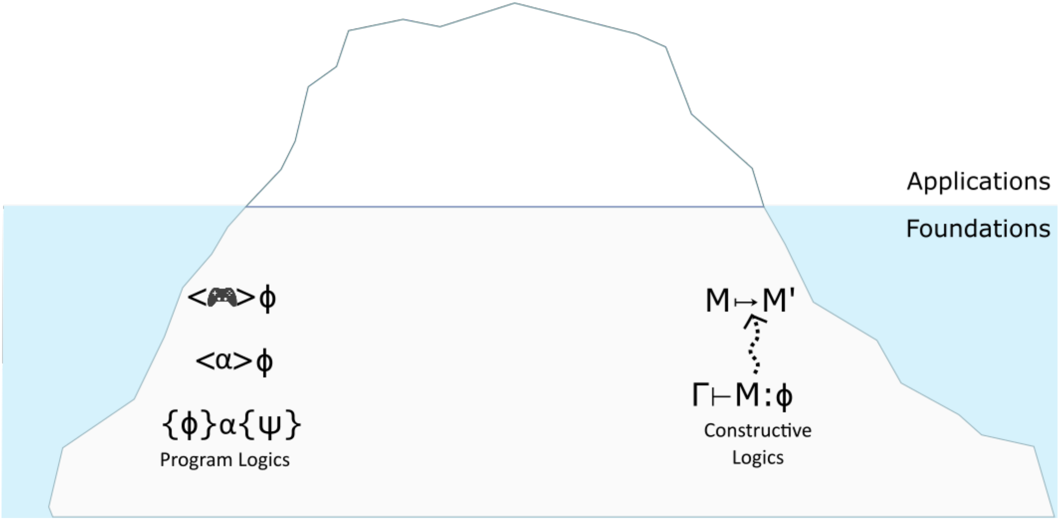
Constructive Game Logic

Brandon Bohrer and André Platzer

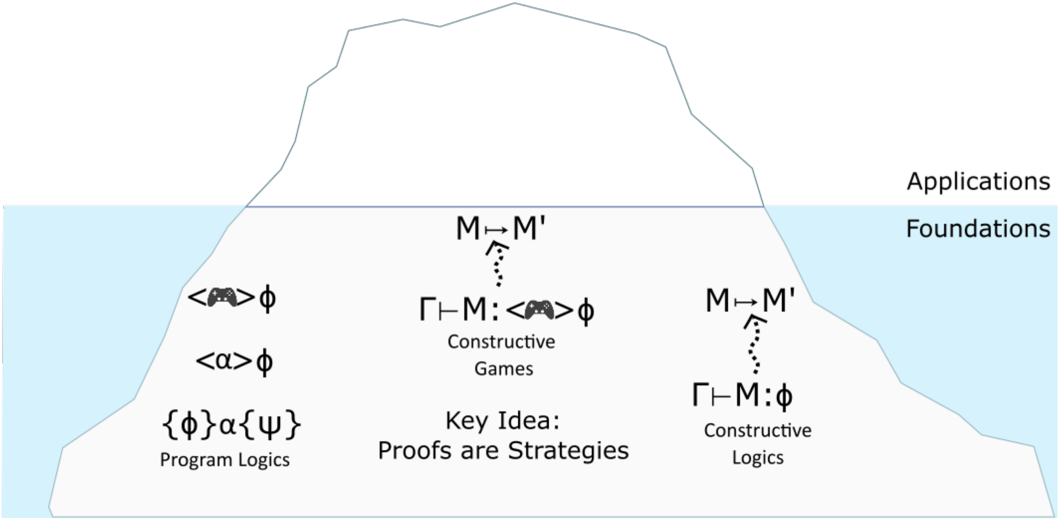
Logical Systems Lab
Computer Science Department
Carnegie Mellon University

ESOP "2020"

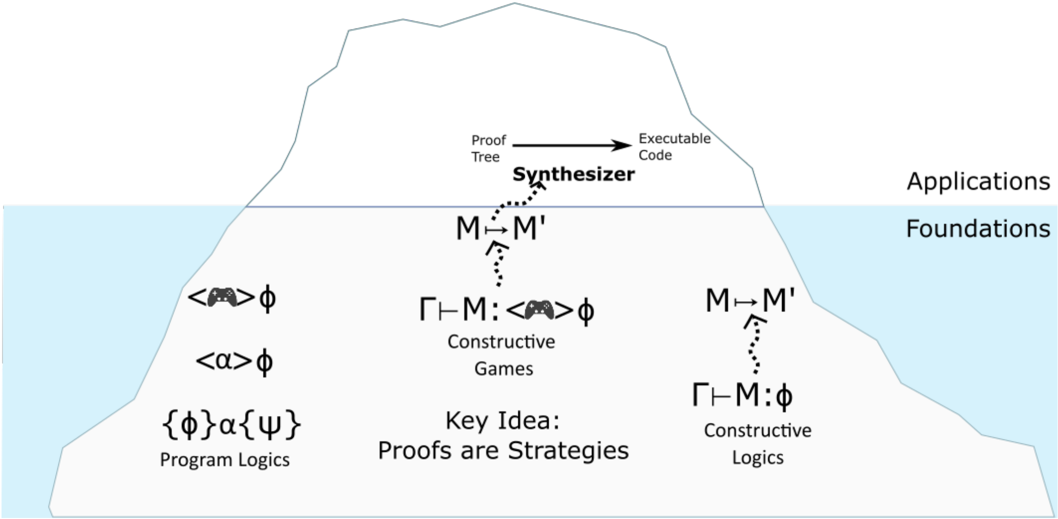
Modal and Constructive Logics Prove Programs



