Parametric Lambda Calculus

A meta-calculus for computation

Luca Paolini
Università di Torino
Dipartimento di Informatica

http://www.di.unito.it/~paolini/papers/parametric09handout.pdf

June 29, 2009

Outline

Parametric Parameter Passing Lambda-Calculus

A Paradigmatic Programming Language

The λ-calculus is the forefather of

modern high-level programming languages

evaluated through activation-records on a stack.

1956-60 John McCarthy introduced LISP, namely a list-processing language with a function-abstraction facility.

LISP's substitution procedure does not respect that of λ-calculus:

α-conversion was replaced by dynamic-binding.

Early 1960 Peter Landin proposed the use of λ-calculus to code constructs of the programming language Algol 60.

- ISWIM is an untyped λ-calculus togheter some constants for numerals.
- SECD (acronym of "Stack, Environment, Code, Dump") is a logical description of a virtual machine formalizing the evaluation of ISWIM.

A call-by-value parameter passing is implemented by SECD, so also ISWIM does not perfectly meet λ-calculus.

Early 1960 Corrado Böhm develops a less know language named CUCH where lambda-calculus and combinatory logic was mixed.

Its evaluation respects λ-calculus,
indeed it implement the leftmost strategy on the call-by-name parameter passing policy.

A bit of References

- John McCarthy. “Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I”. Comm. ACM 3(4):184-195, 1960.
- Peter Landin. “A correspondence between ALGOL 60 and Church’s Lambda-notation”. Comm. ACM 8:89-101, 1965.
- Corrado Böhm. “The CUCH as a formal and description language.” In T. B. Steel, Jr., editor, Formal Language Description Languages for Computer Programming, pages 179-197. North-Holland Co., Amsterdam, 1966. Proc. conference in 1964.
- Felice Cardone and J. Roger Hindley: “Lambda-calculus and Combinators in the 20th Century”, chapter of “The Handbook of the History of Logic” (edited by D. Gabbay and J. Woods), Vol.5:533-627, Elsevier, 2008.

Call by Value Lambda-Calculus

Gordon Plotkin. Call-by-Name, Call-by-Value and the lambda-Calculus. `Theoretical Computer Science` 1(2): 125-159, 1975.

> Our intention is to study call-by-value and call-by-value, in the setting of the lambda-calculus ....
> If the terms of the lambda-calculus (we have in mind the λK-β calculus for the moment) are regarded as [programs][a], with a reduction relation showing how they may be [evaluated][b] and indeed with a normal order reduction sequence capturing, in deterministic fashion, all possible normal forms, then we have already pretty determined a programming language.
> ....
>
> As a primary example of a lambda-calculus programming we consider ISWIM [5,7] without recursion operation and any syntactic sugar. It has an operational semantics which is given by the SECD machine. As primary example of a calculus, take the λK-$\beta\delta$ calculus [2] (the δ-rules make the comparison easier).
>
> Unfortunately, the two are hardly in accord.
>
> (1) Sometimes the SECD machine stops when it should either go on to give a normal form or should not terminate, according to normal order reduction. This is because ISWIM does not simplify procedure bodies.
> (2) Sometimes the SECD machine never stops when, according to normal order reduction, it ought to. This is because ISWIM calls its arguments by value.
>
> So one has to look for other programming-language/calculus pairs.
> ....
>
> Our intention is to study programming mechanisms programming mechanisms and so we accept the SECD machine and we look for the corresponding calculus - called λ_v in the text. The notion of value is changed to that induced by the SECD machine and a normal order reduction sequence theorem is given, which establish a good correspondence between λ_v and ISWIM. In this way we hope to have shown that ISWIM is more than a specification of some characterless reduction sequence. Rather, as well as being computationally natural, it gives rise to an interesting calculus. Its correspondence with this calculus shows it to be less order of reduction dependent than its definition shows.
>
> To study call-by-name, we define a call-by-name ISWIM, corresponding to a certain modification of the SECD machine, which keeps the above notion of value, and show that the usual λK-$\beta\delta$ calculus can be regarder as the call-by-name calculus. This substantiates folklore.
> In both cases the calculi are are seen to be correct from the point of view of the programming languages.

[a] "rules" in the original manuscript of Plotkin.
[b] "carried out " in the original manuscript of Plotkin.

Jokes, Stategies and Observable Outcomes

D. S. Scott. A type-theoretical alternative to ISWIM, CUCH, OWHY, 1969. Late published in Theoretical Computer Science (1&2): 411-440, 1993.

> .....
> The strange title of this paper ought perhaps to be explained . In 1966, Landin published an influential paper [14] which introduced a syntactical design style for programming languages, one of which is called ISWIM, standing for "If you see what I mean". Also Böhm in 1966 published the paper [3] which named a language of combinators called CUCH, standing for "Curry-Church". There seemed to be a worrisome trend in funny acronyms starting here (... GNU, recursively standing for "GNU is not Unix").
> .....
>
> The author hoped to stop some proliferation by suggesting a return to logically standard type-theoretical framework and thereby deter the creation of programming language of doubtful foundation called (as a group) OWHY, standing for "Or What Have You." No one really understood the joke and the effort was doomed to be of no avail. And history proved the author to be too conservative in any case.

C.-H. Luke Ong. Correspondence between Operational and Denotational Semantics. In Handbook of Logic in Computer Science, Volume 4, OUP, pages 269-356, 1995.

> ... Plotkin presented PCF explicitly as a programming language and studied the relationship between its operational semantics and denotational semantics which is based on the Scott-continuous function space model. For ease of exposition, it is good to make a clear distinction between [calculi][a] and programming languages. A [calculus][1] is a (formal) language with rewrite rules. A programming language may be regarded as a [calculus][1] with some additional features. For our purpose, it is an essential feature of a programming language to be equipped with a specified reduction strategy. ...

[a] "reduction system(s)" in the original manuscript of Ong.

Recap

- ☐ A calculus is a formal language endowed by some reductions rules which can be applied to phrases of such a language.
- ☐ A calculus induce a programming language when explicitly endowed with a (deterministic) strategy and a notion of observable outcomes.
- ☐ A calculus is correct w.r.t. a programming language whenever to calculate on a program respects its meaning.

  For instance, since $(\lambda z.M)((\lambda x.xx)(\lambda x.xx)) \to M$ we says that the $\lambda\beta$-calculus is not correct w.r.t ISWIM.

This educational perspective is not fully shared by the scientific community. More pragmatic approach gives rise interesting different perspectives. Mismatches between ISWIM programming language and classical call-by-name lambda-calculus leaded to the introduction of

1. a new calculus, the λ_v-calculus, based on call-by-value parameter passing policy where the "incoming arguments" are the Plotkin's values
2. a notion of observable outcomes (accidentally, being still the Plotkin's values) togheter with a strategy making the λ_v-calculus the basement for ISWIM
3. a new programming language, called lazy lambda-calculus based on the classical λ-calculus where the strategy is the lefmost one and "observable outcomes" are still the Plotkin's values

... the list of scientific results of the Plotkin's paper is not exhaustive, look at the paper :-)

A meta analysis

☞ Question:
Is there a paradigmatic calculus that can meta-model all interesting pairs (programming language/calculus) considered before (and may be that can suggest new such pairs)?

☞ Answer:
The parametric λ-calculus.

- ☐ Luca Paolini and Simona Ronchi Della Rocca. Parametric parameter passing lambda-calculus. Information and Computation, 189(1):87106, February 2004.
- ☐ Simona Ronchi Della Rocca and Luca Paolini. The Parametric Lambda-Calculus. A Metamodel for Computation. Texts in Theoretical Computer Science: An EATCS Series. Springer-Verlag, Berlin, 2004.

The parametric λ-calculus is an abstract calculus with some basic requirements, parametric with respect to a set of terms. Each different instantiation of the parameter produces a calculus.

1. It allows to generalize definitions, properties and proofs
2. It is a natural starting point for the study of new calculi
3. It helps the investigation of relations between different calculi

Parametric Lambda-Calculus

- The set Λ is produced by
$$M ::= x \mid MM \mid \lambda x.M$$

- Let $\Delta \subseteq \Lambda$.
The Δ-reduction ($\rightarrow_\Delta$) is the contextual closure of the following rule:
$$(\lambda x.M)N \rightarrow_\Delta M[N/x] \quad \text{whenever} \quad N \in \Delta$$

☞ When a set Δ induces an interesting calculus ?

It is reasonable to respect some properties.

1. Variables can be considered as placeholder for values.
2. The status of being an input value should be preserved during the evalution process.

Definition 1

$\Delta \subseteq \Lambda$ is a set of input values, when:

- $\mathrm{Var} \subseteq \Delta$ (variable closure)
- $P, Q \in \Delta$ implies $P[Q/x] \in \Delta$, (substitution closure)
- $M \in \Delta$ and $M \rightarrow_\Delta N$ imply $N \in \Delta$ (reduction closure)

Instances of Values

- Λ is a set of input values

(the Λ-calculus is the classical λ-calculus)

- $\Gamma = \mathrm{Var} \cup \{\lambda x.M \mid M \in \Lambda\}$ is a set of input values

(the Γ-calculus is the Plotkin's λ_v-calculus)

- $\Xi = \mathrm{Var} \cup \{M \mid M$ is a closed β normal form $\}$ is a set of input value

- Λ_I defined by the following grammar:

$$M ::= x \mid MM \mid \lambda x.M^{a}$$

is a set of input values

On the othe hand,

- Λ-normal forms are not input values
- Λ-head normal forms are not input values
- Γ-normal forms are not input values

[a] where $x \in \mathrm{FV}(M)$.

Confluence

Confluence

Theorem 2

$M \to^*_\Delta N_1$ and $M \to^*_\Delta N_2$ imply that

there is N_3 such that both $N_1 \to^*_\Delta N_3$ and $N_2 \to^*_\Delta N_3$.

In order to induce a Δ-calculus enjoying the confluence property are sufficient the closure conditions making Δ a set of input values.

Corollary 3

The Δ-normal form of a term, if it exists, is unique.

Let us take Λ-normal forms as input values.

Then the confluence fails, in fact:

$(\lambda x.(\lambda y.z)(x(\lambda x.xx)))(\lambda x.xx)$ reduces to both z and $(\lambda y.z)((\lambda x.xx)(\lambda x.xx))$, which do not have a common reduct.

Parallel Reductions

Let Δ be a set of input values.

Definition 4

The complete reduction $\hookrightarrow_\Delta$ is inductively defined as follows:

1. $x \hookrightarrow_\Delta x$;
2. $M \hookrightarrow_\Delta N$ implies $\lambda x.M \hookrightarrow_\Delta \lambda x.N$;
3. $M \hookrightarrow_\Delta M'$,$N \hookrightarrow_\Delta N'$ and $N \in \Delta$ imply $(\lambda x.M)N \hookrightarrow_\Delta M'[N'/x]$;
4. $M \hookrightarrow_\Delta M'$,$N \hookrightarrow_\Delta N'$ and $N \notin \Delta$ imply $MN \hookrightarrow_\Delta M'N'$.

Definition 5

The parallel reduction $\Rightarrow_\Delta$ is inductively defined as follows:

1. $x \Rightarrow_\Delta x$;
2. $M \Rightarrow_\Delta N$ implies $\lambda x.M \Rightarrow_\Delta \lambda x.N$;
3. $M \Rightarrow_\Delta M'$,$N \Rightarrow_\Delta N'$ and $N \in \Delta$ imply $(\lambda x.M)N \Rightarrow_\Delta M'[N'/x]$;
4. $M \Rightarrow_\Delta M'$,$N \Rightarrow_\Delta N'$ imply $MN \Rightarrow_\Delta M'N'$.

☞ The complete reduction is deterministic, while the parallel reduction is nondeterministic.

Example 6

Let $M \equiv I(II)$.

If $\Delta \equiv \Lambda$ then $M \hookrightarrow_\Delta I$, while $M \Rightarrow_\Delta M$, $M \Rightarrow_\Delta II$ and $M \Rightarrow_\Delta I$.
If $\Delta \equiv \Gamma$ then $M \hookrightarrow_\Delta II$ while $M \Rightarrow_\Delta M$ and $M \Rightarrow_\Delta II$.

Basic Properties

Lemma 7

Let Δ be a set of input values.

1. $M \rightarrow_\Delta N$ implies $M \Rightarrow_\Delta N$.
2. $M \Rightarrow_\Delta N$ implies $M \rightarrow^*_\Delta N$.
3. $\rightarrow^*_\Delta$ is the transitive closure of $\Rightarrow_\Delta$.

Lemma 8

Let $M \Rightarrow_\Delta M'$ and $N \Rightarrow_\Delta N'$.
If $N \in \Delta$ then $M[N/x] \Rightarrow_\Delta M'[N'/x]$.

Property 9

$M \hookrightarrow_\Delta P$ and $M \hookrightarrow_\Delta Q$ implies $P \equiv Q$.

Let $[M]_\Delta$ be the term such $M \hookrightarrow_\Delta [M]_\Delta$. In the literature $[M]_\Delta$ is called the complete development of M.

Diamond Property

Lemma 10

$M \Rightarrow_\Delta N$ implies $N \Rightarrow_\Delta [M]_\Delta$.

Proof. By induction on M.

