

Probabilistic state formulas

 The State Logic: EPPL
 Language

 The Programming Language
 Semantics

 The Hoare Calculus
 Conclusions

 Conclusions
 Properties

Language

Real terms (with $c \in D$)

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

Classical state formulas

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

Probability terms (with $r \in A$) $p ::= r \mid y \mid \tilde{r} \mid (\int \gamma) \mid (p + p) \mid (p p)$

Probabilistic state formulas

Luís Cruz-Filipe Reasoning about Probabilistic Sequential Programs

 The State Logic: EPPL
 Language

 The Programming Language
 Semantics

 The Hoare Calculus
 Conclusions

 Conclusions
 Properties

Language

Real terms (with $c \in D$)

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

Classical state formulas

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

 $\eta ::= (p \le p) \mid \text{fff} \mid (\eta \supset \eta)$

Probability terms (with $r \in A$) $p ::= r \mid y \mid \widetilde{r} \mid (\int \gamma) \mid (p + p) \mid (p p)$

Probabilistic state formulas

Luís Cruz-Filipe Reasoning about Probabilistic Sequential Programs

 The State Logic: EPPL
 Language

 The Programming Language
 Semantics

 The Hoare Calculus
 Conclusions

 Conclusions
 Properties

Language

Real terms (with $c \in D$)

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

Classical state formulas

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

Probability terms (with $r \in A$) $p ::= r \mid y \mid \tilde{r} \mid (\int \gamma) \mid (p + p) \mid (p p)$

Probabilistic state formulas

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

Language Semantics Calculus Properties

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.

$$\kappa ::= (a \le a) \mid \text{fff} \mid (\kappa \supset \kappa)$$

 $(\Box\gamma)$ stands for the formula $((\int\gamma) = (\int t))$ $(\Diamond\gamma)$ stands for the formula $(\ominus(\Box(\neg\gamma)))$

・ロン ・回 と ・ ヨ と ・ ヨ と

Language Semantics Calculus Properties

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.

$$\kappa ::= (a \le a) \mid \text{fff} \mid (\kappa \supset \kappa)$$

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

・ロン ・回 と ・ ヨ と ・ ヨ と

Language Semantics Calculus Properties

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.

$$\kappa ::= (a \le a) \mid \text{fff} \mid (\kappa \supset \kappa)$$

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

・ロン ・回 と ・ ヨ と ・ ヨ と …

Language Semantics Calculus Properties

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 $\llbracket t \rrbracket_{v}$ 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

$|\gamma|_V = \{ v \in V \mid v \Vdash_{\mathsf{c}} \gamma \}.$

・ロン ・雪 と ・ ヨ と ・ ヨ と

Language Semantics Calculus Properties

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

$$|\gamma|_V = \{ v \in V \mid v \Vdash_{\mathsf{c}} \gamma \}.$$

・ロン ・回 と ・ヨン ・ヨン

Э

Language Semantics Calculus Properties

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 $\llbracket t \rrbracket_v$ 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

$$|\gamma|_V = \{ v \in V \mid v \Vdash_{\mathsf{c}} \gamma \}.$$

ヘロン 人間 とくほど 人間 とう

Э

Language Semantics Calculus Properties

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

$$|\gamma|_V = \{ v \in V \mid v \Vdash_{\mathsf{c}} \gamma \}.$$

・ロン ・四マ ・ヨン ・ヨン

Language Semantics Calculus Properties

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

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

3

Language Semantics Calculus Properties

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

・ロン ・回 と ・ ヨン ・ ヨン

Language Semantics Calculus Properties

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 $\mu(X)=1.$

・ロン ・回 と ・ ヨン ・ ヨン

Language Semantics Calculus Properties

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 $\mu(X) = 1$.

・ロン ・回 と ・ ヨ と ・ ヨ と …

Language Semantics Calculus Properties

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 $\rho: Y \to \mathcal{K}$.

・ロト ・回ト ・ヨト ・ヨト

Language Semantics Calculus Properties

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 $\rho: Y \to \mathcal{K}$.

・ロン ・回 と ・ ヨン ・ ヨン

Language Semantics Calculus Properties

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 $\rho: \mathbf{Y} \to \mathcal{K}$.

・ロン ・回 と ・ ヨン ・ ヨン

Language Semantics Calculus Properties

Interpretation

Denotation of probability terms

$$\begin{bmatrix} r \end{bmatrix}_{K,\mu}^{\rho} = r \\ \begin{bmatrix} y \end{bmatrix}_{K,\mu}^{\rho} = \rho(y) \\ \begin{bmatrix} (\int \gamma) \end{bmatrix}_{K,\mu}^{\rho} = \mu(|\gamma|_{\mathcal{V}}) \\ \begin{bmatrix} p_{1} + p_{2} \end{bmatrix}_{K,\mu}^{\rho} = \begin{bmatrix} p_{1} \end{bmatrix}_{K,\mu}^{\rho} + \begin{bmatrix} p_{2} \end{bmatrix}_{K,\mu}^{\rho} \\ \begin{bmatrix} p_{1} p_{2} \end{bmatrix}_{K,\mu}^{\rho} = \begin{bmatrix} p_{1} \end{bmatrix}_{K,\mu}^{\rho} \times \begin{bmatrix} p_{2} \end{bmatrix}_{K,\mu}^{\rho}$$

Satisfaction of probabilistic formulas $(K,\mu)
ho \Vdash (
ho_1 \le
ho_2)$ iff $\llbracket
ho_1 \rrbracket_{K,\mu}^{
ho} \le \llbracket
ho_2 \rrbracket_{K,\mu}^{
ho}$ $(K,\mu)
ho \nVdash$ fff $(K,\mu)
ho \Vdash (\eta_1 \supset \eta_2)$ iff $(K,\mu)
ho \Vdash \eta_2$ or $(K,\mu)
ho$

・ロン ・回 と ・ ヨン ・ ヨン

Language Semantics Calculus Properties

Interpretation

Denotation of probability terms

$$\begin{bmatrix} r \end{bmatrix}_{K,\mu}^{\rho} = r \\ \begin{bmatrix} y \end{bmatrix}_{K,\mu}^{\rho} = \rho(y) \\ \begin{bmatrix} (\int \gamma) \end{bmatrix}_{K,\mu}^{\rho} = \mu(|\gamma|_{\mathcal{V}}) \\ \begin{bmatrix} p_{1} + p_{2} \end{bmatrix}_{K,\mu}^{\rho} = \begin{bmatrix} p_{1} \end{bmatrix}_{K,\mu}^{\rho} + \begin{bmatrix} p_{2} \end{bmatrix}_{K,\mu}^{\rho} \\ \begin{bmatrix} p_{1} p_{2} \end{bmatrix}_{K,\mu}^{\rho} = \begin{bmatrix} p_{1} \end{bmatrix}_{K,\mu}^{\rho} \times \begin{bmatrix} p_{2} \end{bmatrix}_{K,\mu}^{\rho}$$

Satisfaction of probabilistic formulas $(K,\mu)\rho \Vdash (\rho_1 \le \rho_2) \quad \text{iff} \quad \llbracket \rho_1 \rrbracket_{K,\mu}^{\rho} \le \llbracket \rho_2 \rrbracket_{K,\mu}^{\rho}$ $(K,\mu)\rho \Vdash \# \text{fff}$ $(K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad \text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho$

・ロト ・回ト ・ヨト ・ヨト

Language Semantics Calculus Properties

Interpretation

Denotation of probability terms

$$\begin{bmatrix} r \end{bmatrix}_{K,\mu}^{\rho} = r \\ \begin{bmatrix} y \end{bmatrix}_{K,\mu}^{\rho} = \rho(y) \\ \begin{bmatrix} (\int \gamma) \end{bmatrix}_{K,\mu}^{\rho} = \mu(|\gamma|_{\mathcal{V}}) \\ \begin{bmatrix} p_{1} + p_{2} \end{bmatrix}_{K,\mu}^{\rho} = \begin{bmatrix} p_{1} \end{bmatrix}_{K,\mu}^{\rho} + \begin{bmatrix} p_{2} \end{bmatrix}_{K,\mu}^{\rho} \\ \begin{bmatrix} p_{1} p_{2} \end{bmatrix}_{K,\mu}^{\rho} = \begin{bmatrix} p_{1} \end{bmatrix}_{K,\mu}^{\rho} \times \begin{bmatrix} p_{2} \end{bmatrix}_{K,\mu}^{\rho}$$

Satisfaction of probabilistic formulas $(K,\mu)\rho \Vdash (p_1 \le p_2) \quad \text{iff} \quad \llbracket p_1 \rrbracket_{K,\mu}^{\rho} \le \llbracket p_2 \rrbracket_{K,\mu}^{\rho}$ $(K,\mu)\rho \nvDash (H) \quad \text{iff} \quad (K,\mu)\rho \vDash \eta_2 \text{ or } (K,\mu)\rho \amalg \eta_2 \text{ or$

・ロト ・回ト ・ヨト ・ヨト

Language Semantics Calculus Properties

Interpretation

Denotation of probability terms

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

Satisfaction of probabilistic formulas $(K,\mu)\rho \Vdash (p_1 \le p_2) \quad \text{iff} \quad \llbracket p_1 \rrbracket_{K,\mu}^{\rho} \le \llbracket p_2 \rrbracket_{K,\mu}^{\rho}$ $(K,\mu)\rho \nvDash \text{fff}$ $(K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad \text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho$

・ロト ・回ト ・ヨト ・ヨト

э

Language Semantics Calculus Properties

Interpretation

Denotation of probability terms

Satisfaction of probabilistic formulas

 $\begin{aligned} (K,\mu)\rho \Vdash (p_1 \leq p_2) & \text{iff} \quad \llbracket p_1 \rrbracket_{K,\mu}^{\rho} \leq \llbracket p_2 \rrbracket_{K,\mu}^{\rho} \\ (K,\mu)\rho \nvDash \text{ fff} \\ (K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) & \text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho \nvDash \eta_1 \end{aligned}$

・ロン ・回 と ・ ヨ と ・ ヨ と …

э

Language Semantics Calculus Properties

Interpretation

Denotation of probability terms

Satisfaction of probabilistic formulas

 $\begin{aligned} (K,\mu)\rho \Vdash (p_1 \leq p_2) & \text{iff} \quad \llbracket p_1 \rrbracket_{K,\mu}^{\rho} \leq \llbracket p_2 \rrbracket_{K,\mu}^{\rho} \\ (K,\mu)\rho \nvDash \text{fff} \\ (K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) & \text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu) \end{aligned}$