Modal + Constructive is Underexplored

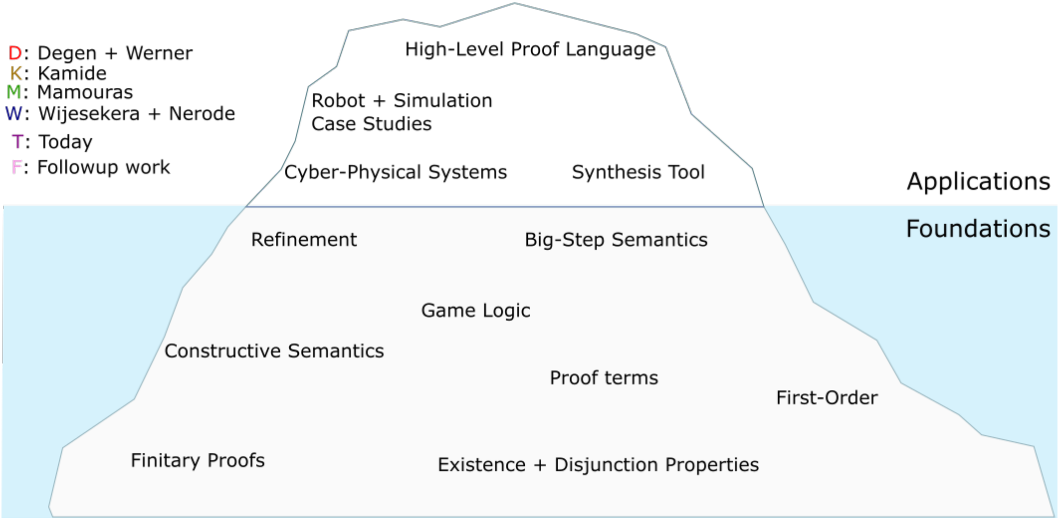


Constructivity Helps Synthesis

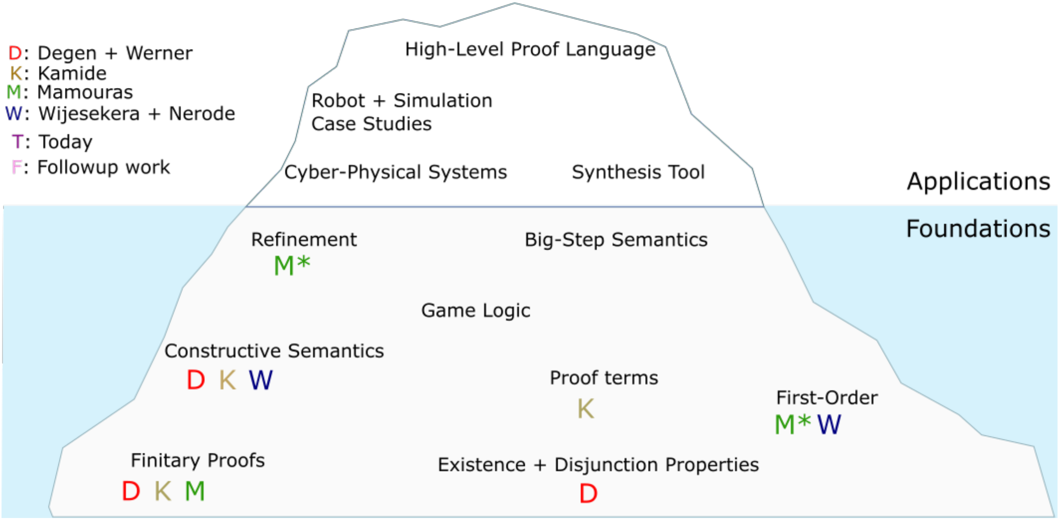


Foundations and Applications are Broad

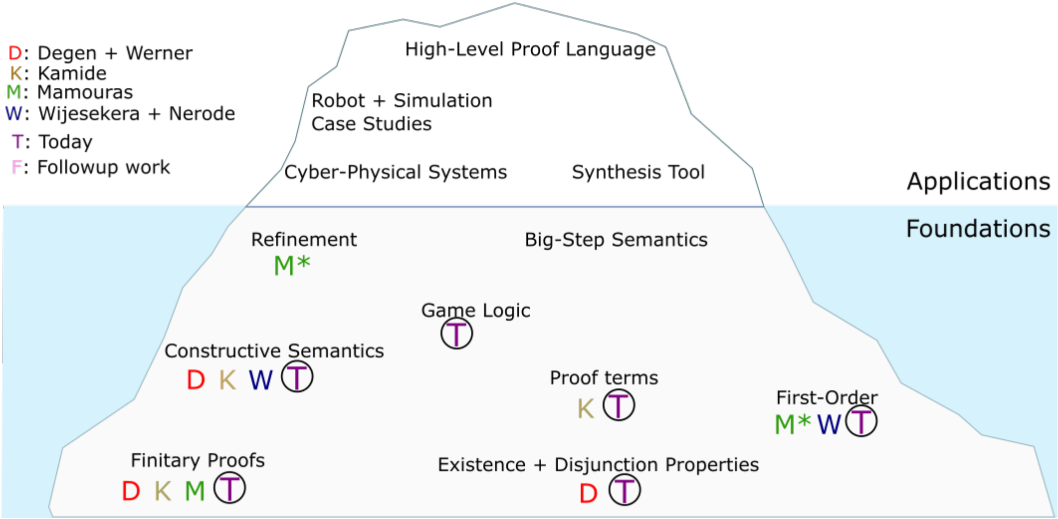
D: Degen + Werner
K: Kamide
M: Mamouras
W: Wijesekera + Nerode
T: Today
F: Followup work



