The λ-calculus, from minimal to classical logic

Webpage of the course

Davide Barbarossa
db2437@bath.ac.uk
Dept of Computer Science
University of Bath

Giulio Guerrieri
g.guerrieri@sussex.ac.uk
Dept of Computer Science
University of Sussex

ESSLLI Summer School, Bochum (Germany) *28/07/2025 – 01/08/2025*

The λ-calculus, from minimal to classical logic

Lecture 5:

Krivine's approach to classical logic

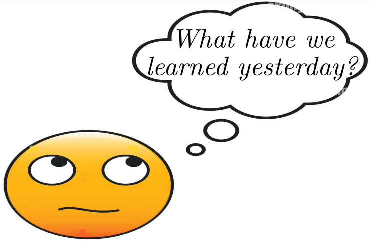
Read the notes: they are full of details, proofs, explanations, exercises, bibliography!

Davide Barbarossa
db2437@bath.ac.uk
Dept of Computer Science
UNIVERSITY OF BATH

ESSLLI Summer School, Bochum (Germany) *01/08/2025*

Previously...

- You have seen minimal logic
- You have seen that it corresponds to the simply-typed λ-calculus
- In the sense that formulas = types *and* cut-elimination = β-reduction
- Computational understanding of logic: proof $\to$ program



Previously...

Curry-Howard formalises the computational understanding (BHK) of logic in the *strongest* sense:
a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ *is* a (typed) program that computes the function

Previously...

Curry-Howard formalises the computational understanding (BHK) of logic in the *strongest* sense:
a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ *is* a (typed) program that computes the function

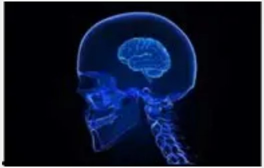
$$\mathtt{N} \in \mathrm{Proofs}(A) \quad \mapsto \quad \mathrm{nf}_\beta(\mathtt{M}\{\mathtt{x} := \mathtt{N}\}) \in \mathrm{Proofs}(B)$$

Previously...

Curry-Howard formalises the computational understanding (BHK) of logic in the *strongest* sense:
a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ *is* a (typed) program that computes the function

$$\mathtt{N} \in \text{Proofs}(A) \quad \mapsto \quad \text{nf}_\beta(\mathtt{M}\{\mathtt{x} := \mathtt{N}\}) \in \text{Proofs}(B)$$

Yesterday: *minimal* logic



Previously...

Curry-Howard formalises the computational understanding (BHK) of logic in the *strongest* sense:
a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ *is* a (typed) program that computes the function

$$\mathtt{N} \in \mathrm{Proofs}(A) \quad \mapsto \quad \mathrm{nf}_{\beta}(\mathtt{M}\{\mathtt{x} := \mathtt{N}\}) \in \mathrm{Proofs}(B)$$

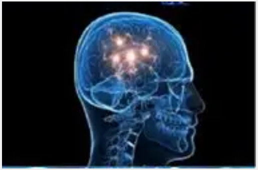
...it actually works for richer and richer *intuitionistic* logics (e.g. system F/2nd-order λ-calculus, MLTT's etc)

Previously...

Curry-Howard formalises the computational understanding (BHK) of logic in the *strongest* sense:
a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ *is* a (typed) program that computes the function

$$\mathtt{N} \in \mathrm{Proofs}(A) \quad \mapsto \quad \mathrm{nf}_\beta(\mathtt{M}\{\mathtt{x} := \mathtt{N}\}) \in \mathrm{Proofs}(B)$$

...it actually works for richer and richer *intuitionistic* logics (e.g. system F/2nd-order λ-calculus, MLTT's etc)



Previously...

Curry-Howard formalises the computational understanding (BHK) of logic in the *strongest* sense:
a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ *is* a (typed) program that computes the function

$$\mathtt{N} \in \mathrm{Proofs}(A) \quad \mapsto \quad \mathrm{nf}_\beta(\mathtt{M}\{\mathtt{x} := \mathtt{N}\}) \in \mathrm{Proofs}(B)$$

...it actually works for richer and richer *intuitionistic* logics (e.g. system F/2nd-order λ-calculus, MLTT's etc)

What about *classical* logic?

Previously...

