System $\mathbf{F}_\omega$ as a Pure Type System: $\boldsymbol{\lambda\omega}$

PTS specification $\omega = (\mathcal{S}_\omega, \mathcal{A}_\omega, \mathcal{R}_\omega)$:

$$\mathcal{S}_\omega \triangleq \{*, \Box\}$$

$$\mathcal{A} \triangleq \{(*, \Box)\}$$

$$\mathcal{R} \triangleq \{(*, *, *), (\Box, *, *), (\Box, \Box, \Box)\}$$

"F_ω is the work horse of modern compilers"

(Greg Morrisett)

System $\mathbf{F}_\omega$ as a Pure Type System: $\boldsymbol{\lambda\omega}$

PTS specification $\omega = (\mathcal{S}_\omega, \mathcal{A}_\omega, \mathcal{R}_\omega)$:

$$\mathcal{S}_\omega \triangleq \{*, \Box\}$$
$$\mathcal{A} \triangleq \{(*, \Box)\}$$
$$\mathcal{R} \triangleq \{(*, *, *), (\Box, *, *), (\Box, \Box, \Box)\}$$

As in $\boldsymbol{\lambda 2}$, sort $*$ is a universe of types; but in $\boldsymbol{\lambda\omega}$, the rule (**prod**) for $(\Box, \Box, \Box)$ means that $\diamond \vdash t : \Box$ holds for all the 'simple types' over the ground type $*$ – the ts generated by the grammar $t ::= * \mid t \to t$

$$(\text{prod})\ \frac{\Gamma \vdash A : \Box \qquad \Gamma, x : A \vdash B : \Box}{\Gamma \vdash \Pi x : A\,(B) : \Box} \quad \text{for } (\Box, \Box, \Box)$$

System $\mathbf{F}_\omega$ as a Pure Type System: $\boldsymbol{\lambda\omega}$

PTS specification $\omega = (\mathcal{S}_\omega, \mathcal{A}_\omega, \mathcal{R}_\omega)$:

$$\mathcal{S}_\omega \triangleq \{*, \square\}$$

$$\mathcal{A} \triangleq \{(*, \square)\}$$

$$\mathcal{R} \triangleq \{(*, *, *), (\square, *, *), (\square, \square, \square)\}$$

As in $\boldsymbol{\lambda 2}$, sort $*$ is a universe of types; but in $\boldsymbol{\lambda\omega}$, the rule (**prod**) for $(\square, \square, \square)$ means that $\diamond \vdash t : \square$ holds for all the 'simple types' over the ground type $*$ – the ts generated by the grammar $t ::= * \mid \underbrace{t \to t}$

$$(\text{prod})\ \frac{\Gamma \vdash A : \square \qquad \Gamma, x : A \vdash B : \square}{\Gamma \vdash \Pi x : A\,(B) : \square} \quad \text{for } (\square, \square, \square)$$

$(A \to B \triangleq \Pi x : A\,(B) \text{ with } x \notin fv(B))$

System $\mathbf{F}_\omega$ as a Pure Type System: $\boldsymbol{\lambda\omega}$

PTS specification $\omega = (\mathcal{S}_\omega, \mathcal{A}_\omega, \mathcal{R}_\omega)$:

$$\mathcal{S}_\omega \triangleq \{*, \Box\}$$
$$\mathcal{A} \triangleq \{(*, \Box)\}$$
$$\mathcal{R} \triangleq \{(*, *, *), (\Box, *, *), (\Box, \Box, \Box)\}$$

As in $\boldsymbol{\lambda 2}$, sort $*$ is a universe of types; but in $\boldsymbol{\lambda\omega}$, the rule (**prod**) for $(\Box, \Box, \Box)$ means that $\diamond \vdash t : \Box$ holds for all the 'simple types' over the ground type $*$ – the ts generated by the grammar $t ::= * \mid t \to t$

