Proof Systems for Retracts in Simply Typed Lambda Calculus

Colin Stirling
cps@inf.ed.ac.uk

LFCS
School of Informatics
University of Edinburgh

ICALP, Riga, 12th July 2013

Simply typed λ-calculus

- types $\sigma ::= b \mid \sigma \to \sigma \qquad b \in B$ base types

Simply typed λ-calculus

- types $\sigma ::= b \mid \sigma \to \sigma$ $\quad b \in B$ base types
  - If $\sigma \notin B$ then has form $\sigma_1 \to \ldots \to \sigma_n \to a$

Simply typed λ-calculus

- types $\sigma ::= b \mid \sigma \to \sigma$ $\quad b \in B$ base types
  - If $\sigma \notin B$ then has form $\sigma_1 \to \ldots \to \sigma_n \to a$
- terms (typed)
  - Variables x have a unique type (Church style) x^σ

Simply typed λ-calculus

- types $\sigma ::= b \mid \sigma \to \sigma$ $b \in B$ base types
  - If $\sigma \notin B$ then has form $\sigma_1 \to \ldots \to \sigma_n \to a$
- terms (typed)
  - Variables x have a unique type (Church style) x^σ
  - if x^σ then $x : \sigma$
  - if $M : \sigma$ and $x : \rho$ then $\lambda x.M : \rho \to \sigma$
  - if $M : \rho \to \sigma$ and $N : \rho$ then $(MN) : \sigma$

Simply typed λ-calculus

- types $\sigma ::= b \mid \sigma \to \sigma \qquad b \in B$ base types
  - If $\sigma \notin B$ then has form $\sigma_1 \to \ldots \to \sigma_n \to a$
- terms (typed)
  - Variables x have a unique type (Church style) x^σ
  - if x^σ then $x : \sigma$
  - if $M : \sigma$ and $x : \rho$ then $\lambda x.M : \rho \to \sigma$
  - if $M : \rho \to \sigma$ and $N : \rho$ then $(MN) : \sigma$
- closed $M : \sigma$ no free variables
- $M, M' : \sigma$ are α-equivalent renamings of each other

Dynamics: reduction

$$(\beta) \quad (\lambda x.M)N \to_\beta M\{N/x\} \qquad \{\cdot/\cdot\} \text{ Substitution}$$

$$(\eta) \quad \lambda x.(Mx) \to_\eta M \qquad x \text{ not free in } M$$

Dynamics: reduction

$$(\beta) \quad (\lambda x.M)N \rightarrow_\beta M\{N/x\} \qquad \{\cdot/\cdot\} \text{ Substitution}$$

$$(\eta) \quad \lambda x.(Mx) \rightarrow_\eta M \qquad x \text{ not free in } M$$

- Facts Strong normalisation and confluence (of $\rightarrow_\beta$, $\rightarrow_\eta$, $\rightarrow_{\beta\eta}$)
- M in β-normal form if no N such that $M \rightarrow_\beta N$

Dynamics: reduction

$$(\beta) \quad (\lambda x.M)N \rightarrow_\beta M\{N/x\} \qquad \{\cdot/\cdot\} \text{ Substitution}$$
$$(\eta) \quad \lambda x.(Mx) \rightarrow_\eta M \qquad x \text{ not free in } M$$

- Facts Strong normalisation and confluence (of $\rightarrow_\beta$, $\rightarrow_\eta$, $\rightarrow_{\beta\eta}$)
- M in β-normal form if no N such that $M \rightarrow_\beta N$
- Equivalence: $M =_{\beta\eta} M'$ if there are N, N'

$$\begin{array}{ccc} M & \rightarrow^*_{\beta\eta} & N \\ & & \shortparallel_\alpha \\ M' & \rightarrow^*_{\beta\eta} & N' \end{array}$$

Dynamics: reduction

$$
\begin{array}{lll}
(\beta) & (\lambda x.M)N \to_\beta M\{N/x\} & \{\cdot/\cdot\} \text{ Substitution} \\
(\eta) & \lambda x.(Mx) \to_\eta M & x \text{ not free in } M
\end{array}
$$

- Facts Strong normalisation and confluence (of $\to_\beta$, $\to_\eta$, $\to_{\beta\eta}$)
- M in β-normal form if no N such that $M \to_\beta N$
- Equivalence: $M =_{\beta\eta} M'$ if there are N, N'

$$
\begin{array}{ccc}
M & \to^*_{\beta\eta} & N \\
 & & \shortparallel_\alpha \\
M' & \to^*_{\beta\eta} & N'
\end{array}
$$

- β-equivalence: $M =_\beta M'$ similar; $\to^*_\beta$ replaces $\to^*_{\beta\eta}$

Retracts in simply typed lambda calculus

ρ is a retract of τ, if there are terms $C : \rho \to \tau$ and $D : \tau \to \rho$ such that $D(C(x^\rho)) =_{\beta\eta} x$

DECISION PROBLEM: given ρ, τ, is ρ a retract of τ ?