1. If $M \equiv x$, then $N \equiv x$ and $[M]_\Delta \equiv x$.
2. If $M \equiv \lambda x.P$ then $N \equiv \lambda x.Q$, for some Q such that $P \Rightarrow_\Delta Q$. By induction $Q \Rightarrow_\Delta [P]_\Delta$, and so $N \Rightarrow_\Delta \lambda x.[P]_\Delta \equiv [M]_\Delta$.
3. If $M \equiv P_1P_2$ and it is not a Δ-redex, then $N \equiv Q_1Q_2$ for some Q_1 and Q_2 such that $P_1 \Rightarrow_\Delta Q_1$ and $P_2 \Rightarrow_\Delta Q_2$. So, by induction, $Q_1 \Rightarrow_\Delta [P_1]_\Delta$ and $Q_2 \Rightarrow_\Delta [P_2]_\Delta$, which implies $N \Rightarrow_\Delta [P_1]_\Delta[P_2]_\Delta \equiv [M]_\Delta$.
4. If $M \equiv (\lambda x.P_1)P_2$ is a redex (i.e. $P_2 \in \Delta$) then either $N \equiv (\lambda x.Q_1)Q_2$ or $N \equiv Q_1[Q_2/x]$, for some Q_i such that $P_i \Rightarrow_\Delta Q_i$ $(1 \leq i \leq 2)$. By induction, $Q_i \Rightarrow_\Delta [P_i]_\Delta$ $(1 \leq i \leq 2)$. Note that $[P_2]_\Delta \in \Delta$ by Lemma 7.(ii). In both cases, $N \Rightarrow_\Delta [P_1]_\Delta[[P_2]_\Delta/x] \equiv [M]_\Delta$, in the former case simply by induction, and in the latter both by induction and by Lemma 8.

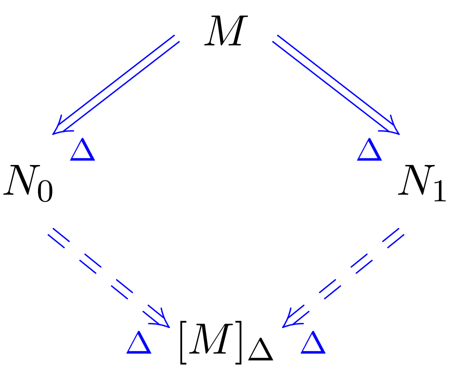
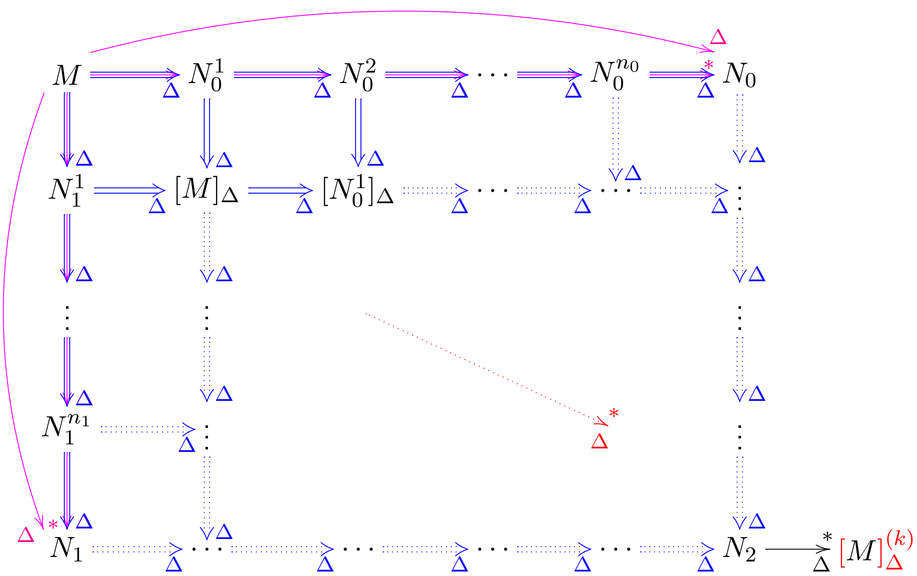


Figure 1: Diamond property.

PROOF of Confluence



On the quality of constraints

Are the closure conditions on input values, given in Definition 1, necessary in order to assure the confluence of the calculus?
No, they are not strictly necessary ...

weaker version of them is needed.

Let $P \in \Delta$ be such that, for every $Q \not\equiv P$ such that $P \to^*_\Delta Q$, $Q \notin \Delta$. Thus $(\lambda x.M)P$ reduces both to $M[P/x]$ and to $(\lambda x.M)Q$, which do not have a common 1-step-reduct, since the last term is NOT a Δ-redex. Thus the weaker version of reduction closure that is necessary is the following: $M \in \Delta$ and $M \to^*_\Delta N$ imply that there is $P \in \Delta$ such that $N \to^*_\Delta P$. On the other hand, let $N, P \in \Delta$ but for all Q such that $N[P/x] \to^*_\Delta Q$, $Q \notin \Delta$. Thus $(\lambda x.(\lambda y.M)N)P$ reduces both to $(\lambda y.M[P/x])N[P/x]$ and to $(M[N/y])[P/x]$, which do not have a common 1-step-reduct. Thus the weaker version of the substitution closure that is necessary is the following: $P, Q \in \Delta$ implies there is $R \in \Delta$ such that $P[Q/x] \to^*_\Delta R$.

Standardization

Standard Strategies

Assume $M \to^*_\Delta N$, and assume that there is more than one Δ-reduction sequence from M to N. The standardization theorem says that, in case the set of input values enjoys a particular property, there is a "standard" reduction sequence from M to N, reducing the redexes in a given order.

In case of Λ-calculus a standard reduction sequence is a strategy choosing redexes from left to right. For instance,

$$(\lambda x.x(KI))(II) \to_\Lambda II(KI) \to_\Lambda I(KI) \to_\Lambda I(\lambda y.I)$$

is a standard reduction sequence in the Λ-calculus.

It is not easy to formalize such a strategy for other sets of input values.
The reduction sequence reducing the same term from left to right in the Γ-calculus:

$$(\lambda x.x(KI))(II) \to_\Gamma (\lambda x.x(KI))(I) \to_\Gamma I(KI)$$

cannot be standardized ordering redexes "leftmost" !

Sequentialization

- A symbol λ in a term M is active if and only if it is the first symbol of a Δ-redex of M.

- The Δ-sequentialization $(M)^\circ$ of a term M is a function from Λ to Λ defined as follows:
  - $(xM_1 ... M_m)^\circ = x(M_1)^\circ ...(M_m)^\circ$;
  - $((\lambda x.P)QM_1 ... M_m)^\circ = (\lambda x.P)^\circ(Q)^\circ(M_1)^\circ ...(M_m)^\circ$ if $Q \in \Delta$;
  - $((\lambda x.P)QM_1 ... M_m)^\circ = (Q)^\circ(\lambda x.P)^\circ(M_1)^\circ ...(M_m)^\circ$ if $Q \notin \Delta$;
  - $(\lambda x.P)^\circ = \lambda x.(P)^\circ$.

- The degree of a redex R in M is the numbers of λ's which both are active in M and occur on the left of $(R)^\circ$ in $(M)^\circ$.

- A sequence $M \equiv P_0 \to_\Delta P_1 \to_\Delta ... \to_\Delta P_n \to_\Delta N$ is standard if and only if the degree of the redex contracted in P_i is less than or equal to the degree of the redex contracted in P_{i+1}, for every $i < n$.

  We denote by $M \to^\circ_\Delta N$ a standard reduction sequence from M to N.

- The principal redex of M, if it exists, is the redex of M with minimum degree. The principal reduction $M \to^p_\Delta N$ denotes that N is obtained from M by reducing the principal redex of M. Moreover, $\to^{*p}_\Delta$ is the reflexive and transitive closure of $\to^p_\Delta$.

Standard Values

$(\lambda x.x(KI))(II) \to_\Gamma (\lambda x.x(KI))I \to_\Gamma I(KI) \to_\Gamma I(\lambda y.I)$ is a standard reduction sequence in the Γ-calculus.

Definition 11
A set Δ of input values is standard if and only if $M \notin \Delta$ and $M \to^*_\Delta N$ by reducing at every step a not principal redex imply $N \notin \Delta$.

- The set of input values Λ, Γ, Ξ are standard.
- The set of input values Λ_I is not standard.

Property 12
The condition that Δ is standard is necessary and sufficient for the Δ-calculus enjoys the standardization property.

Proof. The sufficiency of the condition is a consequence of the Standardization Theorem. To prove its necessity, assume Δ is not standard; we can find a term $M \notin \Delta$ such that $M \to^*_\Delta N \in \Delta$, without reducing the principal redex. Hence $IM \to^*_\Delta IN \to_\Delta N$, by reducing first a redex of degree different from 0 and then a redex of degree 0. Clearly, there is no way of commuting the order of reductions.

Standardization

Theorem 13

Let Δ be standard. $M \to_\Delta^* N$ implies there is a standard reduction sequence from M to N.

Let $M \Rightarrow_\Delta^\circ N$ denote "$M \to_\Delta^\circ N$ and $M \Rightarrow_\Delta N$".

Corollary 14

Let Δ be standard.
If $M \to_\Delta^* N$ and N is a normal form then $M \to_\Delta^{*p} N$.

The main difficulty is to prove the next Lemma.

Lemma 15

Let $\vec{P}, \vec{Q}$ be two sequences of terms such that $\|\vec{P}\| = \|\vec{Q}\|$; moreover, let $P_i \in \Delta$ and $P_i \Rightarrow_\Delta^\circ Q_i$ for all $i \leq \|\vec{P}\|$.

1. If $M \Rightarrow_\Delta^\circ N$ then $M[\vec{P}/\vec{x}] \Rightarrow_\Delta^\circ N[\vec{Q}/\vec{x}]$.
2. If $M \Rightarrow_\Delta N$ then $M \Rightarrow_\Delta^\circ N$.

Proof. (1) and (2) can be proved by mutual induction on M.

An example

Let M be the following term

$$(\lambda^a z.z((\lambda^b x.x)(\lambda^c x.xx)))((\lambda^d x.x)((\lambda^f x.x)(\lambda^g x.x)))((\lambda^h x.xx)(\lambda^i x.x))$$

where we have named lambda-abstractions by letters.

In the Λ-calculus, there 5 active lambda, ordered from the left to the right. In red with the degree of their respective labels:

$$(\lambda^0 z.z((\lambda^1 x.x)(\lambda x.xx)))((\lambda^2 x.x)((\lambda^3 x.x)(\lambda x.x)))((\lambda^4 x.xx)(\lambda x.x)).$$

In the Γ-calculus, there are only 3 active lambda. Since M° is

$$(((\lambda^f x.x)(\lambda^g x.x))(\lambda^d x.x))(\lambda^a z.z((\lambda^b x.x)(\lambda^c x.xx)))((\lambda^h x.xx)(\lambda^i x.x))$$

we obtain the degree of Γ-redexes that of following labels, i.e.

$$(\lambda z.z((\lambda^1 x.x)(\lambda x.xx)))((\lambda x.x)((\lambda^0 x.x)(\lambda^g x.x)))((\lambda^2 x.xx)(\lambda^i x.x)).$$

Proof of Lemma 15.(1)

By Lemma 8, $M[\vec{P}/\vec{x}] \Rightarrow_\Delta N[\vec{Q}/\vec{x}]$, hence it suffices to show that $M[\vec{P}/\vec{x}] \rightarrow^\circ_\Delta N[\vec{Q}/\vec{x}]$.

Let $M \equiv \lambda y_1 ... y_h.\zeta M_1 ... M_m$ $(h, m \in \mathbb{N})$, where either ζ is a variable or $\zeta \equiv (\lambda z.T)U$.

If $h > 0$, then the proof follows by induction.

Let $h = 0$, thus $N \equiv \xi N_1 ... N_m$ such that $\zeta \Rightarrow^\circ_\Delta \xi$ and $M_i \Rightarrow^\circ_\Delta N_i$; furthermore, let $M'_i \equiv M_i[\vec{P}/\vec{x}]$ and $N'_i \equiv N_i[\vec{Q}/\vec{x}]$ $(1 \leq i \leq m)$.

The proof is organized according to the possible shapes of ζ.

☞ 1. Let ζ be a variable. If $m = 0$ then the proof is trivial, so let $m > 0$. There are two cases to be considered.

✗ 1.1. $\zeta \notin \vec{x}$, so $\xi[\vec{Q}/\vec{x}] \equiv \zeta$. By induction $M_i[\vec{P}/\vec{x}] \rightarrow^\circ_\Delta N_i[\vec{Q}/\vec{x}]$ and the standard reduction sequence is

$$\zeta M'_1 ... M'_m \rightarrow^\circ_\Delta \zeta N'_1 M'_2 ... M'_m \rightarrow^\circ_\Delta \; ..... \; \rightarrow^\circ_\Delta \zeta N'_1 ... N'_m.$$

✗ 1.2. $\zeta \equiv x_j \in \vec{x}$ $(1 \leq j \leq l)$, so $\xi[\vec{Q}/\vec{x}] \equiv Q_j$. But $P_j \Rightarrow^\circ_\Delta Q_j$ means that there is a standard sequence $P_j \equiv S_0 \rightarrow_\Delta ..... \rightarrow_\Delta S_n \equiv Q_j$ $(n \in \mathbb{N})$. Two cases can arise.