Hence rule (**prod**) for $(\Box, *, *)$ now gives many more legal pseudo-terms of type $*$ in $\boldsymbol{\lambda\omega}$ compared with $\boldsymbol{\lambda 2}$ (PLC), such as

$$\diamond \vdash (\Pi T : * \to * \, (\Pi \alpha : * \, (\alpha \to T\,\alpha))) : *$$
$$\diamond \vdash (\Pi T : * \to * \, (\Pi \alpha, \beta : * \, ((\alpha \to T\,\beta) \to T\,\alpha \to T\,\beta))) : *$$

types for unit & lift operations, making T a monad

Examples of $\lambda\omega$ type constructions

- Product types (cf. the PLC representation of product types):

$$\mathtt{P} \triangleq \lambda\alpha, \beta : *\,(\Pi\gamma : *\,((\alpha \to \beta \to \gamma) \to \gamma))$$

$$\diamond \vdash \mathtt{P} : * \to * \to *$$

TYPO ALERT!

$$\tau \times \tau' \triangleq \forall\gamma\,((\tau \to \tau' \to \gamma) \to \gamma)$$

where $\gamma \notin ftv(\tau, \tau')$

Examples of $\lambda\omega$ type constructions

- Product types (cf. the PLC representation of product types):

$$\mathtt{P} \triangleq \lambda \alpha, \beta : * \, (\Pi \gamma : * \, ((\alpha \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))$$

$$\diamond \vdash \mathtt{P} : * \rightarrow * \rightarrow *$$

$$\exists \alpha (\tau) \triangleq \forall \beta ((\forall \alpha (\tau \rightarrow \beta)) \rightarrow \beta)$$

where $\beta \notin ftv(\tau)$ & $\beta \neq \alpha$

- Existential types (cf. the PLC representation of existential types):

$$\exists \triangleq \lambda T : * \rightarrow * \, (\Pi \beta : * \, ((\Pi \alpha : * \, (T \, \alpha \rightarrow \beta)) \rightarrow \beta))$$

$$\diamond \vdash \exists : (* \rightarrow *) \rightarrow *$$

Type-checking for $\mathbf{F}_\omega$ $(\lambda\omega)$

Fact (Girard): System $\mathbf{F}_\omega$ is *strongly normalizing* in the sense that for any legal pseudo-term t, there is no infinite chain of beta-reductions $t \to t_1 \to t_2 \to \cdots$.

Type-checking for $\mathbf{F}_\omega$

Fact (Girard): System $\mathbf{F}_\omega$ is *strongly normalizing* in the sense that for any legal pseudo-term t, there is no infinite chain of beta-reductions $t \to t_1 \to t_2 \to \cdots$.

As as corollary we have that the type-checking and typeability problems for $\mathbf{F}_\omega$ are decidable.

$(\lambda\omega)$

Propositions as Types

(sect. 6 of notes)

Curry-Howard correspondence

Logic	$\leftrightarrow$	Type system
propositions ϕ	$\leftrightarrow$	types τ
proofs p	$\leftrightarrow$	expressions M
'p is a proof of ϕ'	$\leftrightarrow$	'M is an expression of type τ'
simplification of proofs	$\leftrightarrow$	reduction of expressions

First arose for <u>constructive</u> logics

Constructive interpretation of logic

- **Implication:** a proof of $\varphi \rightarrow \psi$ is a construction that transforms proofs of φ into proofs of ψ.
- **Negation:** a proof of $\neg\varphi$ is a construction that from any (hypothetical) proof of φ produces a contradiction (= proof of falsity $\perp$)
- **Disjunction:** a proof of $\varphi \vee \psi$ is an object that manifestly is either a proof of φ, or a proof of ψ.
- **For all:** a proof of $\forall x\,(\varphi(x))$ is a construction that transforms the objects a over which x ranges into proofs of $\varphi(a)$.
- **There exists:** a proof of $\exists\,x\,(\varphi(x))$ is given by a pair consisting of an object a and a proof of $\varphi(a)$.