3

Language Semantics Calculus Properties

Interpretation

Denotation of probability terms

$$\begin{bmatrix} r \end{bmatrix}_{K,\mu}^{\rho} = r \\ \begin{bmatrix} y \end{bmatrix}_{K,\mu}^{\rho} = \rho(y) \\ \begin{bmatrix} (\int \gamma) \end{bmatrix}_{K,\mu}^{\rho} = \mu(|\gamma|_{\mathcal{V}}) \\ \begin{bmatrix} p_1 + p_2 \end{bmatrix}_{K,\mu}^{\rho} = \begin{bmatrix} p_1 \end{bmatrix}_{K,\mu}^{\rho} + \begin{bmatrix} p_2 \end{bmatrix}_{K,\mu}^{\rho} \\ \begin{bmatrix} p_1 p_2 \end{bmatrix}_{K,\mu}^{\rho} = \begin{bmatrix} p_1 \end{bmatrix}_{K,\mu}^{\rho} \times \begin{bmatrix} p_2 \end{bmatrix}_{K,\mu}^{\rho}$$

Satisfaction of probabilistic formulas

$$\begin{split} & (K,\mu)\rho \Vdash (p_1 \leq p_2) \quad \text{iff} \quad \llbracket p_1 \rrbracket_{K,\mu}^{\rho} \leq \llbracket p_2 \rrbracket_{K,\mu}^{\rho} \\ & (K,\mu)\rho \nvDash \text{ fff} \\ & (K,\mu)\rho \Vdash (\eta_1 \supset \eta_2) \quad \text{iff} \quad (K,\mu)\rho \Vdash \eta_2 \text{ or } (K,\mu)\rho \nvDash \eta_1 \end{split}$$

・ロト ・回ト ・ヨト ・ヨト

Э

Language Semantics Calculus Properties

Auxiliary notions

Definition

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

Example

$$((\textbf{x1} \le \textbf{x2}) \land (\textbf{x1} > 0)) \Rightarrow (\textbf{x1}^2 \le \textbf{x2}^2)$$

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

・ロ・ ・ 日・ ・ ヨ・ ・ 日・

Э

Language Semantics Calculus Properties

Auxiliary notions

Definition

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

Example

$$((\texttt{x1} \leq \texttt{x2}) \land (\texttt{x1} > \texttt{0})) \Rightarrow (\texttt{x1}^2 \leq \texttt{x2}^2)$$

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

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

3

Language Semantics Calculus Properties

Auxiliary notions

Definition

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

Example

$$((\texttt{x1} \le \texttt{x2}) \land (\texttt{x1} > \texttt{0})) \Rightarrow (\texttt{x1}^2 \le \texttt{x2}^2)$$

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

・ロン ・回 と ・ ヨ と ・ ヨ と

Language Semantics Calculus Properties

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 \bot by fff, \rightarrow by \supset and each propositional symbol (uniformly) by a probabilistic state formula.

Example $((\int (x_1 \le x_2)) < 1) \supset (((\int (x_1 \le x_2)) < 1) \cap \texttt{tt})$

・ロン ・回 と ・ ヨ と ・ ヨ と

Language Semantics Calculus Properties

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 \bot by fff, \rightarrow by \supset and each propositional symbol (uniformly) by a probabilistic state formula.

Example

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

・ロト ・回ト ・ヨト ・ヨト

Language Semantics Calculus Properties

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

 $((y_1 \le y_2) \land (y_1 > 0)) \supset (y_1^2 \le y_2^2)$

The set of valid analytical formulas is decidable.

・ロト ・回ト ・ヨト ・ヨト

Language Semantics Calculus Properties

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

$$((y_1 \le y_2) \land (y_1 > 0)) \supset (y_1^2 \le y_2^2)$$

The set of valid analytical formulas is decidable.

() < </p>

Language Semantics Calculus Properties

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

$$((y_1 \le y_2) \land (y_1 > 0)) \supset (y_1^2 \le y_2^2)$$

The set of valid analytical formulas is decidable.

・ロト ・回ト ・ヨト ・ヨト

Language Semantics Calculus Properties

Calculus

Axioms

- $[extsf{CTaut}] \hspace{0.2cm} \vdash \hspace{0.2cm} (\Box\gamma)$ for each valid state formula γ
- $[{f PTaut}] \hspace{.1in} \vdash \hspace{.1in} \eta$ for each probabilistic tautology η
 - $[\mathbf{RCF}] \vdash \kappa_{\vec{p}}^{\vec{y}}$ for any valid analytical formula κ

$$\begin{split} [\mathsf{Meas}\emptyset] &\vdash ((\int \mathrm{ff}) = 0) \\ [\mathsf{FAdd}] &\vdash (((\int (\gamma_1 \land \gamma_2)) = 0) \supset ((\int (\gamma_1 \lor \gamma_2)) = (\int \gamma_1) + (\int \gamma_2))) \\ [\mathsf{Mon}] &\vdash ((\Box(\gamma_1 \Rightarrow \gamma_2)) \supset ((\int \gamma_1) \le (\int \gamma_2))) \end{split}$$

Inference rule

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

・ロン ・回 と ・ ヨ と ・ ヨ と

3

Language Semantics Calculus Properties

Calculus

Axioms

- $[\textbf{CTaut}] \hspace{0.1in}\vdash \hspace{0.1in} (\Box\gamma) \hspace{0.1in} \text{for each valid state formula} \hspace{0.1in} \gamma$
- $[\textbf{PTaut}] \hspace{.1in} \vdash \hspace{.1in} \eta \hspace{.1in} \text{for each probabilistic tautology} \hspace{.1in} \eta$
 - $[\mathbf{RCF}] \vdash \kappa_{\vec{p}}^{\vec{y}}$ for any valid analytical formula κ

$$\begin{split} [\mathsf{Meas}\emptyset] &\vdash ((\int \mathrm{ff}) = 0) \\ [\mathsf{FAdd}] &\vdash ((((\int (\gamma_1 \land \gamma_2)) = 0) \supset ((\int (\gamma_1 \lor \gamma_2)) = (\int \gamma_1) + (\int \gamma_2))) \\ [\mathsf{Mon}] &\vdash ((\Box(\gamma_1 \Rightarrow \gamma_2)) \supset ((\int \gamma_1) \le (\int \gamma_2))) \end{split}$$

Inference rule

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

(ロ) (同) (E) (E) (E)

Language Semantics Calculus Properties

Calculus

Axioms

- **[CTaut]** \vdash ($\Box \gamma$) for each valid state formula γ
- $[\textbf{PTaut}] \hspace{.1in} \vdash \hspace{.1in} \eta \hspace{.1in} \text{for each probabilistic tautology} \hspace{.1in} \eta$
 - $[\mathbf{RCF}] \vdash \kappa_{\vec{p}}^{\vec{y}}$ for any valid analytical formula κ

$$\begin{split} & [\mathsf{Meas}\emptyset] \ \vdash \ ((\int \mathrm{ff}) = 0) \\ & [\mathsf{FAdd}] \ \vdash \ (((\int (\gamma_1 \land \gamma_2)) = 0) \supset ((\int (\gamma_1 \lor \gamma_2)) = (\int \gamma_1) + (\int \gamma_2))) \\ & [\mathsf{Mon}] \ \vdash \ ((\Box(\gamma_1 \Rightarrow \gamma_2)) \supset ((\int \gamma_1) \le (\int \gamma_2))) \end{split}$$

Inference rule

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

(ロ) (同) (E) (E) (E)

Language Semantics Calculus Properties

Calculus

Axioms

- $[\textbf{CTaut}] \hspace{0.1in}\vdash \hspace{0.1in} (\Box\gamma) \hspace{0.1in} \text{for each valid state formula} \hspace{0.1in} \gamma$
- $[\textbf{PTaut}] \hspace{.1in} \vdash \hspace{.1in} \eta \hspace{.1in} \text{for each probabilistic tautology} \hspace{.1in} \eta$
 - $[\mathbf{RCF}] \vdash \kappa_{\vec{p}}^{\vec{y}}$ for any valid analytical formula κ

$$\begin{split} & [\mathsf{Meas}\emptyset] \ \vdash \ ((\int \mathrm{ff}) = 0) \\ & [\mathsf{FAdd}] \ \vdash \ (((\int (\gamma_1 \land \gamma_2)) = 0) \supset ((\int (\gamma_1 \lor \gamma_2)) = (\int \gamma_1) + (\int \gamma_2))) \\ & [\mathsf{Mon}] \ \vdash \ ((\Box(\gamma_1 \Rightarrow \gamma_2)) \supset ((\int \gamma_1) \le (\int \gamma_2))) \end{split}$$

Inference rule

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

(ロ) (同) (E) (E) (E)

Language Semantics Calculus Properties

Soundness

Theorem

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

Proof.

Straightforward from the definition of the semantics.

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

Э

Language Semantics Calculus Properties

Soundness

Theorem

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

Proof.

Straightforward from the definition of the semantics.

・ロト ・同ト ・ヨト ・ヨト

Language Semantics Calculus Properties

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.

・ロト ・回ト ・ヨト ・ヨト

Language Semantics Calculus Properties

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.

・ロン ・回 と ・ ヨ と ・ ヨ と …

Language Semantics Calculus Properties

Construction of the model

- 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 $\sum_{\nu \Vdash_c \gamma, \nu \in V} y_{\nu}$ where y_{ν} represents the probability of the valuation ν ;
- 3 let κ be the analytical formula $\kappa_1 \cap \bigcap_{y_v | v \in V} (0 \le y_v);$
- η is consistent iff κ is;
- finally, consistency of κ is decided by the axiom RCF and the model is constructed for a consistent κ by solving for y_ν in real closed fields.

・ロト ・同ト ・ヨト ・ヨト

Language Semantics Calculus Properties

Construction of the model

- 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;
- e let κ₁ be the analytical formula obtained from η by effectively replacing measure terms (∫γ) by sums ∑_{v ⊢ cγ, v ∈ V} y_v where y_v represents the probability of the valuation v;
- **③** let κ be the analytical formula $\kappa_1 \cap \bigcap_{v_v | v \in V} (0 \le y_v)$;
- () η is consistent iff κ is;
- Inially, consistency of κ is decided by the axiom RCF and the model is constructed for a consistent κ by solving for y_ν in real closed fields.

・ロト ・同ト ・ヨト ・ヨト

Language Semantics Calculus Properties

Construction of the model

- 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;
- e let κ₁ be the analytical formula obtained from η by effectively replacing measure terms (∫γ) by sums ∑_{v ⊢ cγ, v ∈ V} y_v where y_v represents the probability of the valuation v;
- **③** let κ be the analytical formula $\kappa_1 \cap \bigcap_{y_v | v \in V} (0 \le y_v)$;
- η is consistent iff κ is;
- inally, consistency of κ is decided by the axiom RCF and the model is constructed for a consistent κ by solving for y_ν in real closed fields.

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

Language Semantics Calculus Properties

Construction of the model

- 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;
- e let κ₁ be the analytical formula obtained from η by effectively replacing measure terms (∫γ) by sums ∑_{v ⊢ cγ, v ∈ V} y_v where y_v represents the probability of the valuation v;
- **3** let κ be the analytical formula $\kappa_1 \cap \bigcap_{y_v | v \in V} (0 \leq y_v)$;
- $\ \, {\bf 0} \ \ \eta \ \, {\rm is \ consistent \ \, iff \ \ \kappa \ \, is;}$
- in 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.