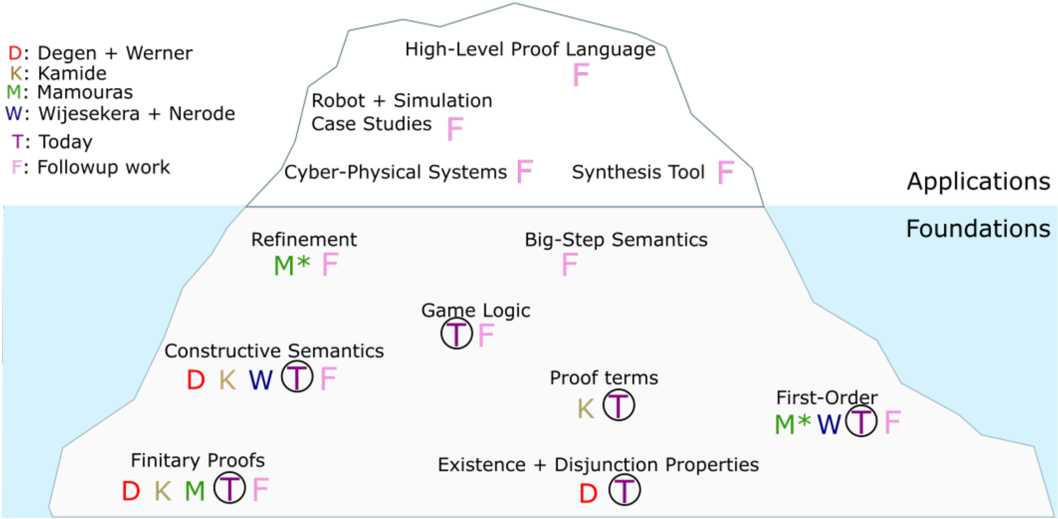
Full Constructive Story is Untold



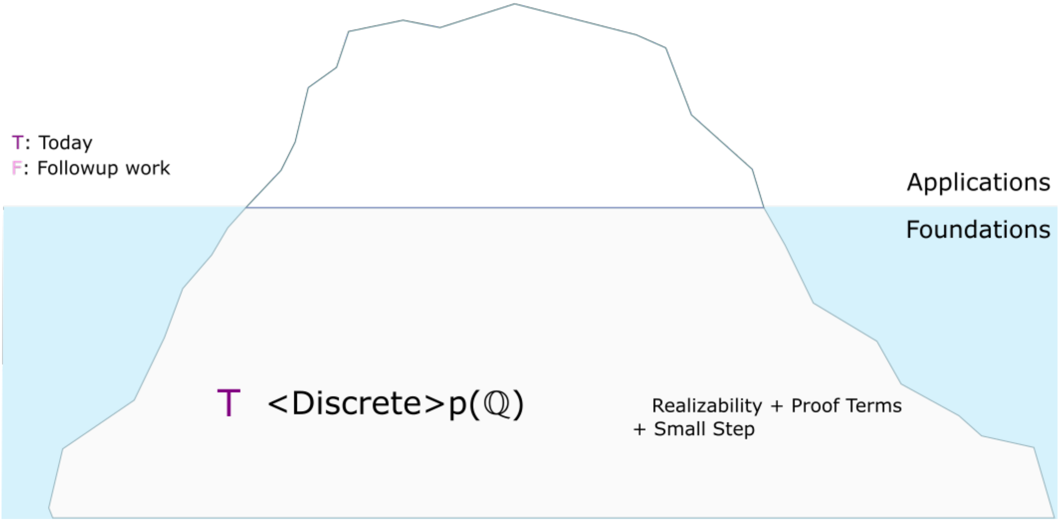
Today's Story is Extensive



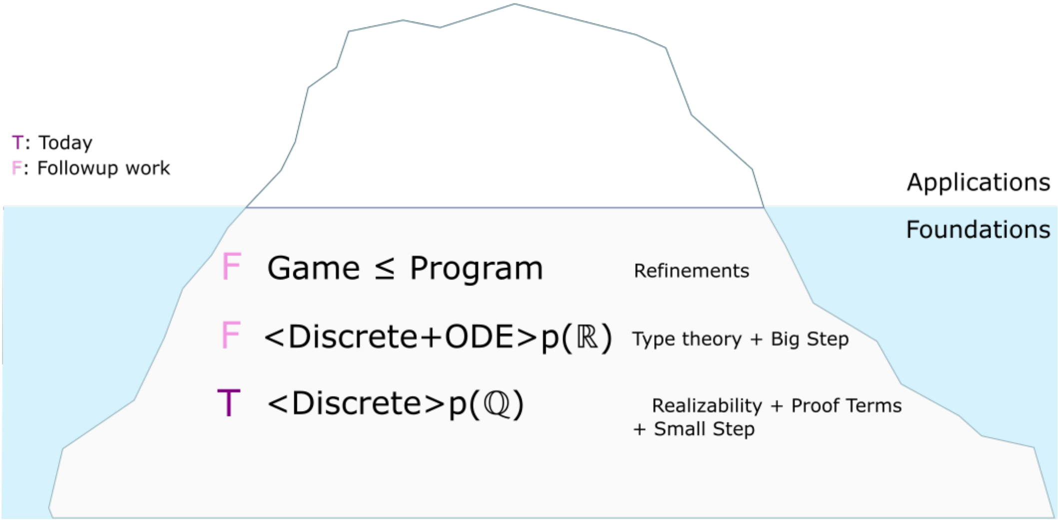
Follow-Up Story is Even Broader



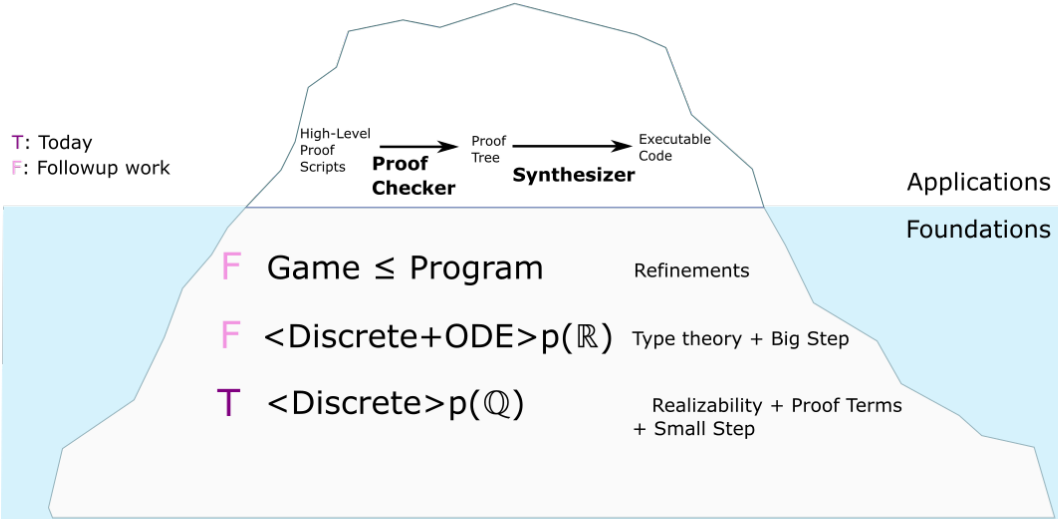
Discrete Foundations Built Today



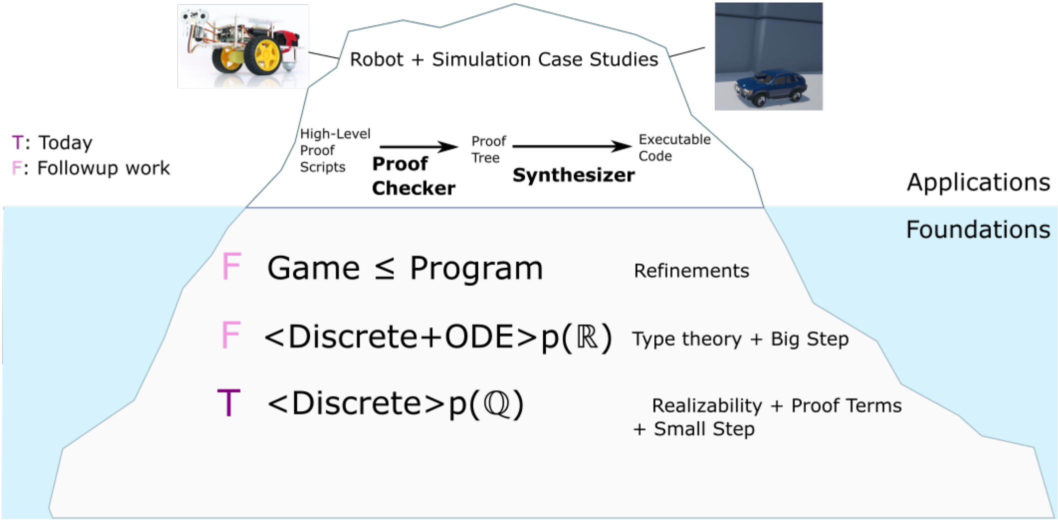
Were Followed by Continuous Systems



Whose Proofchecker and Synthesizer were Implemented



And Applied on Hardware and in Simulation



(Subtraction) Nim is an Introductory Example

Assign Choose move Test

$$\textsc{Nim} = \Big\{ \{\{c := c - 1 \cup c := c - 2 \cup c := c - 3\}; ?c > 0\}; \\ \{\{c := c - 1 \cup c := c - 2 \cup c := c - 3\}; ?c > 0\}^d \Big\}^*$$

(Subtraction) Nim is an Introductory Example

Assign Choose move

Test

$$\textsc{Nim} = \Big\{ \{\{c := c-1 \cup c := c-2 \cup c := c-3\}; ?c > 0\}; \{\{c := c-1 \cup c := c-2 \cup c := c-3\}; ?c > 0\}^d \Big\}^*$$

Take turns

(Subtraction) Nim is an Introductory Example

Assign　　　Choose move

Test