Constructive interpretation of logic

- **Implication:** a proof of $\varphi \to \psi$ is a construction that transforms proofs of φ into proofs of ψ.
- **Negation:** a proof of $\neg\varphi$ is a construction that from any (hypothetical) proof of φ produces a contradiction (= proof of falsity $\perp$)
- **Disjunction:** a proof of $\varphi \vee \psi$ is an object that manifestly is either a proof of φ, or a proof of ψ.
- **For all:** a proof of $\forall x\,(\varphi(x))$ is a construction that transforms the objects a over which x ranges into proofs of $\varphi(a)$.
- **There exists:** a proof of $\exists\, x\,(\varphi(x))$ is given by a pair consisting of an object a and a proof of $\varphi(a)$.

The *Law of Excluded Middle* (LEM) $\boxed{\forall p\,(p \vee \neg p)}$ is a classical tautology (has truth-value `true`), but is rejected by constructivists.

Example of a non-constructive proof

Theorem. There exist two irrational numbers a and b such that b^a is rational.

Example of a non-constructive proof

Theorem. There exist two irrational numbers a and b such that b^a is rational.

Proof. Either $\sqrt{2}^{\sqrt{2}}$ is rational, or it is not (LEM!).

Example of a non-constructive proof

Theorem. There exist two irrational numbers a and b such that b^a is rational.

Proof. Either $\sqrt{2}^{\sqrt{2}}$ is rational, or it is not (LEM!).

If it is, we can take $a = b = \sqrt{2}$, since $\sqrt{2}$ is irrational by a well-known theorem attributed to Euclid.

Example of a non-constructive proof

Theorem. There exist two irrational numbers a and b such that b^a is rational.

Proof. Either $\sqrt{2}^{\sqrt{2}}$ is rational, or it is not (LEM!).

If it is, we can take $a = b = \sqrt{2}$, since $\sqrt{2}$ is irrational by a well-known theorem attributed to Euclid.

If it is not, we can take $a = \sqrt{2}$ and $b = \sqrt{2}^{\sqrt{2}}$, since then $b^a = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\cdot\sqrt{2}} = \sqrt{2}^2 = 2$.

QED

Example of a constructive proof

Theorem. There exist two irrational numbers a and b such that b^a is rational.

Proof. $\sqrt{2}$ is irrational by a well-known constructive proof attributed to Euclid.

$2 \log_2 3$ is irrational, by an easy constructive proof (exercise).

Example of a constructive proof

Theorem. There exist two irrational numbers a and b such that b^a is rational.

Proof. $\sqrt{2}$ is irrational by a well-known constructive proof attributed to Euclid.

$2\log_2 3$ is irrational, by an easy constructive proof (exercise).

So we can take $a = 2\log_2 3$ and $b = \sqrt{2}$, for which we have that $b^a = (\sqrt{2})^{2\log_2 3} = (\sqrt{2}^2)^{\log_2 3} = 2^{\log_2 3} = 3$ is rational.

QED

Curry-Howard correspondence

<u>Logic</u>	$\leftrightarrow$	<u>Type system</u>
propositions ϕ	$\leftrightarrow$	types τ
proofs p	$\leftrightarrow$	expressions M
'p is a proof of ϕ'	$\leftrightarrow$	'M is an expression of type τ'
simplification of proofs	$\leftrightarrow$	reduction of expressions

E.g.

2IPC	$\leftrightarrow$	PLC

Second-order intuitionistic propositional calculus (2IPC)

2IPC propositions: $\boxed{\phi ::= p \mid \phi \to \phi \mid \forall p\,(\phi)}$ where p ranges over an infinite set of propositional variables.

2IPC sequents: $\boxed{\Phi \vdash \phi}$ where Φ is a finite multiset (= unordered list) of 2IPC propositions and ϕ is a 2IPC proposition.

Second-order intuitionistic propositional calculus (2IPC)

2IPC propositions: $\boxed{\phi ::= p \mid \phi \to \phi \mid \forall p\,(\phi)}$ where p ranges over an infinite set of propositional variables.

2IPC sequents: $\boxed{\Phi \vdash \phi}$ where Φ is a finite multiset (= unordered list) of 2IPC propositions and ϕ is a 2IPC proposition.