Curry-Howard formalises the computational understanding (BHK) of logic in the *strongest* sense:
a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ *is* a (typed) program that computes the function

$$\mathtt{N} \in \mathrm{Proofs}(A) \quad \mapsto \quad \mathrm{nf}_\beta(\mathtt{M}\{\mathtt{x} := \mathtt{N}\}) \in \mathrm{Proofs}(B)$$

...it actually works for richer and richer *intuitionistic* logics (e.g. system F/2nd-order λ-calculus, MLTT's etc)

What about *classical* logic?

Taking the above *literally* fails ("Joyal's lemma" in category theory, "Lafont's pairs" in sequent calculus,...) ... which we are not going to see

Previously...

Curry-Howard formalises the computational understanding (BHK) of logic in the *strongest* sense:
a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ *is* a (typed) program that computes the function

$$\mathtt{N} \in \mathrm{Proofs}(A) \quad \mapsto \quad \mathrm{nf}_\beta(\mathtt{M}\{\mathtt{x} := \mathtt{N}\}) \in \mathrm{Proofs}(B)$$

...it actually works for richer and richer *intuitionistic* logics (e.g. system F/2nd-order λ-calculus, MLTT's etc)

What about *classical* logic?

Classical logic still has a computational content indeed!

type	$\rightsquigarrow$	realiser
purely functional	$\rightsquigarrow$	impure functional

Example

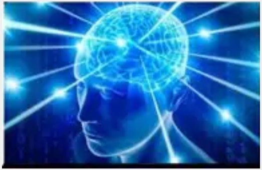
Classical realisability, $\neg\neg$+Dialectica, $\lambda\mu$-calculus, $\overline{\lambda}\mu\widetilde{\mu}$-calculus,...

Previously...

Curry-Howard formalises the computational understanding (BHK) of logic in the *strongest* sense:
a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ *is* a (typed) program that computes the function

$$\mathtt{N} \in \mathrm{Proofs}(A) \quad \mapsto \quad \mathrm{nf}_\beta(\mathtt{M}\{\mathtt{x} := \mathtt{N}\}) \in \mathrm{Proofs}(B)$$

...it actually works for richer and richer *intuitionistic* logics (e.g. system F/2nd-order λ-calculus, MLTT's etc)



What about *classical* logic?

Classical logic still has a computational content indeed!

type	$\rightsquigarrow$	realiser
purely functional	$\rightsquigarrow$	impure functional

Example

Classical realisability, $\neg\neg$+Dialectica, $\lambda\mu$-calculus, $\overline{\lambda}\mu\widetilde{\mu}$-calculus,...

Outline

1. 2nd order classical logic
2. Operational semantics of λ-calculus + `callcc`
3. Realisability and its adequacy to provability
4. Summary, Exercises, Bibliography

2nd order classical logic

Formulas:

$$A \quad ::= \quad X \;\mid\; A \to A$$

Proofs:

$$\frac{}{\underline{A}, \quad B \vdash \quad B}$$

$$\frac{\underline{A} \vdash \quad B \to C \qquad \underline{A} \vdash \quad B}{\underline{A} \vdash \quad C} \qquad\qquad \frac{\underline{A}, \quad B \vdash \quad C}{\underline{A} \vdash \quad B \to C}$$

2nd order classical logic

Formulas:

$$A \quad ::= \quad X \;\mid\; A \to A$$

Proofs:

$$\frac{}{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{y} : B}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B \to C \qquad \underline{\mathtt{x}} : \underline{A} \vdash \mathtt{N} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M\,N} : C} \qquad\qquad \frac{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{M} : C}{\underline{\mathtt{x}} : \underline{A} \vdash \lambda \mathtt{y}.\mathtt{M} : B \to C}$$

Proof-like terms:

$$\mathtt{M} \quad ::= \quad \mathtt{x} \;\mid\; \mathtt{M\,N} \;\mid\; \lambda \mathtt{x}.\mathtt{M}$$

2nd order classical logic

Arithmetic expressions:

$$e \quad ::= \quad n \ | \ a \ | \ f(e,\ldots,e) \qquad (\textit{for } n \in \mathbb{N}, a \in \mathcal{V}_1, f \in \mathcal{S}_k)$$

Formulas:

$$A \quad ::= \quad X(e,\ldots,e) \ | \ A \to A \ | \ \forall c.A \ | \ \forall^k X.A \qquad (\textit{for } X \in \mathcal{V}_2^k)$$