・ロン ・回 と ・ ヨ と ・ ヨ と

Language Semantics Calculus Properties

Construction of the model

- 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;
- e let κ₁ be the analytical formula obtained from η by effectively replacing measure terms (∫γ) by sums ∑_{v ⊢cγ,v∈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;
- finally, consistency of κ is decided by the axiom RCF and the model is constructed for a consistent κ by solving for y_ν in real closed fields.

Syntax Semantics

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).

・ロ・ ・ 日・ ・ ヨ・ ・ 日・

Syntax Semantics

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 Semantics

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).

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

Syntax Semantics

Notation

$\llbracket \gamma \rrbracket_{v} = \texttt{tt} \text{ if } v \Vdash_{\mathsf{c}} \gamma \text{ and } \llbracket \gamma \rrbracket_{v} = \texttt{ff} \text{ otherwise}$

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

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

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

Syntax Semantics

Notation

 $\llbracket \gamma \rrbracket_{v} = \texttt{tt} \text{ if } v \Vdash_{\mathsf{c}} \gamma \text{ and } \llbracket \gamma \rrbracket_{v} = \texttt{ff} \text{ otherwise}$

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

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

Syntax Semantics

Notation

 $\llbracket \gamma \rrbracket_{v} = t$ if $v \Vdash_{c} \gamma$ and $\llbracket \gamma \rrbracket_{v} = f$ otherwise

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

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

Syntax Semantics

Notation

 $\llbracket \gamma \rrbracket_{v} = \texttt{tt} \text{ if } v \Vdash_{\mathsf{c}} \gamma \text{ and } \llbracket \gamma \rrbracket_{v} = \texttt{ff} \text{ otherwise}$

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

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

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

Syntax Semantics

Denotation of programs

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

$$\begin{split} \begin{bmatrix} \mathsf{skip} \end{bmatrix} &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu) \\ \begin{bmatrix} \mathsf{xm} \leftarrow t \end{bmatrix} &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_t^{\mathsf{xm}})^{-1}) \\ \begin{bmatrix} \mathsf{bm} \leftarrow \gamma \end{bmatrix} &= \lambda(\mathcal{K},\mu).(\mathcal{K},\mu \circ (\delta_\gamma^{\mathsf{bm}})^{-1}) \\ \\ \begin{bmatrix} \mathsf{toss}(\mathsf{bm},r) \end{bmatrix} &= \lambda(\mathcal{K},\mu).(\widetilde{r}(\llbracket\mathsf{bm} \leftarrow \mathtt{t} \rrbracket(\mathcal{K},\mu)) + (1-\widetilde{r})(\llbracket\mathsf{bm} \leftarrow \mathtt{ff} \rrbracket(\mathcal{K},\mu))) \\ \\ \\ \begin{bmatrix} \mathsf{s}_1; \mathsf{s}_2 \end{bmatrix} &= \lambda(\mathcal{K},\mu).[\llbracket\mathsf{s}_2 \rrbracket(\llbracket\mathsf{s}_1 \rrbracket(\mathcal{K},\mu)) \\ \\ \mathsf{f} \gamma \text{ then } \mathsf{s}_1 \text{ else } \mathsf{s}_2 \end{bmatrix} &= \lambda(\mathcal{K},\mu).(\llbracket\mathsf{s}_1 \rrbracket(\mathcal{K},\mu_\gamma) + \llbracket\mathsf{s}_2 \rrbracket(\mathcal{K},\mu(-\gamma)))) \end{split}$$

・ロト ・回ト ・ヨト ・ヨト

э

Syntax Semantics

Denotation of programs

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

$$\begin{split} \llbracket \mathsf{skip} \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu) \\ \llbracket \mathsf{xm} \leftarrow t \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_t^{\mathsf{xm}})^{-1}) \\ \llbracket \mathsf{bm} \leftarrow \gamma \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_{\gamma}^{\mathsf{bm}})^{-1}) \\ \llbracket \mathsf{toss}(\mathsf{bm}, r) \rrbracket &= \lambda(\mathcal{K}, \mu).(\widetilde{r}(\llbracket \mathsf{bm} \leftarrow \mathtt{t} \rrbracket (\mathcal{K}, \mu)) + (1 - \widetilde{r})(\llbracket \mathsf{bm} \leftarrow \mathtt{ff} \rrbracket (\mathcal{K}, \mu))) \\ \llbracket \mathsf{s}_1; \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).[\llbracket \mathsf{s}_2 \rrbracket (\llbracket \mathsf{s}_1 \rrbracket (\mathcal{K}, \mu))) \\ \mathsf{f} \gamma \text{ then } \mathsf{s}_1 \text{ else } \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).(\llbracket \mathsf{s}_1 \rrbracket (\mathcal{K}, \mu_{\gamma}) + \llbracket \mathsf{s}_2 \rrbracket (\mathcal{K}, \mu_{(-\gamma)}))) \end{split}$$

・ロン ・回 と ・ ヨ と ・ ヨ と

Syntax Semantics

Denotation of programs

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

$$\begin{split} \llbracket \mathsf{skip} \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu) \\ \llbracket \mathsf{xm} \leftarrow t \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_t^{\mathsf{xm}})^{-1}) \\ \llbracket \mathsf{bm} \leftarrow \gamma \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_{\gamma}^{\mathsf{bm}})^{-1}) \\ \llbracket \mathsf{toss}(\mathsf{bm}, r) \rrbracket &= \lambda(\mathcal{K}, \mu).(\widetilde{r}(\llbracket \mathsf{bm} \leftarrow \mathsf{tm} \rrbracket (\mathcal{K}, \mu)) + (1 - \widetilde{r})(\llbracket \mathsf{bm} \leftarrow \mathsf{ff} \rrbracket (\mathcal{K}, \mu))) \\ \llbracket \mathsf{s}_1; \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).\llbracket \mathsf{s}_2 \rrbracket (\llbracket \mathsf{s}_1 \rrbracket (\mathcal{K}, \mu)) \\ \llbracket \mathsf{s}_1 \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).\llbracket \mathsf{s}_2 \rrbracket (\llbracket \mathsf{s}_1 \rrbracket (\mathcal{K}, \mu)) \\ \llbracket \mathsf{s}_1 \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).\llbracket \mathsf{s}_2 \rrbracket (\llbracket \mathsf{s}_1 \rrbracket (\mathcal{K}, \mu)) \\ \llbracket \mathsf{s}_1 \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).\llbracket \mathsf{s}_2 \rrbracket (\mathcal{K}, \mu) \\ \llbracket \mathsf{s}_1 \mathsf{s}_2 \rrbracket (\mathcal{K}, \mu) + \llbracket \mathsf{s}_2 \rrbracket (\mathcal{K}, \mu_{(\neg \gamma)})) \end{split}$$

・ロン ・回 と ・ ヨン ・ ヨン

Syntax Semantics

Denotation of programs

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

$$\begin{split} \llbracket \mathsf{skip} \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu) \\ \llbracket \mathsf{xm} \leftarrow t \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_t^{\mathsf{xm}})^{-1}) \\ \llbracket \mathsf{bm} \leftarrow \gamma \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_{\gamma}^{\mathsf{bm}})^{-1}) \\ \llbracket \mathsf{toss}(\mathsf{bm}, r) \rrbracket &= \lambda(\mathcal{K}, \mu).(\widetilde{r}(\llbracket \mathsf{bm} \leftarrow \mathsf{tt} \rrbracket(\mathcal{K}, \mu)) + (1 - \widetilde{r})(\llbracket \mathsf{bm} \leftarrow \mathsf{ft} \rrbracket(\mathcal{K}, \mu))) \\ \llbracket \mathsf{s}_1; \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).\llbracket \mathsf{s}_2 \rrbracket(\llbracket \mathsf{s}_1 \rrbracket(\mathcal{K}, \mu)) \\ \gamma \text{ then } \mathsf{s}_1 \text{ else } \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).(\llbracket \mathsf{s}_1 \rrbracket(\mathcal{K}, \mu_{\gamma}) + \llbracket \mathsf{s}_2 \rrbracket(\mathcal{K}, \mu_{(-\gamma)}))) \end{split}$$

・ロト ・回ト ・ヨト ・ヨト

Syntax Semantics

Denotation of programs

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

$$\begin{split} \llbracket \mathsf{skip} \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu) \\ \llbracket \mathsf{xm} \leftarrow t \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_t^{\mathsf{xm}})^{-1}) \\ \llbracket \mathsf{bm} \leftarrow \gamma \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_{\gamma}^{\mathsf{bm}})^{-1}) \\ \llbracket \mathsf{toss}(\mathsf{bm}, r) \rrbracket &= \lambda(\mathcal{K}, \mu).(\widetilde{r}(\llbracket \mathsf{bm} \leftarrow \mathsf{tt} \rrbracket(\mathcal{K}, \mu)) + (1 - \widetilde{r})(\llbracket \mathsf{bm} \leftarrow \mathsf{ft} \rrbracket(\mathcal{K}, \mu))) \\ \llbracket \mathfrak{s}_1; \mathfrak{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).\llbracket \mathfrak{s}_2 \rrbracket(\llbracket \mathfrak{s}_1 \rrbracket(\mathcal{K}, \mu)) \\ \mathsf{then} \ \mathfrak{s}_1 \ \mathsf{else} \ \mathfrak{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).(\llbracket \mathfrak{s}_1 \rrbracket(\mathcal{K}, \mu_{\gamma}) + \llbracket \mathfrak{s}_2 \rrbracket(\mathcal{K}, \mu_{(\neg \gamma)})) \end{split}$$

・ロン ・回 と ・ ヨ と ・ ヨ と

Syntax Semantics

Denotation of programs

∏it

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

$$\begin{split} \llbracket \mathsf{skip} \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu) \\ \llbracket \mathsf{xm} \leftarrow t \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_t^{\mathsf{xm}})^{-1}) \\ \llbracket \mathsf{bm} \leftarrow \gamma \rrbracket &= \lambda(\mathcal{K}, \mu).(\mathcal{K}, \mu \circ (\delta_{\gamma}^{\mathsf{bm}})^{-1}) \\ \llbracket \mathsf{toss}(\mathsf{bm}, r) \rrbracket &= \lambda(\mathcal{K}, \mu).(\widetilde{r}(\llbracket \mathsf{bm} \leftarrow \mathsf{tt} \rrbracket(\mathcal{K}, \mu)) + (1 - \widetilde{r})(\llbracket \mathsf{bm} \leftarrow \mathsf{ft} \rrbracket(\mathcal{K}, \mu))) \\ \llbracket \mathsf{s}_1; \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).\llbracket \mathsf{s}_2 \rrbracket(\llbracket \mathsf{s}_1 \rrbracket(\mathcal{K}, \mu)) \\ \mathsf{f} \gamma \text{ then } \mathsf{s}_1 \text{ else } \mathsf{s}_2 \rrbracket &= \lambda(\mathcal{K}, \mu).(\llbracket \mathsf{s}_1 \rrbracket(\mathcal{K}, \mu_{\gamma}) + \llbracket \mathsf{s}_2 \rrbracket(\mathcal{K}, \mu_{(\neg \gamma)}))) \end{split}$$

・ロン ・回 と ・ ヨ と ・ ヨ と

The calculus Soundness Completeness

Hoare assertions

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

$(\mathcal{K},\mu)\rho \Vdash_{h} \eta \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta$ $(\mathcal{K},\mu)\rho \Vdash_{h} \{\eta_{1}\} s \{\eta_{2}\} \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta_{2} \text{ whenever } \llbracket s \rrbracket (\mathcal{K},\mu)\rho \Vdash \eta_{1}$

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}, μ) and any \mathcal{K} -assignment ρ .