$\Phi \vdash \phi$ is *provable* if it is in the set of sequents inductively generated by:

$$(\mathbf{Id})\ \Phi \vdash \phi \quad \text{if } \phi \in \Phi$$

$$(\to\mathbf{I})\ \frac{\Phi, \phi \vdash \phi'}{\Phi \vdash \phi \to \phi'} \qquad (\to\mathbf{E})\ \frac{\Phi \vdash \phi \to \phi' \qquad \Phi \vdash \phi}{\Gamma \vdash \phi'}$$

$$(\forall\mathbf{I})\ \frac{\Phi \vdash \phi}{\Phi \vdash \forall p\,(\phi)} \ \text{if } p \notin fv(\Phi) \qquad (\forall\mathbf{E})\ \frac{\Phi \vdash \forall p\,(\phi)}{\Phi \vdash \phi[\phi'/p]}$$

Logical operations definable in 2IPC

- *Truth* $\top \triangleq \forall p\,(p \to p)$
- *Falsity* $\bot \triangleq \forall p\,(p)$

Logical operations definable in 2IPC

- *Truth* $\top \triangleq \forall p\,(p \to p)$
- *Falsity* $\bot \triangleq \forall p\,(p)$
- *Conjunction* $\phi \wedge \psi \triangleq \forall p\,((\phi \to \psi \to p) \to p)$
  (where $p \notin fv(\phi, \psi)$)

Logical operations definable in 2IPC

- *Truth* $\top \triangleq \forall p\,(p \to p)$
- *Falsity* $\bot \triangleq \forall p\,(p)$
- *Conjunction* $\phi \wedge \psi \triangleq \forall p\,((\phi \to \psi \to p) \to p)$ (where $p \notin fv(\phi, \psi)$)
- *Disjunction* $\phi \vee \psi \triangleq \forall p\,((\phi \to p) \to (\psi \to p) \to p)$ (where $p \notin fv(\phi, \psi)$)

Logical operations definable in 2IPC

- *Truth* $\top \triangleq \forall p\,(p \to p)$
- *Falsity* $\bot \triangleq \forall p\,(p)$
- *Conjunction* $\phi \wedge \psi \triangleq \forall p\,((\phi \to \psi \to p) \to p)$ (where $p \notin fv(\phi, \psi)$)
- *Disjunction* $\phi \vee \psi \triangleq \forall p\,((\phi \to p) \to (\psi \to p) \to p)$ (where $p \notin fv(\phi, \psi)$)
- *Negation* $\neg\phi \triangleq \phi \to \bot$
- *Bi-implication* $\phi \leftrightarrow \psi \triangleq (\phi \to \psi) \wedge (\psi \to \phi)$

Logical operations definable in 2IPC

- *Truth* $\top \triangleq \forall p\,(p \to p)$
- *Falsity* $\bot \triangleq \forall p\,(p)$
- *Conjunction* $\phi \wedge \psi \triangleq \forall p\,((\phi \to \psi \to p) \to p)$ (where $p \notin \mathit{fv}(\phi, \psi)$)
- *Disjunction* $\phi \vee \psi \triangleq \forall p\,((\phi \to p) \to (\psi \to p) \to p)$ (where $p \notin \mathit{fv}(\phi, \psi)$)
- *Negation* $\neg\phi \triangleq \phi \to \bot$
- *Bi-implication* $\phi \leftrightarrow \psi \triangleq (\phi \to \psi) \wedge (\psi \to \phi)$
- *Existential quantification* $\exists\, p\,(\phi) \triangleq \forall q\,(\forall p\,(\phi \to q) \to q)$ (where $q \notin \mathit{fv}(\phi, p)$)

A 2IPC proof

Writing $p \wedge q$ as an abbreviation for $\forall r\,((p \to q \to r) \to r)$, the sequent

$$\{\} \vdash \forall p\,(\forall q\,((p \wedge q) \to p))$$

is provable in 2IPC:

