On the Computational Content of Classical Logic

Alex Summers
Transfer Presentation
Imperial College
Spring Term 2006

Overview

- Background

Overview

- Background
- The $\mathcal{X}$ calculus

Overview

- Background
- The $\mathcal{X}$ calculus
- Aims

Overview

- Background
- The $\mathcal{X}$ calculus
- Aims
- Type Assignment for $\mathcal{X}$

Overview

- Background
- The $\mathcal{X}$ calculus
- Aims
- Type Assignment for $\mathcal{X}$
- Polymorphism

Overview

- Background
- The $\mathcal{X}$ calculus
- Aims
- Type Assignment for $\mathcal{X}$
- Polymorphism
- Shallow Polymorphism

Overview

- Background
- The $\mathcal{X}$ calculus
- Aims
- Type Assignment for $\mathcal{X}$
- Polymorphism
- Shallow Polymorphism
- Future Work

Overview

- Background
- The $\mathcal{X}$ calculus
- Aims
- Type Assignment for $\mathcal{X}$
- Polymorphism
- Shallow Polymorphism
- Future Work
- Questions

Background

Background

- Curry-Howard Correspondence

Background

- Curry-Howard Correspondence
  - Intuitionistic Implicative Logic

Background

- Curry-Howard Correspondence
  - Intuitionistic Implicative Logic
  - Lambda Calculus

Background

- Curry-Howard Correspondence
  - Intuitionistic Implicative Logic
  - Lambda Calculus
- Sequent Calculi

Background

- Curry-Howard Correspondence
  - Intuitionistic Implicative Logic
  - Lambda Calculus
- Sequent Calculi
- Cut Elimination

Curry-Howard Correspondence

Curry-Howard Correspondence

Intuitionistic Implicative Logic

Curry-Howard Correspondence

Intuitionistic Implicative Logic

$$\frac{}{\Gamma, A \vdash A}\,(ax)$$

Curry-Howard Correspondence

Intuitionistic Implicative Logic

$$\frac{}{\Gamma, A \vdash A}\,(ax)$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$$

Curry-Howard Correspondence

Intuitionistic Implicative Logic

$$\frac{}{\Gamma, A \vdash A}\,(\mathit{ax})$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$$

$$\frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$$

Curry-Howard Correspondence

Intuitionistic Implicative Logic

$$\frac{}{\Gamma, A \vdash A}\,(\mathit{ax})$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$$

$$\frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$$

λ-calculus

Curry-Howard Correspondence

Intuitionistic Implicative Logic	λ-calculus
$\dfrac{}{\Gamma, A \vdash A}\,(\mathit{ax})$	x
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$	
$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$	

Curry-Howard Correspondence

Intuitionistic Implicative Logic	λ-calculus
$\dfrac{}{\Gamma, A \vdash A}\,(\mathit{ax})$	x
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$	$\lambda x.M$
$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$	

Curry-Howard Correspondence

Intuitionistic Implicative Logic	λ-calculus
$\dfrac{}{\Gamma, A \vdash A}\,(\mathit{ax})$	x
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$	$\lambda x.M$
$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$	$(M\ N)$

Curry-Howard Correspondence

Intuitionistic Implicative Logic	λ-calculus
$\dfrac{}{\Gamma, A \vdash A}\,(ax)$	$\dfrac{}{\Gamma, x:A \vdash_\lambda x:A}\,(ax)$
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$	$\lambda x.M$
$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$	$(M\ N)$

Curry-Howard Correspondence

Intuitionistic Implicative Logic	λ-calculus
$\dfrac{}{\Gamma, A \vdash A}\,(\mathit{ax})$	$\dfrac{}{\Gamma, x : A \vdash_\lambda x : A}\,(\mathit{ax})$
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$	$\dfrac{\Gamma, x : A \vdash_\lambda M : B}{\Gamma \vdash_\lambda \lambda x.M : A \to B}\,(\to\mathcal{I})$
$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$	$(M\ N)$

Curry-Howard Correspondence

Intuitionistic Implicative Logic	λ-calculus
$\dfrac{}{\Gamma, A \vdash A}\,(\mathit{ax})$	$\dfrac{}{\Gamma, x : A \vdash_\lambda x : A}\,(\mathit{ax})$
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$	$\dfrac{\Gamma, x : A \vdash_\lambda M : B}{\Gamma \vdash_\lambda \lambda x.M : A \to B}\,(\to\mathcal{I})$
$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$	$\dfrac{\Gamma \vdash_\lambda M : A \to B \quad \Gamma \vdash_\lambda N : A}{\Gamma \vdash_\lambda (M\ N) : B}\,(\to\mathcal{E})$