☑ 1.2.1. $\forall i \leq n$, $S_i \not\equiv \lambda z.S'$. Then the following reduction sequence

$$\sigma : S_0 M'_1 ... M'_m \rightarrow_\Delta \; ..... \; \rightarrow_\Delta S_n M'_1 ... M'_m$$

is standard. Since by induction $M_i[\vec{P}/\vec{x}] \rightarrow^\circ_\Delta N_i[\vec{Q}/\vec{x}]$, there is a standard reduction sequence

$$\tau : S_n M'_1 ... M'_m \rightarrow^\circ_\Delta S_n N'_1 M'_2 ... M'_m \rightarrow^\circ_\Delta \; ..... \; \rightarrow^\circ_\Delta S_n N'_1 ... N'_m.$$

Note that $S_0 M'_1 ... M'_m \equiv M[\vec{P}/\vec{x}]$ and $S_n N'_1 ... N'_m \equiv N[\vec{Q}/\vec{x}]$, so σ followed by τ is the desired standard reduction sequence.

☑ 1.2.2. There is a minimum $k \leq n$ such that $S_k \equiv \lambda z.S'$.
By induction on (ii), $M_1 \Rightarrow^\circ_\Delta N_1$. Therefore, by induction $M_1[\vec{P}/\vec{x}] \Rightarrow^\circ_\Delta N_1[\vec{Q}/\vec{x}]$, where $M_1[\vec{P}/\vec{x}] \rightarrow^\circ_\Delta N_1[\vec{Q}/\vec{x}]$ is $M_1[\vec{P}/\vec{x}] \equiv R_0 \rightarrow_\Delta ..... \rightarrow_\Delta R_p \equiv N_1[\vec{Q}/\vec{x}]$ $(p \in \mathbb{N})$. There are two subcases:

⚽ 1.2.2.1. $\forall i \leq p$, $R_i \notin \Delta$. Then the following reduction sequence:

$$\begin{aligned} \sigma' : M[\vec{P}/\vec{x}] \equiv S_0 R_0 M'_2 ... M'_m &\rightarrow_\Delta ..... \rightarrow_\Delta S_k R_0 M'_2 ... M'_m \\ &\rightarrow_\Delta ..... \rightarrow_\Delta S_k R_p M'_2 ... M'_m \\ \rightarrow_\Delta S_{k+1} R_p M'_2 ... M'_m &\rightarrow_\Delta ..... \rightarrow_\Delta S_n R_p M'_2 ... M'_m \end{aligned}$$

is also standard. Moreover, since $M_i[\vec{P}/\vec{x}] \rightarrow^\circ_\Delta N_i[\vec{P}/\vec{x}]$, the following reduction sequence:

$$\begin{aligned} \tau' : \; & S_n R_p M'_2 ... M'_m \rightarrow^\circ_\Delta \\ & S_n R_p N'_2 M'_3 ... M'_m \rightarrow^\circ_\Delta \; ..... \; \rightarrow^\circ_\Delta S_n R_p N'_2 ... N'_m \end{aligned}$$

is also standard. Clearly σ' followed by τ' is the desired standard reduction sequence.

Proof of Lemma 15.(1) - Continue

- 1.2.2.2. There is a minimum $q \leq p$ such that $R_q \in \Delta$. So

$$\begin{aligned} \sigma'' : \quad M[\vec{P}/\vec{x}] \equiv S_0 R_0 M'_2 ... M'_m \rightarrow_\Delta ..... \rightarrow_\Delta S_k R_0 M'_2 ... M'_m \\ \rightarrow_\Delta ..... \rightarrow_\Delta S_k R_q M'_2 ... M'_m \rightarrow_\Delta S_{k+1} R_q M'_2 ... M'_m \\ \rightarrow_\Delta ..... \rightarrow_\Delta S_n R_q M'_2 ... M'_m \rightarrow_\Delta ..... \rightarrow_\Delta S_n R_p M'_2 ... M'_m \end{aligned}$$

is a standard reduction sequence. The desired standard reduction sequence is σ'' followed by τ'.

- 2. Let $\zeta \equiv (\lambda z.T)U$. Thus $N \equiv (\lambda z.\bar{T})\bar{U} N_1 ... N_m$ or $N \equiv \bar{T}[\bar{U}/z] N_1 ... N_m$, where $T \Rightarrow_\Delta \bar{T}$, $U \Rightarrow_\Delta \bar{U}$ and $M_i \Rightarrow_\Delta N_i$ $(1 \leq i \leq m)$.
By induction, $U' \equiv U[\vec{P}/\vec{x}] \Rightarrow^\circ_\Delta \bar{U}[\vec{Q}/\vec{x}] \equiv U''$, $T' \equiv T[\vec{P}/\vec{x}] \Rightarrow^\circ_\Delta \bar{T}[\vec{Q}/\vec{x}] \equiv T''$ and $M'_i \equiv M_i[\vec{P}/\vec{x}] \Rightarrow^\circ_\Delta N_i[\vec{Q}/\vec{x}] \equiv N'_i$ $(1 \leq i \leq m)$.
Let $U' \equiv R_0 \rightarrow_\Delta ... \rightarrow_\Delta R_p \equiv U''$ $(p \in \mathbb{N})$ be the standard sequence $U' \rightarrow^\circ_\Delta U''$. Without loss of generality let us assume $z \notin \vec{x}$.

- 2.1. Let $N \equiv (\lambda z.\bar{T})\bar{U} N_1 ... N_m$. There are two cases.
- 2.1.1. $\forall i \leq p \;\; R_i \notin \Delta$. Then the standard reduction sequence $M[\vec{P}/\vec{x}] \rightarrow^\circ_\Delta N[\vec{Q}/\vec{x}]$ is

$$\begin{aligned} (\lambda z.T')R_0 M'_1 ... M'_m \rightarrow_\Delta ..... \rightarrow_\Delta (\lambda z.T')R_p M'_1 ... M'_m \\ \rightarrow^\circ_\Delta (\lambda z.T'')R_p M'_1 ... M'_m \rightarrow^\circ_\Delta (\lambda z.T'')R_p N'_1 M'_2 ... M'_m \\ \rightarrow^\circ_\Delta \;\; ..... \;\; \rightarrow^\circ_\Delta (\lambda z.T'')R_p N'_1 ... N'_m. \end{aligned}$$

- 2.1.2. There is a minimum $q \leq p$ such that $R_q \in \Delta$. Thus the desired standard reduction sequence is:

$$\begin{aligned} (\lambda z.T')R_0 M'_1 ... M'_m \rightarrow_\Delta ..... \rightarrow_\Delta (\lambda z.T')R_q M'_1 ... M'_m \\ \rightarrow^\circ_\Delta (\lambda z.T'')R_q M'_1 ... M'_m \rightarrow_\Delta ..... \rightarrow_\Delta (\lambda z.T'')R_p M'_1 ... M'_m \\ \rightarrow^\circ_\Delta (\lambda z.T'')R_p N'_1 M'_2 ... M'_m \rightarrow^\circ_\Delta \;\; ..... \;\; \rightarrow^\circ_\Delta (\lambda z.T'')R_p N'_1 ... N'_m. \end{aligned}$$

- 2.2. Let $N \equiv \bar{T}[\bar{U}/z] N_1 ... N_m$. So, there is a minimum $q \leq p$ such that $R_q \in \Delta$; let μ be the standard reduction sequence:

$$\begin{aligned} M[\vec{P}/\vec{x}] \equiv (\lambda z.T')R_0 M'_1 ... M'_m \rightarrow_\Delta ..... \quad \rightarrow_\Delta (\lambda z.T')R_q M'_1 ... M'_m \\ \rightarrow_\Delta T'[R_q/z] M'_1 ... M'_m. \end{aligned}$$

$T \Rightarrow^\circ_\Delta \bar{T}$, by induction on (ii). Furthermore, since $R_q \Rightarrow^\circ_\Delta U''$, it follows by induction that $T[\vec{P}/\vec{x}][R_q/z] \Rightarrow^\circ_\Delta \bar{T}[\vec{Q}/\vec{x}][U''/z]$.
Let $T[\vec{P}/\vec{x}][R_q/z] \equiv T_0 \rightarrow_\Delta ..... \rightarrow_\Delta T_t \equiv \bar{T}[\vec{Q}/\vec{x}][U''/z]$ be the corresponding standard reduction sequence. Two subcases can arise:

- 2.2.1. $\forall i \leq t$, $T_i \not\equiv \lambda z.S'$. The desired standard reduction sequence is μ followed by:

$$\begin{aligned} T'[R_p/z] M'_1 ... M'_m \equiv T[\vec{P}/\vec{x}][R_p/z] M'_1 ... M'_m \rightarrow_\Delta T_1 M'_1 ... M'_m \\ \rightarrow_\Delta ..... \rightarrow_\Delta T_t M'_1 ... M'_m \rightarrow^\circ_\Delta \;\; ..... \;\; \rightarrow^\circ_\Delta T_t N'_1 ... N'_m \equiv [\vec{Q}/\vec{x}] \end{aligned}$$

- 2.2.2. Let $k \leq t$ be the minimum index such that $T_k \equiv \lambda y.T'_k$. The construction of the standard reduction sequence depends on the fact that M_2 may or may not become an input value, but, in every case, it can be easily built as in the previous cases.

Proof of Lemma 15.(2)

The cases $M \equiv x$ and $M \equiv \lambda z.M'$ are easy.

☞ 1. Let $M \equiv PQ \Rightarrow_\Delta P'Q' \equiv N$, $P \Rightarrow_\Delta P'$ and $Q \Rightarrow_\Delta Q'$.
By induction, there are standard sequences $P \equiv P_0 \rightarrow_\Delta \ldots \rightarrow_\Delta P_p \equiv P'$ and $Q \equiv Q_0 \rightarrow_\Delta \ldots \rightarrow_\Delta Q_q \equiv Q'$.
If $\forall i \leq p \;\; P_i \not\equiv \lambda z.P_i'$, then $M \rightarrow^\circ_\Delta N$ is $P_0Q_0 \rightarrow^\circ_\Delta P_pQ_0 \rightarrow^\circ_\Delta P_pQ_q$.
Otherwise, let k be the minimum index such that $P_k \equiv \lambda z.P_k'$.

- If $\forall j \leq q \;\; Q_j \notin \Delta$, then $M \rightarrow^\circ_\Delta N$ is

$$P_0Q_0 \rightarrow_\Delta \ldots\ldots \rightarrow_\Delta P_kQ_0 \rightarrow^\circ_\Delta P_kQ_q \rightarrow_\Delta P_{k+1}Q_q \rightarrow_\Delta \ldots\ldots \rightarrow_\Delta P_pQ_q.$$

- If there is a minimum h such that $Q_h \in \Delta$, the standard sequence is $P_0Q_0 \rightarrow^\circ_\Delta P_kQ_0 \rightarrow^\circ_\Delta P_kQ_h \rightarrow_\Delta P_{k+1}Q_h \rightarrow^\circ_\Delta P_pQ_h \rightarrow^\circ_\Delta P_pQ_q$.

☞ 2. Let $M \equiv (\lambda x.P)Q \Rightarrow_\Delta P'[Q'/x] \equiv N$ where $P \Rightarrow_\Delta P'$, $Q \Rightarrow_\Delta Q'$ and $Q \in \Delta$. Hence $P \Rightarrow^\circ_\Delta P'$ and $Q \Rightarrow^\circ_\Delta Q'$ follow by induction, so $P[Q/x] \Rightarrow^\circ_\Delta P'[Q'/x]$, by induction on (i). Thus, the desired standard reduction sequence is $(\lambda x.P)Q \rightarrow_\Delta P[Q/x] \rightarrow^\circ_\Delta P'[Q'/x]$.

Standardization 2

Definition 16

Let $M, N \in \Lambda$.

1. $M \rightarrow^i_\Delta N$ denotes that N is obtained from M by reducing a redex that is not the principal redex.
2. $M \Rightarrow^i_\Delta N$ denotes $M \Rightarrow_\Delta N$ and $M \rightarrow^{*i}_\Delta N$.

Corollary 17

$M \Rightarrow_\Delta N$ implies there is P such that $M \rightarrow^{*p}_\Delta P \Rightarrow^i_\Delta N$.

Example 18

Let $M \equiv (\lambda xy.I(\lambda z.IK(II)))I \Rightarrow_\Gamma \lambda yz.IKI$.
Therefore $M \rightarrow^p_\Gamma \lambda y.I(\lambda z.IK(II)) \rightarrow^p_\Gamma \lambda yz.IK(II) \Rightarrow^i_\Gamma \lambda yz.IKI$ and clearly $\lambda yz.IK(II) \in \Gamma$.

Standardization 3

Lemma 19

Let Δ be standard.
$M \Rightarrow^i_\Delta P \rightarrow^p_\Delta N$ implies $M \rightarrow^{*p}_\Delta Q \Rightarrow^i_\Delta N$, for some Q.

Proof. By induction on M. If either $M \equiv \lambda x.M'$, or the head of M is a variable, then the proof follows by induction. Otherwise, let $M \equiv (\lambda y.M_0)M_1...M_m$; thus it must be $P \equiv (\lambda y.P_0)P_1...P_m$. Note that $M \Rightarrow^i_\Delta P$ implies $M_i \Rightarrow_\Delta P_i$ $(1 \leq i \leq m)$. Now there are two cases, according to whether $P_1 \in \Delta$ or not.