Proofs:

$$\frac{}{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{y} : B}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B \to C \qquad \underline{\mathtt{x}} : \underline{A} \vdash \mathtt{N} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M}\,\mathtt{N} : C} \qquad\qquad \frac{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{M} : C}{\underline{\mathtt{x}} : \underline{A} \vdash \lambda\mathtt{y}.\mathtt{M} : B \to C}$$

Proof-like terms:

$$\mathtt{M} \quad ::= \quad \mathtt{x} \ | \ \mathtt{M}\,\mathtt{N} \ | \ \lambda\mathtt{x}.\mathtt{M}$$

2nd order classical logic

Arithmetic expressions:

$$e \quad ::= \quad n \;\mid\; a \;\mid\; f(e,\ldots,e) \qquad (\textit{for } n \in \mathbb{N}, a \in \mathcal{V}_1, f \in \mathcal{S}_k)$$

Formulas:

$$A \quad ::= \quad X(e,\ldots,e) \;\mid\; A \to A \;\mid\; \forall c.A \;\mid\; \forall^k X.A \qquad (\textit{for } X \in \mathcal{V}_2^k)$$

Proofs:

$$\frac{}{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{y} : B}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B \to C \qquad \underline{\mathtt{x}} : \underline{A} \vdash \mathtt{N} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M\,N} : C} \qquad \frac{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{M} : C}{\underline{\mathtt{x}} : \underline{A} \vdash \lambda \mathtt{y}.\,\mathtt{M} : B \to C}$$

$$\frac{\underline{A} \vdash \quad B}{\underline{A} \vdash \quad \forall c.B} \qquad \frac{\underline{A} \vdash \quad B}{\underline{A} \vdash \quad \forall^k Y.B}$$

$$\frac{\underline{A} \vdash \quad \forall c.B}{\underline{A} \vdash \quad B\{c := e\}} \qquad \frac{\underline{A} \vdash \quad \forall^k Y.B}{\underline{A} \vdash \quad B\{Y := \langle C, c_1, \ldots, c_k \rangle\}}$$

Proof-like terms:

$$\mathtt{M} \quad ::= \quad \mathtt{x} \;\mid\; \mathtt{M\,N} \;\mid\; \lambda \mathtt{x}.\mathtt{M}$$

2nd order classical logic

Arithmetic expressions:

$$e \quad ::= \quad n \ \mid \ a \ \mid \ f(e, \dots, e) \qquad (\mathit{for}\ n \in \mathbb{N}, a \in \mathcal{V}_1, f \in \mathcal{S}_k)$$

Formulas:

$$A \quad ::= \quad X(e, \dots, e) \ \mid \ A \to A \ \mid \ \forall c.A \ \mid \ \forall^k X.A \qquad (\mathit{for}\ X \in \mathcal{V}_2^k)$$

Proofs:

$$\frac{}{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{y} : B}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B \to C \qquad \underline{\mathtt{x}} : \underline{A} \vdash \mathtt{N} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M\,N} : C} \qquad \frac{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{M} : C}{\underline{\mathtt{x}} : \underline{A} \vdash \lambda \mathtt{y}.\,\mathtt{M} : B \to C}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall c.B} \qquad \frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall^k Y.B}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall c.B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B\{c := e\}} \qquad \frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall^k Y.B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B\{Y := \langle C, c_1, \dots, c_k \rangle\}}$$

Proof-like terms:

$$\mathtt{M} \quad ::= \quad \mathtt{x} \ \mid \ \mathtt{M\,N} \ \mid \ \lambda \mathtt{x}.\mathtt{M}$$

2nd order classical logic

Arithmetic expressions:

$$e \quad ::= \quad n \;\mid\; a \;\mid\; f(e,\ldots,e) \qquad (\textit{for } n \in \mathbb{N}, a \in \mathcal{V}_1, f \in \mathcal{S}_k)$$

Formulas:

$$A \quad ::= \quad X(e,\ldots,e) \;\mid\; A \to A \;\mid\; \forall c.A \;\mid\; \forall^k X.A \qquad (\textit{for } X \in \mathcal{V}_2^k)$$

Proofs:

$$\frac{}{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{y} : B} \qquad \frac{}{\underline{A} \vdash \qquad ((B \to C) \to B) \to B}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B \to C \qquad \underline{\mathtt{x}} : \underline{A} \vdash \mathtt{N} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M}\,\mathtt{N} : C} \qquad \frac{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{M} : C}{\underline{\mathtt{x}} : \underline{A} \vdash \lambda \mathtt{y}.\,\mathtt{M} : B \to C}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall c.B} \qquad \frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall^k Y.B}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall c.B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B\{c := e\}} \qquad \frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall^k Y.B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B\{Y := \langle C, c_1, \ldots, c_k \rangle\}}$$