(ロ) (同) (E) (E)

臣

The calculus Soundness Completeness

Hoare assertions

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

$(\mathcal{K},\mu)\rho \Vdash_{h} \eta \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta$ $(\mathcal{K},\mu)\rho \Vdash_{h} \{\eta_{1}\} s \{\eta_{2}\} \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta_{2} \text{ whenever } \llbracket s \rrbracket (\mathcal{K},\mu)\rho \Vdash \eta_{1}$

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}, μ) and any \mathcal{K} -assignment ρ .

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Hoare assertions

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

$$(\mathcal{K},\mu)\rho \Vdash_{h} \eta \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta$$
$$(\mathcal{K},\mu)\rho \Vdash_{h} \{\eta_{1}\} s \{\eta_{2}\} \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta_{2} \text{ whenever } \llbracket s \rrbracket (\mathcal{K},\mu)\rho \Vdash \eta_{1}$$

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}, μ) and any \mathcal{K} -assignment ρ .

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

The calculus Soundness Completeness

Hoare assertions

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

$$(\mathcal{K},\mu)\rho \Vdash_{h} \eta \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta$$
$$(\mathcal{K},\mu)\rho \Vdash_{h} \{\eta_{1}\} s \{\eta_{2}\} \quad \text{if} \quad (\mathcal{K},\mu)\rho \Vdash \eta_{2} \text{ whenever } \llbracket s \rrbracket (\mathcal{K},\mu)\rho \Vdash \eta_{1}$$

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}, μ) and any \mathcal{K} -assignment ρ .

(ロ) (同) (E) (E)

臣

The calculus Soundness Completeness

Tossed terms

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

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

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

The calculus Soundness Completeness

Tossed terms

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

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

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

The calculus Soundness Completeness

Tossed terms

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

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

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

The calculus Soundness Completeness

Tossed formulas

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

The formula toss(**bm**, r; η) is the formula obtained from η by replacing every occurrence of each measure term $(\int \gamma)$ by $\tilde{r}(\int \gamma_{\rm tt}^{\rm bm}) + (1 - \tilde{r})(\int \gamma_{\rm ff}^{\rm bm})$.

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

The calculus Soundness Completeness

Tossed formulas

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

The formula toss(**bm**, $r; \eta$) is the formula obtained from η by replacing every occurrence of each measure term $(\int \gamma)$ by $\tilde{r}(\int \gamma_{\rm tt}^{\rm bm}) + (1 - \tilde{r})(\int \gamma_{\rm ft}^{\rm bm})$.

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

The calculus Soundness Completeness

Tossed formulas

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

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

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

The calculus Soundness Completeness

Conditioned terms

Let γ be classical state formula and ${\it p}$ 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 $(\int (\gamma' \land \gamma))$.

$$r/\gamma = r$$

$$y/\gamma = y$$

$$(\int \gamma')/\gamma = (\int (\gamma \land \gamma'))$$

$$(p + p')/\gamma = (p/\gamma + p'/\gamma)$$

$$(pp')/\gamma = ((p/\gamma)(p'/\gamma))$$

・ロン ・回 と ・ ヨ と ・ ヨ と

The calculus Soundness Completeness

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 $(\int (\gamma' \wedge \gamma))$.

$$r/\gamma = r$$

$$y/\gamma = y$$

$$(\int \gamma')/\gamma = (\int (\gamma \land \gamma'))$$

$$(p + p')/\gamma = (p/\gamma + p'/\gamma)$$

$$(pp')/\gamma = ((p/\gamma)(p'/\gamma))$$

・ロン ・回 と ・ ヨ と ・ ヨ と

The calculus Soundness Completeness

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 $(\int (\gamma' \wedge \gamma))$.

$$r/\gamma = r$$

$$y/\gamma = y$$

$$(\int \gamma')/\gamma = (\int (\gamma \land \gamma'))$$

$$(p + p')/\gamma = (p/\gamma + p'/\gamma)$$

$$(pp')/\gamma = ((p/\gamma)(p'/\gamma))$$

イロト イポト イヨト イヨト

The calculus Soundness Completeness

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 $(\int (\gamma' \land \gamma))$.

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

 $(\eta_1 \uparrow_{\gamma} \eta_2)$ stands for $((\eta_1/\gamma) \cap (\eta_2/(\neg \gamma)))$.

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

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 $(\int (\gamma' \wedge \gamma))$.

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

 $(\eta_1 \uparrow_{\gamma} \eta_2)$ stands for $((\eta_1/\gamma) \cap (\eta_2/(\neg \gamma)))$.

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

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 $(\int (\gamma' \wedge \gamma))$.

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

 $(\eta_1 \Upsilon_{\gamma} \eta_2)$ stands for $((\eta_1/\gamma) \cap (\eta_2/(\neg \gamma)))$.

The calculus Soundness Completeness

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 $(\int (\gamma' \wedge \gamma))$.

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

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

・ロト ・ 同ト ・ ヨト ・ ヨト

The calculus Soundness Completeness

Axioms

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

[SKIP] [ASGR] [ASGB] [TOSS] $\vdash \{\eta\} \operatorname{skip} \{\eta\}$

- $\vdash \{\eta_t^{\mathsf{xm}}\}\,\mathsf{xm} \leftarrow t\,\{\eta\}$
- $Dash \{\eta^{\mathsf{bm}}_\gamma\}\,\mathsf{bm} \leftarrow \gamma\,\{\eta\}$
 - $\vdash \{\mathsf{toss}(\mathsf{bm},\eta;r)\} \mathsf{toss}(\mathsf{bm},r) \{\eta\}$

・ロン ・回 と ・ ヨ と ・ ヨ と

臣

Axioms

The State Logic: EPPL The Programming Language The Hoare Calculus Conclusions

The calculus Soundness Completeness

 $[\textbf{TAUT}] \qquad \vdash \eta \qquad \text{if } \eta \text{ is an EPPL theorem} \\ [\int \textbf{FREE}] \qquad \vdash \{\kappa\} s \{\kappa\} \qquad \text{if } \kappa \text{ is an analytical formula}$

[SKIP] [ASGR] [ASGB] [TOSS]

 $\vdash \{\eta\} \text{ skip } \{\eta\}$ $\vdash \{\eta_t^{\text{xm}}\} \text{ xm } \leftarrow t \{\eta\}$ $\vdash \{\eta_{\gamma}^{\text{bm}}\} \text{ bm } \leftarrow \gamma \{\eta\}$ $\vdash \{\text{toss}(\text{bm}, \eta; r)\} \text{ toss}(\text{bm}, r) \{$

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

Axioms

The State Logic: EPPL The Programming Language The Hoare Calculus Conclusions

The calculus Soundness Completeness

 $\begin{array}{ll} [\textbf{TAUT}] & \vdash \eta & \text{ if } \eta \text{ is an EPPL theorem} \\ [\int \textbf{FREE}] & \vdash \{\kappa\} \, s \, \{\kappa\} & \text{ if } \kappa \text{ is an analytical formula} \end{array}$

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

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

臣

Axioms

The State Logic: EPPL The Programming Language The Hoare Calculus Conclusions

The calculus Soundness Completeness

 $[\textbf{TAUT}] \qquad \vdash \eta \qquad \text{if } \eta \text{ is an EPPL theorem} \\ [\int \textbf{FREE}] \qquad \vdash \{\kappa\} s \{\kappa\} \qquad \text{if } \kappa \text{ is an analytical formula} \end{cases}$

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

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

臣

Axioms

The State Logic: EPPL The Programming Language The Hoare Calculus Conclusions

The calculus Soundness Completeness

 $\begin{array}{ll} [\textbf{TAUT}] & \vdash \eta & \text{if } \eta \text{ is an EPPL theorem} \\ [\int \textbf{FREE}] & \vdash \{\kappa\} \, s \, \{\kappa\} & \text{if } \kappa \text{ is an analytical formula} \end{array}$

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

・ロン ・回 と ・ ヨ と ・ ヨ と

3

Axioms

The State Logic: EPPL The Programming Language The Hoare Calculus Conclusions

The calculus Soundness Completeness

 $\begin{array}{ll} [\textbf{TAUT}] & \vdash \eta & \text{if } \eta \text{ is an EPPL theorem} \\ [\int \textbf{FREE}] & \vdash \{\kappa\} \, s \, \{\kappa\} & \text{if } \kappa \text{ is an analytical formula} \end{array}$

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

イロン イヨン イヨン イヨン

3

The calculus Soundness Completeness

Inference rules

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

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

 $[\textbf{ELIMV}] \qquad \{\eta_1 \cap (y = p)\} s \{\eta_2\} \vdash \{\eta_1_p^y\} s \{\eta_2\}$ y does not occur in p or η_2

 $\begin{bmatrix} \text{CONS} \end{bmatrix} \quad \eta_0 \supset \eta_1, \{\eta_1\} \ s \ \{\eta_2\}, \eta_2 \supset \eta_3 \vdash \{\eta_0\} \ s \ \{\eta_3\} \\ \begin{bmatrix} \text{OR} \end{bmatrix} \quad \{\eta_0\} \ s \ \{\eta_2\}, \{\eta_1\} \ s \ \{\eta_2\} \vdash \{\eta_0 \cup \eta_1\} \ s \ \{\eta_2\} \\ \begin{bmatrix} \text{AND} \end{bmatrix} \quad \{\eta_0\} \ s \ \{\eta_1\}, \{\eta_0\} \ s \ \{\eta_2\} \vdash \{\eta_0\} \ s \ \{\eta_1 \bigcap \eta_2\} \\ \end{bmatrix}$