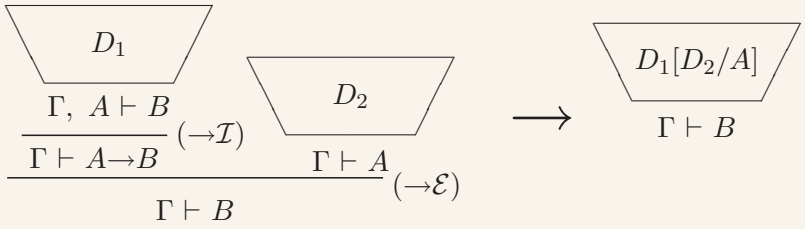
Curry-Howard Correspondence

Intuitionistic Implicative Logic	λ-calculus
$\dfrac{}{\Gamma, A \vdash A}\,(ax)$	$\dfrac{}{\Gamma, x:A \vdash_\lambda x:A}\,(ax)$
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$	$\dfrac{\Gamma, x:A \vdash_\lambda M:B}{\Gamma \vdash_\lambda \lambda x.M : A \to B}\,(\to\mathcal{I})$
$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$	$\dfrac{\Gamma \vdash_\lambda M : A \to B \quad \Gamma \vdash_\lambda N : A}{\Gamma \vdash_\lambda (M\ N) : B}\,(\to\mathcal{E})$

$$\dfrac{\dfrac{\begin{array}{c}D_1\\ \Gamma,\ A \vdash B\end{array}}{\Gamma \vdash A \to B}\,(\to\mathcal{I}) \qquad \begin{array}{c}D_2\\ \Gamma \vdash A\end{array}}{\Gamma \vdash B}\,(\to\mathcal{E})$$

Curry-Howard Correspondence

Intuitionistic Implicative Logic	λ-calculus
$\dfrac{}{\Gamma, A \vdash A}\,(ax)$	$\dfrac{}{\Gamma, x:A \vdash_\lambda x:A}\,(ax)$
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$	$\dfrac{\Gamma, x:A \vdash_\lambda M:B}{\Gamma \vdash_\lambda \lambda x.M : A \to B}\,(\to\mathcal{I})$
$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$	$\dfrac{\Gamma \vdash_\lambda M : A \to B \quad \Gamma \vdash_\lambda N : A}{\Gamma \vdash_\lambda (M\ N) : B}\,(\to\mathcal{E})$



Curry-Howard Correspondence

Intuitionistic Implicative Logic	λ-calculus
$\dfrac{}{\Gamma, A \vdash A}\,(ax)$	$\dfrac{}{\Gamma, x:A \vdash_\lambda x:A}\,(ax)$
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to\mathcal{I})$	$\dfrac{\Gamma, x:A \vdash_\lambda M:B}{\Gamma \vdash_\lambda \lambda x.M : A \to B}\,(\to\mathcal{I})$
$\dfrac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to\mathcal{E})$	$\dfrac{\Gamma \vdash_\lambda M : A \to B \quad \Gamma \vdash_\lambda N : A}{\Gamma \vdash_\lambda (M\ N) : B}\,(\to\mathcal{E})$
$\dfrac{\dfrac{\overset{D_1}{\Gamma,\ A \vdash B}}{\Gamma \vdash A \to B}(\to\mathcal{I}) \quad \overset{D_2}{\Gamma \vdash A}}{\Gamma \vdash B}(\to\mathcal{E}) \;\longrightarrow\; \overset{D_1[D_2/A]}{\Gamma \vdash B}$	$((\lambda x.M_1)\ M_2) \to M_1[M_2/x]$

Classical Logic

Classical Logic

- Intuitionistic Logic requires all proofs are “constructive”.

Classical Logic

- Intuitionistic Logic requires all proofs are "constructive".
  - e.g. can only prove $A \vee B$ if we can prove either A or B seperately.

Classical Logic

- Intuitionistic Logic requires all proofs are "constructive".
  - e.g. can only prove $A \vee B$ if we can prove either A or B seperately.
- In Classical Logic, we accept 'only two truth values'.

Classical Logic

- Intuitionistic Logic requires all proofs are "constructive".
  - e.g. can only prove $A \vee B$ if we can prove either A or B seperately.
- In Classical Logic, we accept 'only two truth values'.
  - Indirect proof (by contradiction).

Classical Logic

- Intuitionistic Logic requires all proofs are "constructive".
  - e.g. can only prove $A \vee B$ if we can prove either A or B seperately.
- In Classical Logic, we accept 'only two truth values'.
  - Indirect proof (by contradiction).
  - e.g. can prove $A \vee B$ if $\neg A \wedge \neg B$ gives a contradiction.