A 2IPC proof

Writing $p \wedge q$ as an abbreviation for $\forall r\,((p \to q \to r) \to r)$, the sequent

$$\{\} \vdash \forall p\,(\forall q\,((p \wedge q) \to p))$$

is provable in 2IPC:

$$
(\forall\mathbf{I})\,\dfrac{(\forall\mathbf{I})\,\dfrac{(\to\mathbf{I})\,\dfrac{(\to\mathbf{E})\,\dfrac{(\to\mathbf{I})\,\dfrac{(\to\mathbf{I})\,\dfrac{(\mathbf{Id})\,\dfrac{}{\{p \wedge q, p, q\} \vdash p}}{\{p \wedge q, p\} \vdash q \to p}}{\{p \wedge q\} \vdash p \to q \to p} \qquad (\forall\mathbf{E})\,\dfrac{(\mathbf{Id})\,\dfrac{}{\{p \wedge q\} \vdash \forall r\,((p \to q \to r) \to r)}}{\{p \wedge q\} \vdash (p \to q \to q) \to q}}{\{p \wedge q\} \vdash p}}{\{\} \vdash (p \wedge q) \to p}}{\{\} \vdash \forall q\,((p \wedge q) \to p)}}{\{\} \vdash \forall p\,(\forall q\,((p \wedge q) \to p))}
$$

Curry-Howard correspondence

2IPC		PLC
<u>Logic</u>	$\leftrightarrow$	<u>Type system</u>

Curry-Howard correspondence

2IPC		PLC
Logic	$\leftrightarrow$	Type system
propositions ϕ	$\leftrightarrow$	types τ

Curry-Howard correspondence

2IPC		PLC
Logic	$\leftrightarrow$	Type system
propositions ϕ	$\leftrightarrow$	types τ
proofs p	$\leftrightarrow$	expressions M
'p is a proof of ϕ'	$\leftrightarrow$	'M is an expression of type τ'

Mapping 2IPC proofs to PLC expressions

$$(\mathbf{Id})\ \Phi, \phi \vdash \phi \quad \mapsto \quad (\mathbf{id})\ \overline{x} : \Phi, x : \phi \vdash x : \phi$$

$$(\to\mathbf{I})\ \frac{\Phi, \phi \vdash \phi'}{\Phi \vdash \phi \to \phi'} \quad \mapsto \quad (\mathbf{fn})\ \frac{\overline{x} : \Phi, x : \phi \vdash M : \phi'}{\overline{x} : \Phi \vdash \lambda x : \phi\,(M) : \phi \to \phi'}$$

$$(\to\mathbf{E})\ \frac{\begin{array}{c}\Phi \vdash \phi \to \phi' \\ \Phi \vdash \phi\end{array}}{\Phi \vdash \phi'} \quad \mapsto \quad (\mathbf{app})\ \frac{\begin{array}{c}\overline{x} : \Phi \vdash M_1 : \phi \to \phi' \\ \overline{x} : \Phi \vdash M_2 : \phi\end{array}}{\overline{x} : \Phi \vdash M_1\, M_2 : \phi'}$$

$$(\forall\mathbf{I})\ \frac{\Phi \vdash \phi}{\Phi \vdash \forall p\,(\phi)} \quad \mapsto \quad (\mathbf{gen})\ \frac{\overline{x} : \Phi \vdash M : \phi}{\overline{x} : \Phi \vdash \Lambda p\,(M) : \forall p\,(\phi)}$$

$$(\forall\mathbf{E})\ \frac{\Phi \vdash \forall p\,(\phi)}{\Phi \vdash \phi[\phi'/p]} \quad \mapsto \quad (\mathbf{spec})\ \frac{\overline{x} : \Phi \vdash M : \forall p\,(\phi)}{\overline{x} : \Phi \vdash M\,\phi' : \phi[\phi'/p]}$$