Retracts in simply typed lambda calculus

ρ is a retract of τ, if there are terms $C : \rho \to \tau$ and $D : \tau \to \rho$ such that $D(C(x^\rho)) =_{\beta\eta} x$

DECISION PROBLEM: given ρ, τ, is ρ a retract of τ ?

- Bruce and Longo [STOC 85] solve problem for $=_\beta$ by providing a simple proof system. Much harder for $=_{\beta\eta}$

Retracts in simply typed lambda calculus

ρ is a retract of τ, if there are terms $C : \rho \rightarrow \tau$ and $D : \tau \rightarrow \rho$ such that $D(C(x^\rho)) =_{\beta\eta} x$

DECISION PROBLEM: given ρ, τ, is ρ a retract of τ ?

- Bruce and Longo [STOC 85] solve problem for $=_\beta$ by providing a simple proof system. Much harder for $=_{\beta\eta}$
- De Liguro, Piperno and Statman [LICS 92] solve affine case for $=_{\beta\eta}$ when B is a singleton by providing a proof system
- Generalised by Regnier and Urzyczyn [2002] to arbitrary B: proof system for affine case provides NP decision procedure

Retracts in simply typed lambda calculus

ρ is a retract of τ, if there are terms $C : \rho \to \tau$ and $D : \tau \to \rho$ such that $D(C(x^\rho)) =_{\beta\eta} x$

DECISION PROBLEM: given ρ, τ, is ρ a retract of τ ?

- Bruce and Longo [STOC 85] solve problem for $=_\beta$ by providing a simple proof system. Much harder for $=_{\beta\eta}$
- De Liguro, Piperno and Statman [LICS 92] solve affine case for $=_{\beta\eta}$ when B is a singleton by providing a proof system
- Generalised by Regnier and Urzyczyn [2002] to arbitrary B: proof system for affine case provides NP decision procedure
- Padovani [TLCA 01] shows decidability for general case when B is a singleton (no complexity bound)

Retracts in simply typed lambda calculus

ρ is a retract of τ, if there are terms $C : \rho \to \tau$ and $D : \tau \to \rho$ such that $D(C(x^\rho)) =_{\beta\eta} x$

DECISION PROBLEM: given ρ, τ, is ρ a retract of τ ?

- Bruce and Longo [STOC 85] solve problem for $=_\beta$ by providing a simple proof system. Much harder for $=_{\beta\eta}$
- De Liguro, Piperno and Statman [LICS 92] solve affine case for $=_{\beta\eta}$ when B is a singleton by providing a proof system
- Generalised by Regnier and Urzyczyn [2002] to arbitrary B: proof system for affine case provides NP decision procedure
- Padovani [TLCA 01] shows decidability for general case when B is a singleton (no complexity bound)
- Decidability of general case follows from decidability of higher-order matching (Stirling [ICALP 06, LMCS 09]); non-elementary complexity bound

Retracts in simply typed lambda calculus

ρ is a retract of τ, if there are terms $C : \rho \to \tau$ and $D : \tau \to \rho$ such that $D(C(x^\rho)) =_{\beta\eta} x$

DECISION PROBLEM: given ρ, τ, is ρ a retract of τ ?

- Bruce and Longo [STOC 85] solve problem for $=_\beta$ by providing a simple proof system. Much harder for $=_{\beta\eta}$
- De Liguro, Piperno and Statman [LICS 92] solve affine case for $=_{\beta\eta}$ when B is a singleton by providing a proof system
- Generalised by Regnier and Urzyczyn [2002] to arbitrary B: proof system for affine case provides NP decision procedure
- Padovani [TLCA 01] shows decidability for general case when B is a singleton (no complexity bound)
- Decidability of general case follows from decidability of higher-order matching (Stirling [ICALP 06, LMCS 09]); non-elementary complexity bound
- Here: proof system for general case with EXPSPACE upper bound

Some known properties: [LPS92, RU02]

Let $\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

- Retracts must share "target" type a. So, ρ retract of τ if there is $D : \tau \to \rho$ and $C : \rho \to \tau$ and instantiating variables

$$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})(C(x^\rho)) =_{\beta\eta} x z_1 \ldots z_l$$

Some known properties: [LPS92, RU02]

Let $\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

- Retracts must share "target" type a. So, ρ retract of τ if there is $D : \tau \to \rho$ and $C : \rho \to \tau$ and instantiating variables

$$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})(C(x^\rho)) =_{\beta\eta} x z_1 \ldots z_l$$

- We can restrict $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$ to $\lambda f^\tau . f\ S_1^{\tau_1} \ldots S_n^{\tau_n}$; variables $f, z_1, \ldots, z_l$ occur only once in $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$

Some known properties: [LPS92, RU02]

Let $\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

- Retracts must share "target" type a. So, ρ retract of τ if there is $D : \tau \to \rho$ and $C : \rho \to \tau$ and instantiating variables

$$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})(C(x^\rho)) =_{\beta\eta} x z_1 \ldots z_l$$

- We can restrict $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$ to $\lambda f^\tau . f\, S_1^{\tau_1} \ldots S_n^{\tau_n}$; variables $f, z_1, \ldots, z_l$ occur only once in $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$
- We can restrict $C(x^\rho)$ to $\lambda y_1^{\tau_1} \ldots y_n^{\tau_n} . H(x T_1^{\rho_1} \ldots T_l^{\rho_l})$ where

Some known properties: [LPS92, RU02]

Let $\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

- Retracts must share "target" type a. So, ρ retract of τ if there is $D : \tau \to \rho$ and $C : \rho \to \tau$ and instantiating variables

$$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})(C(x^\rho)) =_{\beta\eta} x z_1 \ldots z_l$$

- We can restrict $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$ to $\lambda f^\tau . f\ S_1^{\tau_1} \ldots S_n^{\tau_n}$; variables $f, z_1, \ldots, z_l$ occur only once in $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$
- We can restrict $C(x^\rho)$ to $\lambda y_1^{\tau_1} \ldots y_n^{\tau_n} . H(x T_1^{\rho_1} \ldots T_l^{\rho_l})$ where
  - x occurs only once in $C(x^\rho)$

Some known properties: [LPS92, RU02]

Let $\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

- Retracts must share "target" type a. So, ρ retract of τ if there is $D : \tau \to \rho$ and $C : \rho \to \tau$ and instantiating variables

$$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})(C(x^\rho)) =_{\beta\eta} x z_1 \ldots z_l$$

- We can restrict $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$ to $\lambda f^\tau . f\ S_1^{\tau_1} \ldots S_n^{\tau_n}$; variables $f, z_1, \ldots, z_l$ occur only once in $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$
- We can restrict $C(x^\rho)$ to $\lambda y_1^{\tau_1} \ldots y_n^{\tau_n} . H(x T_1^{\rho_1} \ldots T_l^{\rho_l})$ where
  - x occurs only once in $C(x^\rho)$
  - H is ε if ρ and τ are built from a single base type

Some known properties: [LPS92, RU02]

Let $\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

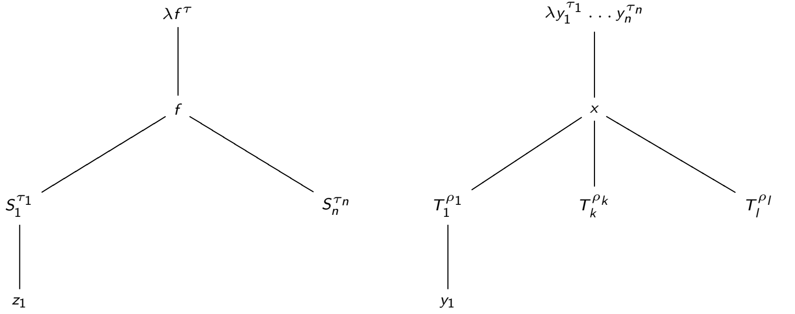
- Retracts must share "target" type a. So, ρ retract of τ if there is $D : \tau \to \rho$ and $C : \rho \to \tau$ and instantiating variables

$$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})(C(x^\rho)) =_{\beta\eta} x z_1 \ldots z_l$$

- We can restrict $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$ to $\lambda f^\tau . f\, S_1^{\tau_1} \ldots S_n^{\tau_n}$; variables $f, z_1, \ldots, z_l$ occur only once in $D(z_1^{\rho_1}, \ldots, z_l^{\rho_l})$
- We can restrict $C(x^\rho)$ to $\lambda y_1^{\tau_1} \ldots y_n^{\tau_n} . H(x T_1^{\rho_1} \ldots T_l^{\rho_l})$ where
  - x occurs only once in $C(x^\rho)$
  - H is ε if ρ and τ are built from a single base type
  - if $T_i^{\rho_i}$ contains an occurrence of y_j then it is the head variable of $T_i^{\rho_i}$, z_i occurs in $S_j^{\tau_j}$ and $T_i^{\rho_i}$ contains no other occurrences of any y_k, $1 \leq k \leq n$

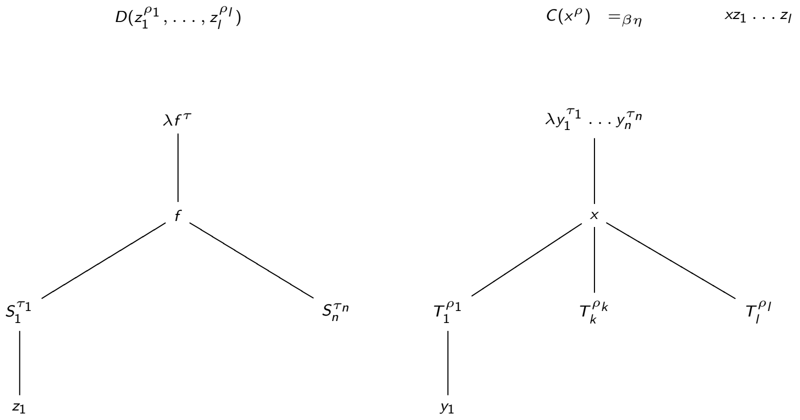
$\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

$$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l}) \qquad C(x^{\rho}) =_{\beta\eta} xz_1 \ldots z_l$$



- Only z_1 occurs in S_1; so y_1 occurs in T_1

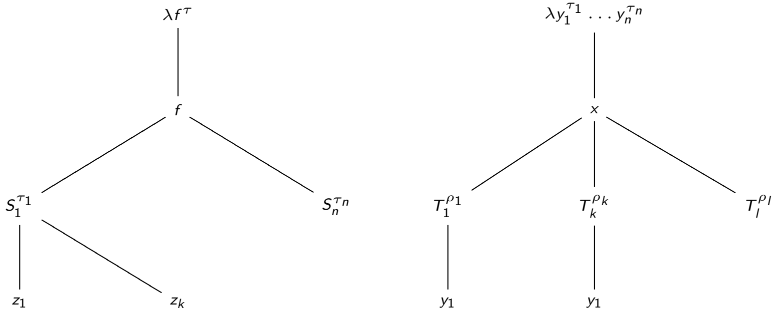
$\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$



- Only z_1 occurs in S_1; so y_1 occurs in T_1
- So, $T_1(\overline{v}_1)(S_1(x_1/z_1)) =_{\beta\eta} x_1\overline{v}_1$; i.e., ρ_1 is retract of τ_1 and $\rho_2 \to \ldots \to \rho_l \to a$ is retract of $\tau_2 \to \ldots \to \tau_n \to a$

$\rho = \rho_1 \rightarrow \ldots \rightarrow \rho_l \rightarrow a$ and $\tau = \tau_1 \rightarrow \ldots \rightarrow \tau_n \rightarrow a$

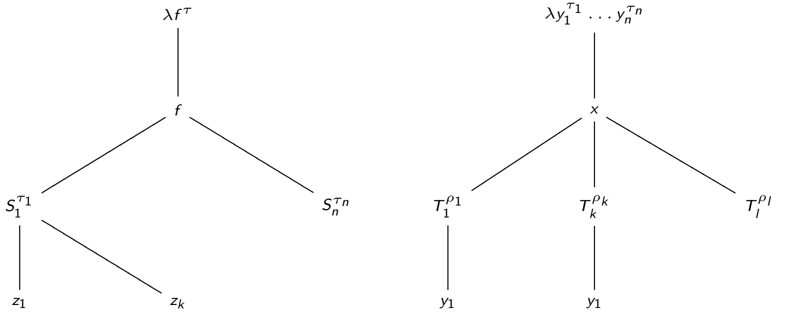
$D(z_1^{\rho 1}, \ldots, z_l^{\rho l})$ $\qquad\qquad$ $C(x^{\rho}) \; =_{\beta\eta} \; x z_1 \ldots z_l$



- Now $z_1, \ldots, z_k$ occur in S_1; so y_1 occurs in $T_1, \ldots, T_k$

$\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

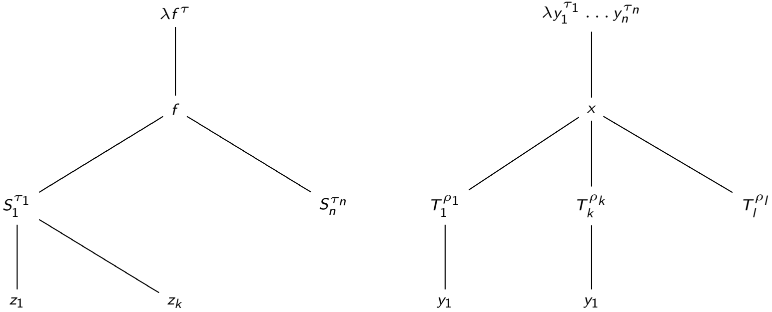
$D(z_1^{\rho 1}, \ldots, z_l^{\rho l}) \qquad C(x^{\rho}) \;=_{\beta\eta}\; x z_1 \ldots z_l$



- Now $z_1, \ldots, z_k$ occur in S_1; so y_1 occurs in $T_1, \ldots, T_k$
- So, one $S_1' = S_1(x_1/z_1, \ldots, x_k/z_k)$ and k terms $T_i(\overline{v}_i)$ with $T_i(\overline{v}_i)(S_1') =_{\beta\eta} x_i \overline{v}_i$;

$\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

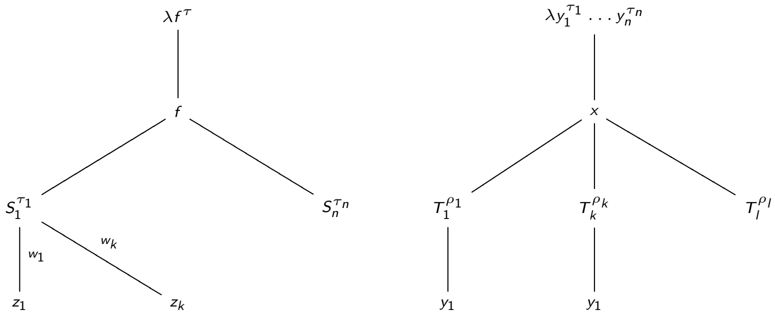
$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l}) \qquad C(x^{\rho}) =_{\beta\eta} x z_1 \ldots z_l$



- Now $z_1, \ldots, z_k$ occur in S_1; so y_1 occurs in $T_1, \ldots, T_k$
- So, one $S_1' = S_1(x_1/z_1, \ldots, x_k/z_k)$ and k terms $T_i(\overline{v}_i)$ with $T_i(\overline{v}_i)(S_1') =_{\beta\eta} x_i \overline{v}_i$;
- May not guarantee ρ_1 is a retract of τ_1 and ... and ρ_k is a retract of τ_1; do have $\rho_{k+1} \to \ldots \to \rho_l \to a$ is retract of $\tau_2 \to \ldots \to \tau_n \to a$

$\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

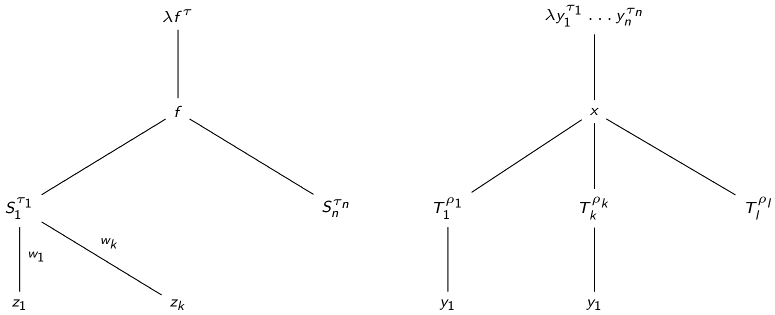
$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l}) \qquad C(x^{\rho}) \quad =_{\beta\eta} \qquad xz_1 \ldots z_l$



- Now $z_1, \ldots, z_k$ occur in S_1; so y_1 occurs in $T_1, \ldots, T_k$
- So, one $S_1' = S_1(x_1/z_1, \ldots, x_k/z_k)$ and k terms $T_i(\overline{v}_i)$ with $T_i(\overline{v}_i)(S_1') =_{\beta\eta} x_i\overline{v}_i$;

$\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

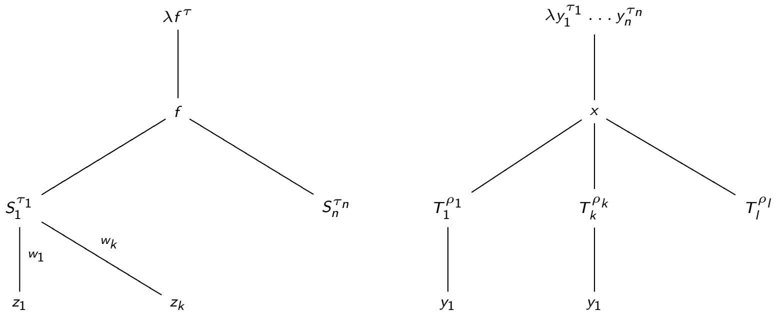
$D(z_1^{\rho_1}, \ldots, z_l^{\rho_l}) \qquad C(x^\rho) \quad =_{\beta\eta} \qquad x z_1 \ldots z_l$



- Now $z_1, \ldots, z_k$ occur in S_1; so y_1 occurs in $T_1, \ldots, T_k$
- So, one $S_1' = S_1(x_1/z_1, \ldots, x_k/z_k)$ and k terms $T_i(\overline{v}_i)$ with $T_i(\overline{v}_i)(S_1') =_{\beta\eta} x_i \overline{v}_i$;
- Must be distinct paths w_i to z_i in S_1; path w_i may "preclude some components of" τ_1.

$\rho = \rho_1 \to \ldots \to \rho_l \to a$ and $\tau = \tau_1 \to \ldots \to \tau_n \to a$

$D(z_1^{\rho 1}, \ldots, z_l^{\rho l}) \qquad C(x^{\rho}) \quad =_{\beta\eta} \qquad x z_1 \ldots z_l$



- Now $z_1, \ldots, z_k$ occur in S_1; so y_1 occurs in $T_1, \ldots, T_k$
- So, one $S_1' = S_1(x_1/z_1, \ldots, x_k/z_k)$ and k terms $T_i(\overline{v}_i)$ with $T_i(\overline{v}_i)(S_1') =_{\beta\eta} x_i \overline{v}_i$;
- Must be distinct paths w_i to z_i in S_1; path w_i may "preclude some components of" τ_1.
- Captured using operator $\tau_1 \upharpoonright w_i$
- Guarantees ρ_i is a retract of $\tau_1 \upharpoonright w_i$

Goal directed proof system (for singleton B)

$$I \quad \rho \trianglelefteq \rho$$

$$W \; \frac{\rho \trianglelefteq \sigma \to \tau}{\rho \trianglelefteq \tau}$$

$$C \; \frac{\delta \to \rho \trianglelefteq \sigma \to \tau}{\delta \trianglelefteq \sigma \qquad \rho \trianglelefteq \tau}$$

$$P_1 \; \frac{\rho_1 \to \ldots \to \rho_k \to \rho \trianglelefteq \sigma \to \tau}{[\rho_1, \ldots, \rho_k] \trianglelefteq \sigma \qquad \rho \trianglelefteq \tau}$$

$$P_2 \; \frac{[\rho_1, \ldots, \rho_k] \trianglelefteq \sigma}{\rho_1 \trianglelefteq \sigma \restriction w_1 \quad \ldots \quad \rho_k \trianglelefteq \sigma \restriction w_k}$$

$w_1 \sqsubset \ldots \sqsubset w_k$ are k-minimal realisable paths of type σ

Example proof tree

$$
\dfrac{\dfrac{(\sigma \to o) \to (\sigma \to o) \to o \trianglelefteq (\sigma \to (o \to o \to o) \to o) \to o}{[\sigma \to o, \sigma \to o] \trianglelefteq \sigma \to (o \to o \to o) \to o} \quad o \trianglelefteq o}{\sigma \to o \trianglelefteq \sigma \to o \quad \sigma \to o \trianglelefteq \sigma \to o}
$$

- Let $\sigma' = \sigma \to (o \to o \to o) \to o$
- There are paths w_1 and w_2 where $\sigma' \upharpoonright w_1 = \sigma \to o = \sigma' \upharpoonright w_2$

Example proof tree

$$\dfrac{\dfrac{(\sigma \to o) \to (\sigma \to o) \to o \trianglelefteq (\sigma \to (o \to o \to o) \to o) \to o}{[\sigma \to o, \sigma \to o] \trianglelefteq \sigma \to (o \to o \to o) \to o} \quad o \trianglelefteq o}{\sigma \to o \trianglelefteq \sigma \to o \quad \sigma \to o \trianglelefteq \sigma \to o}$$

- Let $\sigma' = \sigma \to (o \to o \to o) \to o$
- There are paths w_1 and w_2 where $\sigma' \upharpoonright w_1 = \sigma \to o = \sigma' \upharpoonright w_2$
- In both cases the paths preclude second component of σ'

General case: multiple base types

- Further operation on paths: $w(\sigma)$ subtype after w

General case: multiple base types

- Further operation on paths: $w(\sigma)$ subtype after w
- Proof system just differs in one rule

$$P_2' \; \frac{[\rho_1, \ldots, \rho_k] \trianglelefteq \sigma}{\rho_1 \trianglelefteq v_1(\sigma) \restriction w_1 \qquad \ldots \qquad \rho_k \trianglelefteq v_k(\sigma) \restriction w_k}$$

where the realisable paths are $v_1 w_1, \ldots, v_k w_k$

Soundness and completeness of proof systems

- Define a dialogue game between potential witnesses D and C for a retract

Soundness and completeness of proof systems

- Define a dialogue game between potential witnesses D and C for a retract
- Define a canonical presentation of paths in terms

Soundness and completeness of proof systems

- Define a dialogue game between potential witnesses D and C for a retract
- Define a canonical presentation of paths in terms
- Uniformity poperties of game play underpin combinatorics on paths: notions of realisable (families of) paths, k-minimal paths, . . .

Soundness and completeness of proof systems

- Define a dialogue game between potential witnesses D and C for a retract
- Define a canonical presentation of paths in terms
- Uniformity poperties of game play underpin combinatorics on paths: notions of realisable (families of) paths, k-minimal paths, ...
- Main proofs of soundness and completeness are then inductive: see full version

Complexity of decision procedure

- Each subgoal of a proof rule has smaller size than goal

Complexity of decision procedure

- Each subgoal of a proof rule has smaller size than goal
- A proof witness can then be presented in PSPACE

Complexity of decision procedure

- Each subgoal of a proof rule has smaller size than goal
- A proof witness can then be presented in PSPACE
- BUT checking that a subgoal obeys the side conditions in the case of the product rules requires exponential space

Complexity of decision procedure

- Each subgoal of a proof rule has smaller size than goal
- A proof witness can then be presented in PSPACE
- BUT checking that a subgoal obeys the side conditions in the case of the product rules requires exponential space
- So, EXPSPACE decision procedure

Complexity of decision procedure

- Each subgoal of a proof rule has smaller size than goal
- A proof witness can then be presented in PSPACE
- BUT checking that a subgoal obeys the side conditions in the case of the product rules requires exponential space
- So, EXPSPACE decision procedure
- Can this be reduced to PSPACE?

Examples of when ρ is retract of τ

1. $\rho = (\sigma \to o) \to (\sigma \to o) \to o$
   $\tau = (\sigma \to (o \to o \to o) \to o) \to o$ where σ is arbitrary

Examples of when ρ is retract of τ

1. $\rho = (\sigma \to o) \to (\sigma \to o) \to o$
   $\tau = (\sigma \to (o \to o \to o) \to o) \to o$ where σ is arbitrary
   - Let $\rho_1 = \rho_2 = \sigma \to o$ and let $\tau_1 = \sigma \to (o \to o \to o) \to o$.

Examples of when ρ is retract of τ

1. $\rho = (\sigma \to o) \to (\sigma \to o) \to o$
   $\tau = (\sigma \to (o \to o \to o) \to o) \to o$ where σ is arbitrary
   - Let $\rho_1 = \rho_2 = \sigma \to o$ and let $\tau_1 = \sigma \to (o \to o \to o) \to o$.
   - $D(z_1^{\rho_1}, z_2^{\rho_2})$ is $\lambda f^{\tau}.f(\lambda u^{\sigma} v^{o \to o \to o}.v(z_1 u)(z_2 u))$
   - $C(x^{\rho})$ is $\lambda y^{\tau_1}.x(\lambda w^{\sigma}.yw(\lambda s^o t^o.s))(\lambda w^{\sigma}.yw(\lambda s^o t^o.t))$;

Examples of when ρ is retract of τ

1. $\rho = (\sigma \to o) \to (\sigma \to o) \to o$
   $\tau = (\sigma \to (o \to o \to o) \to o) \to o$ where σ is arbitrary
   - Let $\rho_1 = \rho_2 = \sigma \to o$ and let $\tau_1 = \sigma \to (o \to o \to o) \to o$.
   - $D(z_1^{\rho_1}, z_2^{\rho_2})$ is $\lambda f^{\tau}.f(\lambda u^{\sigma} v^{o \to o \to o}.v(z_1 u)(z_2 u))$
   - $C(x^{\rho})$ is $\lambda y^{\tau_1}.x(\lambda w^{\sigma}.yw(\lambda s^o t^o.s))(\lambda w^{\sigma}.yw(\lambda s^o t^o.t))$;
   - $(D(z_1, z_2))C(x) \to_{\beta}^{*} x(\lambda w^{\sigma}.z_1 w)(\lambda w^{\sigma}.z_2 w) =_{\beta\eta} xz_1 z_2$

Examples of when ρ is retract of τ

1. $\rho = (\sigma \to o) \to (\sigma \to o) \to o$
   $\tau = (\sigma \to (o \to o \to o) \to o) \to o$ where σ is arbitrary
   - Let $\rho_1 = \rho_2 = \sigma \to o$ and let $\tau_1 = \sigma \to (o \to o \to o) \to o$.
   - $D(z_1^{\rho_1}, z_2^{\rho_2})$ is $\lambda f^{\tau}.f(\lambda u^{\sigma} v^{o \to o \to o}.v(z_1 u)(z_2 u))$
   - $C(x^{\rho})$ is $\lambda y^{\tau_1}.x(\lambda w^{\sigma}.yw(\lambda s^o t^o.s))(\lambda w^{\sigma}.yw(\lambda s^o t^o.t))$;
   - $(D(z_1, z_2))C(x) \to_{\beta}^{*} x(\lambda w^{\sigma}.z_1 w)(\lambda w^{\sigma}.z_2 w) =_{\beta\eta} x z_1 z_2$
2. $\rho = (b \to a) \to a \to a$
   $\tau = (b \to (a \to o \to a) \to a) \to a$

Examples of when ρ is retract of τ

1. $\rho = (\sigma \to o) \to (\sigma \to o) \to o$
   $\tau = (\sigma \to (o \to o \to o) \to o) \to o$ where σ is arbitrary
   - Let $\rho_1 = \rho_2 = \sigma \to o$ and let $\tau_1 = \sigma \to (o \to o \to o) \to o$.
   - $D(z_1^{\rho_1}, z_2^{\rho_2})$ is $\lambda f^{\tau}.f(\lambda u^{\sigma} v^{o \to o \to o}.v(z_1 u)(z_2 u))$
   - $C(x^{\rho})$ is $\lambda y^{\tau_1}.x(\lambda w^{\sigma}.yw(\lambda s^o t^o.s))(\lambda w^{\sigma}.yw(\lambda s^o t^o.t))$;
   - $(D(z_1, z_2))C(x) \to_{\beta}^{*} x(\lambda w^{\sigma}.z_1 w)(\lambda w^{\sigma}.z_2 w) =_{\beta\eta} xz_1 z_2$
2. $\rho = (b \to a) \to a \to a$
   $\tau = (b \to (a \to o \to a) \to a) \to a$
   - let $\rho_1 = b \to a$, $\rho_2 = a$ and $\tau_1 = b \to (a \to o \to a) \to a$

Examples of when ρ is retract of τ

1. $\rho = (\sigma \to o) \to (\sigma \to o) \to o$
   $\tau = (\sigma \to (o \to o \to o) \to o) \to o$ where σ is arbitrary
   - Let $\rho_1 = \rho_2 = \sigma \to o$ and let $\tau_1 = \sigma \to (o \to o \to o) \to o$.
   - $D(z_1^{\rho_1}, z_2^{\rho_2})$ is $\lambda f^\tau . f(\lambda u^\sigma v^{o \to o \to o} . v(z_1 u)(z_2 u))$
   - $C(x^\rho)$ is $\lambda y^{\tau_1} . x(\lambda w^\sigma . yw(\lambda s^o t^o . s))(\lambda w^\sigma . yw(\lambda s^o t^o . t))$;
   - $(D(z_1, z_2))C(x) \to_\beta^* x(\lambda w^\sigma . z_1 w)(\lambda w^\sigma . z_2 w) =_{\beta\eta} x z_1 z_2$
2. $\rho = (b \to a) \to a \to a$
   $\tau = (b \to (a \to o \to a) \to a) \to a$
   - let $\rho_1 = b \to a$, $\rho_2 = a$ and $\tau_1 = b \to (a \to o \to a) \to a$
   - $D(z_1^{\rho_1}, z_2^{\rho_2})$ is $\lambda f^\tau . f(\lambda u_1^b u_2^{a \to o \to a} . u_2(z_1 u_1) z_2)$
   - $C(x^\rho)$ is $\lambda y^{\tau_1} . y s^b (\lambda w_1^a w_2^o . x(\lambda v^b . yv(\lambda w_1^a w_2^o . w_1)) w_2)$;

Examples of when ρ is retract of τ

1. $\rho = (\sigma \to o) \to (\sigma \to o) \to o$
   $\tau = (\sigma \to (o \to o \to o) \to o) \to o$ where σ is arbitrary
   - Let $\rho_1 = \rho_2 = \sigma \to o$ and let $\tau_1 = \sigma \to (o \to o \to o) \to o$.
   - $D(z_1^{\rho_1}, z_2^{\rho_2})$ is $\lambda f^{\tau}.f(\lambda u^{\sigma} v^{o \to o \to o}.v(z_1 u)(z_2 u))$
   - $C(x^{\rho})$ is $\lambda y^{\tau_1}.x(\lambda w^{\sigma}.yw(\lambda s^o t^o.s))(\lambda w^{\sigma}.yw(\lambda s^o t^o.t))$;
   - $(D(z_1, z_2))C(x) \to_{\beta}^{*} x(\lambda w^{\sigma}.z_1 w)(\lambda w^{\sigma}.z_2 w) =_{\beta\eta} xz_1 z_2$
2. $\rho = (b \to a) \to a \to a$
   $\tau = (b \to (a \to o \to a) \to a) \to a$
   - let $\rho_1 = b \to a$, $\rho_2 = a$ and $\tau_1 = b \to (a \to o \to a) \to a$
   - $D(z_1^{\rho_1}, z_2^{\rho_2})$ is $\lambda f^{\tau}.f(\lambda u_1^b u_2^{a \to o \to a}.u_2(z_1 u_1) z_2)$
   - $C(x^{\rho})$ is $\lambda y^{\tau_1}.ys^b(\lambda w_1^a w_2^o.x(\lambda v^b.yv(\lambda w_1^a w_2^o.w_1))w_2)$;
   - $(D(z_1, z_2))C(x) \to_{\beta}^{*} x(\lambda v^b.z_1 v)z_2 =_{\beta\eta} xz_1 z_2$