Classical Logic

- Intuitionistic Logic requires all proofs are "constructive".
  - e.g. can only prove $A \vee B$ if we can prove either A or B seperately.
- In Classical Logic, we accept 'only two truth values'.
  - Indirect proof (by contradiction).
  - e.g. can prove $A \vee B$ if $\neg A \wedge \neg B$ gives a contradiction.
- Even in the implicative case, there is a difference

Classical Logic

- Intuitionistic Logic requires all proofs are "constructive".
  - e.g. can only prove $A \vee B$ if we can prove either A or B seperately.
- In Classical Logic, we accept 'only two truth values'.
  - Indirect proof (by contradiction).
  - e.g. can prove $A \vee B$ if $\neg A \wedge \neg B$ gives a contradiction.
- Even in the implicative case, there is a difference
  - $((A \to B) \to A) \to A$ (Peirce's Law).

Classical Logic

- Intuitionistic Logic requires all proofs are "constructive".
  - e.g. can only prove $A \vee B$ if we can prove either A or B seperately.
- In Classical Logic, we accept 'only two truth values'.
  - Indirect proof (by contradiction).
  - e.g. can prove $A \vee B$ if $\neg A \wedge \neg B$ gives a contradiction.
- Even in the implicative case, there is a difference
  - $((A \to B) \to A) \to A$ (Peirce's Law).
  - Reading B as $\bot$, "If assuming $\neg A$ lets us deduce A (contradiction) then A must be true".

Sequent Calculus

Sequent Calculus

- Natural Deduction provides intuitive arguments.

Sequent Calculus

- Natural Deduction provides intuitive arguments.
- Not very methodical, difficult to prove meta-theoretical results.

Sequent Calculus

- Natural Deduction provides intuitive arguments.
- Not very methodical, difficult to prove meta-theoretical results.
- Gentzen (also) invented the Sequent Calculus for this purpose.

Sequent Calculus

- Natural Deduction provides intuitive arguments.
- Not very methodical, difficult to prove meta-theoretical results.
- Gentzen (also) invented the Sequent Calculus for this purpose.
- Idea: Carry all assumptions with you as you go.

Sequent Calculus

- Natural Deduction provides intuitive arguments.
- Not very methodical, difficult to prove meta-theoretical results.
- Gentzen (also) invented the Sequent Calculus for this purpose.
- Idea: Carry all assumptions with you as you go.
- Sequents $A_1, A_2, \ldots, A_m \vdash B$. "If A_i are all true then B is true".

Sequent Calculus

- Natural Deduction provides intuitive arguments.
- Not very methodical, difficult to prove meta-theoretical results.
- Gentzen (also) invented the Sequent Calculus for this purpose.
- Idea: Carry all assumptions with you as you go.
- Sequents $A_1, A_2, \ldots, A_m \vdash B$. "If A_i are all true then B is true".
- For Classical Logic, allow multiple conclusions

Sequent Calculus

- Natural Deduction provides intuitive arguments.
- Not very methodical, difficult to prove meta-theoretical results.
- Gentzen (also) invented the Sequent Calculus for this purpose.
- Idea: Carry all assumptions with you as you go.
- Sequents $A_1, A_2, \ldots, A_m \vdash B$. "If A_i are all true then B is true".
- For Classical Logic, allow multiple conclusions
  - $A_1, A_2, \ldots, A_m \vdash B_1, B_2, \ldots, B_n$

Sequent Calculus

- Natural Deduction provides intuitive arguments.
- Not very methodical, difficult to prove meta-theoretical results.
- Gentzen (also) invented the Sequent Calculus for this purpose.
- Idea: Carry all assumptions with you as you go.
- Sequents $A_1, A_2, \ldots, A_m \vdash B$. "If A_i are all true then B is true".
- For Classical Logic, allow multiple conclusions
  - $A_1, A_2, \ldots, A_m \vdash B_1, B_2, \ldots, B_n$
  - "If A_i are all true then at least one of B_j is true".

Sequent Calculus

- Natural Deduction provides intuitive arguments.
- Not very methodical, difficult to prove meta-theoretical results.
- Gentzen (also) invented the Sequent Calculus for this purpose.
- Idea: Carry all assumptions with you as you go.
- Sequents $A_1, A_2, \ldots, A_m \vdash B$. "If A_i are all true then B is true".
- For Classical Logic, allow multiple conclusions
  - $A_1, A_2, \ldots, A_m \vdash B_1, B_2, \ldots, B_n$
  - "If A_i are all true then at least one of B_j is true".
  - Abbreviate to $\Gamma \vdash \Delta$.

Sequent Calculus

Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\ (\mathrm{Ax})$$

Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\,(\mathrm{Ax}) \qquad\qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\,(\to\mathcal{R})$$

Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\,(\mathrm{Ax})$$

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\,(\to\mathcal{R})$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\,(\to\mathcal{L})$$

Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\,(\mathrm{Ax}) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\,(\to\mathcal{R})$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\,(\to\mathcal{L}) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\,(\mathrm{cut})$$

Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\,(\mathrm{Ax}) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\,(\to\mathcal{R})$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\,(\to\mathcal{L}) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\,(\mathrm{cut})$$

- No rules for elimination; introduction on left and right.

Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\,(\mathrm{Ax}) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\,(\to\mathcal{R})$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\,(\to\mathcal{L}) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\,(\mathrm{cut})$$

- No rules for elimination; introduction on left and right.
- Cut allows two proofs to be 'connected' together.

Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\,(\mathrm{Ax}) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\,(\to\mathcal{R})$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\,(\to\mathcal{L}) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\,(\mathrm{cut})$$

- No rules for elimination; introduction on left and right.
- Cut allows two proofs to be 'connected' together.
- Subformula property (apart from the cut rule).

Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\,(\mathrm{Ax}) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\,(\to\mathcal{R})$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\,(\to\mathcal{L}) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\,(\mathrm{cut})$$

- No rules for elimination; introduction on left and right.
- Cut allows two proofs to be 'connected' together.
- Subformula property (apart from the cut rule).
- Cuts are useful, but make meta-theoretical results awkward.

Cut Elimination

Cut Elimination

Theorem (Gentzen): *The cut is redundant in the sequent calculus.*

Cut Elimination

Theorem (Gentzen): *The cut is redundant in the sequent calculus.*

- A set of rules for *cut elimination* was defined.

Cut Elimination

Theorem (Gentzen): *The cut is redundant in the sequent calculus.*

- A set of rules for *cut elimination* was defined.
- These rules give a notion of *reduction* on sequent proofs.

Cut Elimination

Theorem (Gentzen): *The cut is redundant in the sequent calculus.*

- A set of rules for *cut elimination* was defined.
- These rules give a notion of *reduction* on sequent proofs.
- Any proof can be reduced to a cut-free proof of the same sequent.

Cut Elimination

Theorem (Gentzen): *The cut is redundant in the sequent calculus.*

- A set of rules for *cut elimination* was defined.
- These rules give a notion of *reduction* on sequent proofs.
- Any proof can be reduced to a cut-free proof of the same sequent.
- The cut-free sequent calculus has the subformula property.

Cut Elimination

Theorem (Gentzen): *The cut is redundant in the sequent calculus.*

- A set of rules for *cut elimination* was defined.
- These rules give a notion of *reduction* on sequent proofs.
- Any proof can be reduced to a cut-free proof of the same sequent.
- The cut-free sequent calculus has the subformula property.
  - Easy consistency proof.

Cut Elimination

Theorem (Gentzen): *The cut is redundant in the sequent calculus.*

- A set of rules for *cut elimination* was defined.
- These rules give a notion of *reduction* on sequent proofs.
- Any proof can be reduced to a cut-free proof of the same sequent.
- The cut-free sequent calculus has the subformula property.
  - Easy consistency proof.
  - Proof-search is straightforward.

Cut Elimination

Theorem (Gentzen): *The cut is redundant in the sequent calculus.*

- A set of rules for *cut elimination* was defined.
- These rules give a notion of *reduction* on sequent proofs.
- Any proof can be reduced to a cut-free proof of the same sequent.
- The cut-free sequent calculus has the subformula property.
  - Easy consistency proof.
  - Proof-search is straightforward.
- What does this notion of reduction mean computationally?

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$).

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$).
Four basic syntax constructs:

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets $(x, y, z, \ldots)$ and plugs $(\alpha, \beta, \gamma, \ldots)$.
Four basic syntax constructs:

$$P, Q \ ::= \ \langle x.\alpha \rangle \qquad (\text{capsule})$$

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets $(x, y, z, \ldots)$ and plugs $(\alpha, \beta, \gamma, \ldots)$.
Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \end{array}$$

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$).
Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha}\, [y]\, \widehat{x} Q & (\text{mediator}) \end{array}$$

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$).
Four basic syntax constructs:

$$
\begin{array}{rcll}
P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\
& | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\
& | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\
& | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut})
\end{array}
$$

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Capsules* Simplest terms, connecting one input (socket) directly to one output (plug).

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Capsules* Simplest terms, connecting one input (socket) directly to one output (plug).
- *Exports* Reminiscent of function abstractions; binding an input and an output of P, and exporting the corresponding 'function'.

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Capsules* Simplest terms, connecting one input (socket) directly to one output (plug).
- *Exports* Reminiscent of function abstractions; binding an input and an output of P, and exporting the corresponding 'function'.
- *Mediators* Two subterms P and Q, which require a term to 'mediate' in between.

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Cuts* An active computation, attempting to connect each α in P with each x in Q.

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Cuts* An active computation, attempting to connect each α in P with each x in Q.
- Simplest case: exactly one α and x are immediately exhibited by P and Q.

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Cuts* An active computation, attempting to connect each α in P with each x in Q.
- Simplest case: exactly one α and x are immediately exhibited by P and Q.
  - The cut can be eliminated, possibly creating new ones in the process.

