A solution to Curry and Hindley's problem on combinatory strong reduction

Pierluigi Minari

Department of Philosophy, University of Florence
minari@unifi.it

WORKSHOP ON RECENT TRENDS IN PROOF THEORY

(University of Bern, July 9-11, 2008)

Outline

1. The problem

Outline

Outline

Outline

1. The problem
   - Combinatory strong reduction
   - Curry's *indirect* confluence proof
   - Statement of the problem
2. Analytic proof systems for combinatory logic and λ-calculus
3. Solution to the problem
4. Proving transitivity elimination for $\mathbf{G_{ext}}[\mathbb{X}]$ systems

Combinatory strong reduction

Primitive combinators: I , K , S

$$\frac{}{t \succ t}\,\rho \qquad \frac{}{\mathsf{I}t \succ t}\,\mathsf{I} \qquad \frac{}{\mathsf{K}ts \succ t}\,\mathsf{K} \qquad \frac{}{\mathsf{S}tsr \succ tr(sr)}\,\mathsf{S}$$

$$\frac{t \succ s}{rt \succ rs}\,\mu \qquad \frac{t \succ s}{tr \succ sr}\,\nu \qquad \frac{t \succ r \quad r \succ s}{t \succ s}\,\tau$$

$$\frac{t \succ s}{\lambda^* x.t \succ \lambda^* x.s}\,\xi$$

Combinatory strong reduction

Primitive combinators: I , K , S

$$\frac{}{t \succ t}\,\rho \qquad \frac{}{\mathsf{I}t \succ t}\,\mathsf{I} \qquad \frac{}{\mathsf{K}ts \succ t}\,\mathsf{K} \qquad \frac{}{\mathsf{S}tsr \succ tr(sr)}\,\mathsf{S}$$

$$\frac{t \succ s}{rt \succ rs}\,\mu \qquad \frac{t \succ s}{tr \succ sr}\,\nu \qquad \frac{t \succ r \quad r \succ s}{t \succ s}\,\tau$$

$$\frac{t \succ s}{\lambda^* x.t \succ \lambda^* x.s}\,\xi$$

Combinatory strong reduction

Primitive combinators: I, K, S

$$\frac{}{t \succ t}\ \rho \qquad \frac{}{\mathsf{I}t \succ t}\ \mathsf{I} \qquad \frac{}{\mathsf{K}ts \succ t}\ \mathsf{K} \qquad \frac{}{\mathsf{S}tsr \succ tr(sr)}\ \mathsf{S}$$

$$\frac{t \succ s}{rt \succ rs}\ \mu \qquad \frac{t \succ s}{tr \succ sr}\ \nu \qquad \frac{t \succ r \quad r \succ s}{t \succ s}\ \tau$$

$$\frac{t \succ s}{\lambda^* x.t \succ \lambda^* x.s}\ \xi$$

Abstraction is defined according to the *strong* algorithm.

Abstraction

Abstraction

(a) $\lambda^* x.x := \mathsf{I}$

Abstraction

(a) $\lambda^* x.x := \mathsf{I}$

(b) $\lambda^* x.t := \mathsf{K}t$, if $x \notin V(t)$

Abstraction

(a) $\lambda^* x.x := \mathsf{I}$

(b) $\lambda^* x.t := \mathsf{K}t$, if $x \notin V(t)$

(c) $\lambda^* x.sx := s$, if $x \notin V(s)$

Abstraction

(a) $\lambda^* x.x :=$ I

(b) $\lambda^* x.t :=$ Kt, if $x \notin V(t)$

(c) $\lambda^* x.sx := s$, if $x \notin V(s)$

(d) $\lambda^* x.ts :=$ S$(\lambda^* x.t)(\lambda^* x.s)$, if (b) and (c) do not apply

Abstraction

(a) $\lambda^* x.x := \mathsf{I}$

(b) $\lambda^* x.t := \mathsf{K}t$, if $x \notin V(t)$

(c) $\lambda^* x.sx := s$, if $x \notin V(s)$

(d) $\lambda^* x.ts := \mathsf{S}(\lambda^* x.t)(\lambda^* x.s)$, if (b) and (c) do not apply

Remark

The combinator I is taken as primitive just to avoid having a *trivial* example of a term in strong normal form which is not strongly irreducible.
Indeed, notice that SK $\succ$ KI. So, by defining I $:=$ SKK, we would have:

$$\mathsf{I} \equiv \mathsf{SKK} \succ \mathsf{KIK} \succ \mathsf{K(KIK)K} \succ \dots$$

Notwithstanding its bad reputation of being quite messy, much is known about the metatheory of strong reduction

Notwithstanding its bad reputation of being quite messy, much is known about the metatheory of strong reduction

1. $\succ\!-$ is Church-Rosser [Curry, 1958]

Notwithstanding its bad reputation of being quite messy, much is known about the metatheory of strong reduction

1. $\succ\!-$ is Church-Rosser [Curry, 1958]
2. strongly irreducible terms are in strong normal form [Curry 1958, Hindley & Lercher 1970]

Notwithstanding its bad reputation of being quite messy, much is known about the metatheory of strong reduction

1. $\succ\!-$ is Church-Rosser [Curry, 1958]
2. strongly irreducible terms are in strong normal form [Curry 1958, Hindley & Lercher 1970]
3. . . . and conversely [Lercher 1967]

Notwithstanding its bad reputation of being quite messy, much is known about the metatheory of strong reduction

1. $\succ\!\!-$ is Church-Rosser [Curry, 1958]
2. strongly irreducible terms are in strong normal form [Curry 1958, Hindley & Lercher 1970]
3. . . . and conversely [Lercher 1967]
4. there is a recursive set of axiom schemas axiomatizing $\succ\!\!-$ over weak reduction $\twoheadrightarrow_{w}$ [Hindley 1967, Lercher 1967]

Notwithstanding its bad reputation of being quite messy, much is known about the metatheory of strong reduction

1. $\succ\!\!-$ is Church-Rosser [Curry, 1958]
2. strongly irreducible terms are in strong normal form [Curry 1958, Hindley & Lercher 1970]
3. . . . and conversely [Lercher 1967]
4. there is a recursive set of axiom schemas axiomatizing $\succ\!\!-$ over weak reduction $\twoheadrightarrow_w$ [Hindley 1967, Lercher 1967]

We shall be concerned with point 1, or better with the proof of CR($\succ$).

Curry's proof of the confluence of strong reduction

Curry's proof of the confluence of strong reduction

$(\ )_\lambda : \mathbf{T}_{\{\mathsf{I},\mathsf{K},\mathsf{S}\}} \longrightarrow \Lambda$ and $(\ )_H : \Lambda \longrightarrow \mathbf{T}_{\{\mathsf{I},\mathsf{K},\mathsf{S}\}}$

Standard translations between combinatory terms and λ-terms.

Curry's proof of the confluence of strong reduction

$(\ )_\lambda : \mathbf{T}_{\{\mathsf{I},\mathsf{K},\mathsf{S}\}} \longrightarrow \Lambda$ and $(\ )_H : \Lambda \longrightarrow \mathbf{T}_{\{\mathsf{I},\mathsf{K},\mathsf{S}\}}$

Standard translations between combinatory terms and λ-terms.

These satisfy:

Curry's proof of the confluence of strong reduction

$(\ )_\lambda : \mathbf{T}_{\{\mathsf{I,K,S}\}} \longrightarrow \Lambda$ and $(\ )_H : \Lambda \longrightarrow \mathbf{T}_{\{\mathsf{I,K,S}\}}$

Standard translations between combinatory terms and λ-terms.

These satisfy:

(P1) for $t \in \mathbf{T}_{\{\mathsf{I,K,S}\}}$: $(t_\lambda)_H \equiv t\,,$

Curry's proof of the confluence of strong reduction

$(\ )_\lambda : \mathbf{T}_{\{\mathrm{I,K,S}\}} \longrightarrow \Lambda$ and $(\ )_H : \Lambda \longrightarrow \mathbf{T}_{\{\mathrm{I,K,S}\}}$

Standard translations between combinatory terms and λ-terms.

These satisfy:

(P1) for $t \in \mathbf{T}_{\{\mathrm{I,K,S}\}}$: $(t_\lambda)_H \equiv t$,

(P2) for $t, s \in \Lambda$: $t \twoheadrightarrow_{\beta\eta} s \Rightarrow t_H \succ s_H$,

Curry's proof of the confluence of strong reduction

$(\ )_\lambda : \mathbf{T}_{\{\mathsf{I,K,S}\}} \longrightarrow \Lambda$ and $(\ )_H : \Lambda \longrightarrow \mathbf{T}_{\{\mathsf{I,K,S}\}}$

Standard translations between combinatory terms and λ-terms.

These satisfy:

(P1) for $t \in \mathbf{T}_{\{\mathsf{I,K,S}\}}$: $(t_\lambda)_H \equiv t$,

(P2) for $t, s \in \Lambda$: $t \twoheadrightarrow_{\beta\eta} s \Rightarrow t_H \succ s_H$,

(P3) for $t, s \in \mathbf{T}_{\{\mathsf{I,K,S}\}}$: $t =_{c\beta\eta} s \Rightarrow t_\lambda =_{\beta\eta} s_\lambda$.

Curry's proof of the confluence of strong reduction

$(\ )_\lambda : \mathbf{T}_{\{\mathsf{I,K,S}\}} \longrightarrow \Lambda$ and $(\ )_H : \Lambda \longrightarrow \mathbf{T}_{\{\mathsf{I,K,S}\}}$

Standard translations between combinatory terms and λ-terms.

These satisfy:

(P1) for $t \in \mathbf{T}_{\{\mathsf{I,K,S}\}}$: $(t_\lambda)_H \equiv t$,

(P2) for $t, s \in \Lambda$: $t \twoheadrightarrow_{\beta\eta} s \Rightarrow t_H \succ s_H$,

(P3) for $t, s \in \mathbf{T}_{\{\mathsf{I,K,S}\}}$: $t =_{c\beta\eta} s \Rightarrow t_\lambda =_{\beta\eta} s_\lambda$.