The calculus Soundness Completeness

Inference rules

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

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

- $[\textbf{ELIMV}] \qquad \{\eta_1 \cap (y = p)\} s \{\eta_2\} \vdash \{\eta_1_p^{\mathcal{V}}\} s \{\eta_2\}$ y does not occur in p or η_2
 - $\begin{array}{ll} [\text{CONS}] & \eta_0 \supset \eta_1, \{\eta_1\} \ s \ \{\eta_2\}, \eta_2 \supset \eta_3 \vdash \{\eta_0\} \ s \ \{\eta_3\} \\ [\text{OR}] & \{\eta_0\} \ s \ \{\eta_2\}, \{\eta_1\} \ s \ \{\eta_2\} \vdash \{\eta_0 \cup \eta_1\} \ s \ \{\eta_2\} \\ [\text{AND}] & \{\eta_0\} \ s \ \{\eta_1\}, \{\eta_0\} \ s \ \{\eta_2\} \vdash \{\eta_0\} \ s \ \{\eta_1 \bigcap \eta_2\} \\ \end{array}$

The calculus Soundness Completeness

Inference rules

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

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

$$[\textbf{ELIMV}] \qquad \{\eta_1 \cap (y = p)\} s \{\eta_2\} \vdash \{\eta_1_p^y\} s \{\eta_2\}$$

y does not occur in p or η_2

Luís Cruz-Filipe Reasoning about Probabilistic Sequential Programs

The calculus Soundness Completeness

Inference rules

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

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

$$[\textbf{ELIMV}] \qquad \{\eta_1 \cap (y = p)\} s \{\eta_2\} \vdash \{\eta_1_p^y\} s \{\eta_2\}$$

y does not occur in p or η_2

$$\begin{array}{ll} [\textbf{CONS}] & \eta_0 \supset \eta_1, \{\eta_1\} \, s \, \{\eta_2\}, \eta_2 \supset \eta_3 \vdash \{\eta_0\} \, s \, \{\eta_3\} \\ [\textbf{OR}] & \{\eta_0\} \, s \, \{\eta_2\}, \{\eta_1\} \, s \, \{\eta_2\} \vdash \{\eta_0 \cup \eta_1\} \, s \, \{\eta_2\} \\ [\textbf{AND}] & \{\eta_0\} \, s \, \{\eta_1\}, \{\eta_0\} \, s \, \{\eta_2\} \vdash \{\eta_0\} \, s \, \{\eta_1 \cap \eta_2\} \end{array}$$

The calculus Soundness Completeness

Substitution Lemma for classical valuations

_emma

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

 $v^m_{\llbracket e \rrbracket_v} \Vdash_{\mathsf{c}} \gamma \text{ iff } v \Vdash_{\mathsf{c}} \gamma^m_e.$

Proof.

Induction on the structure of γ .

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Substitution Lemma for classical valuations

Lemma

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

$$v^m_{\llbracket e \rrbracket_v} \Vdash_{\mathsf{c}} \gamma \text{ iff } v \Vdash_{\mathsf{c}} \gamma^m_e.$$

Proof.

Induction on the structure of γ .

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

The calculus Soundness Completeness

Substitution Lemma for classical valuations

Lemma

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

$$v^m_{\llbracket e \rrbracket_v} \Vdash_{\mathsf{c}} \gamma \text{ iff } v \Vdash_{\mathsf{c}} \gamma^m_e.$$

Proof.

Induction on the structure of γ .

・ロ・ ・ 日・ ・ ヨ・ ・ 日・

The calculus Soundness Completeness

Substitution Lemma for assignment

Lemma

Let (\mathcal{K}, μ) 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

 $\llbracket (\int \gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \llbracket (\int \gamma^m_e) \rrbracket^{\rho}_{(\mathcal{K},\mu)}$

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

 $\llbracket p \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \llbracket p_e^m \rrbracket^{\rho}_{(\mathcal{K},\mu)},$

and, for any probabilistic formula η ,

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

The calculus Soundness Completeness

Substitution Lemma for assignment

Lemma

Let (\mathcal{K}, μ) 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

$$\llbracket (\int \gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \llbracket (\int \gamma^m_e) \rrbracket^{\rho}_{(\mathcal{K},\mu)}$$

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

 $\llbracket p \rrbracket_{(\mathcal{K},\mu')}^{\rho} = \llbracket p_e^m \rrbracket_{(\mathcal{K},\mu)}^{\rho},$

and, for any probabilistic formula η ,

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

The calculus Soundness Completeness

Substitution Lemma for assignment

Lemma

Let (\mathcal{K}, μ) 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

$$\llbracket (\int \gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \llbracket (\int \gamma^m_e) \rrbracket^{\rho}_{(\mathcal{K},\mu)}$$

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

$$\llbracket p \rrbracket_{(\mathcal{K},\mu')}^{\rho} = \llbracket p_e^m \rrbracket_{(\mathcal{K},\mu)}^{\rho},$$

and, for any probabilistic formula η ,

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

The calculus Soundness Completeness

Substitution Lemma for assignment

Proof.

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

Therefore, by definition

$$\llbracket (\int \gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \mu \circ (\delta^m_e)^{-1} (|\gamma|_{\mathcal{V}}) = \mu (|\gamma^m_e|_{\mathcal{V}}) = \llbracket (\int \gamma^m_e) \rrbracket^{\rho}_{(\mathcal{K},\mu)}$$

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

Corollary

Axioms ASGB and ASGR are sound.

The calculus Soundness Completeness

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

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

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

Corollary

Axioms ASGB and ASGR are sound

The calculus Soundness Completeness

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,

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

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

Corollary

Axioms ASGB and ASGR are sound.

The calculus Soundness Completeness

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,

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

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

Corollary

Axioms ASGB and ASGR are sound.

・ロン ・四 ・ ・ ヨ ・ ・ ヨ ・ ・

Э

The calculus Soundness Completeness

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,

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

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

Corollary

Axioms ASGB and ASGR are sound.

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

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,

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

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

Corollary

Axioms ASGB and ASGR are sound

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

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,

$$\llbracket (\int \gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \mu \circ (\delta^{m}_{e})^{-1} (|\gamma|_{\mathcal{V}}) = \mu (|\gamma^{m}_{e}|_{\mathcal{V}}) = \llbracket (\int \gamma^{m}_{e}) \rrbracket^{\rho}_{(\mathcal{K},\mu)}.$$

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

Corollary

Axioms ASGB and ASGR are sound.

<ロ> (日) (日) (日) (日) (日)

Э

The calculus Soundness Completeness

Substitution Lemma for probabilistic tosses

Lemma

Let (K, μ) and ρ be as before, $r \in \mathcal{A}$ be a constant and $\mu' = \tilde{r}\mu \circ (\delta_{t}^{bm})^{-1} + (1 - \tilde{r})\mu \circ (\delta_{ff}^{bm})^{-1}$.

For any classical state formula γ ,

 $\llbracket (\int \gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r} \llbracket (\int \gamma^{\mathsf{bm}}_{\mathsf{t}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)} + (1 - \widetilde{r}) \llbracket (\int \gamma^{\mathsf{bm}}_{\mathrm{ff}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)}$

Furthermore, for any probabilistic term p,

 $\llbracket p \rrbracket_{(\mathcal{K},\mu')}^{\rho} = \llbracket \mathsf{toss}(\mathsf{bm}, r; p) \rrbracket_{(\mathcal{K},\mu)}^{\rho},$

and, for any probabilistic formula η ,

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

The calculus Soundness Completeness

Substitution Lemma for probabilistic tosses

Lemma

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

For any classical state formula γ ,

 $\llbracket (\int \gamma) \rrbracket_{(\mathcal{K},\mu')}^{\rho} = \widetilde{r} \llbracket (\int \gamma_{\texttt{tt}}^{\texttt{bm}}) \rrbracket_{(\mathcal{K},\mu)}^{\rho} + (1 - \widetilde{r}) \llbracket (\int \gamma_{\texttt{ff}}^{\texttt{bm}}) \rrbracket_{(\mathcal{K},\mu)}^{\rho}$

Furthermore, for any probabilistic term p,

 $\llbracket p \rrbracket_{(\mathcal{K},\mu')}^{\rho} = \llbracket \mathsf{toss}(\mathsf{bm}, r; p) \rrbracket_{(\mathcal{K},\mu)}^{\rho},$

and, for any probabilistic formula η ,

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

The calculus Soundness Completeness

Substitution Lemma for probabilistic tosses

Lemma

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

For any classical state formula γ ,

$$\llbracket (\int \gamma) \rrbracket_{(\mathcal{K},\mu')}^{\rho} = \widetilde{r} \llbracket (\int \gamma_{\mathfrak{t}}^{\mathsf{bm}}) \rrbracket_{(\mathcal{K},\mu)}^{\rho} + (1-\widetilde{r}) \llbracket (\int \gamma_{\mathrm{ff}}^{\mathsf{bm}}) \rrbracket_{(\mathcal{K},\mu)}^{\rho}.$$

Furthermore, for any probabilistic term p,

$$\llbracket p \rrbracket_{(\mathcal{K},\mu')}^{\rho} = \llbracket \mathsf{toss}(\mathsf{bm}, r; p) \rrbracket_{(\mathcal{K},\mu)}^{\rho},$$

and, for any probabilistic formula η ,

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

The calculus Soundness Completeness

Substitution Lemma for probabilistic tosses

Lemma

Let
$$(K, \mu)$$
 and ρ be as before, $r \in \mathcal{A}$ be a constant and $\mu' = \tilde{r}\mu \circ (\delta_{\mathtt{t}}^{\mathtt{bm}})^{-1} + (1 - \tilde{r})\mu \circ (\delta_{\mathtt{ff}}^{\mathtt{bm}})^{-1}$.

For any classical state formula γ ,

$$\llbracket (\int \gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r} \llbracket (\int \gamma^{\mathsf{bm}}_{\mathsf{tt}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)} + (1 - \widetilde{r}) \llbracket (\int \gamma^{\mathsf{bm}}_{\mathsf{ff}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)}.$$

Furthermore, for any probabilistic term p,

$$\llbracket p \rrbracket_{(\mathcal{K},\mu')}^{\rho} = \llbracket \mathsf{toss}(\mathbf{bm}, r; p) \rrbracket_{(\mathcal{K},\mu)}^{\rho},$$

and, for any probabilistic formula η ,

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

The calculus Soundness Completeness

Substitution Lemma for probabilistic tosses

Proof.

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

 $\llbracket (\int \gamma) \rrbracket_{(\mathcal{K},\mu')}^{\rho} = \widetilde{r} \llbracket (\int \gamma) \rrbracket_{(\mathcal{K},\mu_1)}^{\rho} + (1 - \widetilde{r}) \llbracket (\int \gamma) \rrbracket_{(\mathcal{K},\mu_2)}^{\rho}$

by definition. Also

 $\llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_1)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathtt{t}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)} \text{ and } \llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_2)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathtt{ff}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)}$

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

Corollary

Axiom TOSS is sound.

The calculus Soundness Completeness

Substitution Lemma for probabilistic tosses

Proof.