(i) Let $P_1 \in \Delta$; it follows that P_1 is the argument of the principal redex of P, thus $N \equiv P_0[P_1/y]P_2...P_m$.
Let $M_1 \in \Delta$. Then we can build the following reduction sequence:
$M \equiv (\lambda y.M_0)M_1...M_m \rightarrow^p_\Delta M_0[M_1/y]...M_m \Rightarrow_\Delta P_0[P_1/y]P_2...P_m$, which can be transformed into a standard one by Lemma 17.
Let $M_1 \notin \Delta$ and $P_1 \in \Delta$; since the set Δ is standard, $M_1 \Rightarrow_\Delta P_1 \in \Delta$ if and only if $M_1 \rightarrow^{*p}_\Delta P_1' \Rightarrow^i_\Delta P_1$, where $P_1' \in \Delta$. But this would imply that in the reduction $M \Rightarrow^i_\Delta P$ the principal redex of M_1 has been reduced; but by definition the principal redex of M_1 coincides with the principal redex of M, against the hypothesis that $M \Rightarrow^i_\Delta P$. So this case is not possible.

(ii) Let $P_1 \notin \Delta$. Then there is $j \geq 0$ such that the principal redex of P_j is the principal redex of P.
Let $j \geq 2$; so $\forall k \leq j$ P_k is a normal form. So $N \equiv (\lambda y.P_0)P_1...P_j'..P_m$, where $P_j \rightarrow^p_\Delta P_j'$. From the hypothesis that $M \Rightarrow^i_\Delta P$, it follows that $M_i \equiv P_i$ $(0 \leq i \leq j-1)$, and $M_i \Rightarrow_\Delta P_i$ $(j < i \leq m)$. Then by induction there is P_j^* such that $M_j \rightarrow^{*p}_\Delta P_j^* \Rightarrow^i_\Delta P_j'$, and we can build the following reduction sequence:

$$(\lambda y.M_0)M_1...M_m \rightarrow^{*p}_\Delta (\lambda y.M_0)M_1...P_j^* P_{j+1}...P_m \Rightarrow_\Delta (\lambda y.M_0)M_1...P_j'...P_m$$

which can be transformed into a standard one by Lemma 17.
The case $j < 2$ is similar. ∎

Corollary 20

Let Δ be standard.
If $M \rightarrow^*_\Delta N$ then $M \rightarrow^{*p}_\Delta Q \underbrace{\Rightarrow^i_\Delta \ldots \Rightarrow^i_\Delta}_{k} N$, for some Q and some k.

Proof of Standardization Theorem

The proof is given by induction on N. From Corollary 20, $M \to^*_\Delta N$ implies $M \to^{*p}_\Delta Q \to^{*i}_\Delta N$ for some Q. Obviously, the reduction sequence $\sigma : M \to^{*p}_\Delta Q$ is standard by definition of $\to^{p}_\Delta$. Note that, by definition of $\to^{*i}_\Delta$, $Q \to^{*i}_\Delta N$ implies that Q and N have the same structure, i.e.
$Q \equiv \lambda x_1 ... x_n.\zeta Q_1 ... Q_n$ and $N \equiv \lambda x_1 ... x_n.\zeta' N_1 ... N_n$, where $Q_i \to^*_\Delta N_i$ $(i \leq n)$ and either ζ and ζ' are the same variable, or $\zeta \equiv (\lambda x.R)S$, $\zeta' \equiv (\lambda x.R')S'$, $R \to^*_\Delta R'$ and $S \to^*_\Delta S'$.
The case when ζ is a variable follows by induction. Otherwise, by induction there are standard reduction sequences $\sigma_i : Q_i \to^\circ_\Delta N_i$ $(1 \leq i \leq n)$, $\tau_R : R \to^\circ_\Delta R'$ and $\tau_S : S \to^\circ_\Delta S'$. Let
$S \equiv S_0 \to_\Delta ..... \to_\Delta S_k \equiv S'$ $(k \in \mathbb{N})$.
If $\forall i \leq k$ $S_i \notin \Delta$ then the desired standard reduction sequence is σ followed by $\tau_S, \tau_R, \sigma_1, ..., \sigma_n$.
Otherwise, there is $S_h \in \Delta$ $(h \leq k)$. In this case, let $\tau^0_S : S_0 \to_\Delta ..... \to_\Delta S_h$ and
$\tau^1_S : S_{h+1} \to_\Delta ..... \to_\Delta S_k$; the desired standard reduction sequence is σ followed by
$\tau^0_S, \tau_R, \tau^1_S, \sigma_1, ..., \sigma_n$.

On the quality of constraints

Let $\Delta \subseteq \Lambda$ and let $\mathrm{Var} \subseteq \Delta$. In order for the Δ-reduction enjoy the standardization property it is necessary for Δ to be closed under substitution.

Let $M, N \in \Delta$ and $M[N/x] \notin \Delta$. The following non-standard reduction sequence
$(\lambda x.IM)N \to_\Delta (\lambda x.M)N \to_\Delta M[N/x]$ has not a standard counterpart, in fact
$I(M[N/x]) \not\to_\Delta M[N/x]$.

The investigation on the reduction closure is more complex and it needs some additional definitions and remarks. Morally, the reduction closure is necessary, except in some *degenerated* cases of input values. Details can be found in
Luca Paolini and Simona Ronchi Della Rocca. Parametric parameter passing lambda-calculus.
Information and Computation, 189(1):87-106, February 2004.

Parametric Principal Reduction Machine

The principal evaluation is a sequence of reduction performing at every step the redex of minimum degree, when it exists. The principal evaluation is normalizing.

There is a Parametric Principal Reduction Machine, parametric with respect to the set of input values Δ, reducing a term according to the principal evaluation.

$$\frac{M \rightarrow^p_\Delta N}{\lambda x.M \rightarrow^p_\Delta \lambda x.N}\ p1 \qquad \frac{i = \min\{j \leq m | M_i \notin \Delta\text{-nf}\} \quad M_i \rightarrow^p_\Delta N_i}{xM_1...M_m \rightarrow^p_\Delta xM_1...N_i...M_m}\ p2$$

$$\frac{Q \in \Delta}{(\lambda x.P)QM_1...M_m \rightarrow^p_\Delta P[Q/x]M_1...M_m}\ p3$$

$$\frac{Q \notin \Delta \quad Q \notin \Delta\text{-nf} \quad Q \rightarrow^p_\Delta Q'}{(\lambda x.P)QM_1...M_m \rightarrow^p_\Delta (\lambda x.P)Q'M_1...M_m}\ p4$$

$$\frac{Q \notin \Delta \quad Q \in \Delta\text{-nf} \quad P \notin \Delta\text{-nf} \quad P \rightarrow^p_\Delta P'}{(\lambda x.P)QM_1...M_m \rightarrow^p_\Delta (\lambda x.P')QM_1...M_m}\ p5$$

$$\frac{Q \notin \Delta \quad P, Q \in \Delta\text{-nf} \quad i = \min\{j \leq m | M_i \notin \Delta\text{-nf}\} \quad M_i \rightarrow^p_\Delta N_i}{(\lambda x.P)QM_1...M_m \rightarrow^p_\Delta (\lambda x.P)QM_1...N_i...M_m}\ p6$$

Output values

Definition 21
Let Δ be a set of input values.
A set of output values with respect to Δ is any set $\Theta \subseteq \Lambda$ such that:

- Θ contains all the Δ-normal forms;
- If $M =_\Delta N$ and $N \in \Theta$ then there is $P \in \Theta$ such that $M \rightarrow^{*p}_\Delta P$.

Examples.

- Δ-normal forms is a set of output values w.r.t Δ, for all Δ.
- Λ-head normal forms is a set of output values w.r.t Λ.
- Λ-weak head normal forms is a set of output values w.r.t Λ.
- Γ togheter with Γ-normal forms are a set of output values w.r.t Γ.

Operational Semantics

- Let Θ be a set of output values with respect to the set of input value Δ. $\Downarrow_{\Delta,\Theta}$ is the evaluation relation defined through the following rules:

$$\frac{M \in \Theta}{M \Downarrow_{\Delta,\Theta} M}\ (axiom) \qquad\qquad \frac{M \to^{p}_{\Delta} P \qquad P \Downarrow_{\Delta,\Theta} N}{M \Downarrow_{\Delta,\Theta} N}\ (eval)$$

- Operational pre-order:
$M \preceq_{\Delta,\Theta} N$ if and only if, for all contexts $C[.]$ such that $C[M], C[N] \in \Lambda^0$, $(C[M] \Downarrow_{\Delta,\Theta}$ implies $C[N] \Downarrow_{\Delta,\Theta})$.

- Operational Equivalence:
$M \approx_{\Delta,\Theta} N$ if and only if $M \preceq_{\Delta,\Theta} N$ and $N \preceq_{\Delta,\Theta} M$.

Lazy Operational Instances

$\mathbf{L} \in \mathcal{E}(\Lambda, \Lambda\text{-LHNF})$ is the evaluation relation induced by the formal system proving judgments of the shape $M \Downarrow_{\mathbf{L}} N$ where $M \in \Lambda$ and $N \in \Lambda$-LHNF. It consists of the following rules:

$$\frac{m \geq 0}{xM_1 \ldots M_m \Downarrow_{\mathbf{L}} xM_1 \ldots M_m}\ (var)$$

$$\frac{}{\lambda x.M \Downarrow_{\mathbf{L}} \lambda x.M}\ (lazy)$$

$$\frac{P[Q/x]M_1 \ldots M_m \Downarrow_{\mathbf{L}} N}{(\lambda x.P)QM_1 \ldots M_m \Downarrow_{\mathbf{L}} N}\ (head)$$

$\mathbf{V} \in \mathcal{E}(\Gamma, \Gamma\text{-LBNF})$ is the evaluation relation induced by the formal system proving judgments of the shape $M \Downarrow_{\mathbf{V}} N$ where $M \in \Lambda$ and $N \in \Gamma$-LBNF. It consists of the following rules:

$$\frac{}{xM_1 \ldots M_m \Downarrow_{\mathbf{V}} xM_1 \ldots M_m}\ (var)$$

$$\frac{}{\lambda x.M \Downarrow_{\mathbf{V}} \lambda x.M}\ (lazy)$$

$$\frac{Q \Downarrow_{\mathbf{V}} Q' \quad Q' \in \Gamma \quad P[Q'/x]M_1 \ldots M_m \Downarrow_{\mathbf{V}} N}{(\lambda x.P)QM_1 \ldots M_m \Downarrow_{\mathbf{V}} N}\ (head)$$

$$\frac{Q \Downarrow_{\mathbf{V}} Q' \quad Q' \notin \Gamma}{(\lambda x.P)QM_1 \ldots M_m \Downarrow_{\mathbf{V}} (\lambda x.P)Q'M_1 \ldots M_m}\ (block)$$

Sometimes, in literature, lazy is replaced by weak! Albeit, the weak reduction is also used as nomenclature in combinatory logic and the induced reduction on lambda-terms.

The previous operational evaluation machine can be put in exact correspondence with the two programming languages considered in:

Gordon Plotkin. Call-by-Name, Call-by-Value and the lambda-Calculus. Theoretical Computer Science 1(2): 125-159, 1975.

Operational Instances, again!

$\mathbf{H} \in \mathcal{E}(\Lambda, \Lambda\text{-HNF})$ is the evaluation relation induced by the formal system proving judgments of the shape $M \Downarrow_{\mathbf{H}} N$ where $M \in \Lambda$ and $N \in \Lambda$-HNF. It consists of the following rules:

$$\frac{m \geq 0}{xM_1 \ldots M_m \Downarrow_{\mathbf{H}} xM_1 \ldots M_m} \; (var)$$

$$\frac{M \Downarrow_{\mathbf{H}} N}{\lambda x.M \Downarrow_{\mathbf{H}} \lambda x.N} \; (abs)$$

$$\frac{P[Q/x]M_1 \ldots M_m \Downarrow_{\mathbf{H}} N}{(\lambda x.P)QM_1 \ldots M_m \Downarrow_{\mathbf{H}} N} \; (head)$$

$\mathbf{N} \in \mathcal{E}(\Lambda, \Lambda\text{-NF})$ is the evaluation relation induced by the formal system proving judgments of the shape $M \Downarrow_{\mathbf{N}} N$ where $M \in \Lambda$ and $N \in \Lambda$-NF. It consists of the following rules:

$$\frac{(M_i \Downarrow_{\mathbf{N}} N_i)_{(i \leq m)}}{xM_1 \ldots M_m \Downarrow_{\mathbf{N}} xN_1 \ldots N_m} \; (var)$$

$$\frac{M \Downarrow_{\mathbf{N}} N}{\lambda x.M \Downarrow_{\mathbf{N}} \lambda x.N} \; (abs)$$

$$\frac{P[Q/x]M_1 \ldots M_m \Downarrow_{\mathbf{N}} N}{(\lambda x.P)QM_1 \ldots M_m \Downarrow_{\mathbf{N}} N} \; (head)$$

Since we interested only in the evaluation of closed terms, operational evaluations often are presented by considering the rules needed to the evaluation of closed terms.

Theories

In order to model computation, Δ-equality is too weak.

As an example, let Δ be either Λ or Γ.
If we want to model the termination property, both the terms DD and $(\lambda x.xxx)(\lambda x.xxx)$ represent programs that run forever, while the two terms are $\neq_\Delta$ each other. Indeed $DD \to_\Delta DD$ and $(\lambda x.xxx)(\lambda x.xxx) \to_\Delta (\lambda x.xxx)(\lambda x.xxx)(\lambda x.xxx)$. So it would be natural to consider them equal in this particular setting.

On the other hand, if we want to take into account not only termination but also the size of terms, they need to be different; in fact, the first one reduces to itself while the second increases its size during the reduction. As we will see in the following, for all instances of Δ we will consider, all interesting interpretations of the calculus also equate terms that are not $=_\Delta$.