Then:

$$
\begin{array}{lcll}
t =_{c\beta\eta} s & \Rightarrow & t_\lambda =_{\beta\eta} s_\lambda & \text{by (P3)} \\
 & \Rightarrow & \exists r \in \Lambda :\ t_\lambda \twoheadrightarrow_{\beta\eta} r \ {}_{\beta\eta}\twoheadleftarrow s_\lambda & \text{by CR}(\twoheadrightarrow_{\beta\eta}) \\
 & \Rightarrow & t \succ r_H \prec s & \text{by (P2) and (P1)}
\end{array}
$$

Curry's statement of the problem

H. B. Curry and R. Feys, Combinatory Logic, Vol. I, 1958

List of "Unsolved problems" in § 6 F.5

Curry’s statement of the problem

H. B. Curry and R. Feys, Combinatory Logic, Vol. I, 1958

List of “Unsolved problems” in § 6 F.5

Curry's statement of the problem

H. B. Curry and R. Feys, Combinatory Logic, Vol. I, 1958

List of "Unsolved problems" in § 6 F.5

"c. Is it possible to prove the Church-Rosser property directly for strong reduction, without having recourse to transformations between that theory and the theory of λ-conversion? . . . "

Curry's statement of the problem

H. B. Curry and R. Feys, Combinatory Logic, Vol. I, 1958

List of "Unsolved problems" in § 6 F.5

> *"c. Is it possible to prove the Church-Rosser property directly for strong reduction, without having recourse to transformations between that theory and the theory of λ-conversion? . . ."*

Remark

A solution was advanced by K. Loewen in 1968.
His proof, however, seems to contain an error — as pointed out in Hindley's MR review (1970).

Hindley's statement of the problem

Problem #1 — TLCA List of Open Problems, http://tlca.di.unito.it/opltlca/

Submitted by Roger Hindley *Date:* Known since 1958!

Statement. Is there a direct proof of the confluence of $\beta\eta$-strong reduction?

Problem Origin. First posed by Haskell Curry and Roger Hindley.

Hindley's statement of the problem

Problem #1 — TLCA List of Open Problems, http://tlca.di.unito.it/opltlca/

Submitted by Roger Hindley *Date:* Known since 1958!

Statement. Is there a direct proof of the confluence of $\beta\eta$-strong reduction?

Problem Origin. First posed by Haskell Curry and Roger Hindley.

Hindley's statement of the problem

Problem #1 — TLCA List of Open Problems, http://tlca.di.unito.it/opltlca/

Submitted by Roger Hindley *Date:* Known since 1958!

Statement. Is there a direct proof of the confluence of $\beta\eta$-strong reduction?

Problem Origin. First posed by Haskell Curry and Roger Hindley.

The $\beta\eta$-strong reduction is the combinatory analogue of $\beta\eta$-reduction in λ-calculus. It is confluent. Its only known confluence-proof is very easy, [Curry and Feys, 1958, 6F, p. 221 Theorem 3], but it depends on the having already proved the confluence of $\lambda\beta\eta$-reduction. Thus the theory of combinators is not self-contained at present. **Is there a confluence proof independent of λ-calculus?**

1. The problem
2. Analytic proof systems for combinatory logic and λ-calculus
   - Synthetic vs analytic equational proof systems
   - G-systems
   - Main results
3. Solution to the problem
4. Proving transitivity elimination for $\mathbf{G}_{\mathbf{ext}}[\mathbb{X}]$ systems

Motivations

- Standard presentations of equational proof systems:

Motivations

- Standard presentations of equational proof systems:
  - *specific axioms* (a set of equation schemas)

Motivations

- Standard presentations of equational proof systems:
  - *specific axioms* (a set of equation schemas)
  - the usual inference rules for equality (*reflexivity*, *symmetry*, *transitivity* and *congruence*)

Motivations

- Standard presentations of equational proof systems:
  - *specific axioms* (a set of equation schemas)
  - the usual inference rules for equality (*reflexivity*, *symmetry*, *transitivity* and *congruence*)

Motivations

- Standard presentations of equational proof systems:
  - *specific axioms* (a set of equation schemas)
  - the usual inference rules for equality (*reflexivity*, *symmetry*, *transitivity* and *congruence*)

- The **transitivity rule**

$$\frac{t = \mathbf{r} \quad \mathbf{r} = s}{t = s}$$

Motivations

- Standard presentations of equational proof systems:
  - *specific axioms* (a set of equation schemas)
  - the usual inference rules for equality (*reflexivity*, *symmetry*, *transitivity* and *congruence*)
- The **transitivity rule**

$$\frac{t = \mathbf{r} \quad \mathbf{r} = s}{t = s}$$

Motivations

- Standard presentations of equational proof systems:
  - *specific axioms* (a set of equation schemas)
  - the usual inference rules for equality (*reflexivity*, *symmetry*, *transitivity* and *congruence*)

- The **transitivity rule**

$$\frac{t = \boldsymbol{r} \quad \boldsymbol{r} = s}{t = s}$$

  (which cannot be dispensed with, except that in trivial cases) **has an inherently *synthetic* character** in combining derivations, like *modus ponens* in Hilbert-style proof systems

- *Naive* proof-theoretic arguments are usually impossible (e.g.: syntactic consistency proofs by induction on the length of derivations)

- *Naive* proof-theoretic arguments are usually impossible (e.g.: syntactic consistency proofs by induction on the length of derivations)

- *Naive* proof-theoretic arguments are usually impossible (e.g.: syntactic consistency proofs by induction on the length of derivations)

- No kind of *“subterm property”*

- *Naive* proof-theoretic arguments are usually impossible (e.g.: syntactic consistency proofs by induction on the length of derivations)
- No kind of *“subterm property”*

- *Naive* proof-theoretic arguments are usually impossible (e.g.: syntactic consistency proofs by induction on the length of derivations)

- No kind of *“subterm property”*

- In general, derivations lack any significant mathematical structure

- *Naive* proof-theoretic arguments are usually impossible (e.g.: syntactic consistency proofs by induction on the length of derivations)
- No kind of *“subterm property”*
- In general, derivations lack any significant mathematical structure

- *Naive* proof-theoretic arguments are usually impossible (e.g.: syntactic consistency proofs by induction on the length of derivations)

- No kind of *"subterm property"*

- In general, derivations lack any significant mathematical structure

- As a consequence, 'synthetic' equational calculi do not lend themselves *directly* to proof-theoretical analysis

Question

Are there significant cases in which it is both *possible* and *useful*

to turn a ‘synthetic’ equational proof system into

an **equivalent** ‘analytic’ proof system,

where the transitivity rule is provably redundant?

Recent work

Recent work

Recent work

- **Combinatory logic**: CL (& generalizations)

  P. M., *Analytic combinatory calculi and the elimination of transitivity*, Arch. Math. Logic 43 (2004), 159-191.

Recent work

- **Combinatory logic**: CL (& generalizations)

  P. M., *Analytic combinatory calculi and the elimination of transitivity*, Arch. Math. Logic 43 (2004), 159-191.

Recent work

- **Combinatory logic**: CL (& generalizations)

  P. M., *Analytic combinatory calculi and the elimination of transitivity*, Arch. Math. Logic 43 (2004), 159-191.

- **Lambda-Calculus**: $\boldsymbol{\lambda\beta}$, $\boldsymbol{\lambda\beta\eta}$

  P. M., *Analytic proof systems for λ-calculus: the elimination of transitivity, and why it matters*, Arch. Math. Logic 46 (2007), 385-424.

Recent work

- **Combinatory logic**: CL (& generalizations)

  P. M., *Analytic combinatory calculi and the elimination of transitivity*, Arch. Math. Logic 43 (2004), 159-191.

- **Lambda-Calculus**: $\lambda\beta$, $\lambda\beta\eta$

  P. M., *Analytic proof systems for λ-calculus: the elimination of transitivity, and why it matters*, Arch. Math. Logic 46 (2007), 385-424.

Recent work

- **Combinatory logic**: CL (& generalizations)

  P. M., *Analytic combinatory calculi and the elimination of transitivity*, Arch. Math. Logic 43 (2004), 159-191.

- **Lambda-Calculus**: $\lambda\beta$, $\lambda\beta\eta$

  P. M., *Analytic proof systems for λ-calculus: the elimination of transitivity, and why it matters*, Arch. Math. Logic 46 (2007), 385-424.

- **Extensional Combinatory logic**: $CL_{\mathbf{ext}}$ (& generalizations)

  P. M., *A solution to Curry and Hindley's problem on combinatory strong reduction*, submitted.

Overwiew

Overwiew

$\boxed{\textit{synthetic}\text{ proof-systems}}$

Overwiew

$\boxed{\textit{synthetic}\text{ proof-systems}}$

Overwiew

$\boxed{\textit{synthetic}\text{ proof-systems}}$

$\Downarrow$

$\boxed{\text{equivalent (candidate) }\textit{analytic}\text{ proof-systems (“G-systems”)}}$

Overwiew

synthetic proof-systems

$\Downarrow$

equivalent (candidate) *analytic* proof-systems ("G-systems")

$\Downarrow$

(effective) *transitivity elimination* for G-systems

Overview

synthetic proof-systems

$\Downarrow$

equivalent (candidate) *analytic* proof-systems ("G-systems")

$\Downarrow$

(effective) *transitivity elimination* for G-systems $\Rightarrow$ **consistency**

Overview

synthetic proof-systems

$\Downarrow$

equivalent (candidate) *analytic* proof-systems ("G-systems")

$\Downarrow$

(effective) *transitivity elimination* for G-systems $\Rightarrow$ **consistency**

$\Downarrow$

"*normalizability*" of transitivity-free derivations

Overwiew

synthetic proof-systems

$\Downarrow$

equivalent (candidate) *analytic* proof-systems ("G-systems")

$\Downarrow$

(effective) *transitivity elimination* for G-systems $\Rightarrow$ **consistency**

$\Downarrow$

"*normalizability*" of transitivity-free derivations

$\Downarrow$

applications to combinatory / lambda reductions

Main features of G-systems

Main features of G-systems

- **combinatory axiom schemas / β-conversion schema**
  - turned into pairs of suitable **introduction rules** C β

Main features of G-systems

- **combinatory axiom schemas / β-conversion schema**
  - turned into pairs of suitable **introduction rules** C β

Main features of G-systems

- **combinatory axiom schemas / β-conversion schema**
  ▶ turned into pairs of suitable **introduction rules** C β