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

$$\llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r} \llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_1)} + (1-\widetilde{r}) \llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_2)}$$

by definition. Also

 $\llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_1)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathsf{tt}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)} \text{ and } \llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_2)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathrm{ff}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)}.$

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

Corollary Axiom **TOSS** is sound.

The calculus Soundness Completeness

Substitution Lemma for probabilistic tosses

Proof.

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

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

by definition. Also

$$\llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_1)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathsf{tt}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)} \text{ and } \llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_2)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathsf{ff}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)}.$$

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

Corollary Axiom **TOSS** is sound.

-

The calculus Soundness Completeness

Substitution Lemma for probabilistic tosses

Proof.

Let
$$\mu_1 = \mu \circ (\delta_t^{bm})^{-1}$$
 and $\mu_2 = \mu \circ (\delta_f^{bm})^{-1}$. Then

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

by definition. Also

$$\llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_1)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathfrak{t}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)} \text{ and } \llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_2)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathrm{ff}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)}.$$

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

Corollary

Axiom **TOSS** is sound.

-

The calculus Soundness Completeness

Substitution Lemma for probabilistic tosses

Proof.

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

$$\llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu')} = \widetilde{r} \llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_1)} + (1-\widetilde{r}) \llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_2)}$$

by definition. Also

$$\llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_1)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathfrak{t}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)} \text{ and } \llbracket (f\gamma) \rrbracket^{\rho}_{(\mathcal{K},\mu_2)} = \llbracket (f\gamma^{\mathsf{bm}}_{\mathrm{ff}}) \rrbracket^{\rho}_{(\mathcal{K},\mu)}.$$

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

Corollary Axiom TOSS is sound.

The calculus Soundness Completeness

Soundness of $\int FREE$

_emma

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

$(\llbracket s \rrbracket(\mathcal{K},\mu))\rho \Vdash \kappa \text{ iff } (\mathcal{K},\mu)\rho \Vdash \kappa.$

Proof.

The interpretation of analytical formulas depends only on ho_2

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Soundness of $\int FREE$

Lemma

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

$(\llbracket s \rrbracket(\mathcal{K},\mu))\rho \Vdash \kappa \text{ iff } (\mathcal{K},\mu)\rho \Vdash \kappa.$

Proof.

The interpretation of analytical formulas depends only on ho.

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Soundness of $\int FREE$

Lemma

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

 $(\llbracket s \rrbracket(\mathcal{K},\mu))\rho \Vdash \kappa \text{ iff } (\mathcal{K},\mu)\rho \Vdash \kappa.$

Proof.

The interpretation of analytical formulas depends only on ρ .

・ロン ・回 と ・ ヨン ・ ヨン

The calculus Soundness Completeness

Soundness of $\ensuremath{\mathsf{IF}}$

_emma

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

 $\llbracket (\int \gamma') / \gamma \rrbracket^{\rho}_{(\mathcal{K},\mu)} = \llbracket (\int \gamma') \rrbracket^{\rho}_{(\mathcal{K},\mu_{\gamma})}$

Furthermore, for any probability term p,

$$\llbracket p/\gamma \rrbracket^{\rho}_{(\mathcal{K},\mu)} = \llbracket p \rrbracket^{\rho}_{(\mathcal{K},\mu_{\gamma})},$$

and, for any probabilistic formula η ,

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

・ロン ・雪と ・ほと ・ほと

Э

The calculus Soundness Completeness

Soundness of $\ensuremath{\mathsf{IF}}$

Lemma

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

 $\llbracket (\int \gamma') / \gamma \rrbracket^{\rho}_{(\mathcal{K},\mu)} = \llbracket (\int \gamma') \rrbracket^{\rho}_{(\mathcal{K},\mu_{\gamma})}.$

Furthermore, for any probability term p,

 $\llbracket p/\gamma \rrbracket^{\rho}_{(\mathcal{K},\mu)} = \llbracket p \rrbracket^{\rho}_{(\mathcal{K},\mu_{\gamma})},$

and, for any probabilistic formula η ,

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

・ロン ・四マ ・ヨン ・ヨン

The calculus Soundness Completeness

Soundness of $\ensuremath{\mathsf{IF}}$

Lemma

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

$$\llbracket (\int \gamma') / \gamma \rrbracket^{\rho}_{(\mathcal{K},\mu)} = \llbracket (\int \gamma') \rrbracket^{\rho}_{(\mathcal{K},\mu_{\gamma})}$$

Furthermore, for any probability term p,

$$\llbracket p/\gamma \rrbracket^{\rho}_{(\mathcal{K},\mu)} = \llbracket p \rrbracket^{\rho}_{(\mathcal{K},\mu_{\gamma})},$$

and, for any probabilistic formula η ,

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

・ロン ・回 と ・ 回 と ・ 回 と

э

The calculus Soundness Completeness

Soundness of IF

Proof.

By definition,

$$\llbracket (\int \gamma') \rrbracket_{(\mathcal{K},\mu_{\gamma})}^{\rho} = \mu_{\gamma}(|\gamma'|_{\mathcal{V}}) = \mu(|\gamma'|_{\mathcal{V}} \cap |\gamma|_{\mathcal{V}}) = \mu(|\gamma' \wedge \gamma|_{\mathcal{V}}) = \\ \llbracket (\int \gamma')/\gamma \rrbracket_{(\mathcal{K},\mu)}^{\rho}.$$

The claims for probabilistic terms and formulas follow by induction.

The calculus Soundness Completeness

Soundness of IF

Proof.

By definition,

$$\llbracket (\int \gamma') \rrbracket^{\rho}_{(\mathcal{K},\mu_{\gamma})} = \mu_{\gamma} (|\gamma'|_{\mathcal{V}}) = \mu(|\gamma'|_{\mathcal{V}} \cap |\gamma|_{\mathcal{V}}) = \mu(|\gamma' \wedge \gamma|_{\mathcal{V}}) = \\ \llbracket (\int \gamma') / \gamma \rrbracket^{\rho}_{(\mathcal{K},\mu)}.$$

The claims for probabilistic terms and formulas follow by induction.

The calculus Soundness Completeness

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 ,

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

iff, for any classical state formula γ_0 ,

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

・ロン ・四 と ・ ヨ と ・ ヨ と

The calculus Soundness Completeness

Soundness of $\ensuremath{\mathsf{IF}}$

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 ,

$$\vDash_{h} \{\eta_{1}\} s_{1} \{y_{1} = (\int \gamma)\} \text{ and } \vDash_{h} \{\eta_{2}\} s_{2} \{y_{2} = (\int \gamma)\}$$

iff, for any classical state formula γ_0 ,

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

The calculus Soundness Completeness

Soundness of IF

Proof.

Suppose that $(\mathcal{K},\mu)\rho \Vdash \eta_1 \Upsilon_{\gamma_0} \eta_2$. 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 $\ln_{\mathcal{H}}(\eta_1) = (\eta_1)$ and $(\mathcal{K},\mu_{\gamma_0})\mu \vdash \eta_1$, it follows that $(\mathcal{K},\mu_1) \vdash \mu_2 = (\eta_1)$. Thus, by definition $\rho(\eta_1) = \mu_1(\eta_1 \mu)$. Similarly, $\rho(\eta_2) = \mu_2(\eta_1 \mu)$. Hence, $\mu'(\eta_1 \mu) = \mu_1(\eta_1 \mu) + \mu_2(\eta_1 \mu) = \rho(\eta_1) + \rho(\eta_2) = \rho(\eta_1 + \eta_2)$ and

・ロ・ ・ 日・ ・ ヨ・ ・ 日・

3

The calculus Soundness Completeness

Soundness of $\ensuremath{\mathsf{IF}}$

Proof.

Suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1 \Upsilon_{\gamma_0} \eta_2$. 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 $(\mathcal{K}, \mu_1) = \llbracket s_1 \rrbracket (\mathcal{K}, \mu_{\gamma_0}), (\mathcal{K}, \mu_2) = \llbracket s_2 \rrbracket (\mathcal{K}, \mu_{(\neg \gamma_0)})$ and $\mu' = \mu_1 + \mu_2$. Since $\Vdash_h \{\eta_1\} s_1 \{\eta_1 = (\int \gamma)\}$ and $(\mathcal{K}, \mu_{\gamma_0})\rho \Vdash \eta_1$, it follows that $(\mathcal{K}, \mu_1) \Vdash_h \eta = (\int \gamma)$. Thus, by definition $\mu(\gamma) = \mu_1(\gamma_1)$ and $\mu' = \mu_1 + \mu_2$.

The calculus Soundness Completeness

Soundness of IF

Proof.

Suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1 \Upsilon_{\gamma_0} \eta_2$. 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 $(\mathcal{K}, \mu_1) = \llbracket s_1 \rrbracket (\mathcal{K}, \mu_{\gamma_0}), (\mathcal{K}, \mu_2) = \llbracket s_2 \rrbracket (\mathcal{K}, \mu_{(\neg \gamma_0)})$ 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 $\rho(y_1) = \mu_1(|\gamma|\nu)$.

The calculus Soundness Completeness

Soundness of IF

Proof.

Suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1 \Upsilon_{\gamma_0} \eta_2$. 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 $(\mathcal{K}, \mu_1) = \llbracket s_1 \rrbracket (\mathcal{K}, \mu_{\gamma_0}), (\mathcal{K}, \mu_2) = \llbracket s_2 \rrbracket (\mathcal{K}, \mu_{(\neg \gamma_0)})$ 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, $\rho(y_2) = \mu_2(|\gamma|\nu)$.

The calculus Soundness Completeness

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 $\Vdash_h \{\eta_1\} s_1 \{y_1 = (\lceil \gamma)\}$ and $(\mathcal{K}, \mu_{\gamma_0}) \rho \Vdash \eta_1$, it follows that $(\mathcal{K}, \mu_1) \Vdash_h \gamma_1 = (\int \gamma)$. Thus, by definition $\rho(\gamma_1) = \mu_1(|\gamma|_{\mathcal{V}})$.

The calculus Soundness Completeness

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 $\rho(y_1) = \mu_1(|\gamma|_{\mathcal{V}})$.

The calculus Soundness Completeness

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 $(\mathcal{K}, \mu_1) = \llbracket s_1 \rrbracket (\mathcal{K}, \mu_{\gamma_0})$, $(\mathcal{K}, \mu_2) = \llbracket s_2 \rrbracket (\mathcal{K}, \mu_{(\neg \gamma_0)})$ 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, $\rho(y_2) = \mu_2(|\gamma|_{\mathcal{V}})$.