Proof-like terms:

$$\mathtt{M} \quad ::= \quad \mathtt{x} \;\mid\; \mathtt{M}\,\mathtt{N} \;\mid\; \lambda \mathtt{x}.\mathtt{M}$$

2nd order classical logic

Arithmetic expressions:

$$e \quad ::= \quad n \;\mid\; a \;\mid\; f(e,\dots,e) \qquad (\textit{for } n \in \mathbb{N}, a \in \mathcal{V}_1, f \in \mathcal{S}_k)$$

Formulas:

$$A \quad ::= \quad X(e,\dots,e) \;\mid\; A \to A \;\mid\; \forall c.A \;\mid\; \forall^k X.A \qquad (\textit{for } X \in \mathcal{V}_2^k)$$

Proofs:

$$\frac{}{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{y} : B} \qquad \frac{}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{callcc} : ((B \to C) \to B) \to B}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B \to C \qquad \underline{\mathtt{x}} : \underline{A} \vdash \mathtt{N} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M}\,\mathtt{N} : C} \qquad \frac{\underline{\mathtt{x}} : \underline{A}, \mathtt{y} : B \vdash \mathtt{M} : C}{\underline{\mathtt{x}} : \underline{A} \vdash \lambda \mathtt{y}.\,\mathtt{M} : B \to C}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall c.B} \qquad \frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall^k Y.B}$$

$$\frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall c.B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B\{c := e\}} \qquad \frac{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : \forall^k Y.B}{\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B\{Y := \langle C, c_1, \dots, c_k \rangle\}}$$

Proof-like terms:

$$\mathtt{M} \quad ::= \quad \mathtt{x} \;\mid\; \mathtt{M}\,\mathtt{N} \;\mid\; \lambda \mathtt{x}.\mathtt{M} \;\mid\; \mathtt{callcc}$$

Operational semantics of λ-calculus + callcc

OPS via:	*programs run...*	*E.g.*	*Logic*
β-reduction	by themselves	$\mathtt{M} \twoheadrightarrow_\beta \mathtt{N}$	intuitionistic
Abstract Machine	by interaction with execution stacks	$\mathtt{M} \star \pi \to \mathtt{N} \star \rho$	classical

Operational semantics of λ-calculus + $\texttt{callcc}$

OPS via:	*programs run...*	*E.g.*	*Logic*
β-reduction	by themselves	$\mathtt{M} \twoheadrightarrow_\beta \mathtt{N}$	intuitionistic
Abstract Machine	by interaction with execution stacks	$\mathtt{M} \star \pi \rightarrow \mathtt{N} \star \rho$	classical

$$\textit{Proof-Like terms} \quad \mathtt{P} \quad ::= \quad \mathtt{x} \mid \lambda\mathtt{x}.\,\mathtt{P} \mid \mathtt{P}\,\mathtt{P} \mid \mathtt{callcc}$$

$$\textit{Terms} \quad \mathtt{M} \quad ::= \quad \mathtt{x} \mid \lambda\mathtt{x}.\,\mathtt{M} \mid \mathtt{M}\,\mathtt{M} \mid \mathtt{callcc} \mid \mathtt{k}_\pi$$

$$\textit{Stacks} \quad \pi \quad ::= \quad [\,] \mid \mathtt{M} :: \pi$$

Operational semantics of λ-calculus + callcc

OPS via:	*programs run...*	*E.g.*	*Logic*
β-reduction	by themselves	$\mathtt{M} \twoheadrightarrow_\beta \mathtt{N}$	intuitionistic
Abstract Machine	by interaction with execution stacks	$\mathtt{M} \star \pi \rightarrow \mathtt{N} \star \rho$	classical

$$\textit{Proof-Like terms} \quad \mathtt{P} \quad ::= \quad \mathtt{x} \mid \lambda\mathtt{x}.\,\mathtt{P} \mid \mathtt{P}\,\mathtt{P} \mid \mathtt{callcc}$$

$$\textit{Terms} \quad \mathtt{M} \quad ::= \quad \mathtt{x} \mid \lambda\mathtt{x}.\,\mathtt{M} \mid \mathtt{M}\,\mathtt{M} \mid \mathtt{callcc} \mid \mathtt{k}_\pi$$