The proof of the 2IPC sequent

$$\{\} \vdash \forall p\,(\forall q\,((p \wedge q) \to p))$$

given before is transformed by the mapping of 2IPC proofs to PLC expressions to

$$\{\} \vdash \Lambda p, q\,(\lambda z : p \wedge q\,(z\,p\,(\lambda x : p, y : q\,(x))))$$
$$: \forall p\,(\forall q\,((p \wedge q) \to p))$$

with typing derivation:

$$
\text{(gen)}\,\dfrac{\text{(gen)}\,\dfrac{\text{(fn)}\,\dfrac{\text{(app)}\,\dfrac{\text{(fn)}\,\dfrac{\text{(fn)}\,\dfrac{\text{(id)}\,\dfrac{}{\{z : p \wedge q, x : p, y : q\} \vdash x : p}}{\{z : p \wedge q, x : p\} \vdash \lambda y : q\,(x) : q \to p}}{\{z : p \wedge q\} \vdash \lambda x : p, y : q\,(x) : p \to q \to p} \qquad \text{(spec)}\,\dfrac{\text{(id)}\,\dfrac{}{\{z : p \wedge q\} \vdash z : \forall r\,((p \to q \to r) \to r)}}{\{z : p \wedge q\} \vdash z\,p : (p \to q \to p) \to p}}{\{z : p \wedge q\} \vdash z\,p\,(\lambda x : p, y : q\,(x)) : p}}{\{\} \vdash \lambda z : p \wedge q\,(z\,p\,(\lambda x : p, y : q\,(x))) : (p \wedge q) \to p}}{\{\} \vdash \Lambda q\,(\lambda z : p \wedge q\,(z\,p\,(\lambda x : p, y : q\,(x)))) : \forall q\,((p \wedge q) \to p)}}{\{\} \vdash \Lambda p, q\,(\lambda z : p \wedge q\,(z\,p\,(\lambda x : p, y : q\,(x)))) : \forall p, q\,((p \wedge q) \to p)}
$$

Curry-Howard correspondence

Logic	$\leftrightarrow$	Type system
propositions ϕ	$\leftrightarrow$	types τ
proofs p	$\leftrightarrow$	expressions M
'p is a proof of ϕ'	$\leftrightarrow$	'M is an expression of type τ'
simplification of proofs	$\leftrightarrow$	reduction of expressions

Proof simplification $\leftrightarrow$ Expression reduction

$$
(\to\mathbf{E})\;\dfrac{(\to\mathbf{I})\;\dfrac{\dfrac{\vdots}{\Phi, \phi \vdash \psi}}{\Phi \vdash \phi \to \psi} \qquad \dfrac{\vdots}{\Phi \vdash \phi}}{\Phi \vdash \psi}
\quad \mapsto \quad
\dfrac{\dfrac{\dfrac{\vdots}{\overline{x} : \Phi, x : \phi \vdash M : \psi}}{\overline{x} : \Phi \vdash \lambda x : \phi\,(M) : \phi \to \psi} \qquad \dfrac{\vdots}{\overline{x} : \Phi \vdash N : \phi}}{\overline{x} : \Phi \vdash (\lambda x : \phi\,(M))\,N : \psi}
$$

Proof simplification $\leftrightarrow$ Expression reduction

$$
(\to\mathrm{I})\,\dfrac{\dfrac{\vdots}{\Phi, \phi \vdash \psi}}{\Phi \vdash \phi \to \psi}
\qquad
\dfrac{\vdots}{\Phi \vdash \phi}
\;(\to\mathrm{E})\;
\Rightarrow \dfrac{\quad}{\Phi \vdash \psi}
$$

$$
\dfrac{\dfrac{\dfrac{\vdots}{\overline{x} : \Phi, x : \phi \vdash M : \psi}}{\overline{x} : \Phi \vdash \lambda x : \phi\,(M) : \phi \to \psi} \qquad \dfrac{\vdots}{\overline{x} : \Phi \vdash N : \phi}}{\overline{x} : \Phi \vdash (\lambda x : \phi\,(M))\,N : \psi}
$$

$$
\Big\downarrow \ \text{beta-reduce expression}
$$

$$
\dfrac{\dfrac{\vdots}{\overline{x} : \Phi, x : \phi \vdash M : \psi} \qquad \dfrac{\vdots}{\overline{x} : \Phi \vdash N : \phi}}{\overline{x} : \Phi \vdash M[N/x] : \psi}\;(\mathbf{subst})
$$

The rule $(\mathbf{subst})$ for PLC is *admissible*: if its hypotheses are valid PLC typing judgements, then so is its conclusion.

Proof simplification $\leftrightarrow$ Expression reduction

$$
(\to\mathbf{I})\;\dfrac{\dfrac{\dfrac{\vdots}{\Phi, \phi \vdash \psi}}{\Phi \vdash \phi \to \psi} \qquad \dfrac{\vdots}{\Phi \vdash \phi}}{\Phi \vdash \psi}\;(\to\mathbf{E})
\quad\mapsto\quad
\dfrac{\dfrac{\dfrac{\vdots}{\overline{x} : \Phi, x : \phi \vdash M : \psi}}{\overline{x} : \Phi \vdash \lambda x : \phi\,(M) : \phi \to \psi} \qquad \dfrac{\vdots}{\overline{x} : \Phi \vdash N : \phi}}{\overline{x} : \Phi \vdash (\lambda x : \phi\,(M))\,N : \psi}
$$

$$\Big\downarrow \;\text{beta-reduce expression}$$

$$
(\mathbf{cut})\;\dfrac{\dfrac{\vdots}{\Phi, \phi \vdash \psi} \qquad \dfrac{\vdots}{\Phi \vdash \phi}}{\Phi \vdash \psi}
\quad\longleftarrow\!\!\!\!\!\!\!\mid\quad
\dfrac{\dfrac{\vdots}{\overline{x} : \Phi, x : \phi \vdash M : \psi} \qquad \dfrac{\vdots}{\overline{x} : \Phi \vdash N : \phi}}{\overline{x} : \Phi \vdash M[N/x] : \psi}\;(\mathbf{subst})
$$

The rule $(\mathbf{subst})$ for PLC is *admissible*: if its hypotheses are valid PLC typing judgements, then so is its conclusion.

Proof simplification $\leftrightarrow$ Expression reduction

$$\dfrac{(\to\mathbf{I})\,\dfrac{\dfrac{\vdots}{\Phi, \phi \vdash \psi}}{\Phi \vdash \phi \to \psi} \qquad \dfrac{\vdots}{\Phi \vdash \phi}}{\Phi \vdash \psi}\,(\to\mathbf{E}) \quad \mapsto \quad \dfrac{\dfrac{\dfrac{\vdots}{\overline{x} : \Phi, x : \phi \vdash M : \psi}}{\overline{x} : \Phi \vdash \lambda x : \phi\,(M) : \phi \to \psi} \qquad \dfrac{\vdots}{\overline{x} : \Phi \vdash N : \phi}}{\overline{x} : \Phi \vdash (\lambda x : \phi\,(M))\,N : \psi}$$

simplify proof $\downarrow$ $\qquad\qquad$ $\downarrow$ beta-reduce expression

$$(\mathbf{cut})\ \dfrac{\dfrac{\vdots}{\Phi, \phi \vdash \psi} \qquad \dfrac{\vdots}{\Phi \vdash \phi}}{\Phi \vdash \psi} \quad \longleftarrow \quad \dfrac{\dfrac{\vdots}{\overline{x} : \Phi, x : \phi \vdash M : \psi} \qquad \dfrac{\vdots}{\overline{x} : \Phi \vdash N : \phi}}{\overline{x} : \Phi \vdash M[N/x] : \psi}\ (\mathbf{subst})$$

The rule $(\mathbf{subst})$ for PLC is *admissible*: if its hypotheses are valid PLC typing judgements, then so is its conclusion.

Hence, the rule $(\mathbf{cut})$ is admissible for 2IPC.

Type-inference versus proof search

Type-inference: given $\boldsymbol{\Gamma}$ and $\boldsymbol{M}$, is there a type $\boldsymbol{\tau}$ such that $\boldsymbol{\Gamma} \vdash \boldsymbol{M} : \boldsymbol{\tau}$?

(For PLC/2IPC this is decidable.)

Type-inference versus proof search

Type-inference: given Γ and M, is there a type τ such that $\Gamma \vdash M : \tau$?

(For PLC/2IPC this is decidable.)

Proof-search: given Γ and ϕ, is there a proof term M such that $\Gamma \vdash M : \phi$?

(For PLC/2IPC this is undecidable.)