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

The calculus Soundness Completeness

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 $(\mathcal{K}, \mu_1) = \llbracket s_1 \rrbracket (\mathcal{K}, \mu_{\gamma_0})$, $(\mathcal{K}, \mu_2) = \llbracket s_2 \rrbracket (\mathcal{K}, \mu_{(\neg \gamma_0)})$ 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. \Box

イロン イ団ン イヨン イヨン 三日

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Let
$$k = \llbracket p \rrbracket_{(\mathcal{K},\mu)}^{\rho}$$
 and $\rho_1 = \rho_k^y$. Then:

• for any probabilistic term p_0 , $\llbracket p_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = \llbracket p_0 P_p \rrbracket_{(\mathcal{K},\mu)}^{\rho};$

• for any probabilistic formula η , $(\mathcal{K},\mu)\rho_1 \Vdash \eta$ iff $(\mathcal{K},\mu)\rho \Vdash \eta_p^{y}$.

Proof.

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

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Let
$$k = \llbracket p \rrbracket_{(\mathcal{K},\mu)}^{
ho}$$
 and $ho_1 =
ho_k^y$. Then:

• for any probabilistic term p_0 , $\llbracket p_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = \llbracket p_0 \lor_P \rrbracket_{(\mathcal{K},\mu)}^{\rho};$

• for any probabilistic formula η , $(\mathcal{K}, \mu)\rho_1 \Vdash \eta$ iff $(\mathcal{K}, \mu)\rho \Vdash \eta_p^y$.

Proof.

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

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Let
$$k = \llbracket p \rrbracket_{(\mathcal{K},\mu)}^{
ho}$$
 and $ho_1 =
ho_k^y$. Then:

- for any probabilistic term p_0 , $\llbracket p_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = \llbracket p_0 \H_P^y \rrbracket_{(\mathcal{K},\mu)}^{\rho};$
- for any probabilistic formula η, (K, μ)ρ₁ ⊢ η iff (K, μ)ρ ⊢ η^y_p.

Proof.

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

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Let
$$k = \llbracket p \rrbracket_{(\mathcal{K},\mu)}^{
ho}$$
 and $ho_1 =
ho_k^y$. Then:

- for any probabilistic term p_0 , $\llbracket p_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = \llbracket p_0 \H_P^y \rrbracket_{(\mathcal{K},\mu)}^{\rho};$
- for any probabilistic formula η, (K, μ)ρ₁ ⊢ η iff (K, μ)ρ ⊢ η^y_p.

Proof.

Let p_0 be a variable y_0 .

If y_0 is y, then $\llbracket y \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = k = \llbracket p \rrbracket_{(\mathcal{K},\mu)}^{\rho} = \llbracket y_p^{\vee} \rrbracket_{(\mathcal{K},\mu)}^{\rho}$. Otherwise, $\llbracket y_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = \rho_1(y_0) = \rho(y_0) = \llbracket y_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho} = \llbracket y_0 \rrbracket_{(\mathcal{K},\mu)}^{\vee}$. The rest follows by induction.

・ロン ・回 と ・ ヨン ・ ヨン

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Let
$$k = \llbracket p \rrbracket_{(\mathcal{K},\mu)}^{
ho}$$
 and $ho_1 =
ho_k^y$. Then:

- for any probabilistic term p_0 , $\llbracket p_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = \llbracket p_0 \H_P^y \rrbracket_{(\mathcal{K},\mu)}^{\rho};$
- for any probabilistic formula η, (K, μ)ρ₁ ⊢ η iff (K, μ)ρ ⊢ η^y_p.

Proof.

Let p_0 be a variable y_0 . If y_0 is y, then $\llbracket y \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = k = \llbracket p \rrbracket_{(\mathcal{K},\mu)}^{\rho} = \llbracket y_p^y \rrbracket_{(\mathcal{K},\mu)}^{\rho}$. Otherwise, $\llbracket y_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = \rho_1(y_0) = \rho(y_0) = \llbracket y_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho} = \llbracket y_{0p} \rrbracket_{(\mathcal{K},\mu)}^{\rho}$. The rest follows by induction.

・ロン ・回 と ・ ヨン ・ ヨン

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Let
$$k = \llbracket p \rrbracket_{(\mathcal{K},\mu)}^{
ho}$$
 and $ho_1 =
ho_k^{
m y}$. Then:

- for any probabilistic term p_0 , $\llbracket p_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = \llbracket p_0 \H_P^y \rrbracket_{(\mathcal{K},\mu)}^{\rho};$
- for any probabilistic formula η, (K, μ)ρ₁ ⊢ η iff (K, μ)ρ ⊢ η^y_p.

Proof.

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

イロン イヨン イヨン イヨン

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Let
$$k = \llbracket p \rrbracket_{(\mathcal{K},\mu)}^{
ho}$$
 and $ho_1 =
ho_k^y$. Then:

- for any probabilistic term p_0 , $\llbracket p_0 \rrbracket_{(\mathcal{K},\mu)}^{\rho_1} = \llbracket p_0 \H_P^y \rrbracket_{(\mathcal{K},\mu)}^{\rho};$
- for any probabilistic formula η, (K, μ)ρ₁ ⊢ η iff (K, μ)ρ ⊢ η^y_p.

Proof.

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

イロン イヨン イヨン イヨン

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η ,

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

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\} s \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1^{p}$. Let $k = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho}$ and $\rho_1 = \rho_k^{y}$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $\llbracket y \rrbracket_{(\mathcal{K}, \mu)}^{\rho_1} = k$. Also $\llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho_1} = \llbracket p_p^{y} \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = k$. 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 , $(\llbracket s \rrbracket (\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required.

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η ,

if
$$\Vdash_h \{\eta_1 \cap (y = p)\} s \{\eta_2\}$$
 then $\Vdash_h \{\eta_1_p^y\} s \{\eta_2\}$.

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\} s \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1\rho$. Let $k = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho_1}$ and $\rho_1 = \rho_k^{\gamma}$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $\llbracket y \rrbracket_{(\mathcal{K}, \mu)}^{\rho_1} = k$. Also $\llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho_1} = \llbracket p_p^{\gamma} \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = k$. 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 , $(\llbracket s \rrbracket (\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required.

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η ,

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

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\} s \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1^{\mathcal{Y}}$. Let $k = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho}$ and $\rho_1 = \rho_k^{\mathcal{Y}}$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $\llbracket y \rrbracket_{(\mathcal{K}, \mu)}^{\rho_1} = k$. Also $\llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho_1} = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = k$. Therefore, $(\mathcal{K}, \mu)\rho_1 \Vdash (y = p)$. Since $\ln | m \cap (y = p)$.

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η ,

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

Proof.

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

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η ,

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

Proof.

Assume that $\Vdash_{h} \{\eta_{1} \cap (y = p)\} s \{\eta_{2}\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_{1}^{y}$. Let $k = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho}$ and $\rho_{1} = \rho_{k}^{y}$. Then $(\mathcal{K}, \mu)\rho_{1} \Vdash \eta_{1}$ and $\llbracket y \rrbracket_{(\mathcal{K}, \mu)}^{\rho_{1}} = k$. Also $\llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho_{1}} = \llbracket p_{p}^{y} \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = k$. 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} , $(\llbracket s \rrbracket (\mathcal{K}, \mu))\rho \Vdash \eta_{2}$ as required.

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η ,

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

Proof.

Assume that $\Vdash_h \{\eta_1 \cap (y = p)\} s \{\eta_2\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_1^{\mathcal{Y}}$. Let $k = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho}$ and $\rho_1 = \rho_k^{\mathcal{Y}}$. Then $(\mathcal{K}, \mu)\rho_1 \Vdash \eta_1$ and $\llbracket y \rrbracket_{(\mathcal{K}, \mu)}^{\rho_1} = k$. Also $\llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho_1} = \llbracket p_p^{\mathcal{Y}} \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = \llbracket p \rrbracket_{(\mathcal{K}, \mu)}^{\rho} = k$. 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 , $(\llbracket s \rrbracket (\mathcal{K}, \mu))\rho \Vdash \eta_2$ as required.

The calculus Soundness Completeness

Soundness of **ELIMV**

Lemma

Given y not occurring in either p or in η ,

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

Proof.

Assume that $\Vdash_{h} \{\eta_{1} \cap (y = p)\} s \{\eta_{2}\}$ and suppose that $(\mathcal{K}, \mu)\rho \Vdash \eta_{1}p^{\rho}$. Let $k = \llbracket p \rrbracket^{\rho}_{(\mathcal{K}, \mu)}$ and $\rho_{1} = \rho_{k}^{y}$. Then $(\mathcal{K}, \mu)\rho_{1} \Vdash \eta_{1}$ and $\llbracket y \rrbracket^{\rho_{1}}_{(\mathcal{K}, \mu)} = k$. Also $\llbracket p \rrbracket^{\rho_{1}}_{(\mathcal{K}, \mu)} = \llbracket p_{p}^{y} \rrbracket^{\rho}_{(\mathcal{K}, \mu)} = \llbracket p \rrbracket^{\rho}_{(\mathcal{K}, \mu)} = k$. 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} , $(\llbracket s \rrbracket (\mathcal{K}, \mu))\rho \Vdash \eta_{2}$ as required.

The calculus Soundness Completeness

Soundness of the calculus

Theorem

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

Proof.

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

The calculus Soundness Completeness

Soundness of the calculus

Theorem

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

Proof.

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

・ロン ・四 と ・ ヨ と ・ ヨ と

3

The calculus Soundness Completeness

Preterms

pt(skip, p) = p $pt(bm \leftarrow \gamma, p) = p_{\gamma}^{bm}$ $pt(xm \leftarrow t, p) = p_{t}^{xm}$ pt(toss(bm, r), p) = toss(bm, r; p) $pt(s_{1}; s_{2}, p) = pt(s_{1}, pt(s_{2}, p))$

The calculus Soundness Completeness

Preterms

pt(skip, p) = p $pt(bm \leftarrow \gamma, p) = p_{\gamma}^{bm}$ $pt(xm \leftarrow t, p) = p_{t}^{xm}$ pt(toss(bm, r), p) = toss(bm, r; p) $pt(s_{1}; s_{2}, p) = pt(s_{1}, pt(s_{2}, p))$

The calculus Soundness Completeness

Preterms

pt(skip, p) = p $pt(bm \leftarrow \gamma, p) = p_{\gamma}^{bm}$ $pt(xm \leftarrow t, p) = p_{t}^{xm}$ pt(toss(bm, r), p) = toss(bm, r; p) $pt(s_{1}; s_{2}, p) = pt(s_{1}, pt(s_{2}, p))$

The calculus Soundness Completeness

Preterms

pt(skip, p) = p $pt(\mathbf{bm} \leftarrow \gamma, p) = p_{\gamma}^{\mathbf{bm}}$ $pt(\mathbf{xm} \leftarrow t, p) = p_{t}^{\mathbf{xm}}$ $pt(toss(\mathbf{bm}, r), p) = toss(\mathbf{bm}, r; p)$ $pt(s_{1}; s_{2}, p) = pt(s_{1}, pt(s_{2}, p))$