$$\textit{Stacks} \quad \pi \quad ::= \quad [\,] \mid \mathtt{M} :: \pi$$

Operational Semantics via a (simplified) KAM:

$$\begin{array}{ll} \textit{Weak Head} & \textbf{(push)} \quad \mathtt{M}\,\mathtt{N} \star \pi \rightarrow \mathtt{M} \star \mathtt{N} :: \pi \\ \textit{β-reduction} & \textbf{(pop)} \quad \lambda\mathtt{x}.\,\mathtt{M} \star \mathtt{N} :: \pi \rightarrow \mathtt{M}\{\mathtt{x} := \mathtt{N}\} \star \pi \end{array}$$

$$\begin{array}{ll} \textit{Continuation} & \textbf{(save)} \quad \mathtt{callcc} \star \mathtt{M} :: \pi \rightarrow \mathtt{M} \star \mathtt{k}_\pi :: \pi \\ \textit{passing style} & \textbf{(restore)} \quad \mathtt{k}_\pi \star \mathtt{M} :: \rho \rightarrow \mathtt{M} \star \pi \end{array}$$

Operational semantics of λ-calculus + `callcc`

	Arena	
Player		Opponent
$\mathtt{k}_\pi$	$\star$	$\mathtt{M} :: \rho$
	$\downarrow$	
$\mathtt{M}$	$\star$	π

Operational semantics of λ-calculus + callcc

Arena			Weapons	
Player		Opponent	Witnesses	Counterwitnesses
$\mathtt{k}_\pi$	$\star$	$\mathtt{M} :: \rho$		
	$\downarrow$		$\mathtt{M} \in \mathcal{W}(A)$	$\pi \in \mathcal{C}(A)$
$\mathtt{M}$	$\star$	π		

Operational semantics of λ-calculus + callcc

Arena			Weapons	
Player		Opponent	Witnesses	Counterwitnesses
$\mathtt{k}_\pi$	$\star$	$\mathtt{M} :: \rho$		
	$\downarrow$		$\mathtt{M} \in \mathcal{W}(A)$	$\pi \in \mathcal{C}(A)$
$\mathtt{M}$	$\star$	π		
	$\pitchfork$			
	$\perp\!\!\!\perp$			

Operational semantics of λ-calculus + callcc

Arena			Weapons	
Player		Opponent	Witnesses	Counterwitnesses
$\mathtt{k}_\pi$	$\star$	$\mathtt{M} :: \rho$		
	$\downarrow$		$\mathtt{M} \in \mathcal{W}(A)$ $\mathtt{k}_\pi \in \mathcal{W}(A \to \bot)$	$\pi \in \mathcal{C}(A)$
$\mathtt{M}$	$\star$	π		

$$\Longrightarrow \qquad \mathtt{k}_\pi \in \mathcal{W}(\neg A) \quad \textit{for all } \pi \in \mathcal{C}(A)$$

Operational semantics of λ-calculus + callcc

Arena

Player		Opponent
callcc	$\star$	$\texttt{M} :: \pi$
	$\downarrow$	
M	$\star$	$\texttt{k}_\pi :: \pi$

Operational semantics of λ-calculus + callcc

Arena			Weapons	
Player		Opponent	Witnesses	Counterwitnesses
$\texttt{callcc}$	$\star$	$\texttt{M} :: \pi$		
	$\downarrow$		$\texttt{M} \in \mathcal{W}(\neg\neg A)$	$\pi \in \mathcal{C}(A)$
$\texttt{M}$	$\star$	$\texttt{k}_\pi :: \pi$		

Operational semantics of λ-calculus + callcc

Arena			Weapons	
Player		Opponent	Witnesses	Counterwitnesses
$\texttt{callcc}$	$\star$	$\texttt{M} :: \pi$		
	$\downarrow$		$\texttt{M} \in \mathcal{W}(\neg A \to \bot)$	$\pi \in \mathcal{C}(A)$
$\texttt{M}$	$\star$	$\texttt{k}_\pi :: \pi$		
	$\pitchfork$			
	$\perp\!\!\!\perp$			