$$\textsc{Nim} = \Big\{\{\{c := c - 1 \cup c := c - 2 \cup c := c - 3\}; ?c > 0\};$$
$$\{\{c := c - 1 \cup c := c - 2 \cup c := c - 3\}; ?c > 0\}^d\Big\}^*$$

Repeat

Take turns

(Subtraction) Nim is an Introductory Example

$$\textsc{Nim} = \Big\{ \{\{ \overbrace{c := c - 1}^{\text{Assign}} \overbrace{\cup}^{\text{Choose move}} c := c - 2 \cup c := c - 3\}; \overbrace{?c > 0}^{\text{Test}}\};$$

$$\underbrace{\{\{c := c - 1 \cup c := c - 2 \cup c := c - 3\}; ?c > 0\}^{d}}_{\text{Take turns}} \Big\}^{\overbrace{*}^{\text{Repeat}}}$$

- If $c \in \{0, 2, 3\} \pmod 4$, the first player can achieve $c \in \{2, 3, 4\}$

$$c > 0 \to c \bmod 4 \in \{0, 2, 3\} \to \underbrace{\langle \textsc{Nim} \rangle (c \in \{2, 3, 4\})}_{\text{First player wins}}$$

- If $c \equiv 1 \pmod 4$, the second player can maintain $c \equiv 1 \pmod 4$:

$$c > 0 \to c \bmod 4 = 1 \to \underbrace{[\textsc{Nim}](c \bmod 4 = 1)}_{\text{Second player wins}}$$

Stateful Realizers Define + Play Games

If X : *Region* then $X\langle\!\langle\alpha\rangle\!\rangle$: *Region* and $X[\![\alpha]\!]$: *Region*. Won