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Cuts* An active computation, attempting to connect each α in P with each x in Q.
- Simplest case: exactly one α and x are immediately exhibited by P and Q.
  - The cut can be eliminated, possibly creating new ones in the process.
  - This is called a *logical rule*

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Cuts* An active computation, attempting to connect each α in P with each x in Q.

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Cuts* An active computation, attempting to connect each α in P with each x in Q.
- If a logical rule cannot be applied, the cut must *propagate* into the structure of P and/or Q, to 'single out' the αs and xs.

The $\mathcal{X}$-calculus

Two kinds of *connectors*: sockets ($x, y, z, \ldots$) and plugs ($\alpha, \beta, \gamma, \ldots$). Four basic syntax constructs:

$$\begin{array}{rcll} P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\ & | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\ & | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\ & | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \end{array}$$

- *Cuts* An active computation, attempting to connect each α in P with each x in Q.
- If a logical rule cannot be applied, the cut must *propagate* into the structure of P and/or Q, to 'single out' the αs and xs.
- Sometimes cuts may propagate into P (left propagation) or Q (right propagation).

The $\mathcal{X}$-calculus

We extend the syntax:

$$
\begin{array}{rcll}
P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\
& | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\
& | & P \widehat{\alpha} \, [y] \, \widehat{x} Q & (\text{mediator}) \\
& | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut})
\end{array}
$$

The $\mathcal{X}$-calculus

We extend the syntax:

$$
\begin{array}{rcll}
P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\
& | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\
& | & P \widehat{\alpha}\, [y]\, \widehat{x} Q & (\text{mediator}) \\
& | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \\
& | & P \widehat{\alpha} \not\nearrow \widehat{x} Q & (\text{left cut}) \\
& | & P \widehat{\alpha} \not\nwarrow \widehat{x} Q & (\text{right cut})
\end{array}
$$

The $\mathcal{X}$-calculus

We extend the syntax:

$$
\begin{array}{rcll}
P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\
& | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\
& | & P \widehat{\alpha}\, [y]\, \widehat{x} Q & (\text{mediator}) \\
& | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \\
& | & P \widehat{\alpha} \not\nearrow \widehat{x} Q & (\text{left cut}) \\
& | & P \widehat{\alpha} \not\nwarrow \widehat{x} Q & (\text{right cut})
\end{array}
$$

- These *active cuts* indicate the direction of propagation.

The $\mathcal{X}$-calculus

We extend the syntax:

$$
\begin{array}{rcll}
P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\
& | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\
& | & P \widehat{\alpha}\, [y]\, \widehat{x} Q & (\text{mediator}) \\
& | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \\
& | & P \widehat{\alpha} \not\swarrow \widehat{x} Q & (\text{left cut}) \\
& | & P \widehat{\alpha} \not\searrow \widehat{x} Q & (\text{right cut})
\end{array}
$$

- These *active cuts* indicate the direction of propagation.
- An active cut propagates until it reaches a capsule.

The $\mathcal{X}$-calculus

We extend the syntax:

$$
\begin{array}{rcll}
P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\
& | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\
& | & P \widehat{\alpha}\, [y]\, \widehat{x} Q & (\text{mediator}) \\
& | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \\
& | & P \widehat{\alpha} \not\nearrow \widehat{x} Q & (\text{left cut}) \\
& | & P \widehat{\alpha} \not\nwarrow \widehat{x} Q & (\text{right cut})
\end{array}
$$

- These *active cuts* indicate the direction of propagation.
- An active cut propagates until it reaches a capsule.
  - If capsule introduces the connective the cut is seeking, it deactivates

The $\mathcal{X}$-calculus

We extend the syntax:

$$
\begin{array}{rcll}
P, Q & ::= & \langle x.\alpha \rangle & (\text{capsule}) \\
& | & \widehat{x} P \widehat{\alpha} \cdot \beta & (\text{export}) \\
& | & P \widehat{\alpha}\, [y]\, \widehat{x} Q & (\text{mediator}) \\
& | & P \widehat{\alpha} \dagger \widehat{x} Q & (\text{cut}) \\
& | & P \widehat{\alpha} \not\swarrow \widehat{x} Q & (\text{left cut}) \\
& | & P \widehat{\alpha} \not\searrow \widehat{x} Q & (\text{right cut})
\end{array}
$$

- These *active cuts* indicate the direction of propagation.
- An active cut propagates until it reaches a capsule.
  - If capsule introduces the connective the cut is seeking, it deactivates
  - Otherwise, the cut is destroyed (the capsule remains).