Operational semantics of λ-calculus + callcc

Arena			Weapons	
Player		Opponent	Witnesses	Counterwitnesses
$\texttt{callcc}$	$\star$	$\texttt{M} :: \pi$		
	$\downarrow$		$\texttt{M} \in \mathcal{W}(\neg\neg A)$ $\texttt{callcc} \in \mathcal{W}(\neg\neg A \to A)$	$\pi \in \mathcal{C}(A)$
$\texttt{M}$	$\star$	$\texttt{k}_\pi :: \pi$		

$$\Longrightarrow \qquad \texttt{callcc} \Vdash \neg\neg A \to A$$

Operational semantics of λ-calculus + callcc

Arena			Weapons	
Player		Opponent	Witnesses	Counterwitnesses
$\texttt{callcc}$	$\star$	$\texttt{M} :: \pi$	$\texttt{M} \in \mathcal{W}(\neg\neg A)$	$\pi \in \mathcal{C}(A)$
	$\downarrow$		$\texttt{callcc} \in \mathcal{W}(\neg\neg A \to A)$	
$\texttt{M}$	$\star$	$\texttt{k}_\pi :: \pi$		

$$\Longrightarrow \qquad \texttt{callcc} \Vdash \neg\neg A \to A$$

Similar argument for

$$\lambda \texttt{x}.\, \texttt{callcc}\, \texttt{x} \Vdash \neg\neg A \to A \quad \textit{and} \quad \lambda \texttt{x}.\, \texttt{callcc}(\lambda \texttt{y}.\, \texttt{x}\, \texttt{y}) \Vdash \neg\neg A \to A$$

Definition (Realisability semantics of formulas)

$$C^{\perp\!\!\!\perp} := \{t \in \mathbb{W} \mid t \perp\!\!\!\perp \pi \textit{ for all } \pi \in C\}$$

$$\mathcal{W}(\_) := \mathcal{C}(\_)^{\perp\!\!\!\perp}$$

	$\mathbb{W}$	$\mathbb{C}$	$\perp\!\!\!\perp \subseteq \mathbb{W}\times\mathbb{C}$	$*$
Tarski	$\{\Box\}$	$\{\dagger\}$	$\emptyset$	$\Rightarrow$
Krivine	Λ	Λ^*	*pole*	*cons*

$$\mathcal{C}(X(e_1,\ldots,e_k)) = [\![X]\!]\,([\![e_1]\!],\ldots,[\![e_k]\!])$$

$$\mathcal{C}(A \to B) = \mathcal{W}(A) * \mathcal{C}(B)$$

$$\mathcal{C}(\forall c.A) = \bigcup_{m\in\mathbb{N}} \mathcal{C}(A\{c := m\})$$

$$\mathcal{C}(\forall^m Y.A) = \bigcup_{Q:\mathbb{N}^m\to\mathcal{P}(\mathbb{C})} \mathcal{C}(A\{Y := Q\})$$

Realisability relation: $\mathtt{M} \Vdash A$ whenever $\mathtt{M}$ is a closed proof like term in $\mathcal{W}(A)$

Not only $\lambda\mathtt{x}.\,\mathtt{callcc}(\lambda\mathtt{y}.\,\mathtt{x\,y}) \Vdash \neg\neg A \to A$ but even:

$$\vdash \lambda\mathtt{x}.\,\mathtt{callcc}(\lambda\mathtt{y}.\,\mathtt{x\,y}) \;:\; \neg\neg A \to A$$

In fact, typing $\Rightarrow$ realising. The converse fails: that's precisely what we want!

Adequacy Theorem

Let

$$\mathtt{x}_1 : A_1, \ldots, \mathtt{x}_m : A_m \vdash \mathtt{M} : B$$

and fix an interpretation of the (free) 1st and 2nd order variables of $A_1, \ldots, A_m, B$.
For all closed terms $\mathtt{N}_1, \ldots, \mathtt{N}_m$, we have:

$$\mathtt{N}_1 \in \mathcal{W}(A_1)\,,\;\ldots\;,\,\mathtt{N}_m \in \mathcal{W}(A_m) \quad\Longrightarrow\quad \mathtt{M}\{\vec{\mathtt{x}} := \vec{\mathtt{N}}\} \in \mathcal{W}(B).$$

Realisability and its adequacy to provability

In fact, typing ⇒ realising. The converse fails: that's precisely what we want!