Regions X defined by $X \subseteq (\mathit{Realizer} \times \mathit{State}) \cup \{\top\} \cup \{\bot\}$. Lost
Realizers a, b, c are higher-order, continuation-passing programs.

Stateful Realizers Define + Play Games

If X : *Region* then $X\langle\langle\alpha\rangle\rangle$: *Region* and $X[[\alpha]]$: *Region*. Won Lost

Regions X defined by $X \subseteq (\mathit{Realizer} \times \mathit{State}) \cup \{\top\} \cup \{\bot\}$.

Realizers a, b, c are higher-order, continuation-passing programs.

Example Angelic semantics cases:

Chose α Chose β

$$X\langle\langle\alpha \cup \beta\rangle\rangle = X_{\langle 0\rangle}\langle\langle\alpha\rangle\rangle \cup X_{\langle 1\rangle}\langle\langle\beta\rangle\rangle$$

Stateful Realizers Define + Play Games

If X : *Region* then $X\langle\!\langle\alpha\rangle\!\rangle$: *Region* and $X[\![\alpha]\!]$: *Region*. Won

Regions X defined by $X \subseteq (\mathit{Realizer} \times \mathit{State}) \cup \{\top\} \cup \{\bot\}$. Lost

Realizers a, b, c are higher-order, continuation-passing programs.