The calculus Soundness Completeness

Preterms

 $\begin{array}{rcl} \mathsf{pt}(\mathsf{skip},\,\rho) &=& \rho\\ \mathsf{pt}(\mathbf{bm}\leftarrow\gamma,\,\rho) &=& \rho_{\gamma}^{\mathbf{bm}}\\ \mathsf{pt}(\mathbf{xm}\leftarrow t,\,\rho) &=& \rho_{t}^{\mathbf{xm}}\\ \mathsf{pt}(\mathsf{toss}(\mathbf{bm},r),\,\rho) &=& \mathsf{toss}(\mathbf{bm},r;\rho)\\ \mathsf{pt}(s_{1};s_{2},\,\rho) &=& \mathsf{pt}(s_{1},\,\mathsf{pt}(s_{2},\,\rho)) \end{array}$

Luís Cruz-Filipe Reasoning about Probabilistic Sequential Programs

The calculus Soundness Completeness

Preterms

 $pt(if \gamma then s_1 else s_2, r) = r$ $pt(if \gamma then s_1 else s_2, y) = y$ $pt(if \gamma then s_1 else s_2, (\int \gamma_0)) = (pt(s_1, (\int \gamma_0))/\gamma + pt(s_2, (\int \gamma_0))/(\neg \gamma))$ $pt(if \gamma then s_1 else s_2, (p_1 + p_2)) = (pt(if \gamma then s_1 else s_2, p_1) + pt(if \gamma then s_1 else s_2, p_2))$ $pt(if \gamma then s_1 else s_2, (p_1 p_2)) = (pt(if \gamma then s_1 else s_2, p_1) \times pt(if \gamma then s_1 else s_2, p_2))$

・ロン ・回 と ・ヨン ・ヨン

크

The calculus Soundness Completeness

Preterms

 $pt(if \gamma then s_1 else s_2, r) = r$ $pt(if \gamma then s_1 else s_2, y) = y$ $pt(if \gamma then s_1 else s_2, (\int \gamma_0)) = (pt(s_1, (\int \gamma_0))/\gamma + pt(s_2, (\int \gamma_0))/(\neg \gamma))$ $t(if \gamma then s_1 else s_2, (p_1 + p_2)) = (pt(if \gamma then s_1 else s_2, p_1) + pt(if \gamma then s_1 else s_2, p_2))$ $pt(if \gamma then s_1 else s_2, (p_1 p_2)) = (pt(if \gamma then s_1 else s_2, p_1) \times pt(if \gamma then s_1 else s_2, p_1) \times pt(if \gamma then s_1 else s_2, p_1)$

3

The calculus Soundness Completeness

Preterms

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

イロン イ団ン イヨン イヨン 三日

The calculus Soundness Completeness

Properties of preterms

Lemma

$$\llbracket \mathsf{pt}(s, \, \rho) \rrbracket^{\rho}_{(\mathcal{K}, \mu)} = \llbracket \rho \rrbracket^{\rho}_{\llbracket s \rrbracket(\mathcal{K}, \mu)}.$$

・ロト ・ 同 ト ・ ヨ ト ・ ヨ ト … ヨ

The calculus Soundness Completeness

Weakest preconditions

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

Theorem

 $(\mathcal{K},\mu)\rho \Vdash_h wp(s,\eta) \text{ iff } (\llbracket s \rrbracket (\mathcal{K},\mu))\rho \Vdash_h \eta.$

The calculus Soundness Completeness

Weakest preconditions

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

Theorem

 $(\mathcal{K},\mu)\rho \Vdash_h wp(s,\eta) \text{ iff } (\llbracket s \rrbracket (\mathcal{K},\mu))\rho \Vdash_h \eta.$

・ロト ・日 ・ ・ ヨ ・ ・ ヨ ・ ・ ヨ

The calculus Soundness Completeness

Weakest preconditions

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

Theorem

 $(\mathcal{K},\mu)\rho \Vdash_h wp(s,\eta) \text{ iff } (\llbracket s \rrbracket(\mathcal{K},\mu))\rho \Vdash_h \eta.$

(ロ) (同) (E) (E) (E)

The calculus Soundness Completeness

Weakest preconditions, semantically

Corollary

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

Proof.

(⇒) 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 $(\mathcal{K}, \mu)\rho \Vdash wp(s, \eta)$. Therefore $\vDash (\eta' \supset wp(s, \eta))$.

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

・ロト ・同ト ・ヨト ・ヨト

臣

The calculus Soundness Completeness

Weakest preconditions, semantically

Corollary

$$\vDash_{h} \{\eta'\} s \{\eta\} iff \vDash (\eta' \supset wp(s, \eta)).$$

Proof.

(⇒) 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 $(\mathcal{K}, \mu)\rho \Vdash wp(s, \eta)$. Therefore $\vDash (\eta' \supset wp(s, \eta))$.

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

・ロシ ・ 日 ・ ・ ヨ ・ ・ ヨ ・

臣

The calculus Soundness Completeness

Weakest preconditions, semantically

Corollary

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

Proof.

(⇒) 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 $(\mathcal{K}, \mu)\rho \Vdash wp(s, \eta)$. Therefore $\vDash (\eta' \supset wp(s, \eta))$.

(⇐) Suppose that \vDash ($\eta' \supset$ wp(s, η)) and (\mathcal{K}, μ) $\rho \Vdash \eta'$. Then (\mathcal{K}, μ) $\rho \Vdash$ wp(s, η) and hence ($\llbracket s \rrbracket (\mathcal{K}, \mu)$) $\rho \Vdash \eta$. Therefore =_h { η' } s { η }.

・ロ・ ・ 日・ ・ ヨ・ ・ 日・

3

The calculus Soundness Completeness

Weakest preconditions, semantically

Corollary

$$\vDash_{h} \{\eta'\} s \{\eta\} iff \vDash (\eta' \supset wp(s, \eta)).$$

Proof.

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

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

(ロ) (同) (E) (E) (E)

The calculus Soundness Completeness

Weakest preconditions, semantically

Corollary

$$\vDash_{h} \{\eta'\} s \{\eta\} iff \vDash (\eta' \supset wp(s, \eta)).$$

Proof.

(⇒) 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 $(\mathcal{K}, \mu)\rho \Vdash wp(s, \eta)$. Therefore $\vDash (\eta' \supset wp(s, \eta))$.

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

The calculus Soundness Completeness

Weakest preconditions, semantically

Corollary

$$\vDash_{h} \{\eta'\} s \{\eta\} iff \vDash (\eta' \supset wp(s, \eta)).$$

Proof.

(⇒) 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 $(\mathcal{K}, \mu)\rho \Vdash wp(s, \eta)$. Therefore $\vDash (\eta' \supset wp(s, \eta))$.

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

(ロ) (同) (E) (E) (E)

The calculus Soundness Completeness

Weakest preconditions, semantically

Corollary

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

Proof.

(⇒) 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 $(\mathcal{K}, \mu)\rho \Vdash wp(s, \eta)$. Therefore $\vDash (\eta' \supset wp(s, \eta))$.

(⇐) Suppose that \vDash ($\eta' \supset$ wp(s, η)) and (\mathcal{K}, μ) $\rho \Vdash \eta'$. Then (\mathcal{K}, μ) $\rho \Vdash$ wp(s, η) and hence ($\llbracket s \rrbracket (\mathcal{K}, \mu)$) $\rho \Vdash \eta$. Therefore $\vDash_{h} \{\eta'\} s \{\eta\}.$

The calculus Soundness Completeness

Weakest preconditions, semantically

Corollary

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

Proof.

(⇒) 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 $(\mathcal{K}, \mu)\rho \Vdash wp(s, \eta)$. Therefore $\vDash (\eta' \supset wp(s, \eta))$.

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

イロト イポト イヨト イヨト 二日

The calculus Soundness Completeness

Weakest preconditions, semantically

Corollary

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

Proof.

(⇒) 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 $(\mathcal{K}, \mu)\rho \Vdash wp(s, \eta)$. Therefore $\vDash (\eta' \supset wp(s, \eta))$.

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

イロト イポト イヨト イヨト 二日

The calculus Soundness Completeness

Weakest preconditions, sintactically

Lemma

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

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

Theorem

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

 $- \{ wp(s, \eta) \} s \{ \eta \}.$

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Weakest preconditions, sintactically

Lemma

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

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

Theorem

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

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

・ロン ・回 と ・ ヨン ・ ヨン

3

The calculus Soundness Completeness

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.

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

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.

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

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.

<ロ> (日) (日) (日) (日) (日)

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\} s \{\eta\}$. Then $\vDash (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\models_h \{\eta'\} s \{\eta\}$. Then $\models (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

(ロ) (同) (E) (E) (E)

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\models_h \{\eta'\} s \{\eta\}$. Then $\models (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, whence $\vdash \{\eta'\} s \{\eta\}$ is the product of the second secon

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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

(ロ) (同) (E) (E) (E)

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\models_h \{\eta'\} s \{\eta\}$. Then $\models (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

・ロ・ ・ 日・ ・ ヨ・ ・ 日・

Э

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\} s \{\eta\}$. Then $\vDash (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

(ロ) (同) (E) (E) (E)

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\} s \{\eta\}$. Then $\vDash (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s,\eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s,\eta)$ can be computed algorithmically.

・ロン ・四 と ・ ヨ と ・ ヨ と

Э

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\} s \{\eta\}$. Then $\vDash (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

・ロン ・四 と ・ ヨ と ・ ヨ と

3

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\} s \{\eta\}$. Then $\vDash (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

・ロン ・回 と ・ ヨン ・ ヨン

3

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\} s \{\eta\}$. Then $\vDash (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\} s \{\eta\}$. Then $\vDash (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

・ロト ・回ト ・ヨト ・ヨト

The calculus Soundness Completeness

Completeness and decidability

Proof.

Completeness. Suppose that $\vDash_h \{\eta'\} s \{\eta\}$. Then $\vDash (\eta' \supset wp(s, \eta))$. By completeness of EPPL, $\vdash (\eta' \supset wp(s, \eta))$. On the other hand, $\vdash \{wp(s, \eta)\} s \{\eta\}$, 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 $\vdash (\eta' \supset wp(s, \eta))$. The decidability is now a consequence of the decidability of EPPL and the fact that $wp(s, \eta)$ can be computed algorithmically.

・ロト ・ 同ト ・ ヨト ・ ヨト

Achievements

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

・ロト ・回ト ・ヨト ・ヨト

э

Achievements

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

・ロン ・回 と ・ ヨン ・ ヨン

Achievements

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

・ロン ・回 と ・ ヨン ・ ヨン

Future work

• unbounded iteration (while)

quantum programming languages

・ロト ・回ト ・ヨト ・ヨト

Э

Future work

- unbounded iteration (while)
- quantum programming languages

・ロト ・回ト ・ヨト ・ヨト

臣