Adequacy Theorem

Let

$$\mathtt{x}_1 : A_1, \ldots, \mathtt{x}_m : A_m \vdash \mathtt{M} : B$$

and fix an interpretation of the (free) 1st and 2nd order variables of $A_1, \ldots, A_m, B$.
For all closed terms $\mathtt{N}_1, \ldots, \mathtt{N}_m$, we have:

$$\mathtt{N}_1 \in \mathcal{W}(A_1)\,,\ \ldots\ ,\ \mathtt{N}_m \in \mathcal{W}(A_m) \quad \Longrightarrow \quad \mathtt{M}\{\vec{\mathtt{x}} := \vec{\mathtt{N}}\} \in \mathcal{W}(B).$$

Corollary

$$\vdash \mathtt{M} : A \implies \mathtt{M} \Vdash A$$

Realisability and its adequacy to provability

In fact, typing ⇒ realising. The converse fails: that's precisely what we want!

Adequacy Theorem

Let

$$\mathtt{x}_1 : A_1, \ldots, \mathtt{x}_m : A_m \vdash \mathtt{M} : B$$

and fix an interpretation of the (free) 1st and 2nd order variables of $A_1, \ldots, A_m, B$.
For all closed terms $\mathtt{N}_1, \ldots, \mathtt{N}_m$, we have:

$$\mathtt{N}_1 \in \mathcal{W}(A_1)\,,\ \ldots\ ,\ \mathtt{N}_m \in \mathcal{W}(A_m) \quad \Longrightarrow \quad \mathtt{M}\{\vec{\mathtt{x}} := \vec{\mathtt{N}}\} \in \mathcal{W}(B).$$

In other words, a proof $\mathtt{x} : A \vdash \mathtt{M} : B$ defines a proof-term $\lambda\mathtt{x}.\,\mathtt{M}$ that computes (for each interpretation of variables and notion of winning process), the function

$$\mathtt{N} \in \mathcal{W}(A) \quad \mapsto \quad \mathtt{M}\{\mathtt{x} := \mathtt{N}\} \in \quad \mathcal{W}(B)$$

Realisability formalises BHK by extending the literal Curry-Howard one!

Realisability and its adequacy to provability

In fact, typing $\Rightarrow$ realising. The converse fails: that's precisely what we want!

Adequacy Theorem

Let

$$\mathtt{x}_1 : A_1, \ldots, \mathtt{x}_m : A_m \vdash \mathtt{M} : B$$

and fix an interpretation of the (free) 1st and 2nd order variables of $A_1, \ldots, A_m, B$.
For all closed terms $\mathtt{N}_1, \ldots, \mathtt{N}_m$, we have:

$$\mathtt{N}_1 \in \mathcal{W}(A_1)\,,\ \ldots\ ,\, \mathtt{N}_m \in \mathcal{W}(A_m) \quad \Longrightarrow \quad \mathtt{M}\{\vec{\mathtt{x}} := \vec{\mathtt{N}}\} \in \mathcal{W}(B).$$

Corollary

Let $\mathcal{T}$ be a theory of PA2 (or ZF/+CH/+C+...). If all axioms A of $\mathcal{T}$ are realised by programs $\mathtt{N}_A \Vdash A$, then: $\underline{\mathtt{x}} : \underline{A} \vdash \mathtt{M} : B \Longrightarrow \mathtt{M}\{\underline{\mathtt{x}} := \underline{\mathtt{N}}\} \Vdash B$.

E.g., $A =$*countable/dependent axiom of choice, ultrafilter axiom on* $\mathbb{N}$, *Continuum Hypothesis,...*

Moral of the story 1:

(The computational content of) Classical logic is about the interaction with some notion of environment

Realisability and its adequacy to provability

Moral of the story 1:

(The computational content of) Classical logic is about the interaction with some notion of environment

Proof.

True for all (that I know) computational approaches of classical logic: classical realisability, Game Semantics, Linear Logic, $\lambda\mu$-calculus, $\neg\neg$+Dialectica $\square$

Realisability and its adequacy to provability

Moral of the story 1:

(The computational content of) Classical logic is about the interaction with some notion of environment

Moral of the story 2:

Now that we disclosed the "hidden" computational content of classical logic, we want to realise axioms, not only theorems!

Realisability and its adequacy to provability

Moral of the story 1:

(The computational content of) Classical logic is about the interaction with some notion of environment

Moral of the story 2:

Now that we disclosed the "hidden" computational content of classical logic, we want to realise axioms, not only theorems!

Example

The theory of the formulas which admit a realiser is deductively closed.
Under mild assumption on the pole $\perp\!\!\!\perp$, it is also non-contradictory.
Study its models.
In the case for ZFC, this is stronger than forcing!

Realisability and its adequacy to provability

Moral of the story 1:

(The computational content of) Classical logic is about the interaction with some notion of environment

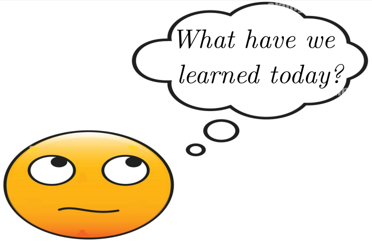
Moral of the story 2:

Now that we disclosed the "hidden" computational content of classical logic, we want to realise axioms, not only theorems!

Example

The theory of the formulas which admit a realiser is deductively closed.
Under mild assumption on the pole $\perp\!\!\!\perp$, it is also non-contradictory.
Study its models.
In the case for ZFC, this is stronger than forcing!

- We have seen proof-terms for classical 2nd order logic
- We have given them an operational semantics in terms of a KAM which manipulates continuations
- We have defined the realisability semantics by refining Tarski
- We have seen that realisability is adequate wrt provability



Summary, Exercises, Bibliography

- Look at our **notes** on the webpage of the course, there are plenty of **details**, **proofs** and **exercises**.

- Look at our **notes** on the webpage of the course, there are plenty of **details**, **proofs** and **exercises**. Here's one

Exercise

The 2nd order encoding of $A \vee B$ is:
$A \vee B \;:=\; \forall^0 X.\,(A \to X) \to (B \to X) \to X$. With that, show (by hand) that:

$$\mathtt{callcc}(\lambda \mathtt{yvh}.\,\mathtt{h}(\lambda \mathtt{x}.\,\mathtt{y}(\lambda \mathtt{zw}.\,\mathtt{zx}))) \;\Vdash\; A \vee \neg A.$$

This is actually the proof-term of a derivation of the excluded middle from *Consequentia Mirabilis* (itself an instance of Peirce's law). Do *not* use adequacy though.

- Look at our **notes** on the webpage of the course, there are plenty of **details**, **proofs** and **exercises**. Here's one

Exercise

The 2nd order encoding of $A \vee B$ is:
$A \vee B \ := \ \forall^0 X.\ (A \to X) \to (B \to X) \to X$. With that, show (by hand) that:

$$\texttt{callcc}(\lambda \texttt{yvh}.\, \texttt{h}(\lambda \texttt{x}.\, \texttt{y}(\lambda \texttt{zw}.\, \texttt{zx}))) \ \Vdash \ A \vee \neg A.$$

This is actually the proof-term of a derivation of the excluded middle from *Consequentia Mirabilis* (itself an instance of Peirce's law). Do *not* use adequacy though.

- The exercises have **solutions** (but try to do them by yourself before looking at them!).

- Look at our **notes** on the webpage of the course, there are plenty of **details**, **proofs** and **exercises**. Here's one

Exercise

The 2nd order encoding of $A \vee B$ is:
$A \vee B \;:=\; \forall^0 X.\, (A \to X) \to (B \to X) \to X$. With that, show (by hand) that:

$$\texttt{callcc}(\lambda \texttt{yvh.}\, \texttt{h}(\lambda \texttt{x.}\, \texttt{y}(\lambda \texttt{zw.}\, \texttt{zx}))) \;\Vdash\; A \vee \neg A.$$

This is actually the proof-term of a derivation of the excluded middle from *Consequentia Mirabilis* (itself an instance of Peirce's law). Do *not* use adequacy though.

- The exercises have **solutions** (but try to do them by yourself before looking at them!).

One million dollars exercise

Find a program $\texttt{M}$ such that $\texttt{M} \Vdash$ full Axiom of Choice

[*Hint (?): Krivine proved that one exists.*]

- Where the idea of callcc with Peirce law was introduced:
  A formulae-as-type notion of control, Timothy G. Griffin, 1990,
  https://dl.acm.org/doi/10.1145/96709.96714
- A standard introduction to the topic:
  Realizability in classical logic, Jean-Louis Krivine, 2004,
  https://www.irif.fr/~krivine/articles/Luminy04.pdf
- A very nice and clear PhD manuscript on the topic:
  On Forcing and Classical Realizability, Lionel Rieg, 2014,
  https://www-verimag.imag.fr/~riegl/assets/thesis-color.pdf
- To go further (one cool example among many possible):
  A program for the full Axiom of Choice, Jean-Louis Krivine, 2021,
  https://www.irif.fr/~krivine/articles/A_program_for_full_AC.pdf