Reduction Rules

Logical Rules

Reduction Rules

Logical Rules

"Capsules rename, Exports insert into Mediators"

Reduction Rules

Logical Rules

"Capsules rename, Exports insert into Mediators"

$$(cap): \qquad \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle \rightarrow \langle y.\beta\rangle$$

Reduction Rules

Logical Rules

"Capsules rename, Exports insert into Mediators"

$$(cap): \quad \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle \rightarrow \langle y.\beta\rangle$$

$$(exp): \quad (\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x.\gamma\rangle \rightarrow \widehat{y}P\widehat{\beta}\cdot\gamma \qquad \alpha \notin fs(P)$$

Reduction Rules

Logical Rules

"Capsules rename, Exports insert into Mediators"

$$
\begin{array}{lcll}
(cap): & \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle \rightarrow \langle y.\beta\rangle & & \\
(exp): & (\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x.\gamma\rangle \rightarrow \widehat{y}P\widehat{\beta}\cdot\gamma & & \alpha \notin fs(P) \\
(med): & \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}(P\widehat{\beta}\,[x]\,\widehat{z}Q) \rightarrow P\widehat{\beta}\,[y]\,\widehat{z}Q & & x \notin fs(P,Q)
\end{array}
$$

Reduction Rules

Logical Rules

"Capsules rename, Exports insert into Mediators"

$$
\begin{array}{lll}
(cap): & \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle \rightarrow \langle y.\beta\rangle & \\
(exp): & (\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x.\gamma\rangle \rightarrow \widehat{y}P\widehat{\beta}\cdot\gamma & \alpha \notin fs(P) \\
(med): & \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}(P\widehat{\beta}\,[x]\,\widehat{z}Q) \rightarrow P\widehat{\beta}\,[y]\,\widehat{z}Q & x \notin fs(P,Q) \\
(exp\text{-}med): & (\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}\,[x]\,\widehat{z}R) \rightarrow &
\end{array}
$$

Reduction Rules

Logical Rules

"Capsules rename, Exports insert into Mediators"

$$(cap): \quad \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}\langle x.\beta\rangle \to \langle y.\beta\rangle$$

$$(exp): \quad (\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}\langle x.\gamma\rangle \to \widehat{y}P\widehat{\beta}\cdot\gamma \qquad \alpha \notin fs(P)$$

$$(med): \quad \langle y.\alpha\rangle\widehat{\alpha} \dagger \widehat{x}(P\widehat{\beta}\,[x]\,\widehat{z}Q) \to P\widehat{\beta}\,[y]\,\widehat{z}Q \qquad x \notin fs(P,Q)$$

$$(exp\text{-}med): (\widehat{y}P\widehat{\beta}\cdot\alpha)\widehat{\alpha} \dagger \widehat{x}(Q\widehat{\gamma}\,[x]\,\widehat{z}R) \to$$

$$\textit{either} \begin{cases} Q\widehat{\gamma} \dagger \widehat{y}(P\widehat{\beta} \dagger \widehat{z}R) \\ (Q\widehat{\gamma} \dagger \widehat{y}P)\widehat{\beta} \dagger \widehat{z}R \end{cases} \qquad \alpha \notin fs(P), x \notin fs(Q,R)$$

Reduction Rules

Activation Rules

$(act\text{-}\mathrm{L}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\prec \widehat{x}Q$ *if* P *does not freshly introduce* α

$(act\text{-}\mathrm{R}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\succ \widehat{x}Q$ *if* Q *does not freshly introduce* x

Propagation

Reduction Rules

Activation Rules

$(act\text{-}\mathrm{L}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\prec \widehat{x}Q$ *if* P *does not freshly introduce* α

$(act\text{-}\mathrm{R}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\succ \widehat{x}Q$ *if* Q *does not freshly introduce* x

Propagation

- A right cut $P\widehat{\alpha} \not\succ \widehat{x}Q$ propagates inside structure of Q.

Reduction Rules

Activation Rules

$(act\text{-L}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\prec \widehat{x}Q$ *if* P *does not freshly introduce* α

$(act\text{-R}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\succ \widehat{x}Q$ *if* Q *does not freshly introduce* x

Propagation

- A right cut $P\widehat{\alpha} \not\succ \widehat{x}Q$ propagates inside structure of Q.
  - Creates a cut between P and each x found.

Reduction Rules

Activation Rules

$(act\text{-}\mathrm{L}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\prec \widehat{x}Q$ *if* P *does not freshly introduce* α

$(act\text{-}\mathrm{R}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\succ \widehat{x}Q$ *if* Q *does not freshly introduce* x

Propagation

- A right cut $P\widehat{\alpha} \not\succ \widehat{x}Q$ propagates inside structure of Q.
  - Creates a cut between P and each x found.
  - Reminiscent of term substitution $Q[P/x]$.

Reduction Rules

Activation Rules

$(act\text{-}\mathrm{L}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\prec \widehat{x}Q$ *if* P *does not freshly introduce* α

$(act\text{-}\mathrm{R}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\succ \widehat{x}Q$ *if* Q *does not freshly introduce* x

Propagation

- A right cut $P\widehat{\alpha} \not\succ \widehat{x}Q$ propagates inside structure of Q.
  - Creates a cut between P and each x found.
  - Reminiscent of term substitution $Q[P/x]$.
- Left cuts propagate inside P.

Reduction Rules

Activation Rules

$(act\text{-L}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\prec \widehat{x}Q$ *if* P *does not freshly introduce* α

$(act\text{-R}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\succ \widehat{x}Q$ *if* Q *does not freshly introduce* x

Propagation

- A right cut $P\widehat{\alpha} \not\succ \widehat{x}Q$ propagates inside structure of Q.
  - Creates a cut between P and each x found.
  - Reminiscent of term substitution $Q[P/x]$.
- Left cuts propagate inside P.
  - 'Dual' notion of substitution ($P[Q/\alpha]$?).

Reduction Rules

Activation Rules

$(act\text{-}\mathrm{L}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\prec \widehat{x}Q$ *if* P *does not freshly introduce* α

$(act\text{-}\mathrm{R}) : P\widehat{\alpha} \dagger \widehat{x}Q \to P\widehat{\alpha} \not\succ \widehat{x}Q$ *if* Q *does not freshly introduce* x

Propagation

- A right cut $P\widehat{\alpha} \not\succ \widehat{x}Q$ propagates inside structure of Q.
  - Creates a cut between P and each x found.
  - Reminiscent of term substitution $Q[P/x]$.
- Left cuts propagate inside P.
  - 'Dual' notion of substitution ($P[Q/\alpha]$?).
  - Take an evaluation context Q to the 'calls' in P.

Aims

Aims

- Better understanding of the $\mathcal{X}$-calculus.

Aims

- Better understanding of the $\mathcal{X}$-calculus.
  - How does it compare to ‘similar’ calculi?

Aims

- Better understanding of the $\mathcal{X}$-calculus.
  - How does it compare to 'similar' calculi?
  - What can be achieved computationally?

Aims

- Better understanding of the $\mathcal{X}$-calculus.
  - How does it compare to 'similar' calculi?
  - What can be achieved computationally?
- Extending the calculus with programming features

Aims

- Better understanding of the $\mathcal{X}$-calculus.
  - How does it compare to 'similar' calculi?
  - What can be achieved computationally?
- Extending the calculus with programming features
  - Polymorphism

Aims

- Better understanding of the $\mathcal{X}$-calculus.
  - How does it compare to 'similar' calculi?
  - What can be achieved computationally?
- Extending the calculus with programming features
  - Polymorphism
  - Recursion

Aims

- Better understanding of the $\mathcal{X}$-calculus.
  - How does it compare to 'similar' calculi?
  - What can be achieved computationally?
- Extending the calculus with programming features
  - Polymorphism
  - Recursion
  - Other logical connectives (continuations).

Polymorphism (on the whiteboard)

Polymorphism (on the whiteboard)

- System F style

Polymorphism (on the whiteboard)

- System F style
- Shallow Polymorphism

Polymorphism (on the whiteboard)

- System F style
- Shallow Polymorphism
  - Encoding ML.

Polymorphism (on the whiteboard)

- System F style
- Shallow Polymorphism
  - Encoding ML.
  - Taking multiple instances.

Polymorphism (on the whiteboard)

- System F style
- Shallow Polymorphism
  - Encoding ML.
  - Taking multiple instances.
  - Symmetric Shallow Polymorphism.

Other Work

Other Work

- Extending $\mathcal{X}$ with other logical connectives.

Other Work

- Extending $\mathscr{X}$ with other logical connectives.
  - Negation ($\neg$), for continuations.

Other Work

- Extending $\mathscr{X}$ with other logical connectives.
  - Negation ($\neg$), for continuations.
  - Conjunction and Disjunction ($\wedge, \vee$), for pairing.

Other Work

- Extending $\mathscr{X}$ with other logical connectives.
  - Negation ($\neg$), for continuations.
  - Conjunction and Disjunction ($\wedge, \vee$), for pairing.
- Strong Normalisation.

Other Work

- Extending $\mathscr{X}$ with other logical connectives.
  - Negation ($\neg$), for continuations.
  - Conjunction and Disjunction ($\wedge, \vee$), for pairing.
- Strong Normalisation.
- Comparisons with other calculi.

Questions

Questions

Are there any?