Example Angelic semantics cases: Chose α Chose β

$$
\begin{aligned}
X\langle\!\langle\alpha \cup \beta\rangle\!\rangle &= X_{\langle 0\rangle}\langle\!\langle\alpha\rangle\!\rangle \cup X_{\langle 1\rangle}\langle\!\langle\beta\rangle\!\rangle \\
X\langle\!\langle ?\phi\rangle\!\rangle &\ni (b, \omega) && \leftarrow ((a, b), \omega) \in X \text{ and } (a, \omega) \in [\![\phi]\!] \\
X\langle\!\langle ?\phi\rangle\!\rangle &\ni \bot && \leftarrow ((a, b), \omega) \in X \text{ and } (a, \omega) \notin [\![\phi]\!]
\end{aligned}
$$

Stateful Realizers Define + Play Games

If $X : \mathit{Region}$ then $X\langle\!\langle\alpha\rangle\!\rangle : \mathit{Region}$ and $X[\![\alpha]\!] : \mathit{Region}$. Won
Regions X defined by $X \subseteq (\mathit{Realizer} \times \mathit{State}) \cup \{\top\} \cup \{\bot\}$. Lost
Realizers a, b, c are higher-order, continuation-passing programs.
Example Angelic semantics cases:

Chose α Chose β

$$X\langle\!\langle\alpha \cup \beta\rangle\!\rangle = X_{\langle 0\rangle}\langle\!\langle\alpha\rangle\!\rangle \cup X_{\langle 1\rangle}\langle\!\langle\beta\rangle\!\rangle$$

$$X\langle\!\langle ?\phi\rangle\!\rangle \ni (b, \omega) \quad \leftarrow ((a, b), \omega) \in X \text{ and } (a, \omega) \in [\![\phi]\!]$$

$$X\langle\!\langle ?\phi\rangle\!\rangle \ni \bot \quad \leftarrow ((a, b), \omega) \in X \text{ and } (a, \omega) \notin [\![\phi]\!]$$

$$X\langle\!\langle x := f\rangle\!\rangle \ni (a, \omega[x \mapsto f(\omega)]) \quad \leftarrow (a, \omega) \in X$$

Modify

Stateful Realizers Define + Play Games

If X : *Region* then $X\langle\!\langle\alpha\rangle\!\rangle$: *Region* and $X[\![\alpha]\!]$: *Region*.
Regions X defined by $X \subseteq (\mathit{Realizer} \times \mathit{State}) \cup \{\top\} \cup \{\bot\}$.

Won

Lost

Realizers a, b, c are higher-order, continuation-passing programs.

Example Angelic semantics cases:

Chose α

Chose β

$$X\langle\!\langle\alpha \cup \beta\rangle\!\rangle = X_{\langle 0\rangle}\langle\!\langle\alpha\rangle\!\rangle \cup X_{\langle 1\rangle}\langle\!\langle\beta\rangle\!\rangle$$

$$X\langle\!\langle ?\phi\rangle\!\rangle \ni (b, \omega) \qquad \leftarrow ((a, b), \omega) \in X \text{ and } (a, \omega) \in [\![\phi]\!]$$

$$X\langle\!\langle ?\phi\rangle\!\rangle \ni \bot \qquad \leftarrow ((a, b), \omega) \in X \text{ and } (a, \omega) \notin [\![\phi]\!]$$

$$X\langle\!\langle x := f\rangle\!\rangle \ni (a, \omega[x \mapsto f(\omega)]) \qquad \leftarrow (a, \omega) \in X$$

$$X\langle\!\langle\alpha^d\rangle\!\rangle = X[\![\alpha]\!]$$

Modify

Natural Deduction Makes Proofs Functional Programs

$$\langle\cup\rangle\mathrm{E}\quad \frac{\Gamma \vdash \langle\alpha \cup \beta\rangle\phi \quad \Gamma, \langle\alpha\rangle\phi \vdash \psi \quad \Gamma, \langle\beta\rangle\phi \vdash \psi}{\Gamma \vdash \psi}$$

$$\langle?\rangle\mathrm{I}\quad \frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \langle?\phi\rangle\psi}$$

$$\langle?\rangle\mathrm{E1}\quad \frac{\Gamma \vdash \langle?\phi\rangle\psi}{\Gamma \vdash \phi}$$

$$\langle?\rangle\mathrm{E2}\quad \frac{\Gamma \vdash \langle?\phi\rangle\psi}{\Gamma \vdash \psi}$$

$$\langle\cup\rangle\mathrm{I1}\quad \frac{\Gamma \vdash \langle\alpha\rangle\phi}{\Gamma \vdash \langle\alpha \cup \beta\rangle\phi}$$

$$\langle\cup\rangle\mathrm{I2}\quad \frac{\Gamma \vdash \langle\beta\rangle\phi}{\Gamma \vdash \langle\alpha \cup \beta\rangle\phi}$$

$$[?]\mathrm{I}\quad \frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash [?\phi]\psi}$$

$$[?]\mathrm{E}\quad \frac{\Gamma \vdash [?\phi]\psi \quad \Gamma \vdash \phi}{\Gamma \vdash \psi}$$

Natural Deduction Can Prove Games

$$[;]\mathrm{I} \quad \frac{\Gamma \vdash [\alpha][\beta]\phi}{\Gamma \vdash [\alpha;\beta]\phi}$$

$$[:=]\mathrm{I} \quad \frac{\Gamma_x^y, (x = f_x^y) \vdash \phi}{\Gamma \vdash [x := f]\phi}$$

$$[^d]\mathrm{I} \quad \frac{\Gamma \vdash \langle\alpha\rangle\phi}{\Gamma \vdash [\alpha^d]\phi}$$

Natural Deduction Can Prove Games

$$[;]\mathrm{I}\quad \frac{\Gamma \vdash [\alpha][\beta]\phi}{\Gamma \vdash [\alpha;\beta]\phi} \qquad [:=]\mathrm{I}\quad \frac{\Gamma_x^y, (x = f_x^y) \vdash \phi}{\Gamma \vdash [x := f]\phi} \qquad [^d]\mathrm{I}\quad \frac{\Gamma \vdash \langle\alpha\rangle\phi}{\Gamma \vdash [\alpha^d]\phi}$$

$$[*]\mathrm{I}\quad \frac{\Gamma \vdash \psi \quad \psi \vdash [\alpha]\psi \quad \psi \vdash \phi}{\Gamma \vdash [\alpha^*]\phi}$$

$$\langle * \rangle\mathrm{I}\quad \frac{\Gamma \vdash \varphi \quad \varphi, \mathcal{M}_0 = \mathcal{M} \succ \mathbf{0} \vdash \langle\alpha\rangle(\varphi \land \mathcal{M}_0 \succ \mathcal{M}) \quad \varphi, \mathcal{M} = \mathbf{0} \vdash \phi}{\Gamma \vdash \langle\alpha^*\rangle\phi}$$

Proof Calculus is Sound

Theorem (Soundness of proof calculus)

Every provable sequent $(\Gamma \vdash \phi)$ *is valid.*

Proof Calculus is Sound

Theorem (Soundness of proof calculus)

Every provable sequent $(\Gamma \vdash \phi)$ *is valid.*

Lemma (Arithmetic-term substitution)

If $\Gamma \vdash \phi$ *then* $\sigma(\Gamma) \vdash \sigma(\phi)$ *for admissible substitutions* σ*.*

Lemma (Coincidence)

The semantics of formula ϕ *depends only on free variables of* ϕ*.*

Lemma (Bound effect)

Only bound variables of game α *are modified by execution.*

Proofs Are *Imperative* Programs

Lemma (Weak Existence Property)

If $\Gamma \vdash (\exists x : \mathbb{Q}\ \phi)$, *there exists* $f : State \to \mathbb{Q}$ *which witnesses* ϕ.

Lemma (Weak Disjunction Property)

If $\Gamma \vdash \phi \vee \psi$ *there exists* $f : State \to Bool$ *which chooses a branch of* $\phi \vee \psi$. *In each case,* ϕ *or* ψ *has a realizer.*

Proofs Are *Imperative* Programs

Lemma (Weak Existence Property)

If $\Gamma \vdash (\exists x : \mathbb{Q}\ \phi)$, *there exists* $f : State \to \mathbb{Q}$ *which witnesses* ϕ.

Lemma (Weak Disjunction Property)

If $\Gamma \vdash \phi \vee \psi$ *there exists* $f : State \to Bool$ *which chooses a branch of* $\phi \vee \psi$. *In each case,* ϕ *or* ψ *has a realizer.*

Theorem (Strategy Property for Angel's Turn)

If $\Gamma \vdash \langle\alpha\rangle\phi$, *there exists a realizer that wins* $\langle\langle\alpha\rangle\rangle$ *with goal* ϕ *assuming* Γ *initially.*

Theorem (Strategy Property for Demon's Turn)

If $\Gamma \vdash [\alpha]\phi$, *there exists a realizer that wins* $[[\alpha]]$ *with goal* ϕ *assuming* Γ *initially.*

Realizability Reduces Constructivity to Soundness

Lemma (Weak Existence Property)

If $\Gamma \vdash (\exists x : \mathbb{Q}\ \phi)$, *there exists* $f : \mathit{State} \to \mathbb{Q}$ *which witnesses* ϕ.

Lemma (Weak Disjunction Property)

If $\Gamma \vdash \phi \vee \psi$ *there exists* $f : \mathit{State} \to \mathit{Bool}$ *which chooses a branch of* $\phi \vee \psi$. *In each case,* ϕ *or* ψ *has a realizer.*

Theorem (Strategy Property for Angel's Turn)

If $\Gamma \vdash \langle\alpha\rangle\phi$, *there exists a realizer that wins* $\langle\langle\alpha\rangle\rangle$ *with goal* ϕ *assuming* Γ *initially.*

Theorem (Strategy Property for Demon's Turn)

If $\Gamma \vdash [\alpha]\phi$, *there exists a realizer that wins* $[[\alpha]]$ *with goal* ϕ *assuming* Γ *initially.*

Theorem (Soundness of proof calculus)

Every provable sequent $(\Gamma \vdash \phi)$ *is valid.*

Proofs Terms Show *Functional* Interpretation

- Interpret explicit proof syntax as pure functional program
- Modal separation: proof about program $\approx$ monadic program
- Application: normalize proofs to simplify further processing

Proofs Terms Show *Functional* Interpretation

- Interpret explicit proof syntax as pure functional program
- Modal separation: proof about program $\approx$ monadic program
- Application: normalize proofs to simplify further processing

Definition (Proof term grammar)

Propositional

$$
\begin{aligned}
M, N, O ::= {} & (\lambda p : \phi.\ M) \mid \langle M, N \rangle \mid \langle \ell \cdot M \rangle \mid \langle r \cdot M \rangle \\
& \mid (M \text{ rep } p : \psi.\ N \text{ in } O) \\
& \mid \langle \iota\ M \rangle \mid \langle \text{yield } M \rangle \mid \langle x := f\,{}^{y}_{x} \text{ in } p.\ M \rangle
\end{aligned}
$$

Proofs Terms Show *Functional* Interpretation

- Interpret explicit proof syntax as pure functional program
- Modal separation: proof about program $\approx$ monadic program
- Application: normalize proofs to simplify further processing

Definition (Proof term grammar)

Propositional

$$
\begin{aligned}
M, N, O ::= {} & (\lambda p : \phi.\ M) \mid \langle M, N \rangle \mid \langle \ell \cdot M \rangle \mid \langle r \cdot M \rangle \\
& \mid (M \text{ rep } p : \psi.\ N \text{ in } O) \\
& \mid \langle \iota\ M \rangle \mid \langle \text{yield } M \rangle \mid \langle x := f\,{}^{y}_{x} \text{ in } p.\ M \rangle
\end{aligned}
$$

(Co)induction

Proofs Terms Show *Functional* Interpretation

- Interpret explicit proof syntax as pure functional program
- Modal separation: proof about program $\approx$ monadic program
- Application: normalize proofs to simplify further processing

Definition (Proof term grammar)

Propositional

$$
\begin{aligned}
M, N, O ::= & \ (\lambda p : \phi.\ M) \mid \langle M, N \rangle \mid \langle \ell \cdot M \rangle \mid \langle r \cdot M \rangle \\
& \mid (M \text{ rep } p : \psi.\ N \text{ in } O) \\
& \mid \langle \iota\ M \rangle \mid \langle \text{yield } M \rangle \mid \langle x := f\,{}^{y}_{x} \text{ in } p.\ M \rangle
\end{aligned}
$$

(Co)induction

Games

Proof Terms Execute By Simplifying

Definition (Operational semantics)

$M \mapsto M'$ if M reduces to M' in one step.

Definition (Normal forms)

Normal proof terms M consist of canonical forms and case analyses.

Proof Terms Execute By Simplifying

Definition (Operational semantics)

$M \mapsto M'$ if M reduces to M' in one step.

Definition (Normal forms)

Normal proof terms M consist of canonical forms and case analyses.

Lemma (Progress)

If $\cdot \vdash M : \phi$, then either M is normal or $M \mapsto M'$ for some M'.

Lemma (Preservation)

If $\cdot \vdash M : \phi$ and $M \mapsto^* M'$, then $\cdot \vdash M' : \phi$.

Propositional Connectives are an Example

$$\lambda\phi\beta \quad (\lambda p : \phi.\ M)\ N \mapsto [N/p]M$$

$$\pi_L\beta \quad [\pi_1[M, N]] \mapsto M$$

$$\lambda\beta \quad (\lambda x : \mathbb{Q}.\ M)\ f \mapsto M_x^f$$

$$\pi_R\beta \quad [\pi_2[M, N]] \mapsto N$$

$$\pi_1\mathsf{S} \quad \frac{M \mapsto M'}{[\pi_1 M] \mapsto [\pi_1 M']}$$

$$\pi_2\mathsf{S} \quad \frac{M \mapsto M'}{[\pi_2 M] \mapsto [\pi_2 M']}$$

$$[\pi_1]\mathsf{C} \quad [\pi_1\langle \mathsf{case}\ M\ \mathrm{of}\ \ell \Rightarrow\ N \mid r \Rightarrow\ O\rangle] \mapsto \langle \mathsf{case}\ M\ \mathrm{of}\ \ell \Rightarrow\ [\pi_1 N] \mid r \Rightarrow\ [\pi_1 O]\rangle$$

$$[\pi_2]\mathsf{C} \quad [\pi_2\langle \mathsf{case}\ M\ \mathrm{of}\ \ell \Rightarrow\ N \mid r \Rightarrow\ O\rangle] \mapsto \langle \mathsf{case}\ M\ \mathrm{of}\ \ell \Rightarrow\ [\pi_2 N] \mid r \Rightarrow\ [\pi_2 O]\rangle$$

These Foundations Have Been Built On