Definition 22

Let Δ be a set of input values.
$\approx_T \subseteq \Lambda \times \Lambda$ is a Δ-theory whenever

- $\approx_T$ is a preorder relation, namely it is reflexive and transitive;
- $P \approx_T Q$ and $C[P], C[Q] \in \Lambda^0$ imply $C[P] \approx_T C[Q]$, for each context $C[.]$;
- $P =_\Delta Q$ implies $P \approx_T Q$.

Calculi vs. Programming Languages

Let Θ be a set of output values w.r.t. the set of input value Δ. **Property** 23

If $M \to^*_\Delta N \in \Theta$ then $M \Downarrow_{\mathcal{U}^\Delta_\Theta}$.

Proof. Since Θ satisfies the principality condition, $M \to^*_\Delta N \in \Theta$ implies there is $N' \in \Theta$ such that $M \to^{*p}_\Delta N'$. Then the proof follows by induction on the length of the reduction sequence $M \to^{*p}_\Delta N' \in \Theta$. If $M \in \Theta$, then the proof follows by rule (*axiom*) of the formal system defining $\mathcal{U}^\Delta_\Theta$. Otherwise, $M \to^{*p}_\Delta N'$ means $M \to^p_\Delta N'' \to^{*p}_\Delta N'$, so the proof follows by induction.

Theorem 24 **$\mathcal{U}^\Delta_\Theta$-Correctness**

The Δ-calculus is correct with respect to the $\mathcal{U}^\Delta_\Theta$-operational semantics.

Proof. $M =_\Delta N$ implies $C[M] =_\Delta C[N]$, for all contexts $C[.]$.
If there is $P \in \Theta$ such that $C[M] \to^*_\Delta P$, then $C[M] \Downarrow_{\mathcal{U}^\Delta_\Theta}$, by Theorem 23. Clearly $P =_\Delta C[N]$; thus, by principality, there is $P' \in \Theta$ such that $C[N] \to^{*p}_\Delta P'$, so $C[N] \Downarrow_{\mathcal{U}^\Delta_\Theta}$.
In case there is not such a P, both $C[M] \Uparrow_{\mathcal{U}^\Delta_\Theta}$ and $C[N] \Uparrow_{\mathcal{U}^\Delta_\Theta}$.

Pretheories

Definition 25

Let Δ be a set of input values.

1. $\preceq_T \subseteq \Lambda \times \Lambda$ is a Δ-pretheory whenever
- $\preceq_T$ is a preorder relation, namely it is reflexive and transitive;
- $P \preceq_T Q$ and $C[P], C[Q] \in \Lambda^0$ imply $C[P] \preceq_T C[Q]$, for each context $C[.]$;
- $P =_\Delta Q$ implies $P \preceq_T Q$.

2. If $\preceq_T$ is a Δ-pretheory then the induced equivalence $\approx_T \subseteq \Lambda \times \Lambda$ is a Δ-theory.

Definition 26

Let Δ be a set of input values.

1. $\preceq_T^0 \subseteq \Lambda^0 \times \Lambda^0$ is a closed Δ-pretheory whenever
- $\preceq_T^0$ is a preorder relation, namely it is reflexive and transitive;
- $P, Q \in \Lambda^0$, $P \preceq_T^0 Q$ and $C[P], C[Q] \in \Lambda^0$ imply $C[P] \preceq_T^0 C[Q]$, for each context $C[.] \in \Lambda_C^0$;
- $P, Q \in \Lambda^0$ and $P =_\Delta Q$ imply $P \preceq_T^0 Q$.

2. $\lessapprox_T \subseteq \Lambda \times \Lambda$ denotes the relation induced by a closed Δ-pretheory $\preceq_T^0$, by putting $M \lessapprox_T N$ if and only if there exists a set of variables $\{x_1, ..., x_n\}$ such that $\mathrm{FV}(M) \cup \mathrm{FV}(N) \subseteq \{x_1, ..., x_n\}$ and $\lambda x_1 ... x_n.M \preceq_T^0 \lambda x_1 ... x_n.N$.

Pretheory Property

The relation defined in the last point of the previous definition is actually a Δ-pretheory, as shown in the next lemma.

Lemma 27

Let $\preceq^0_T$ be a closed Δ-pretheory.

1. Let $P, Q \in \Lambda$, $\mathrm{FV}(P) \cup \mathrm{FV}(Q) \subseteq \{x_1, ..., x_k\}$ and $\{x_1, ..., x_k\} \subseteq \{y_1, ..., y_h\}$ $(k, h \in \mathbb{N})$. If $\lambda x_1...x_k.P \preceq^0_T \lambda x_1...x_k.Q$ then $\lambda y_1...y_h.P \preceq^0_T \lambda y_1...y_h.Q$.
2. $\lessapprox_T$ is a Δ-pretheory.

Proof.

1. Let $C[.] \equiv \lambda y_1...y_h.([.]x_1...x_k)$, thus $C[\lambda x_1...x_k.P], C[\lambda x_1...x_k.Q] \in \Lambda^0$. Therefore $\lambda y_1...y_h.P \preceq^0_T \lambda y_1...y_h.Q$, since $C[\lambda x_1...x_k.P] \preceq^0_T C[\lambda x_1...x_k.Q]$, $C[\lambda x_1...x_k.P] =_\Delta \lambda y_1...y_h.P$ and $C[\lambda x_1...x_k.Q] =_\Delta \lambda y_1...y_h.Q$.
2.
   - If $M \in \Lambda$ and $\mathrm{FV}(M) = \{x_1, ..., x_n\}$ then $\lambda x_1...x_n.M \preceq^0_T \lambda x_1...x_n.M$ by reflexivity, so $M \lessapprox_T M$ by Definition 26.(2). Let $M_0 \lessapprox_T M_1$ and $M_1 \lessapprox_T M_2$; so, there are variables such that:
     - $\mathrm{FV}(M_0) \cup \mathrm{FV}(M_1) \subseteq \{x_1, ..., x_n\}$ and $\lambda x_1...x_n.M_0 \preceq^0_T \lambda x_1...x_n.M_1$,
     - $\mathrm{FV}(M_1) \cup \mathrm{FV}(M_2) \subseteq \{y_1, ..., y_m\}$ and $\lambda y_1...y_m.M_1 \preceq^0_T \lambda y_1...y_m.M_2$.

     Let $\{x_1, ..., x_n\} \cup \{y_1, ..., y_m\} \subseteq \{z_1, ..., z_p\}$; without loss of generality, $\lambda z_1...z_p.M_0 \preceq^0_T \lambda z_1...z_p.M_1$ and $\lambda z_1...z_p.M_1 \preceq^0_T \lambda z_1...z_p.M_2$ by the previous point. Thus $\lambda z_1...z_p.M_0 \preceq^0_T \lambda z_1...z_p.M_2$ since $\preceq^0_T$ is transitive, so $M_0 \lessapprox_T M_2$.
   - Let $M \lessapprox_T N$ and $C[.] \in \Lambda_C$ be such that $C[M], C[N] \in \Lambda^0$.
     Let $\lambda x_1...x_n.M \preceq^0_T \lambda x_1...x_n.N$, $\mathrm{FV}(M) \cup \mathrm{FV}(N) \subseteq \{x_1, ..., x_n\}$ and $C'[.] \equiv C[[.]x_1...x_n] \in \Lambda_C$. Since $\mathrm{FV}(C'[\lambda x_1...x_n.M]) \cup \mathrm{FV}(C'[\lambda x_1...x_n.N])$ can be nonempty, let $C''[.] \equiv \lambda z_1...z_m.C'[.]$ such that $C''[M], C''[N] \in \Lambda^0$. So $C''[\lambda x_1...x_n.M] \preceq^0_T C''[\lambda x_1...x_n.N]$, since $\preceq^0_T$ is a closed Δ-pretheory. But $C''[\lambda x_1...x_n.M] =_\Delta \lambda z_1...z_m.C[M]$ and $C''[\lambda x_1...x_n.N] =_\Delta \lambda z_1...z_m.C[N]$ imply $\lambda z_1...z_m.C[M] \preceq^0_T \lambda z_1...z_m.C[N]$.
     Hence $C[M] \lessapprox_T C[N]$ since $\lambda z_1...z_m.C[M], \lambda z_1...z_m.C[N] \in \Lambda^0$.
   - If $M =_\Delta N$ and $\mathrm{FV}(M) \cup \mathrm{FV}(N) \subseteq \{x_1, ..., x_n\}$ then $\lambda x_1...x_n.M =_\Delta \lambda x_1...x_n.N$. But $\lambda x_1...x_n.M, \lambda x_1...x_n.N \in \Lambda^0$ implies $\lambda x_1...x_n.M \preceq^0_T \lambda x_1...x_n.N$, so $M \lessapprox_T N$.

Unique Open-Extension Property

An interesting corollary holds, in case $\lessapprox_T$ is the Δ-pretheory induced by the closed Δ-pretheory $\preceq_T^0$. Namely, if $P, Q \in \Lambda$ and $P \lessapprox_T Q$ then $\lambda x_1...x_n.P \lessapprox_T \lambda x_1...x_n.Q$ for all sequences $x_1, ..., x_n$, by Lemma 27.(2) and Definition 25.
The following theorem shows a useful relation between closed Δ-pretheories and Δ-pretheories. It implies that a closed Δ-pretheory has a unique extension to open terms, precisely that of the Definition 26.

Property 28

Let $\preceq_T^0$ be a closed Δ-pretheory and let $\preceq_S$ be a Δ-pretheory.
If $\preceq_S$ and $\preceq_T^0$ are the same relation on closed terms then $\preceq_S \equiv \lessapprox_T$.

Proof. Let $P, Q \in \Lambda$.

- If $P \preceq_S Q$ and $\mathrm{FV}(P) \cup \mathrm{FV}(Q) \subseteq \{x_1, ..., x_n\}$ then $\lambda x_1...x_n.P \preceq_S \lambda x_1...x_n.Q$, so $\lambda x_1...x_n.P \preceq_T^0 \lambda x_1...x_n.Q$, thus $P \lessapprox_T Q$.
- $P \lessapprox_T Q$ and $\mathrm{FV}(P) \cup \mathrm{FV}(Q) \subseteq \{x_1, ..., x_n\}$ imply $\lambda x_1...x_n.P \preceq_T^0 \lambda x_1...x_n.Q$, therefore $\lambda x_1...x_n.P \preceq_S \lambda x_1...x_n.Q$. The proof follows immediately, since $P =_\Delta (\lambda x_1...x_n.P)x_1...x_n \preceq_S (\lambda x_1...x_n.Q)x_1...x_n =_\Delta Q$.

The previous theorem assures us that without loss of generality, we can concentrate our attention on closed (pre-)theories.

Denotational Semantics

Some parametric analysis of denotational semantics has been developed by using filter models and intersection typing systems. In case of interest, you can look at:

- Simona Ronchi Della Rocca and Luca Paolini. The Parametric Lambda-Calculus. A Metamodel for Computation. Texts in Theoretical Computer Science: An EATCS Series. Springer-Verlag, Berlin, 2004.

- Luca Paolini and Simona Ronchi Della Rocca. Lazy Logical Semantics. Proceedings of the Final Workshop of the Italian Research Project COMETA, Electronic Notes in Theoretical Computer Science, volume 104C , pages 235–251, 2004, Elsevier.

New calculi

Definition 29

If Δ is a set of input values then

$$\Delta_\star = \mathrm{Var} \cup \{M \in \Lambda \mid M \to^*_\Delta N \in \Delta\}$$

is the set of Δ-valuable terms.

Theorem 30

Let Δ be a set of input values.

1. The equivalences $=_\Delta$ and $=_{\Delta_\star}$ coincide.
2. $\Delta_\star$ is a set of input value, moreover it is standard!
3. Δ-NF$=$ $\Delta_\star$-NF.

Proof.

1. Trivial.
2. The only nontrivial constraint is the possibly closure under substitution, that follows by the previous point.
3. Trivial.

Applicative PreOrder

Often valuable sets are undecidable!

☞ Are set of valuable terms interesting?

Yes, it makes us able to adapt some call-by-name proof-techniques to call-by-value lambda-caluli!

Definition 31

Let $\Downarrow_{\Delta,\Theta}$ is the evaluation relation induced by Θ respecting Δ.
If $M\vec{Q} \Downarrow_{\Delta,\Theta}$ imply $N\vec{Q} \Downarrow_{\Delta,\Theta}$, for each sequence $\vec{Q}$ of closed terms, then $M \trianglelefteq_{\Delta,\Theta} N$. We call $\trianglelefteq_{\Delta,\Theta}$, the applicative preorder induced by the evaluation $\Downarrow_{\Delta,\Theta}$.

We say that an operational semantics is uniform when its applicative preorder coincides with its contextual theory.

It is possible to prove that uniformity holds for $\mathbf{H}$, $\mathbf{N}$, $\mathbf{L}$ adapting a clever synactical technique of Berry, by induction on the rules defining our predicates.

Unfortunately, this technique cannot be applied to call-by-value lambda calculus because its pre-evaluation of arguments.

Another Programming Language

Proposition 32

$\Gamma - LBNF$ is a set of output values w.r.t $\Gamma_\star$.

$\mathbf{V}_\star \in \mathcal{E}(\Gamma_\star, \Gamma\text{-LBNF})$ is the evaluation relation induced by the formal system proving judgments of the shape $M \Downarrow_{\mathbf{V}_\star} N$ where $M \in \Lambda$ and $N \in \Gamma\text{-LBNF}$. It consists of the following rules:

$$\frac{}{xM_1 \ldots M_m \Downarrow_{\mathbf{V}_\star} xM_1 \ldots M_m}\ (var)$$

$$\frac{}{\lambda x.M \Downarrow_{\mathbf{V}_\star} \lambda x.M}\ (lazy)$$

$$\frac{\cancel{Q \Downarrow_{\mathbf{V}} Q'} \quad Q \in \Gamma_\star \quad P[Q/x]M_1 \ldots M_m \Downarrow_{\mathbf{V}_\star} N}{(\lambda x.P)QM_1 \ldots M_m \Downarrow_{\mathbf{V}_\star} N}\ (head)$$

$$\frac{\cancel{Q \Downarrow_{\mathbf{V}} Q'} \quad Q \notin \Gamma_\star}{(\lambda x.P)QM_1 \ldots M_m \Downarrow_{\mathbf{V}_\star} (\lambda x.P)QM_1 \ldots M_m}\ (block)$$

Theorem 33

$M \Downarrow_{\mathbf{V}}$ if and only if $M \Downarrow_{\mathbf{V}_\star}$, for all M.

Proof. The proof follows from the standardization theorems for Γ and $\Gamma_\star$.

Uniformity

Theorem 34

Let $M, N \in \Lambda^0$; $M \preceq_{\mathbf{V}_\star} N$ if and only if $M \trianglelefteq_{\mathbf{V}_\star} N$.

Proof. $(\Rightarrow)$ Trivial. $(\Leftarrow)$ We will show that $C[M] \Downarrow_{\mathbf{V}_\star}$ implies $C[N] \Downarrow_{\mathbf{V}_\star}$; the proof is given by induction on the rules proving that $C[M] \Downarrow_{\mathbf{V}_\star}$.

$(lazy)$ Either $C[.] \equiv [.]$ or $C[.] \equiv \lambda x.C'[.]$. Both cases are trivial.

$(head)$ There are two cases.

1. Let $C[.] \equiv [.]C_1[.]...C_m[.]$ $(m \in \mathbb{N})$ and let $M \equiv (\lambda z.M_0)M_1...M_n$ $(n \in \mathbb{N})$ where $m + n \geq 1$. If $n = 0$ then $m \geq 1$, therefore we can let $D[.] \equiv M_0[C_1[.]/z]C_2[.]...C_m[.]$; otherwise let $D[.] \equiv M_0[M_1/z]M_2...M_nC_1[.]...C_m[.]$. In both cases $D[M] \Downarrow_{\mathbf{V}_\star}$ is the premise of the rule $(head)$, so $D[N] \Downarrow_{\mathbf{V}_\star}$ by induction. Hence $MC_1[N]...C_m[N] \Downarrow_{\mathbf{V}_\star}$ by rule $(head)$, thus the proof follows by the theorem's hypothesis.
2. If $C[.] \equiv (\lambda x.C_0[.])C_1[.]...C_m[.]$ $(m \geq 1)$ then, since $M \in \Lambda^0$, the proof follows by induction on $C[M] \equiv C_0[M][C_1[M]/x]C_2[M]...C_m[M] \Downarrow_{\mathbf{V}_\star}$.

An introduction

In the classical λ-calculus,
β-normal forms are crucial ingredients for semantical analysis.

Instead in Plotkin's cbv λ-calulus we will see that:
Γ-normal forms are essentially meaningless, while potentially valuable terms are relevants.

Let $\Delta \subseteq \Lambda$ be a set of terms.

- The lazy Δ-reduction ($\rightarrow_{\Delta\ell}$) is the applicative closure of the following rule:

$$(\lambda x.M)N \rightarrow M[N/x] \text{ if and only if } N \in \Delta$$

- Let $I \equiv \lambda x.x$; then $\lambda x.II \rightarrow_{\Lambda} \lambda x.I$
  while $\lambda x.II \not\rightarrow_{\Lambda\ell} \lambda x.I$

- Let $\spadesuit \in \{\Delta, \Delta\ell\}$.
  $\rightarrow^{+}_{\spadesuit}$ is the transitive closure of $\rightarrow_{\spadesuit}$
  $\rightarrow^{*}_{\spadesuit}$ is the reflexive and transitive closure of $\rightarrow_{\spadesuit}$
  $=_{\spadesuit}$ is the symmetric, reflexive and transitive closure of $\rightarrow_{\spadesuit}$

Let $\spadesuit \in \{\Delta, \Delta\ell\}$ where $\Delta \subseteq \Lambda$ is a set of input values.

- A term M is in $\spadesuit$-normal form iff
  it has no occurrences of $\spadesuit$-redexes.

- A term M has $\spadesuit$-normal form iff
  $\exists N$ in $\spadesuit$-normal form s.t. $M \rightarrow^{*}_{\spadesuit} N$.

- A term M is $\spadesuit$-strongly normalizing iff
  all sequences of $\spadesuit$-reduction starting from M are finite.

  So, in particular, M has $\spadesuit$-normal form.

Blocked Δ-Normal Forms

If M is a Λ-NF then it has the following shape:

$$\lambda x_1 ... x_n . x M_1 ... M_m \qquad (n, m \geq 0)$$

where $M_i \in \Lambda$-NF, for all $i \leq m$.

Namely, the set Λ-NF of Λ-normal forms is

$$\begin{aligned} \Lambda\text{-NF} = \ \mathrm{Var} \ & \cup \{ x M_1 ... M_n \mid M_k \in \Lambda\text{-NF} \ (1 \leq k \leq n) \} \\ & \cup \{ \lambda x_1 ... x_n . M \mid M \in \Lambda\text{-NF} \}. \end{aligned}$$

As an example, $\lambda x. I(xx) \in \Gamma$-NF which is different from the call-by-name considered cases.

If M is a Δ-NF then it has the following shape:

$$\lambda x_1 ... x_n . \zeta M_1 ... M_m \qquad (n, m \geq 0)$$

where $M_i \in \Delta$-NF, for all $i \leq m$ and

$$\text{either } \zeta = \begin{cases} x \in \mathrm{Var} & \text{or,} \\ (\lambda x. P) Q & \text{where } P, Q \in \Delta\text{-NF}, \ Q \notin \Delta. \end{cases}$$

Namely, the set Δ-NF of Δ-normal forms is

$$\begin{aligned} \Delta\text{-NF} = \mathrm{Var} \ & \cup \{ x M_1 ... M_n \mid M_k \in \Delta\text{-NF} \ (1 \leq k \leq n) \} \\ & \cup \{ \lambda x_1 ... x_n . M \mid M \in \Delta\text{-NF} \} \\ & \cup \{ (\lambda x. P) Q M_1 ... M_n \mid P, Q, M_i \in \Delta\text{-NF}, Q \notin \Delta \}. \end{aligned}$$

Separability

Definition 35

1. A Δ-theory $\mathcal{T}$ is consistent if and only if there are $M, N \in \Lambda$ such that $M \neq_{\mathcal{T}} N$. Otherwise $\mathcal{T}$ is inconsistent.
2. A Δ-theory $\mathcal{T}$ is input consistent if and only if there are $M, N \in \Delta$ such that $M \neq_{\mathcal{T}} N$. Otherwise $\mathcal{T}$ is input inconsistent.

Property 36

Let $\mathcal{T}$ be a Δ-theory.
If $\mathcal{T}$ is input consistent then it is consistent.

Proof. Obvious.

Definition 37

Let Δ be a set of input values.
Two terms M, N are Δ-separable if and only if there is a context $C[.]$ such that $C[M] =_{\Delta} x$ and $C[N] =_{\Delta} y$ for two different variables x and y.

Property 38

Let M, N be Δ-separable.
If $\mathcal{T}$ is a Δ-theory such that $M =_{\mathcal{T}} N$ then $\mathcal{T}$ is input inconsistent.

Proof. Let $C[.]$ be the context separating M and N, i.e. $C[M] =_{\Delta} x$ and $C[N] =_{\Delta} y$ for two different variables x and y. Since $=_{\mathcal{T}}$ is a congruence, $M =_{\mathcal{T}} N$ implies $C[M] =_{\mathcal{T}} C[N]$, and so, since $\mathcal{T}$ is closed under $=_{\Delta}$, $x =_{\mathcal{T}} y$. But this implies $\lambda xy.x =_{\mathcal{T}} \lambda xy.y$, i.e. $K =_{\mathcal{T}} O$. But, since $=_{\mathcal{T}}$ is a congruence, this implies $KMN =_{\mathcal{T}} OMN$ for all terms M, N. In particular, if $M, N \in \Delta$ then $M =_{\mathcal{T}} N$ by Δ-reduction.

We denote by $=_{\eta}$ the minimum congruence relation wich includes $M = \lambda x.Mx$, for all $M \in \Lambda$ and $x \notin \mathrm{FV}(M)$.

Theorem 39 **Böhm's theorem**

Let $M, N \in \Lambda$-NF. If $M \neq_{\Lambda\eta} N$ then M and N are Λ-separable.

Hence, different $\Lambda\eta$-normal forms cannot be equated by a consistent Λ-theory.

On the other hand, we can equates different Δ-NF.
As an example, in the Γ-calculus just consider:

$$(\lambda xy.xy)(uu)(vI)$$
$$(\lambda xy.yx)(vI)(uu)$$

Γ-Liar Normal Forms

- Let $D \equiv \lambda x.xx$, since $(xI) \notin \Gamma$ and $(xI) \in \Gamma$-NF

$$(\lambda y.D)(xI)D \qquad \text{is a } \Gamma\text{-NF}.$$

- Let $C[.]$ be a context:

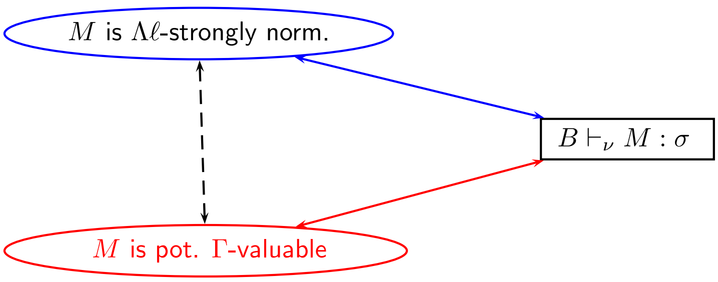
$$C[(\lambda y.D)(xI)D] \to^*_\Gamma I \quad \text{iff} \quad C[DD] \to^*_\Gamma I$$

Definition 40

- A term M is Δ-valuable if and only if there is $N \in \Delta$ such that $M \to^*_\Delta N$.
- A term M is potentially Δ-valuable if and only if there is a (capture free) substitution $\mathbf{s}$, replacing variables by terms belonging to Δ, such that $\mathbf{s}(M)$ is Δ-valuable.
- M is a Δ-liar normal forms iff
  M is a Δ-normal forms non potentially Δ-valuable.

Relating different calculi

A Relating Result



Intersection Types

- Let C be a countable set of type-constants (ranging over $\alpha, \beta, ..$) containing at least the type constant ν.
- The set $T(\mathsf{C})$ of types, ranging over by $\sigma, \tau, \pi, \rho, ..$ is defined by:
  $\sigma ::= \alpha \mid (\sigma \to \tau) \mid (\sigma \wedge \tau)$
- A basis is a partial function from Var to $T(\mathsf{C})$ having a finite domain of definition.

Types will be considered modulo associativity, commutativity and idempotency of $\wedge$.

$[\sigma_1/x_1, ..., \sigma_n/x_n]$ will denote the basis B
s.t. $dom(B) = \{x_1, ..., x_n\}$ and $B(x_i) = \sigma_i$. Typing judgments $B \vdash_\nu M : \sigma$ are proved by the following rules:

$$\frac{}{B[\sigma/x] \vdash_\nu x : \sigma}\,(var) \qquad \frac{}{B \vdash_\nu \lambda x.M : \nu}\,(\nu)$$

$$\frac{B[\sigma/x] \vdash_\nu M : \tau}{B \vdash_\nu \lambda x.M : \sigma \to \tau}\,(\to I) \qquad \frac{B \vdash_\nu M : \sigma \to \tau \quad B \vdash_\nu N : \sigma}{B \vdash_\nu MN : \tau}\,(\to E)$$

$$\frac{B \vdash_\nu M : \sigma \quad B \vdash_\nu M : \tau}{B \vdash_\nu M : \sigma \wedge \tau}\,(\wedge I)$$

$$\frac{B \vdash_\nu M : \sigma \wedge \tau}{B \vdash_\nu M : \sigma}\,(\wedge E_l) \qquad \frac{B \vdash_\nu M : \sigma \wedge \tau}{B \vdash_\nu M : \tau}\,(\wedge E_r)$$

Generation Lemmas

Lemma 41

1. If $B \vdash_\nu M : \sigma$ then $B - \{x : \tau \mid x \notin FV(M)\} \vdash_\nu M : \sigma$.
2. If $B \vdash_\nu M : \sigma$ then $B \cap B' \vdash_\nu M : \sigma$, for any basis B'.
3. If $B \vdash_\nu x : \sigma$ then either $x : \sigma \in B$ or $x : \rho \in B$, where $\rho \simeq \sigma \wedge \tau$, for some τ.
4. If $B \vdash_\nu MN : \sigma$ then there are types ρ_i and τ_i such that $\sigma \simeq \rho_1 \wedge \ldots \wedge \rho_n$, $B \vdash_\nu M : \tau_i \to \rho_i$ and $B \vdash_\nu N : \tau_i$ with $1 \leq i \leq n$.
5. $B \vdash_\nu \lambda x.M : \sigma \to \tau$ if and only if $B[\sigma/x] \vdash_\nu M : \tau$.

If B, B' are bases then $B \cap B'$ is the basis defined as follows:

$$(B \cap B')(y) = \begin{cases} B(y) \wedge B'(y) & \text{if} B(y), B'(y) \text{ are defined,} \\ B(y) & \text{if only } B(y) \text{ is defined,} \\ B'(y) & \text{if only } B'(y) \text{ is defined,} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

System Properties

Proposition 42

If d is a derivation of $B \vdash_\nu M : \sigma$ then every subterm of M, which is not under the scope of a λ-abstraction, is typed by a subderivation of d.

Proposition 43 **Subject-reduction**

If $B \vdash_\nu M : \sigma$ and $M \to_\Lambda N$ then $B \vdash_\nu N : \sigma$.

Proposition 44 **Typed subject-expansion**

Let $C[.]$ be a context. Then $B \vdash_\nu C[P[Q/x]] : \sigma$ and $B' \vdash_\nu Q : \tau$ imply $B \cap B' \vdash_\nu C[(\lambda x.P)Q] : \sigma$.

$\Lambda\ell$-Characterizations

Lemma 45

$M \in \Lambda\ell$-NF implies it is typable in the system $\vdash_\nu$.

Proof. The proof is carried out by induction on the structure of normal forms.
Let $M \equiv xM_1 \ldots M_n$. If $n = 0$ then the proof is trivial, so let $n > 0$. By induction there are B_i and σ_i s.t. $B_i \vdash_\nu M_i : \sigma_i$. Then M has type σ in the basis $B' = B_1 \cap \ldots \cap B_n \cap [\sigma_1 \to \ldots \to \sigma_n \to \sigma / x]$ since:

$$\dfrac{\dfrac{}{B' \vdash x : \sigma_1 \to \ldots \to \sigma_n \to \sigma}(*) \quad \dfrac{\vdots}{B' \vdash M_i : \sigma_i} \cdots}{B' \vdash xM_1 \ldots M_n : \sigma}(\to E)$$

where $(*)$ denotes a sequence of applications of (var), $(\wedge I)$, $(\wedge E_l)$ and $(\wedge E_r)$.
In the case $M \equiv \lambda x.M'$ then $B \vdash_\nu M : \nu$ for any basis B.

Theorem 46

$M \in \Lambda\ell$-SN implies M is typable in $\vdash_\nu$.

Proof. If M is in $\Lambda\ell$-NF, then the proof follows from the previous Lemma. Otherwise, we can assume that there is a Λ-reduction sequence

$$M \equiv M_0 \to_{\Lambda\ell} M_1 \to_{\Lambda\ell} \ldots \to_{\Lambda\ell} M_n \equiv N$$

reducing at each step the leftmost-innermost redex ($n > 0$). This reduction sequence is finite, since M is Λ-strongly normalizing. The proof is given by induction on n.
By induction hypothesis, there are B_1, σ such that $B_1 \vdash M_1 : \sigma$. If $(\lambda x.P)Q$ is the reduced redex then Q is in normal form and so there is a basis B_2 and a type τ such $B_2 \vdash Q : \tau$ by the previous Lemma. The proof follows from Proposition 44, taking into account that the innermost redex cannot occur in a subterm typed by the type ν.

Characterization of Potentially Valuable

Corollary 47

$M \in \Gamma$ implies M is typable in $\vdash_\nu$.

Theorem 48

$M \in \Gamma$-PV implies M is typable in $\vdash_\nu$.

Proof. Let $M \in \Gamma$-PV. Then there is a substitution $\mathtt{s}$, replacing variables by terms belonging to Γ, such that $\mathtt{s}(M) \to_\Gamma^* N \in \Gamma$. Let $FV(M) = \{x_1, \ldots, x_n\}$ $(n \geq 0)$, and let $\mathtt{s}(x_i) = P_i \in \Gamma$. By Corollary 47, there are B and σ such that $B \vdash N : \sigma$. Moreover, by Corollary 47 and Proposition 44 we have $B' \vdash_\nu \mathtt{s}(M) : \sigma$, for some B' such that $B \subseteq B'$.
Again by Corollary 47 and Proposition 44, $d : B'' \vdash_\nu (\lambda x_1 \ldots x_n.M)P_1 \ldots P_n : \sigma$, for some B'' such that $B' \subseteq B''$.
In order to conclude, assume M is not typed by a subderivation of d. But in this case it there must be j such that $d' : B' \vdash_\nu \lambda x_j \ldots x_n.M : \nu$ is a subderivation of d $(1 \leq j \leq n)$. But, since the type ν has not applicative power, only the subterm $(\lambda x_1 ... \lambda x_{j-1}.M)P_1 \ldots P_{j-1}$ can be typed, contrary to what we obtained before.

Saturated Sets

Definition 49

Let a k-lazy saturated set S_k^ℓ be a set such that:

1. $S_k^\ell \subseteq \Lambda\ell$-SN;
2. $x \in \mathrm{Var}$ and $M_i \in \Lambda\ell$-SN imply $xM_1 ... M_n \in S_k^\ell$ $(1 \leq i \leq n)$;
3. $M[P/x]M_1 ... M_n \in S_k^\ell$ and $P \in \Lambda\ell$-SN imply $(\lambda x.M)PM_1 ... M_n \in S_k^\ell$.
4. $\forall h \geq k$, $O^h \in S_k^\ell$, where $O^h \equiv \lambda x_1 \ldots x_h x_{h+1}.x_{h+1}$.

Let SAT_k^ℓ be the set of all k-saturated sets and let $SAT^\ell = \bigcup_{k \in \omega} SAT_k^\ell$.

Lemma 50

1. $M \notin \Lambda\ell$-SN implies that $\mathtt{s}(M) \notin \Lambda\ell$-SN, for every substitution $\mathtt{s}$.
2. $C[(\lambda x.P)Q] \notin \Lambda\ell$-SN, $Q \in \Lambda\ell$-SN and $C[(\lambda x.P)Q] \to_{\Lambda\ell} C[P[Q/x]]$ imply $C[P[Q/x]] \notin \Lambda\ell$-SN.

Theorem 51

$\Lambda\ell$-SN $\subseteq SAT^\ell$.

Saturated Arrow-Set

Let $\mathtt{S}$ and $\mathtt{T}$ be either two saturated sets or lazy-saturated sets. Define: $\mathtt{S} \rightarrow \mathtt{T} = \{M \mid MN \in \mathtt{T}, \text{for all } N \in \mathtt{S}\}$. In order to prove that $\mathtt{S} \rightarrow \mathtt{T}$ is saturated, we need a further property.

Proposition 52

$\mathtt{T} \in SAT_k^\ell$ implies that $O^{k+1}N \in \mathtt{T}$, for all $N \in \Lambda\ell$-SN.

Lemma 53

$\mathtt{T} \in SAT_t^\ell$ implies that $(\mathtt{S} \rightarrow \mathtt{T}) \in SAT_{t+1}^\ell$, for all $\mathtt{S} \in SAT^\ell$.

Types-Interpretation

Definition 54

If $\rho : \mathsf{C} \rightarrow SAT^\ell$ then $[\![.]\!]_\rho^\ell$ is the function from types to lazy saturated set defined as follows,

- $[\![\alpha]\!]_\rho^\ell = \rho(\alpha)$;
- $[\![\nu]\!]_\rho^\ell = \Lambda\ell$-SN;
- $[\![\sigma \rightarrow \tau]\!]_\rho^\ell = [\![\sigma]\!]_\rho^\ell \rightarrow [\![\tau]\!]_\rho^\ell$;
- $[\![\sigma \wedge \tau]\!]_\rho^\ell = [\![\sigma]\!]_\rho^\ell \cap [\![\tau]\!]_\rho^\ell$.

If $B = x_1 : \sigma_1, ..., x_n : \sigma_n$ then $[\![B]\!]_\rho^\ell = \{\mathbf{s} \mid \mathbf{s}(x_i) \in [\![\sigma_i]\!]_\rho^\ell\}$.

Adequateness

Theorem 55

$B \vdash_\nu M : \sigma$ implies $\forall \rho : \mathsf{C} \to SAT^\ell$, $\forall \mathbf{s} \in [\![B]\!]^\ell_\rho$, $\mathbf{s}(M) \in [\![\sigma]\!]^\ell_\rho$.

Proof. By induction on the derivation.
If the last applied rule is (var) then the result is obvious.

In the case the last applied rule is

$$\frac{B[\sigma/x] \vdash M : \tau}{B \vdash \lambda x.M : \sigma \to \tau}\ (\to I)$$

by induction, $\forall \rho, \forall \mathbf{s} \in [\![B[\sigma/x]]\!]_\rho$, $\mathbf{s}(M) \in [\![\tau]\!]_\rho$. Since $\mathbf{s}(x) \in [\![\sigma]\!]_\rho$, the result follows by the definition of $[\![\sigma]\!]_\rho \to [\![\tau]\!]_\rho$.

In case the last applied rule is

$$\frac{B \vdash M : \sigma \to \tau \quad B \vdash N : \sigma}{B \vdash MN : \tau}\ (\to E)$$

by induction, $\forall \rho.\forall \mathbf{s}.\mathbf{s}(M) \in [\![\sigma \to \tau]\!]_\rho$ and $\mathbf{s}(N) \in [\![\sigma]\!]_\rho$. Then $\mathbf{s}(MN) \in [\![\tau]\!]_\rho$, by definition of $[\![\sigma]\!]_\rho \to [\![\tau]\!]_\rho$.

The cases dealing with the rules involving $\wedge$ come immediately by induction.

Last, (ν) is obvious. ■

Completeness

Proposition 56

Let $\mathbf{O}$ be the set of all substitutions mapping each variable x_i to a term of the shape O^{k_i}, for some $k_i \in \mathbb{N}$.
$B \vdash_\nu M : \sigma$ implies that there exists $\mathbf{o} \in \mathbf{O}$ such that $\mathbf{o}(M) \in \Lambda\ell$-SN.

Lemma 57

$M \in \Lambda\ell$-SN and $M \in \Lambda^0$ implies $M \to^*_\Gamma N \in \Gamma$

Proof. The proof is easy by induction on M. ■

Conclusions

Theorem 58

1. $B \vdash_\nu M : \sigma$ implies $M \in \Gamma$-PV.
2. $B \vdash_\nu M : \sigma$ implies $M \in \Lambda\ell$-SN.

☞ We can find another set Δ of input value having $\Lambda\ell$-SN terms has corresponding potentially valuable term?

Theorem 59

If $\Delta_1 = (\Lambda\ell\text{-SN})^0 \cup \mathrm{Var}$ then Δ_1-PV $= \Lambda\ell$-SN.

Proof. $M \in \Delta_1$-PV implies, by definition, that there is a substitution $\mathtt{s} : Var \to \Delta_1$ such that $\mathtt{s}(M) \to^*_{\Delta_1} N \in \Delta_1$, which implies that N is $\Lambda\ell$-strongly normalizing. All redexes in the considered reduction sequence have arguments in $\Delta_1 \subset \Lambda\ell$-SN, so $\mathtt{s}(M) \in \Lambda\ell$-SN by Lemma 50.(ii). Hence $M \in \Lambda\ell$-SN, by Lemma 50.(i). ■

It is interesting to note that Γ is not a minimal solution. Indeed, $\Gamma_D = \Gamma - \{M \in \Gamma | M \to^*_\Gamma \lambda x.xx\}$ is a proper subset of Γ being a solution.

Theorem 60

There is no minimal set of input values having $\Lambda\ell$-SN terms as potentially valuable!

Proof. Let Δ be a such minimal set.

Let $\omega_3^k \equiv \lambda z.\overbrace{(\lambda x.xxx)...(\lambda x.xxx)}^{k+2}$ for all $k \in \mathbb{N}$.

Since $D_3 \equiv \lambda x.xxx$ is a closed normal form, then $D_3 \in \Delta$ by hypothesis. Note that ω_3^k contains a single redex and $\omega_3^k \to_\Delta \omega_3^{k+1}$ for all $k \in \mathbb{N}$, since $D_3 \in \Delta$.

Each ω_3^k must be Δ-valuable, since $\omega_3^k \in \Lambda\ell$-NF^0; thus, there exists $n \in \mathbb{N}$ such that $\omega_3^n \in \Delta$.

Indeed Δ contains an infinite subset of ω_3^k-terms.
Clearly $\Delta^* = \Delta - \{\omega_3^n | n \in \mathbb{N}\}$ is strictly contained in Δ, but it is again a set of input values such that its potentially valuables terms correspond exactly to that of $\Lambda\ell$-strongly normalizing terms. ■

Solvability

- A term M is Δ-solvable if and only if there is a head Δ-context $C[.] \equiv (\lambda\vec{x}.[.])\vec{N}$ such that:

$$\vec{x} \subseteq \mathrm{FV}(M)$$

$$\vec{N} \subseteq \Delta \qquad \text{and} \qquad C[M] =_\Delta I$$

- A term is Λ-solvable if and only if it has a Λ-head normal form.

- The characterization of Γ-solvable terms rest on the notion of potentially Γ-valuable terms.

Solvable of zero-degree

Definition 61

1. M is of Δ-order 0 if and only if there is no P such that $M \to^*_{\Delta\ell} \lambda x.P$;

2. M is of Δ-order $n \geq 1$ if and only if n is the maximum integer such that $M \to^*_{\Delta\ell} \lambda x_1.M_1$, $M_i \to^*_{\Delta\ell} \lambda x_{i+1}.M_{i+1}$ $(1 \leq i \leq n)$ and M_n is Δ-unsolvable of order 0.
   If such an n does not exists M is of Δ-order ∞.

Lemma 62

The class of Γ-solvable terms is properly included in the class of potentially Γ-valuable terms.

Proof. Let us first prove the inclusion. Let M be Γ-solvable, so there is a head context $(\lambda\vec{x}.[.])\vec{N}$ such that $(\lambda\vec{x}.M)\vec{N} \to^*_\Gamma I$ (since I is in normal form). Assume $\|\vec{x}\| \leq \|\vec{N}\|$
(otherwise, we can consider the context $(\lambda\vec{x}.[.])\vec{N}\,\underbrace{I.....I}_{p}$, where $p = \|\vec{x}\| - \|\vec{N}\|$)

and $\vec{N} \equiv \vec{N}_1\vec{N}_2$ such that $\|\vec{x}\| = \|\vec{N}_1\|$. So $M[\vec{N}_1/\vec{x}]\vec{N}_2 \to^*_\Gamma I$.
But $M[\vec{N}_1/\vec{x}]\vec{N}_2$ Γ-reduces to an abstraction implies that $M[\vec{N}_1/\vec{x}]$ reduces to an abstraction too, i.e it is Γ-valuable. So M is Γ-potentially valuable.

The inclusion is proper, since $\lambda x.DD$ is valuable, and so potentially valuable, but clearly Γ-unsolvable.

Γ-Head Normal Forms

Definition 63

1. The relation $\searrow \subseteq \Lambda \times \Lambda$ is defined inductively in the following way:
- $\lambda x.P \searrow \lambda x.Q$ if and only if $P \searrow Q$,
- $xM_1 \ldots M_m \searrow xN_1 \ldots N_m$ if and only if $M_i \to_{\Delta\ell} N_i \in \Lambda\ell$-NF $(1 \leq i \leq m)$,
- $(\lambda x.P)QM_1 \ldots M_m \searrow R$ if and only if $Q \to_{\Delta\ell} \bar{Q} \in \Lambda\ell$-NF and $P[\bar{Q}/x]M_1 \ldots M_m \searrow R$.

2. M is in Γ-head normal form (Γ-hnf) if and only if $M \equiv \lambda\vec{x}.xM_1...M_m$, and for all $1 \leq i \leq m$, $M_i \in \Lambda\ell$-NF;

   Γ-HNF denotes the set of all Γ-head normal forms.

3. M has Γ-head-normal form if and only if $M \searrow \lambda\vec{x}.xM_1...M_m$ and $M_i \in \Xi$, for all $1 \leq i \leq m$. $\|\vec{x}\|$ is the Γ-order and m is the Γ-degree of M.

Γ-Solvabilty Characterization

Note that Γ-HNF is a proper subclass of Λ-HNF.
In fact, $\lambda x.x(DD) \in \Lambda$-HNF, but $\lambda x.x(DD) \notin \Gamma$-HNF since $DD \notin \Xi$.

Theorem 64 **Γ-Solvability**

A term is Γ-solvable if and only if it has Γ-head-normal form.

A strange result: the theory is semi-sensible,
i.e. each term is equated to an unsolvable term.

A call-by-value recursion operator is a term Z such that $ZM =_\Gamma M(\lambda z.ZMz)$, for all Γ-valuable terms M. Thus $\lambda x.(\lambda y.x(\lambda z.yyz))(\lambda y.x(\lambda z.yyz))$ is a such operator.

Theorem 65

Let Z be a call-by-value recursion operator.
If $B \equiv \lambda xyz.x(yz)$ then $I \approx_{\mathbf{V}} ZB$.

Φ-calculus

☞ We are searching for a set of input values Φ having as potentially valuable the Λ-SN one!

Let Φ be defined as follows:
$\Phi = \mathrm{Var} \cup (\Upsilon)^0$ where $\Upsilon = \cup_i \Upsilon_i$ and Υ_i, Φ_i are defined as follows:

$$\Upsilon_0 = \mathrm{Var} \qquad\qquad \Phi_i = \mathrm{Var} \cup (\Upsilon_i)^0$$

$$\begin{aligned}\Upsilon_{i+1} = \mathrm{Var} \cup\ & \{xM_1 \ldots M_n \mid M_k \in \Upsilon_i (1 \le k \le n)\} \\ \cup\ & \{\lambda\vec{x}.M \mid M \in \Upsilon_i\} \\ \cup\ & \left\{ (\lambda x.P)QM_1 \ldots M_n \;\middle|\; \begin{array}{l} Q \in \Upsilon_i - (\Lambda^0 \cup \mathrm{Var}), \\ M_k \in \Upsilon_i (1 \le k \le n) \\ P[Q/x]M_1 \ldots M_n \to^*_{\Phi_i} R \in \Upsilon_i \end{array} \right\}\end{aligned}$$

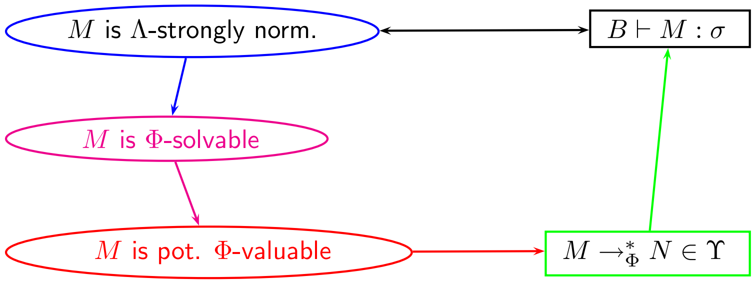
Note that:

1. if $M \in \Phi$ then either $M \in \mathrm{Var}$ or M is closed;
2. if $M \in \Phi$ then M is a Φ-normal form;
3. if $M \in \Phi$ then M is strongly β-normalizing.

Properties

1. Examples: $\Phi_0 = \mathrm{Var}$,
   $\Upsilon_1 = \mathrm{Var} \cup \{xy_1 \ldots y_n \mid y_i \in Var\} \cup \{\lambda\vec{x}.y \mid y \in \mathrm{Var}\}$ and $\Phi_1 = \mathrm{Var} \cup \{\lambda x_1 \ldots x_m.x_i\}$
2. $\Phi = \cup_i \Phi_i$
3. Φ_i is a set of input value, for all $i \in \mathbb{N}$
4. Φ is a set of input values
5. Υ and Υ_i are not sets of input values, for all $i \in \mathbb{N}$
6. $\Upsilon_i \subset \Upsilon_{i+1}$, $\Phi_i \subset \Phi_{i+1}$ and $\to_{\Phi_i} \subseteq \to_{\Phi_{i+1}}$, for all $i \in \mathbb{N}$
7. $\Upsilon_i \subset \Upsilon$, $\Phi_i \subset \Phi$ and $\to_{\Phi_i} \subseteq \to_{\Phi}$, for all $i \in \mathbb{N}$
8. $\Upsilon^0 = \Phi^0$
9. $M \in \Phi^0$ implies $M \equiv \lambda z.P$, for some $z \in \mathrm{Var}$ and $P \in \Upsilon$ (so $\Phi \subseteq \Gamma$)
10. $\Phi \subseteq \Upsilon$ and $\Upsilon \subseteq \Phi$-NF
11. Φ-NF $\not\subseteq \Upsilon$ and Φ-NF $\not\subseteq \Phi$, in fact $\lambda z.(\lambda x.D)(zI)D \in \Phi$-NF since $zI \notin \Phi$,
    but $\lambda z.(\lambda x.D)(zI)D \notin \Upsilon$ and $\lambda z.(\lambda x.D)(zI)D \notin \Phi$

Some Properties of Φ-calculus



The typing system $\vdash$ is obtained from that considered in slide 56 by avoiding the use of the rule ν.

Minimality

Lemma 66

Let Ψ be a solution to our PV-Problem, such that $\Psi = \mathrm{Var} \cup (\Psi)^0$ and $\Psi \subseteq \Psi$-NF. Then Ψ is a minimal solution.

Proof. Let $\Delta^\star$ be a set of input values.
We will prove that, if the set of potentially $\Delta^\star$-valuable terms coincides with the set of the strongly Λ-normalizing terms and $\Delta^\star \subseteq \Psi$ then $\Delta^\star = \Psi$. Clearly $\Delta^\star = \Psi$ if and only if $(\Delta^\star)^0 = (\Psi)^0$, since $\Delta^\star \subseteq \Psi$, $\Psi = \mathrm{Var} \cup (\Psi)^0$ and the Definition 1.(iii) . Let $M \in (\Psi)^0$. Note that M is Ψ-valuable, potentially Ψ-valuable, in Ψ-normal form and also a closed strongly Λ-normalizing term. Thus, M is potentially $\Delta^\star$-valuable by hypothesis and $M \in (\Lambda)^0$ implies that M is $\Delta^\star$-valuable. But $\Delta^\star \subseteq \Psi$ implies Ψ-NF $\subseteq \Delta^\star$-NF, hence $M \in \Delta^\star$-NF. This, together with the fact that M is $\Delta^\star$-valuable implies that M must already be a $\Delta^\star$-value, i.e. $M \in \Delta^\star$ and the proof is done.

Corollary 67

Φ is a minimal solution to our PV-Problem!

It is worthy to say that, although Φ is minimal, it is not the minimum one.

In fact, the minimum solution to the following equations:

$$\begin{aligned}\Theta = \;& \{\lambda x_0 ... x_n.y \mid y \neq x_i \;\; (0 \leq i \leq n)\;\} \cup \\ & \{x M_1 ... M_n \mid M_k \in \Theta \; (1 \leq k \leq n)\} \cup \{\lambda \vec{x}.M \mid M \in \Theta\} \cup \\ & \{(\lambda x.P) Q M_1 ... M_n \mid Q, M_1, ..., M_n \in \Theta,\; Q \notin \Delta,\; P[Q/x] M_1 ... M_n \to_\Delta^* R \in \Theta\} \\ \Delta = \;& \{\lambda x_0 ... x_n.y \mid y \neq x_i \;\; (0 \leq i \leq n)\;\} \cup (\Theta)^0\end{aligned}$$

is also a minimal solution to Problem 1. Sets Θ and Φ are not comparable, in fact $\lambda x.y \in \Theta$ but not in Φ, while $I(\lambda x.y) \in \Phi$ but not to Θ.

Conclusions

Proposition 68

For every term M, there is an effective procedure building two Λ-normal forms, P_M and Q_M, such that $P_M Q_M \to^*_\Lambda M$.

Proof. The proof is by induction on the structure of M. If $M \equiv x$, then $P_x Q_x \equiv (\lambda y.y)x$. If $M \equiv \lambda x.N$, then by induction there are P_N and Q_N such that $P_N Q_N \to^*_\Lambda N$. So $P_M Q_M \equiv (\lambda yx.yP_N(yQ_N))I$, where y is fresh. If $M \equiv NR$, then $P_M Q_M \equiv (\lambda y.yP_N Q_N(yP_R Q_R))I$, where y is fresh.

Theorem 69

There isn't a decidable set of input values which is a solution of our PV-Problem.

Proof. The proof is based on the following remarks. Assume Δ be any set of input values, $M' \in \Lambda$-NF^0 and $M^* \in \Delta$-NF^0. Then

(1) $M' \in \Lambda$-SN^0 and $M' \in \Delta$-NF^0;
(2) $M^* \in \Delta$ if and only if M^* is Δ-valuable (since M^* is in Δ-normal form) if and only if M^* is Δ-potentially valuable (since M^* is closed).

Assume $\Delta^\#$ be a solution of PV-Problem.
Since Remark (1), all closed Λ-normal forms must belong to $\Delta^\#$.
As an example $D \equiv \lambda x.xx$ is a closed Λ-normal form (hence, it belongs to Λ-SN^0) and consequently D is $\Delta^\#$-potentially valuable (by hypothesis on $\Delta^\#$), it is $\Delta^\#$-valuable (since it belongs to $\Delta^\#$-NF) and it belongs to $\Delta^\#$ (since it is closed). Henceforth, xx does not belong to $\Delta^\#$, since input values need to be closed under substitution and DD is not a strongly normalizing term.
Let $M \equiv \lambda x.(\lambda z.P)(xx)Q$ where $P, Q \in \Lambda$-NF^0. Note that M is Λ-strongly normalizing if and only if PQ is Λ-strongly normalizing. Since $M \in \Delta^\#$-NF^0, by Remark (2), M is $\Delta^\#$-potentially valuable if and only if it belongs to $\Delta^\#$, but M belongs to $\Delta^\#$ exactly when PQ is Λ-strongly normalizing.
Thus the problem of $\Delta^\#$ membership is reduced to that one of deciding if a term which is an application of two Λ-normal forms is Λ-strongly normalizing. But, by Property 68, such problem is equivalent to the general Λ-strongly normalization problem, which is well known to be undecidable.

Selected References

- Luca Paolini, Elaine Pimentel and Simona Ronchi Della Rocca. Strong Normalization from an unusual point of view. Theoretical Computer Science, ??(??):??, 2009.
- Luca Paolini, Elaine Pimentel and Simona Ronchi Della Rocca. An Operational Characterization of Strong Normalization. FOSSACS 2006. Lecture Notes in Computer Science, volume 3921, pages 367-381, 2006.
- Luca Paolini and Simona Ronchi Della Rocca. