Hereditary Substitution for the $\lambda\Delta$-Calculus

Harley Eades and Aaron Stump
Computer Science

CL&C 2012

Overview

- The $\lambda\Delta$-Calculus
- Hereditary Substitution
- The problem with defining the hereditary substitution function for the $\lambda\Delta$-calculus
- How we solve this problem
- Properties of the Hereditary Substitution Function
- Concluding Normalization

The λΔ-Calculus

- A type theory correspoding to classical natural deduction.
- Originally defined by J. Rehof and M. Sørensen in 1994.
- Provably equalivalent to M. Parigot's $\lambda\mu$-Calculus.
- The bases of classical pure type systems (G. Barthe, J. Hatcliff, M. Sørensen 1997).

The $\lambda\Delta$-Calculus

- Syntax:

$$\begin{array}{rcl} T, A, B, C & ::= & \perp \mid b \mid A \to B \\ t & ::= & x \mid \lambda x : T.t \mid \Delta x : T.t \mid t_1\, t_2 \\ n, m & ::= & \lambda x : T.n \mid \Delta x : T.n \mid h \\ h & ::= & x \mid h\, n \end{array}$$

  We denote the set of all terms $\mathcal{T}$ and the set of all types Ψ.

- Reduction:

$$\frac{}{(\lambda x : T.t)\, t' \rightsquigarrow [t'/x]t}\ \textsc{Beta}$$

$$\frac{\begin{array}{c} y \text{ fresh in } t \text{ and } t' \\ z \text{ fresh in } t \text{ and } t' \end{array}}{(\Delta x : \neg(T_1 \to T_2).t)\, t' \rightsquigarrow \Delta y : \neg T_2.[\lambda z : T_1 \to T_2.(y\,(z\, t'))/x]t}\ \textsc{StructRed}$$

The $\lambda\Delta$-Calculus

- Typing Rules:

$$\frac{}{\Gamma, x : A \vdash x : A}\ \textsc{Ax} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A.t : A \to B}\ \textsc{Lam}$$

$$\frac{\begin{array}{c}\Gamma \vdash t_2 : A \\ \Gamma \vdash t_1 : A \to B\end{array}}{\Gamma \vdash t_1\ t_2 : B}\ \textsc{App} \qquad \frac{\Gamma, x : \neg A \vdash t : \perp}{\Gamma \vdash \Delta x : \neg A.t : A}\ \textsc{Delta}$$

Hereditary Substitution

- Syntax: $[t/x]^A t' = t''$.
- Usual termination order: (A, t').

- Like ordinary capture avoiding substitution.
- Except, if the substitution introduces a redex, then that redex is recursively reduced.
  - Example: $[\lambda z : \mathsf{b}.z/x]^{\mathsf{b}\to\mathsf{b}}(x\,y)(\approx ((\lambda z : \mathsf{b}.z)\,y \approx [y/z]^{\mathsf{b}}z) = y$.

- The constructive content of normalization proofs dating all the way back to Prawitz (1965).
- First made explicit by K. Watkins for simple types and R. Adams for dependent types.

An Intuition of the Problems Involved

- Recall how hereditary substitution works for β-reduction:
  $[\lambda z : \mathsf{b}.z/x]^{\mathsf{b}\to\mathsf{b}}(x\,y)(\approx ((\lambda z : \mathsf{b}.z)\,y \approx [y/z]^{\mathsf{b}} z) = y$

An Intuition of the Problems Involved

- Recall how hereditary substitution works for β-reduction:
  $[\lambda z : \mathsf{b}.z/x]^{\mathsf{b}\to\mathsf{b}}(x\,y)(\approx ((\lambda z : \mathsf{b}.z)\,y \approx [y/z]^{\mathsf{b}} z) = y$

- The naive solution for structural reduction:
  $[\Delta x : \neg(A'' \to A').(x\,q)/z]^{(A''\to A')}(z\,r) = \Delta y : \neg A'.[(\lambda u : A'' \to A'.(y\,(u\,r)))/x]^{\neg(A''\to A')}(x\,q)$

An Intuition of the Problems Involved

- Recall how hereditary substitution works for β-reduction:
  $[\lambda z : \mathsf{b}.z/x]^{\mathsf{b}\to\mathsf{b}}(x\,y)(\approx ((\lambda z : \mathsf{b}.z)\,y \approx [y/z]^{\mathsf{b}} z) = y$
- The naive solution for structural reduction:
  $[\Delta x : \neg(A'' \to A').(x\,q)/z]^{(A'' \to A')}(z\,r) = \Delta y : \neg A'.[(\lambda u : A'' \to A'.(y\,(u\,r)))/x]^{\neg(A'' \to A')}(x\,q)$
  - The cut type actually increased!
- The problem: The usual termination order (A, t') no longer works.
  - How do we fix this?

A Look at Structural Reduction

Consider: $((\Delta x : \neg(A'' \to A').t)\, t') \rightsquigarrow \Delta y : \neg A'.[(\lambda u : A'' \to A'.(y\,(u\,t')))/x]t$

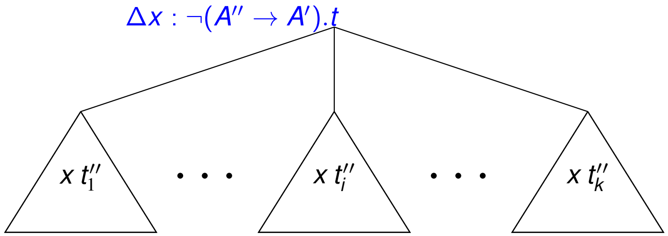
A Look at Structural Reduction

Consider: $((\Delta x : \neg(A'' \to A').t)\, t') \rightsquigarrow \Delta y : \neg A'.[(\lambda u : A'' \to A'.(y\,(u\,t')))/x]t$

A Look at Structural Reduction

Consider: $((\Delta x : \neg(A'' \to A').t)\ t') \rightsquigarrow \Delta y : \neg A'.[(\lambda u : A'' \to A'.(y\,(u\,t')))/x]t$

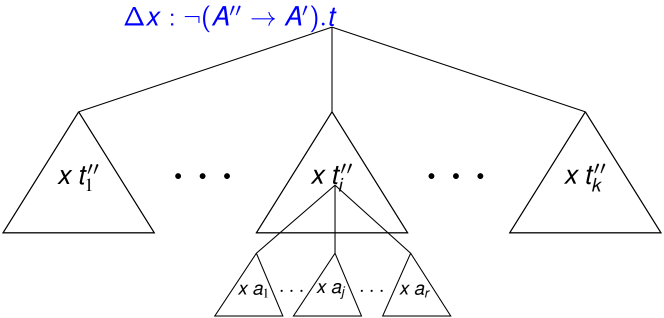
When redexes are created:



A Look at Structural Reduction

Consider: $((\Delta x : \neg(A'' \to A').t)\ t') \rightsquigarrow \Delta y : \neg A'.[(\lambda u : A'' \to A'.(y\,(u\,t')))/x]t$

When redexes are created:



Is Further Reduction the Answer?

- Consider the previous example:
  $[\Delta x : \neg(A'' \to A').(x\,q)/z]^{(A'' \to A')}(z\,r) = \Delta y : \neg A'.[(\lambda u : A'' \to A'.(y\,(u\,r)))/x]^{\neg(A'' \to A')}(x\,q)$
- Recursively reducing the redexes introduced by substituting the linear λ-abstraction:
  $[\Delta x : \neg(A'' \to A').(x\,q)/z]^{(A'' \to A')}(z\,r) = \Delta y : \neg A'.(y\,[q/u]^{(A'' \to A')}(u\,r))$

Is Further Reduction the Answer?

- Consider the previous example:
  $[\Delta x : \neg(A'' \to A').(x\, q)/z]^{(A'' \to A')}(z\, r) = \Delta y : \neg A'.[(\lambda u : A'' \to A'.(y\,(u\, r)))/x]^{\neg(A'' \to A')}(x\, q)$
- Recursively reducing the redexes introduced by substituting the linear λ-abstraction:
  $[\Delta x : \neg(A'' \to A').(x\, q)/z]^{(A'' \to A')}(z\, r) = \Delta y : \neg A'.(y\,[q/u]^{(A'' \to A')}(u\, r))$
  - The cut type stayed the same.
  - But the term we are substituting has decreased.
  - Is this always the case? Basically, it is!

The Final Solution

- The term we are substituting either decrease structurally or decreases contextually.
  - Structural decrease: $\forall t, t'.t < t'$ if t' is a strict subexpression of t.
  - Contextual decrease: A term is considered larger than itself with a hole.
    - $\forall \mathcal{C}, t.C < t$ if $\exists s.\mathcal{C}[s] \equiv t$.

- Using this insight the hereditary substitution function is defineable using the ordering (A, t, t').

Hereditary Substitution

$[t/x]^A\Box = \Box$

$[t/x]^A x = t$

$[t/x]^A y = y$
Where y is a variable distinct from x.

Type: $\mathcal{T} \cup \mathcal{E} \to \mathcal{T} \to \Psi \to \mathcal{T} \cup \mathcal{E} \to \mathcal{T} \cup \mathcal{E}$
Total using the ordering: (A, t, t')

$[t/x]^A(\lambda y : A'.t') = \lambda y : A'.([t/x]^A t')$
Where $\mathrm{FV}(t) \cap \mathrm{FV}(t') = \emptyset$.

$[t/x]^A(\Delta y : A'.t') = \Delta y : A'.([t/x]^A t')$
Where $\mathrm{FV}(t) \cap \mathrm{FV}(t') = \emptyset$.

$[t/x]^A(t_1\ t_2) = ([t/x]^A t_1)\,([t/x]^A t_2)$
Where $([t/x]^A t_1)$ is not a λ-abstraction or Δ-abstraction, or both $([t/x]^A t_1)$ and t_1 are λ-abstractions or Δ-abstractions, or $\mathrm{ctype}_A(x, t_1)$ is undefined.

$[t/x]^A(t_1\ t_2) = [s_2'/y]^{A''} s_1'$
Where $([t/x]^A t_1) = \lambda y : A''.s_1'$ for some y, s_1' and A'',
$[t/x]^A t_2 = s_2'$, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

$[t/x]^A(t_1\ t_2) = \Delta z : \neg A'.([\lambda u : A'' \to A'.(z\,(u\,s_2))/y]s_1)$
Where $([t/x]^A t_1) = \Delta y : \neg(A'' \to A').s_1$ for some y, s_1, A'', and there does not exists any context of s_1 equal to $\mathcal{C}[y\ s_1']$ for some term s_1', $([t/x]^A t_2) = s_2$ for some s_2, z and u are fresh variables of type A' and $A'' \to A'$ respectively, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

$[t/x]^A(t_1\ t_2) = \Delta z : \neg A'.[\lambda u : A'' \to A'.(z\,(u\,s_2))/y](\text{fill}\,\mathcal{C}[\overrightarrow{\Box_i}]\,\overrightarrow{\mathcal{C}[z\,([s_1/q]^{A'' \to A'}(q\,s_2))]})$
Where $([t/x]^A t_1) = \Delta y : \neg(A'' \to A').\mathcal{C}[\overrightarrow{(y\,s_1)_i}]$ for some i, y, s_1 and A'',
$([t/x]^A t_2) = s_2$ for some s_2, z and r are fresh variables of type A' and A'' respectively, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

Hereditary Substitution

Type: $\mathcal{T} \cup \mathcal{E} \to \mathcal{T} \to \Psi \to \mathcal{T} \cup \mathcal{E} \to \mathcal{T} \cup \mathcal{E}$
Total using the ordering: (A, t, t')

$[t/x]^A \Box = \Box$

$[t/x]^A x = t$

$[t/x]^A y = y$
Where y is a variable distinct from x.

$[t/x]^A (\lambda y : A'.t') = \lambda y : A'.([t/x]^A t')$
Where $\mathrm{FV}(t) \cap \mathrm{FV}(t') = \emptyset$.

$[t/x]^A (\Delta y : A'.t') = \Delta y : A'.([t/x]^A t')$
Where $\mathrm{FV}(t) \cap \mathrm{FV}(t') = \emptyset$.

$[t/x]^A (t_1\ t_2) = ([t/x]^A t_1)\ ([t/x]^A t_2)$
Where $([t/x]^A t_1)$ is not a λ-abstraction or Δ-abstraction, or both $([t/x]^A t_1)$ and t_1 are λ-abstractions or Δ-abstractions, or $\mathrm{ctype}_A(x, t_1)$ is undefined.

$[t/x]^A (t_1\ t_2) = [s_2'/y]^{A''} s_1'$
Where $([t/x]^A t_1) = \lambda y : A''.s_1'$ for some y, s_1' and A'', $[t/x]^A t_2 = s_2'$, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

$[t/x]^A (t_1\ t_2) = \Delta z : \neg A'.([\lambda u : A'' \to A'.(z\,(u\,s_2))/y] s_1)$
Where $([t/x]^A t_1) = \Delta y : \neg(A'' \to A').s_1$ for some y, s_1, A'', and there does not exists any context of s_1 equal to $\mathcal{C}[y\ s_1']$ for some term s_1', $([t/x]^A t_2) = s_2$ for some s_2, z and u are fresh variables of type A' and $A'' \to A'$ respectively, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

$[t/x]^A (t_1\ t_2) = \Delta z : \neg A'.[\lambda u : A'' \to A'.(z\,(u\,s_2))/y](\text{fill}\ \mathcal{C}[\overrightarrow{\Box_i}]\ \overrightarrow{\mathcal{C}[z\,([s_1/q]^{A'' \to A'}(q\,s_2))]})$
Where $([t/x]^A t_1) = \Delta y : \neg(A'' \to A').\mathcal{C}[\overrightarrow{(y\ s_1)_i}]$ for some i, y, s_1 and A'', $([t/x]^A t_2) = s_2$ for some s_2, z and r are fresh variables of type A' and A'' respectively, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

Hereditary Substitution

Type: $\mathcal{T} \cup \mathcal{E} \to \mathcal{T} \to \Psi \to \mathcal{T} \cup \mathcal{E} \to \mathcal{T} \cup \mathcal{E}$
Total using the ordering: (A, t, t')

$[t/x]^A \Box = \Box$

$[t/x]^A x = t$

$[t/x]^A y = y$
Where y is a variable distinct from x.

$[t/x]^A (\lambda y : A'.t') = \lambda y : A'.([t/x]^A t')$
Where $\mathsf{FV}(t) \cap \mathsf{FV}(t') = \emptyset$.

$[t/x]^A (\Delta y : A'.t') = \Delta y : A'.([t/x]^A t')$
Where $\mathsf{FV}(t) \cap \mathsf{FV}(t') = \emptyset$.

$[t/x]^A (t_1\ t_2) = ([t/x]^A t_1)\ ([t/x]^A t_2)$
Where $([t/x]^A t_1)$ is not a λ-abstraction or Δ-abstraction, or both $([t/x]^A t_1)$ and t_1 are λ-abstractions or Δ-abstractions, or $\mathrm{ctype}_A(x, t_1)$ is undefined.

$[t/x]^A (t_1\ t_2) = [s_2'/y]^{A''} s_1'$
Where $([t/x]^A t_1) = \lambda y : A''.s_1'$ for some y, s_1' and A'', $[t/x]^A t_2 = s_2'$, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

$[t/x]^A (t_1\ t_2) = \Delta z : \neg A'.([\lambda u : A'' \to A'.(z\,(u\,s_2))/y] s_1)$
Where $([t/x]^A t_1) = \Delta y : \neg(A'' \to A').s_1$ for some y, s_1, A'', and there does not exists any context of s_1 equal to $\mathcal{C}[y\ s_1']$ for some term s_1', $([t/x]^A t_2) = s_2$ for some s_2, z and u are fresh variables of type A' and $A'' \to A'$ respectively, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

$[t/x]^A (t_1\ t_2) = \Delta z : \neg A'.[\lambda u : A'' \to A'.(z\,(u\,s_2))/y](\text{fill}\ \mathcal{C}[\overrightarrow{\Box_i}]\ \overrightarrow{\mathcal{C}[z\,([s_1/q]^{A'' \to A'}(q\,s_2))]})$
Where $([t/x]^A t_1) = \Delta y : \neg(A'' \to A').\mathcal{C}[\overrightarrow{(y\ s_1)_i}]$ for some i, y, s_1 and A'', $([t/x]^A t_2) = s_2$ for some s_2, z and r are fresh variables of type A' and A'' respectively, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

Hereditary Substitution: Handling Structural Reduction

- Case when no further redexes are created:

$[t/x]^A(t_1\ t_2) = \Delta z : \neg A'.([\lambda u : A'' \to A'.(y\,(u\,s_2))/y]s_1)$
Where $([t/x]^A t_1) = \Delta y : \neg(A'' \to A').s_1$ for some y, s_1, A'', and there does not exists any context of s_1 equal to $\mathcal{C}[y\ s_1']$ for some term s_1', $([t/x]^A t_2) = s_2$ for some s_2, z and u are fresh variables of type A' and $A'' \to A'$ respectively, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

- Case when no further redexes are created:

  $[t/x]^A(t_1\ t_2) = \Delta z : \neg A'.([\lambda u : A'' \to A'.(y\,(u\,s_2))/y]s_1)$
  Where $([t/x]^A t_1) = \Delta y : \neg(A'' \to A').s_1$ for some y, s_1, A'', and there does not exists any context of s_1 equal to $\mathcal{C}[y\ s_1']$ for some term s_1' , $([t/x]^A t_2) = s_2$ for some s_2, z and u are fresh variables of type A' and $A'' \to A'$ respectively, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.

- Case when structural reduction will introduce more redexes:

  $[t/x]^A(t_1\ t_2) = \Delta z : \neg A'.[\lambda u : A'' \to A'.(z\,(u\,s_2))/y](\mathrm{fill}\,\mathcal{C}[\overrightarrow{\Box_i}]\ \overrightarrow{\mathcal{C}[z\,([s_1/q]^{A''\to A'}(q\,s_2))]})$
  Where $([t/x]^A t_1) = \Delta y : \neg(A'' \to A').\mathcal{C}[\overrightarrow{(y\,s_1)_i}]$ for some i, y, s_1 and A'', $([t/x]^A t_2) = s_2$ for some s_2, z and r are fresh variables of type A' and A'' respectively, and $\mathrm{ctype}_A(x, t_1) = A'' \to A'$.
  - Do not substitute the linear lambda-abstractions, but reduce them right away.
  - $\overrightarrow{\mathcal{C}[t]}$: Expands the context into a list of lists of subcontexts.
  - If $A \equiv A'' \to A'$ then we know $t_1 \equiv x$ and $t \equiv \Delta y : \neg(A'' \to A').\mathcal{C}[\overrightarrow{(y\,s_1)_i}]$.
    - Hence $s_1 < t$.

Properties of Hereditary Substitution

Lemma (No Holes)

If $\Gamma \vdash t : A$*,* $\Gamma, x : A, \Gamma' \vdash t' : B$ *and* $[t/x]^A t'$ *is defined then* $[t/x]^A t'$ *has no holes.*

Lemma (Totality and Type Preservation)

If $\Gamma \vdash t : A$ *and* $\Gamma, x : A, \Gamma' \vdash t' : B$*, then there exists a term* s *such that* $[t/x]^A t' = s$ *and* $\Gamma, \Gamma' \vdash s : B$*.*

Lemma (Normality Preservation)

If $\Gamma \vdash n : A$ *and* $\Gamma, x : A, \Gamma' \vdash n' : A'$ *then* $[n/x]^A n'$ *is normal.*

Lemma (Soundness with Respect to Reduction)

If $\Gamma \vdash t : A$ *and* $\Gamma, x : A, \Gamma' \vdash t' : B$ *then* $[t/x]t' \rightsquigarrow^* [t/x]^A t'$*.*

Concluding Normalization

Definition

The interpretation of types $[\![T]\!]_\Gamma$ is defined by:

$$n \in [\![T]\!]_\Gamma \quad \iff \quad \Gamma \vdash n : T$$

We extend this definition to non-normal terms t in the following way:

$$t \in [\![T]\!]_\Gamma \quad \iff \quad \exists n.t \rightsquigarrow^! n \in [\![T]\!]_\Gamma$$

Lemma (Hereditary Substitution for the Interpretation of Types)

If $n \in [\![T]\!]_\Gamma$ and $n' \in [\![T']\!]_{\Gamma,x:T,\Gamma'}$, then $[n/x]^T n' \in [\![T']\!]_{\Gamma,\Gamma'}$.

Theorem (Type Soundness)

If $\Gamma \vdash t : T$ then $t \in [\![T]\!]_\Gamma$.

Conclusion

- We defined hereditary substitution function using the ordering (A, t, t').
- It can be used to show normalization of the $\lambda\Delta$-calculus.
- Currently formalizing all of this in the Coq proof assistant.
- Future work:
  - Formulate the canonical predicative classical logical framework.
  - Giving a categorical semantics of hereditary substitution.
    - Potentially usable to define the hereditary substitution function for Girard-Reynolds system F.
  - Formulate the hereditary substitution function for Gödel's system T.

Thank you!

Multi-Holed Contexts

Recall the usual definition of single-hole contexts:

$$\mathcal{C} ::= \Box \mid \lambda x : T.\mathcal{C} \mid \Delta x : T.\mathcal{C} \mid t\,\mathcal{C} \mid \mathcal{C}\,t$$

We extend this definition to multi-holed context as follows:

$$\mathcal{C} ::= \Box_i \mid \lambda x : T.\mathcal{C} \mid \Delta x : T.\mathcal{C} \mid t\,\mathcal{C} \mid \mathcal{C}\,t$$

where $i \in \mathbb{N}$.

Definition (Well-Formed Multi-Holed Context)

A context $\mathcal{C}$ is well formed if $\mathcal{C}$ does not have more than one hole with the same i.

We denote the set of all well-formed contexts as $\mathcal{E}$.

Definition (Context Hole Filling)

If $\mathcal{C}$ is a well-formed context with i holes then $\mathcal{C}[\vec{t}_i] = \mathcal{C}[t_1, \ldots, t_i]$, where t_i fills $\Box_i$.

Hereditary Substitution

Definition (Well-founded ordering on types)

We define an ordering on types T as the compatible closure of the following formulas.

$$\begin{array}{lcl} T_1 \to T_2 & > & T_1 \\ T_1 \to T_2 & > & T_2 \end{array}$$

Absurdity and base types are minimal elements.

We denote the reflexive-transitive closure of $>$ as $\geq$.

Hereditary Substitution

Definition

We define the partial function ctype : $\Psi \to \mathcal{T} \to \mathcal{T} \to \mathcal{T}$ which computes the type of an application in head normal form. It is defined as follows:

$$\begin{array}{l} \mathrm{ctype}_T(x, x) = T \\ \mathrm{ctype}_T(x, t_1\ t_2) = T'' \\ \quad \text{Where } \mathrm{ctype}_T(x, t_1) = T' \to T''. \end{array}$$

Lemma (Properties of ctype*)*

i. *If* $\mathrm{ctype}_T(x, t) = T'$ *then* $head(t) = x$ *and* $T' \leq T$.

ii. *If* $\Gamma, x : T, \Gamma' \vdash t : T'$ *and* $\mathrm{ctype}_T(x, t) = T''$ *then* $T' \equiv T''$.

Full ctype *Properties*

Lemma (Properties of ctype)

i. If $\mathsf{ctype}_T(x, t) = T'$ then $head(t) = x$ and T' is a subexpression of T.

ii. If $\Gamma, x : T, \Gamma' \vdash t : T'$ and $\mathsf{ctype}_T(x, t) = T''$ then $T' \equiv T''$.

iii. If $\Gamma, x : T, \Gamma' \vdash t_1\ t_2 : T'$, $\Gamma \vdash t : T$, $[t/x]^T t_1 = \lambda y : T_1.t'$, and t_1 is not a λ-abstraction, then there exists a type A such that $\mathsf{ctype}_T(x, t_1) = A$.

iv. If $\Gamma, x : T, \Gamma' \vdash t_1\ t_2 : T'$, $\Gamma \vdash t : T$, $[t/x]^T t_1 = \Delta y : \neg(T'' \to T').t'$, and t_1 is not a μ-abstraction, then there exists a type A such that $\mathsf{ctype}_T(x, t_1) = A$.