Sequent Calculus

$$\dfrac{\dfrac{\dfrac{}{\langle y.\pi\rangle \mathrel{\vdots} y:\varphi \vdash_{\text{SP}} \pi:\varphi}\,(cap)}{\widehat{y}\langle y.\pi\rangle\widehat{\pi}\cdot\theta \mathrel{\vdots} \vdash_{\text{SP}} \theta:\varphi\to\varphi}\,(exp)}{\widehat{y}\langle y.\pi\rangle\widehat{\pi}\cdot\theta \mathrel{\vdots} \vdash_{\text{SP}} \theta:\forall X.(X\to X)}\,(\forall\mathcal{R}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\langle x.\gamma\rangle \mathrel{\vdots} x:A\to A \vdash_{\text{SP}} \gamma:A\to A,\alpha:A\to A}\,(cap) \qquad \dfrac{}{\langle p.\alpha\rangle \mathrel{\vdots} p:A\to A, x:A\to A \vdash_{\text{SP}} \alpha:A\to A}\,(cap)}{\langle x.\gamma\rangle\widehat{\gamma}\,[x]\,\widehat{p}\langle p.\alpha\rangle \mathrel{\vdots} x:(A\to A)\to(A\to A), x:A\to A \vdash_{\text{SP}} \alpha:A\to A}\,(med)}{\langle x.\gamma\rangle\widehat{\gamma}\,[x]\,\widehat{p}\langle p.\alpha\rangle \mathrel{\vdots} x:(A\to A)\to(A\to A), x:\forall X.(X\to X) \vdash_{\text{SP}} \alpha:A\to A}\,(\forall\mathcal{L})}{\langle x.\gamma\rangle\widehat{\gamma}\,[x]\,\widehat{p}\langle p.\alpha\rangle \mathrel{\vdots} x:\forall X.(X\to X) \vdash_{\text{SP}} \alpha:A\to A}\,(\forall\mathcal{L})$$

$$\dfrac{\widehat{y}\langle y.\pi\rangle\widehat{\pi}\cdot\theta \mathrel{\vdots} \vdash_{\text{SP}} \theta:\forall X.(X\to X) \qquad \langle x.\gamma\rangle\widehat{\gamma}\,[x]\,\widehat{p}\langle p.\alpha\rangle \mathrel{\vdots} x:\forall X.(X\to X) \vdash_{\text{SP}} \alpha:A\to A}{(\widehat{y}\langle y.\pi\rangle\widehat{\pi}\cdot\theta)\widehat{\theta}\,\dagger\,\widehat{x}(\langle x.\gamma\rangle\widehat{\gamma}\,[x]\,\widehat{p}\langle p.\alpha\rangle) \mathrel{\vdots} \vdash_{\text{SP}} \alpha:A\to A}\,(cut)$$

Sequent Calculus

$$\dfrac{\dfrac{\dfrac{\dfrac{}{y : \varphi \vdash_{\mathrm{ML}} y : \varphi}\,(ax)}{\vdash_{\mathrm{ML}} \lambda y.y : \varphi \to \varphi}\,(\to\mathcal{I})}{\vdash_{\mathrm{ML}} \lambda y.y : \forall Z.Z \to Z}\,(\forall\mathcal{I}) \qquad \dfrac{\dfrac{\dfrac{}{x : \forall Z.Z \to Z \vdash_{\mathrm{ML}} x : \forall Z.Z \to Z}\,(ax)}{x : \forall Z.Z \to Z \vdash_{\mathrm{ML}} x : (A \to A) \to (A \to A)}\,(\forall\mathcal{E}) \qquad \dfrac{\dfrac{}{x : \forall Z.Z \to Z \vdash_{\mathrm{ML}} x : \forall Z.Z \to Z}\,(ax)}{x : \forall Z.Z \to Z \vdash_{\mathrm{ML}} x : A \to A}\,(\forall\mathcal{E})}{x : \forall Z.Z \to Z \vdash_{\mathrm{ML}} x\ x : A \to A}\,(\to\mathcal{E})}{\vdash_{\mathrm{ML}} \mathtt{let}\ x = \lambda y.y\ \mathtt{in}\ x\ x : A \to A}\,(\mathtt{let})$$

Sequent Calculus

$$\frac{}{\Gamma, A \vdash A, \Delta}\,(\mathrm{Ax}) \qquad \frac{}{\langle x.\alpha\rangle :\cdot \Gamma, x : A \vdash \alpha : A, \Delta}\,(cap)$$

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A{\to}B, \Delta}\,(\to\mathcal{R}) \qquad \frac{P :\cdot \Gamma, x : A \vdash \alpha : B, \Delta}{\widehat{x}P\widehat{\alpha}\cdot\beta :\cdot \Gamma \vdash \beta : A{\to}B, \Delta}\,(exp)$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A{\to}B \vdash \Delta}\,(\to\mathcal{L}) \qquad \frac{P :\cdot \Gamma \vdash \alpha : A, \Delta \quad Q :\cdot \Gamma, y : B \vdash \Delta}{P\widehat{\alpha}\,[x]\,\widehat{y}Q :\cdot \Gamma, x : A{\to}B \vdash \Delta}\,(med)$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}\,(\mathrm{cut}) \qquad \frac{P :\cdot \Gamma \vdash \alpha : A, \Delta \quad Q :\cdot \Gamma, x : A \vdash \Delta}{P\widehat{\alpha} \dagger \widehat{x}Q :\cdot \Gamma \vdash \Delta}\,(cut)$$