- **symmetry rule** ▶ dropped

Main features of G-systems

- **combinatory axiom schemas / β-conversion schema**
  ▶ turned into pairs of suitable **introduction rules** C β

- **symmetry rule** ▶ dropped

Main features of G-systems

- **combinatory axiom schemas / β-conversion schema**
  ▶ turned into pairs of suitable **introduction rules** ⏩ C ⏩ β
- **symmetry rule** ▶ dropped
- **reflexivity (0-premises) rule** ▶ restricted to atomic terms

Main features of G-systems

- **combinatory axiom schemas / β-conversion schema**
  ▶ turned into pairs of suitable **introduction rules** ⏩ C ⏩ β
- **symmetry rule** ▶ dropped
- **reflexivity (0-premises) rule** ▶ restricted to atomic terms

Main features of G-systems

- **combinatory axiom schemas / β-conversion schema**
  - turned into pairs of suitable **introduction rules** C β
- **symmetry rule** ▶ dropped
- **reflexivity (0-premises) rule** ▶ restricted to atomic terms
- **monotony rule(s)**
  - taken in the **parallel** version

$$\frac{t = s \quad p = q}{tp = sq}\, App$$

Main features of G-systems

- **combinatory axiom schemas / β-conversion schema**
  - turned into pairs of suitable **introduction rules** C β
- **symmetry rule**
  - dropped
- **reflexivity (0-premises) rule**
  - restricted to atomic terms
- **monotony rule(s)**
  - taken in the **parallel** version

$$\frac{t = s \quad p = q}{tp = sq}\,App$$

Main features of G-systems

- **combinatory axiom schemas / β-conversion schema**
  - turned into pairs of suitable **introduction rules** C β
- **symmetry rule**
  - dropped
- **reflexivity (0-premises) rule**
  - restricted to atomic terms
- **monotony rule(s)**
  - taken in the **parallel** version

$$\frac{t = s \quad p = q}{tp = sq}\ App$$

- **extensionality rule (if any)**
  - taken in the version

$$\frac{tx = sx}{t = s}\ Ext \quad \{x \notin V(ts)\}$$

G-systems for full combinatory logic: $\mathbf{G}[\mathbb{C}]$ / $\mathbf{G}_{\mathbf{ext}}[\mathbb{C}]$

$\mathbf{G}[\mathbb{C}]$ (corresponding to $\mathbf{CL}$)

G-systems for full combinatory logic: $\mathbf{G}[\mathbb{C}]$ / $\mathbf{G}_{\mathbf{ext}}[\mathbb{C}]$

$\mathbf{G}[\mathbb{C}]$ (corresponding to $\mathbf{CL}$)

G-systems for full combinatory logic: $\mathbf{G}[\mathbb{C}]$ / $\mathbf{G}_{\mathbf{ext}}[\mathbb{C}]$

$\mathbf{G}[\mathbb{C}]$ (corresponding to $\mathbf{CL}$)

- "structural rules":

$$\frac{}{t = t}\,\rho' \quad (t \text{ atomic}) \qquad \frac{t = s \quad p = q}{tp = sq}\,App \qquad \frac{t = r \quad r = s}{t = s}\,\tau$$

G-systems for full combinatory logic: $\mathbf{G}[\mathbb{C}]$ / $\mathbf{G}_{\mathbf{ext}}[\mathbb{C}]$

$\mathbf{G}[\mathbb{C}]$ (corresponding to $\mathbf{CL}$)

- “structural rules”:

$$\frac{}{t = t}\,\rho' \text{ (}t\text{ atomic)} \qquad \frac{t = s \quad p = q}{tp = sq}\,App \qquad \frac{t = r \quad r = s}{t = s}\,\tau$$

- *left* and *right* combinatory introduction rules for $\mathsf{I}, \mathsf{K}, \mathsf{S}$

G-systems for full combinatory logic: $\mathbf{G}[\mathbb{C}]$ / $\mathbf{G}_{\mathbf{ext}}[\mathbb{C}]$

$\mathbf{G}[\mathbb{C}]$ (corresponding to $\mathbf{CL}$)

- "structural rules":

$$\frac{}{t = t}\,\rho' \quad \text{(}t\text{ atomic)} \qquad \frac{t = s \quad p = q}{tp = sq}\,App \qquad \frac{t = r \quad r = s}{t = s}\,\tau$$

- *left* and *right* combinatory introduction rules for $\mathrm{I}, \mathrm{K}, \mathrm{S}$

$\mathbf{G}_{\mathbf{ext}}[\mathbb{C}]$ (corresponding to $\mathbf{CL}_{\mathbf{ext}}$)

G-systems for full combinatory logic: $\mathbf{G}[\mathbb{C}]$ / $\mathbf{G_{ext}}[\mathbb{C}]$

$\mathbf{G}[\mathbb{C}]$ (corresponding to $\mathbf{CL}$)

- "structural rules":

$$\frac{}{t = t}\ \rho' \quad (t \text{ atomic}) \qquad \frac{t = s \quad p = q}{tp = sq}\ App \qquad \frac{t = r \quad r = s}{t = s}\ \tau$$

- *left* and *right* combinatory introduction rules for $\mathsf{I}, \mathsf{K}, \mathsf{S}$

$\mathbf{G_{ext}}[\mathbb{C}]$ (corresponding to $\mathbf{CL_{ext}}$)

- + the extensionality rule $[Ext]$

and for arbitrary combinatory systems $\mathbb{X}$: $\mathbf{G}[\mathbb{X}]$ / $\mathbf{G_{ext}}[\mathbb{X}]$

and for arbitrary combinatory systems $\mathbb{X}$: $\mathbf{G}[\mathbb{X}]$ / $\mathbf{G_{ext}}[\mathbb{X}]$

A **combinatory system** $\mathbb{X}$ is a map, defined on a non-empty set $\mathbf{X} = dom(\mathbb{X})$ of primitive combinators $(\mathsf{F}, \mathsf{G} \ldots)$, which associates to each $\mathsf{F} \in \mathbf{X}$ a pair $\langle k_{\mathsf{F}}, d_{\mathsf{F}} \rangle$ s.t.:

and for arbitrary combinatory systems $\mathbb{X}$: $\mathbf{G}[\mathbb{X}]$ / $\mathbf{G_{ext}}[\mathbb{X}]$

A **combinatory system** $\mathbb{X}$ is a map, defined on a non-empty set $\mathbf{X} = dom(\mathbb{X})$ of primitive combinators $(\mathsf{F}, \mathsf{G} \ldots)$, which associates to each $\mathsf{F} \in \mathbf{X}$ a pair $\langle k_{\mathsf{F}}, d_{\mathsf{F}} \rangle$ s.t.:

- k_{F}, the *index* of F under $\mathbb{X}$, is a non negative integer;

and for arbitrary combinatory systems $\mathbb{X}$: $\mathbf{G}[\mathbb{X}]$ / $\mathbf{G_{ext}}[\mathbb{X}]$

A **combinatory system** $\mathbb{X}$ is a map, defined on a non-empty set $\mathbf{X} = dom(\mathbb{X})$ of primitive combinators ($\mathsf{F}, \mathsf{G} \dots$), which associates to each $\mathsf{F} \in \mathbf{X}$ a pair $\langle k_{\mathsf{F}}, d_{\mathsf{F}} \rangle$ s.t.:

- k_{F}, the *index* of F under $\mathbb{X}$, is a non negative integer;
- d_{F}, the *definition* of F under $\mathbb{X}$, is a term with $V(d_{\mathsf{F}}) \subseteq \{v_1, \dots, v_{k_{\mathsf{F}}}\}$.

and for arbitrary combinatory systems $\mathbb{X}$: $\mathbf{G}[\mathbb{X}]$ / $\mathbf{G_{ext}}[\mathbb{X}]$

A **combinatory system** $\mathbb{X}$ is a map, defined on a non-empty set $\mathbf{X} = dom(\mathbb{X})$ of primitive combinators $(\mathsf{F}, \mathsf{G} \dots)$, which associates to each $\mathsf{F} \in \mathbf{X}$ a pair $\langle k_\mathsf{F}, d_\mathsf{F} \rangle$ s.t.:

- k_F, the *index* of F under $\mathbb{X}$, is a non negative integer;
- d_F, the *definition* of F under $\mathbb{X}$, is a term with $V(d_\mathsf{F}) \subseteq \{v_1, \dots, v_{k_\mathsf{F}}\}$.

Intuitively, for each primitive combinator $\mathsf{F} \in \mathbf{X}$:

$$\mathbb{X} : \mathsf{F} \longmapsto \mathsf{F} t_1 \dots t_{k_\mathsf{F}} = d_\mathsf{F}[v_1/t_1, \dots, v_{k_\mathsf{F}}/t_{k_\mathsf{F}}] \qquad (\mathrm{AX}\,\mathsf{F})_\mathbb{X}$$

and for arbitrary combinatory systems $\mathbb{X}$: $\mathbf{G}[\mathbb{X}]$ / $\mathbf{G_{ext}}[\mathbb{X}]$

A **combinatory system** $\mathbb{X}$ is a map, defined on a non-empty set $\mathbf{X} = dom(\mathbb{X})$ of primitive combinators $(\mathsf{F}, \mathsf{G} \dots)$, which associates to each $\mathsf{F} \in \mathbf{X}$ a pair $\langle k_\mathsf{F}, d_\mathsf{F} \rangle$ s.t.:

- k_F, the *index* of F under $\mathbb{X}$, is a non negative integer;
- d_F, the *definition* of F under $\mathbb{X}$, is a term with $V(d_\mathsf{F}) \subseteq \{v_1, \dots, v_{k_\mathsf{F}}\}$.

Intuitively, for each primitive combinator $\mathsf{F} \in \mathbf{X}$:

$$\mathbb{X} : \mathsf{F} \longmapsto \mathsf{F} t_1 \dots t_{k_\mathsf{F}} = d_\mathsf{F}[v_1/t_1, \dots, v_{k_\mathsf{F}}/t_{k_\mathsf{F}}] \qquad (\mathrm{AX}\,\mathsf{F})_\mathbb{X}$$

$\mathbf{G}[\mathbb{X}]$ / $\mathbf{G_{ext}}[\mathbb{X}]$

are defined exactly as $\mathbf{G}[\mathbb{C}]$ / $\mathbf{G_{ext}}[\mathbb{C}]$, except that the introduction rules for $\mathsf{I}, \mathsf{K}, \mathsf{S}$ are replaced by the rules $[\mathsf{F}_l]_\mathbb{X}$, $[\mathsf{F}_r]_\mathbb{X}$, for each $\mathsf{F} \in \mathbf{X}$

G-systems for λ-calculus: $\mathbf{G}[\beta]$ / $\mathbf{G}_{\mathbf{ext}}[\beta]$

$\mathbf{G}[\beta]$ (corresponding to $\boldsymbol{\lambda}\beta$)

G-systems for λ-calculus: $\mathbf{G}[\beta]$ / $\mathbf{G}_{\mathbf{ext}}[\beta]$

$\mathbf{G}[\beta]$ (corresponding to $\boldsymbol{\lambda}\beta$)

G-systems for λ-calculus: $\mathbf{G}[\beta]$ / $\mathbf{G}_{\mathbf{ext}}[\beta]$

$\mathbf{G}[\beta]$ (corresponding to $\boldsymbol{\lambda}\beta$)

- "structural rules":

$$\frac{}{x = x}\,\rho' \qquad \frac{t = s \quad p = q}{tp = sq}\,App \qquad \frac{t = s}{\lambda x.t = \lambda x.s}\,\xi \qquad \frac{t = r \quad r = s}{t = s}\,\tau$$

G-systems for λ-calculus: $\mathbf{G}[\beta]$ / $\mathbf{G}_{\mathbf{ext}}[\beta]$

$\mathbf{G}[\beta]$ (corresponding to $\boldsymbol{\lambda}\beta$)

- "structural rules":

$$\frac{}{x = x}\,\rho' \qquad \frac{t = s \quad p = q}{tp = sq}\,App \qquad \frac{t = s}{\lambda x.t = \lambda x.s}\,\xi \qquad \frac{t = r \quad r = s}{t = s}\,\tau$$

- *left* and *right* β-introduction rules

G-systems for λ-calculus: $\mathbf{G}[\beta]$ / $\mathbf{G}_{\mathbf{ext}}[\beta]$

$\mathbf{G}[\beta]$ (corresponding to $\boldsymbol{\lambda}\beta$)

- "structural rules":

$$\frac{}{x = x}\,\rho' \qquad \frac{t = s \quad p = q}{tp = sq}\,App \qquad \frac{t = s}{\lambda x.t = \lambda x.s}\,\xi \qquad \frac{t = r \quad r = s}{t = s}\,\tau$$

- *left* and *right* β-introduction rules

$\mathbf{G}_{\mathbf{ext}}[\beta]$ (corresponding to $\boldsymbol{\lambda}\beta\eta$)

G-systems for λ-calculus: $\mathbf{G}[\beta]$ / $\mathbf{G}_{\mathbf{ext}}[\beta]$

$\mathbf{G}[\beta]$ (corresponding to $\boldsymbol{\lambda}\beta$)

- "structural rules":

$$\frac{}{x = x}\,\rho' \qquad \frac{t = s \quad p = q}{tp = sq}\,App \qquad \frac{t = s}{\lambda x.t = \lambda x.s}\,\xi \qquad \frac{t = r \quad r = s}{t = s}\,\tau$$

- *left* and *right* β-introduction rules

$\mathbf{G}_{\mathbf{ext}}[\beta]$ (corresponding to $\boldsymbol{\lambda}\beta\eta$)

- + the extensionality rule $[Ext]$

Transitivity elimination

Lemma [Equivalence]

G-systems are equivalent to the corresponding synthetic systems

Transitivity elimination

Lemma [Equivalence]

G-systems are equivalent to the corresponding synthetic systems

Main Theorem [τ-elimination]

G-systems admit (effective) transitivity elimination

Transitivity elimination

Lemma [Equivalence]

G-systems are equivalent to the corresponding synthetic systems

Main Theorem [τ-elimination]

G-systems admit (effective) transitivity elimination

Transitivity elimination

Lemma [Equivalence]

G-systems are equivalent to the corresponding synthetic systems

Main Theorem [τ-elimination]

G-systems admit (effective) transitivity elimination

Proof (in order of increasing complexity):

- $\mathbf{G}[\mathbb{X}]$ ($\mathbb{X}$ arbitrary)   [PM 04]

Transitivity elimination

Lemma [Equivalence]

G-systems are equivalent to the corresponding synthetic systems

Main Theorem [τ-elimination]

G-systems admit (effective) transitivity elimination

Proof (in order of increasing complexity):

- $\mathbf{G}[\mathbb{X}]$ ($\mathbb{X}$ arbitrary) [PM 04]
- $\mathbf{G_{ext}}[\mathbb{X}]$ **($\mathbb{X}$ linear)** [PM 04]

Transitivity elimination

Lemma [Equivalence]

G-systems are equivalent to the corresponding synthetic systems

Main Theorem [τ-elimination]

G-systems admit (effective) transitivity elimination

Proof (in order of increasing complexity):

- $\mathbf{G}[\mathbb{X}]$ ($\mathbb{X}$ arbitrary) [PM 04]
- $\mathbf{G_{ext}}[\mathbb{X}]$ **($\mathbb{X}$ linear)** [PM 04]
- $\mathbf{G}[\beta]$ and $\mathbf{G_{ext}}[\beta]$ [PM 07]

Transitivity elimination

Lemma [Equivalence]

G-systems are equivalent to the corresponding synthetic systems

Main Theorem [τ-elimination]

G-systems admit (effective) transitivity elimination

Proof (in order of increasing complexity):

- $\mathbf{G}[\mathbb{X}]$ ($\mathbb{X}$ arbitrary) [PM 04]
- $\mathbf{G_{ext}}[\mathbb{X}]$ **($\mathbb{X}$ linear)** [PM 04]
- $\mathbf{G}[\beta]$ and $\mathbf{G_{ext}}[\beta]$ [PM 07]
- $\mathbf{G_{ext}}[\mathbb{X}]$ **($\mathbb{X}$ arbitrary)** [PM 08]

Consequences & applications of τ-elimination

Consequences & applications of τ-elimination

- τ-free G-derivations enjoy a kind of **subterm property**

Consequences & applications of τ-elimination

- τ-free G-derivations enjoy a kind of **subterm property**
- This gives, as an immediate consequence

Consequences & applications of τ-elimination

- τ-free G-derivations enjoy a kind of **subterm property**
- This gives, as an immediate consequence
  - the unprovability of $x = y$ (with x distinct from y)

Consequences & applications of τ-elimination

- τ-free G-derivations enjoy a kind of **subterm property**
- This gives, as an immediate consequence
  - the unprovability of $x = y$ (with x distinct from y)
  - so the **consistency** of G-systems and of the corresponding synthetic systems

Consequences & applications of τ-elimination

- τ-free G-derivations enjoy a kind of **subterm property**
- This gives, as an immediate consequence
  - the unprovability of $x = y$ (with x distinct from y)
  - so the **consistency** of G-systems and of the corresponding synthetic systems
- Owing to the nice structural properties of τ-free derivations, we can provide a unified framework in which new **very short** demonstrations of central results concerning **reductions** can be given, including:

Consequences & applications of τ-elimination

- τ-free G-derivations enjoy a kind of **subterm property**
- This gives, as an immediate consequence
  - the unprovability of $x = y$ (with x distinct from y)
  - so the **consistency** of G-systems and of the corresponding synthetic systems
- Owing to the nice structural properties of τ-free derivations, we can provide a unified framework in which new **very short** demonstrations of central results concerning **reductions** can be given, including:
  - Church-Rosser

Consequences & applications of τ-elimination

- τ-free G-derivations enjoy a kind of **subterm property**
- This gives, as an immediate consequence
  - the unprovability of $x = y$ (with x distinct from y)
  - so the **consistency** of G-systems and of the corresponding synthetic systems
- Owing to the nice structural properties of τ-free derivations, we can provide a unified framework in which new **very short** demonstrations of central results concerning **reductions** can be given, including:
  - Church-Rosser
  - Standardization

Consequences & applications of τ-elimination

- τ-free G-derivations enjoy a kind of **subterm property**
- This gives, as an immediate consequence
  - the unprovability of $x = y$ (with x distinct from y)
  - so the **consistency** of G-systems and of the corresponding synthetic systems
- Owing to the nice structural properties of τ-free derivations, we can provide a unified framework in which new **very short** demonstrations of central results concerning **reductions** can be given, including:
  - Church-Rosser
  - Standardization
  - Leftmost reduction (in particular for $\lambda\beta\eta$-reduction)

Consequences & applications of τ-elimination

- τ-free G-derivations enjoy a kind of **subterm property**
- This gives, as an immediate consequence
  - the unprovability of $x = y$ (with x distinct from y)
  - so the **consistency** of G-systems and of the corresponding synthetic systems
- Owing to the nice structural properties of τ-free derivations, we can provide a unified framework in which new **very short** demonstrations of central results concerning **reductions** can be given, including:
  - Church-Rosser
  - Standardization
  - Leftmost reduction (in particular for $\lambda\beta\eta$-reduction)
  - ...

Common $\succ$*-reduct extraction* Lemma

Lemma

From any given τ-free $\mathbf{G_{ext}}[\mathbb{C}]$-derivation

$$\mathcal{D} \vdash^{-} t = s$$

one can effectively extract a term $r_{\mathcal{D}}$ such that $t \succ \boldsymbol{r_{\mathcal{D}}} \prec s$

Common $\succ$-reduct extraction Lemma

Lemma

From any given τ-free $\mathbf{G_{ext}}[\mathbb{C}]$-derivation

$$\mathcal{D} \vdash^{-} t = s$$

one can effectively extract a term $r_{\mathcal{D}}$ such that $t \succ \boldsymbol{r_{\mathcal{D}}} \prec s$

Common $\succ$-reduct extraction Lemma

Lemma

From any given τ-free $\mathbf{G}_{\mathbf{ext}}[\mathbb{C}]$-derivation

$$\mathcal{D} \vdash^{-} t = s$$

one can effectively extract a term $r_{\mathcal{D}}$ such that $\quad t \succ \boldsymbol{r}_{\boldsymbol{\mathcal{D}}} \prec s$

Proof: by straightforward induction on the length of $\mathcal{D}$.

Common $\succ$*-reduct extraction* Lemma

Lemma

From any given τ-free $\mathbf{G_{ext}}[\mathbb{C}]$-derivation

$$\mathcal{D} \vdash^{-} t = s$$

one can effectively extract a term $r_{\mathcal{D}}$ such that $t \succ \boldsymbol{r_{\mathcal{D}}} \prec s$

Proof: by straightforward induction on the length of $\mathcal{D}$.

- $\mathcal{D} \equiv t = t$ [t atomic] $\quad \boldsymbol{r_{\mathcal{D}}} := t$
- $\mathcal{D} \equiv App(\mathcal{D}_1, \mathcal{D}_2)$ $\quad \boldsymbol{r_{\mathcal{D}}} := \boldsymbol{r_{\mathcal{D}_1} r_{\mathcal{D}_2}}$
- $\mathcal{D} \equiv R(\mathcal{D}_1)$ [R a combinatory rule] $\quad \boldsymbol{r_{\mathcal{D}}} := \boldsymbol{r_{\mathcal{D}_1}}$
- $\mathcal{D} \equiv Ext_x(\mathcal{D}_1)$ $\quad \boldsymbol{r_{\mathcal{D}}} := \lambda^* x.\boldsymbol{r_{\mathcal{D}_1}}$

Common $\succ$*-reduct extraction* Lemma

Lemma

From any given τ-free $\mathbf{G}_{\mathbf{ext}}[\mathbb{C}]$-derivation

$$\mathcal{D} \vdash^{-} t = s$$

one can effectively extract a term $r_{\mathcal{D}}$ such that $\quad t \succ \boldsymbol{r}_{\mathcal{D}} \prec s$

Proof: by straightforward induction on the length of $\mathcal{D}$.

- $\mathcal{D} \equiv \ t = t \ \ [t$ atomic] $\qquad \boldsymbol{r}_{\mathcal{D}} := t$
- $\mathcal{D} \equiv \ App(\mathcal{D}_1, \mathcal{D}_2)$ $\qquad \boldsymbol{r}_{\mathcal{D}} := \boldsymbol{r}_{\mathcal{D}_1}\boldsymbol{r}_{\mathcal{D}_2}$
- $\mathcal{D} \equiv \ R(\mathcal{D}_1)$ $[R$ a combinatory rule] $\qquad \boldsymbol{r}_{\mathcal{D}} := \boldsymbol{r}_{\mathcal{D}_1}$
- $\mathcal{D} \equiv \ Ext_x(\mathcal{D}_1)$ $\qquad \boldsymbol{r}_{\mathcal{D}} := \lambda^* x.\boldsymbol{r}_{\mathcal{D}_1}$

As to the last case, indeed:

$$tx \succ r \prec sx \ \ [x \notin V(ts)] \quad \Rightarrow_{\text{rule } \xi} \quad t \equiv \lambda^* x.tx \succ \lambda^* x.r \prec \lambda^* x.sx \equiv s$$

A direct proof of the confluence of strong reduction

Suppose $t =_{c\beta\eta} s$, i.e.

$$\mathbf{CL_{ext}} \vdash t = s\,.$$

A direct proof of the confluence of strong reduction

Suppose $t =_{c\beta\eta} s$, i.e.

$$\mathbf{CL_{ext}} \vdash t = s \,.$$

A direct proof of the confluence of strong reduction

Suppose $t =_{c\beta\eta} s$, i.e.

$$\mathbf{CL_{ext}} \vdash t = s\,.$$

Then, by the *equivalence* Lemma and the τ-*elimination* Theorem, we get a transitivity-free $\mathbf{G_{ext}}[\mathbb{C}]$-derivation

$$\mathcal{D} \vdash^{-} t = s\,.$$

A direct proof of the confluence of strong reduction

Suppose $t =_{c\beta\eta} s$, i.e.

$$\mathbf{CL_{ext}} \vdash t = s\,.$$

Then, by the *equivalence* Lemma and the τ-*elimination* Theorem, we get a transitivity-free $\mathbf{G_{ext}}[\mathbb{C}]$-derivation

$$\mathcal{D} \vdash^{-} t = s\,.$$

A final application of the *extraction* Lemma to $\mathcal{D}$ yields a **common** $\succ$**-reduct** $r_{\mathcal{D}}$ of t and s:

$$t \succ r_{\mathcal{D}} \prec s$$

A direct proof of the confluence of strong reduction

Suppose $t =_{c\beta\eta} s$, i.e.

$$\mathbf{CL_{ext}} \vdash t = s\,.$$

Then, by the *equivalence* Lemma and the τ-*elimination* Theorem, we get a transitivity-free $\mathbf{G_{ext}}[\mathbb{C}]$-derivation

$$\mathcal{D} \vdash^{-} t = s\,.$$

A final application of the *extraction* Lemma to $\mathcal{D}$ yields a **common $\succ$-reduct** $r_{\mathcal{D}}$ of t and s:

$$t \succ r_{\mathcal{D}} \prec s$$

This confluence proof for $\succ$ is independent of λ-calculus!

Terminology and notations

Terminology and notations

- $\|t\| :=$ the depth of t

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \ldots :$ contexts (terms with some holes $*$)

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \ldots :$ contexts (terms with some holes $*$)
- $t, \Phi \mapsto \Phi[\![t]\!]$

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \ldots :$ contexts (terms with some holes $*$)
- $t, \Phi \mapsto \Phi[\![t]\!]$

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \ldots :$ contexts (terms with some holes $*$)
- $t, \Phi \mapsto \Phi[\![t]\!]$

- $\mathcal{D}, \mathcal{D}' \ldots :$ $\mathbf{G}_{\mathbf{ext}}[\mathbb{X}]$-derivations

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \ldots$: contexts (terms with some holes $*$)
- $t, \Phi \mapsto \Phi[\![t]\!]$

- $\mathcal{D}, \mathcal{D}' \ldots$: $\mathbf{G_{ext}}[\mathbb{X}]$-derivations
- $\mathcal{D} \vdash^{-} t = s$: $\mathcal{D}$ is a τ-free derivation of $t = s$

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \ldots$: contexts (terms with some holes $*$)
- $t, \Phi \mapsto \Phi[\![t]\!]$

- $\mathcal{D}, \mathcal{D}' \ldots$: $\mathbf{G_{ext}}[\mathbb{X}]$-derivations
- $\mathcal{D} \vdash^{-} t = s$: $\mathcal{D}$ is a τ-free derivation of $t = s$
- *Left* derivation ($\vdash_L$): no **right** combinatory inferences

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \dots :$ contexts (terms with some holes $*$)
- $t, \Phi \mapsto \Phi[\![t]\!]$

- $\mathcal{D}, \mathcal{D}' \dots :$ $\mathbf{G}_{\mathbf{ext}}[\mathbb{X}]$-derivations
- $\mathcal{D} \vdash^{-} t = s :$ $\mathcal{D}$ is a τ-free derivation of $t = s$
- *Left* derivation ($\vdash_L$): no **right** combinatory inferences
- *Right* derivation ($\vdash_R$): dually

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \ldots$: contexts (terms with some holes $*$)
- $t, \Phi \mapsto \Phi[\![t]\!]$

- $\mathcal{D}, \mathcal{D}' \ldots$: $\mathbf{G_{ext}}[\mathbb{X}]$-derivations
- $\mathcal{D} \vdash^{-} t = s$: $\mathcal{D}$ is a τ-free derivation of $t = s$
- *Left* derivation ($\vdash_L$): no **right** combinatory inferences
- *Right* derivation ($\vdash_R$): dually

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \ldots$: contexts (terms with some holes $*$)
- $t, \Phi \mapsto \Phi[\![t]\!]$

- $\mathcal{D}, \mathcal{D}' \ldots$: $\mathbf{G_{ext}}[\mathbb{X}]$-derivations
- $\mathcal{D} \vdash^{-} t = s$: $\mathcal{D}$ is a τ-free derivation of $t = s$
- *Left* derivation ($\vdash_L$): no **right** combinatory inferences
- *Right* derivation ($\vdash_R$): dually

- $\mathsf{s}(\mathcal{D}) :=$ # of combinatory and $[Ext]$ inferences in $\mathcal{D}$

Terminology and notations

- $\|t\| :=$ the depth of t
- $\Phi, \Psi, \ldots$: contexts (terms with some holes $*$)
- $t, \Phi \mapsto \Phi[\![t]\!]$

- $\mathcal{D}, \mathcal{D}' \ldots$: $\mathbf{G_{ext}}[\mathbb{X}]$-derivations
- $\mathcal{D} \vdash^{-} t = s$: $\mathcal{D}$ is a τ-free derivation of $t = s$
- *Left* derivation ($\vdash_L$): no **right** combinatory inferences
- *Right* derivation ($\vdash_R$): dually

- $\mathsf{s}(\mathcal{D}) :=$ # of combinatory and $[Ext]$ inferences in $\mathcal{D}$
- $\mathsf{h}(\mathcal{D}) :=$ tree-height of $\mathcal{D}$

Proof strategy — $\mathbf{G}[\mathbb{X}]$ systems

Proof strategy — $\mathbf{G}[\mathbb{X}]$ systems

Proof strategy — $\mathbf{G}[\mathbb{X}]$ systems

We show how to eliminate a topmost application of τ :

$$\mathcal{D}_1 \vdash^- t = s\,,\ \mathcal{D}_2 \vdash^- s = r \ \longmapsto\ \mathcal{D}^* \vdash^- t = r$$

Proof strategy — $\mathbf{G}[\mathbb{X}]$ systems

We show how to eliminate a topmost application of τ :

$$\mathcal{D}_1 \vdash^- t = s \,,\; \mathcal{D}_2 \vdash^- s = r \;\longmapsto\; \mathcal{D}^* \vdash^- t = r$$

The *proof* runs by ω^3-induction:

Proof strategy — $\mathbf{G}[\mathbb{X}]$ systems

We show how to eliminate a topmost application of τ :

$$\mathcal{D}_1 \vdash^- t = s\,,\ \mathcal{D}_2 \vdash^- s = r \longmapsto \mathcal{D}^* \vdash^- t = r$$

The *proof* runs by ω^3-induction:

main: $\mathsf{h}'(\mathcal{D}_1) + \mathsf{h}'(\mathcal{D}_2)$

Proof strategy — $\mathbf{G}[\mathbb{X}]$ systems

We show how to eliminate a topmost application of τ :

$$\mathcal{D}_1 \vdash^- t = s\,,\ \mathcal{D}_2 \vdash^- s = r \ \longmapsto\ \mathcal{D}^* \vdash^- t = r$$

The *proof* runs by ω^3-induction:

main: $\mathsf{h}'(\mathcal{D}_1) + \mathsf{h}'(\mathcal{D}_2)$

secondary: $\mathsf{s}(\mathcal{D}_1) + \mathsf{s}(\mathcal{D}_2)$

Proof strategy — $\mathbf{G}[\mathbb{X}]$ systems

We show how to eliminate a topmost application of τ :

$$\mathcal{D}_1 \vdash^- t = s\,,\ \mathcal{D}_2 \vdash^- s = r \longmapsto \mathcal{D}^* \vdash^- t = r$$

The *proof* runs by ω^3-induction:

main: $\mathsf{h}'(\mathcal{D}_1) + \mathsf{h}'(\mathcal{D}_2)$

secondary: $\mathsf{s}(\mathcal{D}_1) + \mathsf{s}(\mathcal{D}_2)$

ternary: $\|s\|$

Proof strategy — $\mathbf{G}[\mathbb{X}]$ systems

We show how to eliminate a topmost application of τ :

$$\mathcal{D}_1 \vdash^- t = s\,,\ \mathcal{D}_2 \vdash^- s = r \ \longmapsto\ \mathcal{D}^* \vdash^- t = r$$

The *proof* runs by ω^3-induction:

main: $\mathrm{h}'(\mathcal{D}_1) + \mathrm{h}'(\mathcal{D}_2)$

secondary: $\mathrm{s}(\mathcal{D}_1) + \mathrm{s}(\mathcal{D}_2)$

ternary: $\|s\|$

This strategy doesn't work when the **extensionality rule** is present, coupled with **non linear** combinators.

Proof strategy — $\mathbf{G_{ext}}[\mathbb{X}]$ systems

Proof strategy — $\mathbf{G}_{\mathbf{ext}}[\mathbb{X}]$ systems

We show that the following **generalized transitivity rule**

$$\frac{t = s \qquad \Phi[\![s]\!] = r}{\Phi[\![t]\!] = r}\ \tau^*$$

is eliminable.

Proof strategy — $\mathbf{G}_{\mathbf{ext}}[\mathbb{X}]$ systems

We show that the following **generalized transitivity rule**

$$\frac{t = \mathbf{s} \qquad \Phi[\![\mathbf{s}]\!] = r}{\Phi[\![t]\!] = r}\ \tau^*$$

is eliminable.

Proof strategy — $\mathbf{G_{ext}}[\mathbb{X}]$ systems

We show that the following **generalized transitivity rule**

$$\frac{t = \mathbf{s} \qquad \Phi[\![\mathbf{s}]\!] = r}{\Phi[\![t]\!] = r}\ \tau^*$$

is eliminable.

The proof consists of **four** main steps (in this order):

Proof strategy — $\mathbf{G_{ext}}[\mathbb{X}]$ systems

We show that the following **generalized transitivity rule**

$$\frac{t = \mathbf{s} \qquad \Phi[\![\mathbf{s}]\!] = r}{\Phi[\![t]\!] = r}\,\tau^*$$

is eliminable.

The proof consists of **four** main steps (in this order):

- generalized F-inversion

Proof strategy — $\mathbf{G_{ext}}[\mathbb{X}]$ systems

We show that the following **generalized transitivity rule**

$$\frac{t = \mathbf{s} \qquad \Phi[\![\mathbf{s}]\!] = r}{\Phi[\![t]\!] = r}\ \tau^*$$

is eliminable.

The proof consists of **four** main steps (in this order):

- generalized F-inversion
- *left* τ-elimination

Proof strategy — $\mathbf{G}_{\mathbf{ext}}[\mathbb{X}]$ systems

We show that the following **generalized transitivity rule**

$$\frac{t = \mathbf{s} \qquad \Phi[\![\mathbf{s}]\!] = r}{\Phi[\![t]\!] = r}\,\tau^*$$

is eliminable.

The proof consists of **four** main steps (in this order):

- generalized F-inversion
- *left* τ-elimination
- generalized F-introduction

Proof strategy — $\mathbf{G}_{\mathbf{ext}}[\mathbb{X}]$ systems

We show that the following **generalized transitivity rule**

$$\frac{t = s \qquad \Phi[\![s]\!] = r}{\Phi[\![t]\!] = r}\,\tau^*$$

is eliminable.

The proof consists of **four** main steps (in this order):

- generalized F-inversion
- *left* τ-elimination
- generalized F-introduction
- elimination of a topmost occurrence of $[\tau^*]$

Step 1: *generalized* F-*inversion* Lemma

For any $\mathsf{F} \in \mathbf{X}$, with $k = k_{\mathsf{F}}$, and any context Φ:

Every τ-free derivation

$$\mathcal{D} \vdash^{-} \Phi[\![\mathsf{F} t_1 \ldots t_k p_1 \ldots p_n]\!] = s$$

can effectively be transformed into a τ-free derivation

$$\mathcal{D}^{*} \vdash^{-} \Phi[\![d_{\mathsf{F}}[t_1, \ldots, t_k] p_1 \ldots p_n]\!] = s$$

which, moreover, is a *right* derivation provided $\mathcal{D}$ is a *right* derivation

Step 1: *generalized* F-*inversion* Lemma

For any $\mathsf{F} \in \mathbf{X}$, with $k = k_\mathsf{F}$, and any context Φ:

Every τ-free derivation

$$\mathcal{D} \vdash^{-} \Phi[\![\mathsf{F} t_1 \dots t_k p_1 \dots p_n]\!] = s$$

can effectively be transformed into a τ-free derivation

$$\mathcal{D}^* \vdash^{-} \Phi[\![d_\mathsf{F}[t_1, \dots, t_k] p_1 \dots p_n]\!] = s$$

which, moreover, is a *right* derivation provided $\mathcal{D}$ is a *right* derivation

Step 1: *generalized* F-*inversion* Lemma

For any $\mathsf{F} \in \mathbf{X}$, with $k = k_{\mathsf{F}}$, and any context Φ:

Every τ-free derivation

$$\mathcal{D} \vdash^{-} \Phi[\![\mathsf{F}t_1 \dots t_k p_1 \dots p_n]\!] = s$$

can effectively be transformed into a τ-free derivation

$$\mathcal{D}^* \vdash^{-} \Phi[\![d_{\mathsf{F}}[t_1, \dots, t_k] p_1 \dots p_n]\!] = s$$

which, moreover, is a *right* derivation provided $\mathcal{D}$ is a *right* derivation

This follows from the following:

Lemma

Lemma

Lemma

Given

- *a τ-free derivation $\mathcal{D} \vdash^{-} t = s$*

Lemma

Given

- *a τ-free derivation $\mathcal{D} \vdash^{-} t = s$*
- *a set S of* F*-redexes occurrences in t*

Lemma

Given

- *a τ-free derivation $\mathcal{D} \vdash^{-} t = s$*
- *a set S of* F*-redexes occurrences in t*

Lemma

Given

- *a τ-free derivation $\mathcal{D} \vdash^{-} t = s$*
- *a set S of F-redexes occurrences in t*

we can construct a τ-free derivation

$$\mathcal{D}^{\sharp} \vdash^{-} \boldsymbol{t^{\sharp}} = s\,,$$

Lemma

Given

- *a τ-free derivation $\mathcal{D} \vdash^{-} t = s$*
- *a set S of* F*-redexes occurrences in t*

we can construct a τ-free derivation

$$\mathcal{D}^{\sharp} \vdash^{-} \boldsymbol{t}^{\sharp} = s\,,$$

where $\boldsymbol{t}^{\sharp}$ is the term obtained from t by minimal-redex-first complete development of S.

Moreover, $\mathcal{D}^{\sharp}$ is a right derivation provided $\mathcal{D}$ is such.

Lemma

Given

- *a τ-free derivation $\mathcal{D} \vdash^{-} t = s$*
- *a set S of F-redexes occurrences in t*

we can construct a τ-free derivation

$$\mathcal{D}^{\sharp} \vdash^{-} \boldsymbol{t^{\sharp}} = s\,,$$

where $\boldsymbol{t^{\sharp}}$ is the term obtained from t by minimal-redex-first complete development of S.

Moreover, $\mathcal{D}^{\sharp}$ is a right derivation provided $\mathcal{D}$ is such.

Proof.

By main induction on $\mathsf{s}(\mathcal{D})$ and secondary induction on $\|t\|$. $\square$

Step 2: **left** τ-*elimination* Lemma

Lemma

Step 2: **left** τ-*elimination* Lemma

Lemma

Step 2: **left** τ*-elimination* Lemma

Lemma

To any given pair

$$\mathcal{D}_1 \vdash_L^- t = s \qquad \text{and} \qquad \mathcal{D}_2 \vdash^- s = r$$

of τ-free derivations, such that $\mathcal{D}_1$ is a left derivation, we can effectively associate a τ-free derivation

$$\mathcal{D}^* \vdash^- t = r$$

which is a left derivation provided $\mathcal{D}_2$ is such.

Step 2: **left** τ-*elimination* Lemma

Lemma

To any given pair

$$\mathcal{D}_1 \vdash^-_L t = s \qquad \textit{and} \qquad \mathcal{D}_2 \vdash^- s = r$$

of τ-free derivations, such that $\mathcal{D}_1$ is a left derivation, we can effectively associate a τ-free derivation

$$\mathcal{D}^* \vdash^- t = r$$

which is a left derivation provided $\mathcal{D}_2$ is such.

Proof.

Main induction on $\mathrm{s}(\mathcal{D}_2)$, secondary induction on $\mathrm{s}(\mathcal{D}_1)$, ternary induction on $\|s\|$, using F-inversion. □

Step 3: *generalized* F-*introduction* Lemma

For any F $\in \mathbf{X}$, with $k = k_{\mathsf{F}}$, and any context Φ:

The following generalized combinatory introduction rules are τ-free admissible:

$$\frac{\Phi[\![d_{\mathsf{F}}[t_1,\ldots,t_k]p_1\ldots p_n]\!] = s}{\Phi[\![\mathsf{F}t_1\ldots t_k p_1\ldots p_n]\!] = s}\;[\mathsf{F}_l^+] \qquad \frac{s = \Phi[\![d_{\mathsf{F}}[t_1,\ldots,t_k]p_1\ldots p_n]\!]}{s = \Phi[\![\mathsf{F}t_1\ldots t_k p_1\ldots p_n]\!]}\;[\mathsf{F}_r^+]$$

Moreover, $[\mathsf{F}_l^+]$ and $[\mathsf{F}_r^+]$ preserve *left-handedness*, resp. *right-handedness*.

Step 3: *generalized* F-*introduction* Lemma

For any $\mathsf{F} \in \mathbf{X}$, with $k = k_{\mathsf{F}}$, and any context Φ:

The following generalized combinatory introduction rules are τ-free admissible:

$$\frac{\Phi[\![d_{\mathsf{F}}[t_1, \ldots, t_k]p_1 \ldots p_n]\!] = s}{\Phi[\![\mathsf{F}t_1 \ldots t_k p_1 \ldots p_n]\!] = s}\,[\mathsf{F}_l^+] \qquad \frac{s = \Phi[\![d_{\mathsf{F}}[t_1, \ldots, t_k]p_1 \ldots p_n]\!]}{s = \Phi[\![\mathsf{F}t_1 \ldots t_k p_1 \ldots p_n]\!]}\,[\mathsf{F}_r^+]$$

Moreover, $[\mathsf{F}_l^+]$ and $[\mathsf{F}_r^+]$ preserve *left-handedness*, resp. *right-handedness*.

Proof.

By left τ-elimination. □

$$\dfrac{\begin{matrix} \divideontimes \\ \vdots \\ \Phi[\![\mathsf{F}t_1 \dots t_k \overline{p}]\!] = \Phi[\![d_{\mathsf{F}}[t_1, \dots, t_k]\overline{p}]\!] \end{matrix} \qquad \begin{matrix} \vdots \\ \Phi[\![d_{\mathsf{F}}[t_1, \dots, t_k]\overline{p}]\!] = s \end{matrix}}{\Phi[\![\mathsf{F}t_1 \dots t_k \overline{p}]\!] = s} \textbf{ Left elim.}$$

$\divideontimes$: structural rules + applications of $[\mathsf{F}_l]$

Final step: *main elimination* Lemma

For any context Φ:

To each pair of τ-free derivations

$$\mathcal{D}_1 \vdash^- t = \mathbf{s} \quad \textit{and} \quad \mathcal{D}_2 \vdash^- \Phi[\![\mathbf{s}]\!] = r$$

we can effectively associate a τ-free derivation

$$\mathcal{D}^* \vdash^- \Phi[\![t]\!] = r$$

Final step: *main elimination* Lemma

For any context Φ:

To each pair of τ-free derivations

$$\mathcal{D}_1 \vdash^- t = \boldsymbol{s} \quad \textit{and} \quad \mathcal{D}_2 \vdash^- \Phi[\![\boldsymbol{s}]\!] = r$$

we can effectively associate a τ-free derivation

$$\mathcal{D}^* \vdash^- \Phi[\![t]\!] = r$$

The proof runs by ω^3-induction

Final step: *main elimination* Lemma

For any context Φ:

To each pair of τ-free derivations

$$\mathcal{D}_1 \vdash^{-} t = \boldsymbol{s} \quad \textit{and} \quad \mathcal{D}_2 \vdash^{-} \Phi[\![\boldsymbol{s}]\!] = r$$

we can effectively associate a τ-free derivation

$$\mathcal{D}^* \vdash^{-} \Phi[\![t]\!] = r$$

The proof runs by ω^3-induction

- main: $\mathsf{s}(\mathcal{D}_1)$

Final step: *main elimination* Lemma

For any context Φ:

To each pair of τ-free derivations

$$\mathcal{D}_1 \vdash^- t = \boldsymbol{s} \quad \textit{and} \quad \mathcal{D}_2 \vdash^- \Phi[\![\boldsymbol{s}]\!] = r$$

we can effectively associate a τ-free derivation

$$\mathcal{D}^* \vdash^- \Phi[\![t]\!] = r$$

The proof runs by ω^3-induction

- main: s$(\mathcal{D}_1)$
- secondary: $\|s\|$

Final step: *main elimination* Lemma

For any context Φ:

To each pair of τ-free derivations

$$\mathcal{D}_1 \vdash^- t = \boldsymbol{s} \quad \textit{and} \quad \mathcal{D}_2 \vdash^- \Phi[\![\boldsymbol{s}]\!] = r$$

we can effectively associate a τ-free derivation

$$\mathcal{D}^* \vdash^- \Phi[\![t]\!] = r$$

The proof runs by ω^3-induction

- main: s$(\mathcal{D}_1)$
- secondary: $\|s\|$
- ternary: h$(\mathcal{D}_2)$

Final step: *main elimination* Lemma

For any context Φ:

To each pair of τ-free derivations

$$\mathcal{D}_1 \vdash^- t = \boldsymbol{s} \quad \textit{and} \quad \mathcal{D}_2 \vdash^- \Phi[\![\boldsymbol{s}]\!] = r$$

we can effectively associate a τ-free derivation

$$\mathcal{D}^* \vdash^- \Phi[\![t]\!] = r$$

The proof runs by ω^3-induction

- main: s$(\mathcal{D}_1)$
- secondary: $\|s\|$
- ternary: h$(\mathcal{D}_2)$

Final step: *main elimination* Lemma

For any context Φ:

To each pair of τ-free derivations

$$\mathcal{D}_1 \vdash^{-} t = s \quad \textit{and} \quad \mathcal{D}_2 \vdash^{-} \Phi[\![s]\!] = r$$

we can effectively associate a τ-free derivation

$$\mathcal{D}^* \vdash^{-} \Phi[\![t]\!] = r$$

The proof runs by ω^3-induction

- main: $\mathsf{s}(\mathcal{D}_1)$
- secondary: $\|s\|$
- ternary: $\mathsf{h}(\mathcal{D}_2)$

taking main cases according to the last inference R of $\mathcal{D}_1$.

Case $R = [\mathsf{F}_r]$

Case $R = [\mathsf{F}_r]$

M.I.H. + generalized F-inversion

Case $R = [\mathsf{F}_r]$

M.I.H. + generalized F-inversion

Case $R = [\mathsf{F}_r]$

M.I.H. + generalized F-inversion

$$\dfrac{\dfrac{\dfrac{\vdots}{t = s'}}{t = s}\,\mathsf{F}_r \qquad \dfrac{\vdots}{\Phi[\![s]\!] = r}}{\Phi[\![t]\!] = r}\,\tau *$$

Case $R = [\mathsf{F}_r]$

M.I.H. $+$ **generalized** F**-inversion**

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ t = s'\end{array}}{t = s}\,\mathsf{F}_r \qquad \begin{array}{c}\vdots\\ \Phi[\![s]\!] = r\end{array}}{\Phi[\![t]\!] = r}\,\tau *$$

$$\blacktriangledown$$

$$\dfrac{\begin{array}{c}\vdots\\ t = s'\end{array} \qquad \dfrac{\begin{array}{c}\vdots\\ \Phi[\![s]\!] = r\end{array}}{\Phi[\![s']\!] = r}\,\mathsf{F}_{inv}}{\Phi[\![t]\!] = r}\,M.I.H$$

Case $R = [\mathsf{F}_l]$

Case $R = [\mathsf{F}_l]$

M.I.H. + generalized F-introduction

Case $R = [\mathsf{F}_l]$

M.I.H. + generalized F-introduction

Case $R = [\mathsf{F}_l]$

M.I.H. + generalized F-introduction

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ t' = s\end{matrix}}{t = s}\,\mathsf{F}_l \qquad \begin{matrix}\vdots\\ \Phi[\![s]\!] = r\end{matrix}}{\Phi[\![t]\!] = r}\,\tau*$$

Case $R = [\mathsf{F}_l]$

M.I.H. + generalized F-introduction

$$\dfrac{\dfrac{\dfrac{\vdots}{t' = s}}{t = s}\,\mathsf{F}_l \qquad \dfrac{\vdots}{\Phi[\![s]\!] = r}}{\Phi[\![t]\!] = r}\,\tau*$$

$$\blacktriangledown$$

$$\dfrac{\dfrac{\dfrac{\vdots}{t' = s} \qquad \dfrac{\vdots}{\Phi[\![s]\!] = r}}{\Phi[\![t']\!] = r}\,M.I.H.}{\Phi[\![t]\!] = r}\,\mathsf{F}_l^+$$

Case $R = [App]$

Case $R = [App]$

S.I.H. + context shifts

Case $R = [App]$

S.I.H. $+$ context shifts

Case $R = [App]$

S.I.H. + **context shifts**

$$
\dfrac{\dfrac{\begin{array}{c}\vdots \\ t_1 = s_1\end{array} \quad \begin{array}{c}\vdots \\ t_2 = s_2\end{array}}{t_1 t_2 = s_1 s_2}\, App \qquad \begin{array}{c}\vdots \\ \Phi[\![s_1 s_2]\!] = r\end{array}}{\Phi[\![t_1 t_2]\!] = r}\, \tau^*
$$

Case $R = [App]$

S.I.H. + context shifts

$$\dfrac{\dfrac{\dfrac{\vdots}{t_1 = s_1} \quad \dfrac{\vdots}{t_2 = s_2}}{t_1 t_2 = s_1 s_2}\,App \qquad \dfrac{\vdots}{\Phi[\![s_1 s_2]\!] = r}}{\Phi[\![t_1 t_2]\!] = r}\,\tau^*$$

▼

Case $R = [App]$

S.I.H. + context shifts

$$
\dfrac{\dfrac{\dfrac{\vdots}{t_1 = s_1} \quad \dfrac{\vdots}{t_2 = s_2}}{t_1 t_2 = s_1 s_2}\,App \qquad \dfrac{\vdots}{\Phi[\![s_1 s_2]\!] = r}}{\Phi[\![t_1 t_2]\!] = r}\,\tau^*
$$

$$\blacktriangledown$$

$$
\dfrac{\dfrac{\vdots}{t_2 = s_2} \qquad \dfrac{\dfrac{\vdots}{t_1 = s_1} \quad \dfrac{\vdots}{\Phi[\![s_1 s_2]\!] = r}}{\Phi[\![t_1 s_2]\!] = r}\,S.I.H.}{\Phi[\![t_1 t_2]\!] = r}\,S.I.H.
$$

Case $R = [App]$

S.I.H. + context shifts

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ t_1 = s_1\end{array} \quad \begin{array}{c}\vdots\\ t_2 = s_2\end{array}}{t_1 t_2 = s_1 s_2}\,App \qquad \begin{array}{c}\vdots\\ \Phi[\![s_1 s_2]\!] = r\end{array}}{\Phi[\![t_1 t_2]\!] = r}\,\tau^*$$

$$\blacktriangledown$$

$$\dfrac{\begin{array}{c}\vdots\\ t_2 = s_2\end{array} \qquad \dfrac{\begin{array}{c}\vdots\\ t_1 = s_1\end{array} \quad \begin{array}{c}\vdots\\ \color{red}{\Psi[\![s_1]\!]} = r\end{array}}{\Phi[\![t_1 s_2]\!] = r}\,S.I.H.}{\Phi[\![t_1 t_2]\!] = r}\,S.I.H.$$

Case $R = [App]$

S.I.H. + context shifts

$$
\dfrac{\dfrac{\dfrac{\vdots}{t_1 = s_1} \quad \dfrac{\vdots}{t_2 = s_2}}{t_1 t_2 = s_1 s_2}\,App \qquad \dfrac{\vdots}{\Phi[\![s_1 s_2]\!] = r}}{\Phi[\![t_1 t_2]\!] = r}\,\tau^*
$$

$$\blacktriangledown$$

$$
\dfrac{\dfrac{\vdots}{t_2 = s_2} \qquad \dfrac{\dfrac{\vdots}{t_1 = s_1} \quad \dfrac{\vdots}{\Phi[\![s_1 s_2]\!] = r}}{\Theta[\![s_2]\!] = r}\,S.I.H.}{\Phi[\![t_1 t_2]\!] = r}\,S.I.H.
$$

Case $R = [Ext]$

Case $R = [Ext]$

This is the most complex case.

Case $R = [Ext]$

This is the most complex case.

Case $R = [Ext]$

This is the most complex case.

We have now to look both at

- the last inference R' of $\mathcal{D}_2$

Case $R = [Ext]$

This is the most complex case.

We have now to look both at

- the last inference R' of $\mathcal{D}_2$
- the form of the context Φ

The case $\Phi \equiv *$ is easily disposed off by the M.I.H.

The case $\Phi \equiv *$ is easily disposed off by the M.I.H.

$$\dfrac{\dfrac{\dfrac{\vdots}{tx = sx}}{t = s}\,Ext \qquad \dfrac{\vdots}{s = r}}{t = r}\,\tau^*$$

The case $\Phi \equiv *$ is easily disposed off by the M.I.H.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ tx = sx\end{matrix}}{t = s}\,Ext \qquad \begin{matrix}\vdots\\ s = r\end{matrix}}{t = r}\,\tau^*$$

The case $\Phi \equiv *$ is easily disposed off by the M.I.H.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ tx = sx\end{matrix}}{t = s}\,\mathit{Ext} \qquad \begin{matrix}\vdots\\ s = r\end{matrix}}{t = r}\,\tau^*$$

▼

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ tx = sx\end{matrix} \qquad \dfrac{\begin{matrix}\vdots\\ s = r\end{matrix} \qquad x = x}{sx = rx}\,\mathit{App}}{tx = rx}\,\mathit{M.I.H.}}{t = r}\,\mathit{Ext}$$

If Φ is distinct from $*$ we look at R'

If Φ is distinct from $*$ we look at R'

$R' = [App]$ / [F_r] / $[Ext]$

Easy, by the ternary I.H.

If Φ is distinct from $*$ we look at R'

$R' = [App] \,/\, [\mathsf{F}_r] \,/\, [Ext]$

Easy, by the ternary I.H.

$R' = [\mathsf{F}_l]$

More delicate: a "cross-cut" is required.

We use the ternary I.H. followed by an application of the M.I.H.

Combinatory introduction rules for the combinator S :

Combinatory introduction rules for the combinator S :

$$\mathsf{S}tsr = tr(sr) \quad [\mathrm{AX}\ \mathsf{S}]$$

Combinatory introduction rules for the combinator S :

$$\mathsf{S}tsr = tr(sr) \quad [\mathrm{AX}\ \mathsf{S}]$$

Combinatory introduction rules for the combinator S :

$$\mathsf{S}tsr = tr(sr) \quad [\text{AX S}]$$

$$\Downarrow\Downarrow$$

$$\frac{tr(sr)p_1 \ldots p_n = q}{\mathsf{S}tsrp_1 \ldots p_n = q}\ [\mathsf{S}_l] \qquad \frac{q = tr(sr)p_1 \ldots p_n}{q = \mathsf{S}tsrp_1 \ldots p_n}\ [\mathsf{S}_r]$$

where $n \geq 0$, i.e.: the "side terms" $p_1, \ldots, p_n$ may be missing

Combinatory introduction rules for the combinator S :

$$\mathsf{S}tsr = tr(sr) \quad [\mathrm{AX}\ \mathsf{S}]$$

$$\Downarrow \Downarrow$$

$$\frac{tr(sr)p_1 \ldots p_n = q}{\mathsf{S}tsrp_1 \ldots p_n = q}\,[\mathsf{S}_l] \qquad \frac{q = tr(sr)p_1 \ldots p_n}{q = \mathsf{S}tsrp_1 \ldots p_n}\,[\mathsf{S}_r]$$

where $n \geq 0$, i.e.: the "side terms" $p_1, \ldots, p_n$ may be missing

Combinatory introduction rules for other primitive combinators F:

[F_l] and [F_r] are defined similarly

β-introduction rules:

β-introduction rules:

$$(\lambda x.t)r = t[x/r] \quad [\beta\mathrm{-conv}]$$

β-introduction rules:

$$(\lambda x.t)r = t[x/r] \quad [\beta\mathrm{-conv}]$$

β-introduction rules:

$$(\lambda x.t)r = t[x/r] \quad [\beta{-}\mathrm{conv}]$$

$$\Downarrow\Downarrow$$

$$\frac{t[x/r]p_1 \dots p_n = q}{(\lambda x.t)rp_1 \dots p_n = q}\,[\beta_l] \qquad \frac{q = t[x/r]p_1 \dots p_n}{q = (\lambda x.t)rp_1 \dots p_n}\,[\beta_r]$$

where $n \geq 0$, i.e.: the “side terms” $p_1, \dots, p_n$ may be missing

β-introduction rules:

β-introduction rules:

$$(\lambda x.t)r = t[x/r] \quad [\beta\mathrm{-conv}]$$

β-introduction rules:

$$(\lambda x.t)r = t[x/r] \quad [\beta\text{−conv}]$$

β-introduction rules:

$$(\lambda x.t)r = t[x/r] \quad [\beta\text{−conv}]$$

$$\Downarrow \Downarrow$$

$$\frac{t[x/r]p_1 \ldots p_n = q}{(\lambda x.t)rp_1 \ldots p_n = q}\,[\beta_l] \qquad \frac{q = t[x/r]p_1 \ldots p_n}{q = (\lambda x.t)rp_1 \ldots p_n}\,[\beta_r]$$

where $n \geq 0$, i.e.: the “side terms” $p_1, \ldots, p_n$ may be missing

$$\frac{tp_1 \ldots p_n = s}{\mathsf{I}tp_1 \ldots p_n = s}[\mathsf{I}_l] \qquad \frac{s = tp_1 \ldots p_n}{s = \mathsf{I}tp_1 \ldots p_n}[\mathsf{I}_r] \qquad (n \geq 0)$$

$$\frac{tp_1 \ldots p_n = s}{\mathsf{K}trp_1 \ldots p_n = s}[\mathsf{K}_l] \qquad \frac{s = tp_1 \ldots p_n}{s = \mathsf{K}trp_1 \ldots p_n}[\mathsf{K}_r] \qquad (n \geq 0)$$

$$\frac{tq(rq)p_1 \ldots p_n = s}{\mathsf{S}trqp_1 \ldots p_n = s}[\mathsf{S}_l] \qquad \frac{s = tq(rq)p_1 \ldots p_n}{s = \mathsf{S}trqp_1 \ldots p_n}[\mathsf{S}_r] \qquad (n \geq 0)$$

Convention

We write $t[s_1, \ldots, s_n]$ short for $t[v_1/s_1, \ldots, v_n/s_n]$

Convention

We write $t[s_1, \ldots, s_n]$ short for $t[v_1/s_1, \ldots, v_n/s_n]$

$$\mathsf{F}t_1 \ldots t_{k_\mathsf{F}} = d_\mathsf{F}[t_1, \ldots, t_{k_\mathsf{F}}] \quad (\mathrm{AX}\,\mathsf{F})_{\mathbb{X}}$$

Convention

We write $t[s_1, \ldots, s_n]$ short for $t[v_1/s_1, \ldots, v_n/s_n]$

$$\mathsf{F}t_1 \ldots t_{k_\mathsf{F}} = d_\mathsf{F}[t_1, \ldots, t_{k_\mathsf{F}}] \quad (\mathrm{AX}\,\mathsf{F})_\mathbb{X}$$

Convention

We write $t[s_1, \ldots, s_n]$ short for $t[v_1/s_1, \ldots, v_n/s_n]$

$$\mathsf{F} t_1 \ldots t_{k_\mathsf{F}} = d_\mathsf{F}[t_1, \ldots, t_{k_\mathsf{F}}] \quad (\mathrm{AX}\,\mathsf{F})_\mathbb{X}$$

$$\Downarrow\Downarrow$$

$$\frac{d_\mathsf{F}[t_1, \ldots, t_{k_\mathsf{F}}] p_1 \ldots p_n = s}{\mathsf{F} t_1 \ldots t_{k_\mathsf{F}} p_1 \ldots p_n = s}\,[\mathsf{F}_l]_\mathbb{X} \qquad \frac{s = d_\mathsf{F}[t_1, \ldots, t_{k_\mathsf{F}}] p_1 \ldots p_n}{s = \mathsf{F} t_1 \ldots t_{k_\mathsf{F}} p_1 \ldots p_n}\,[\mathsf{F}_r]_\mathbb{X}$$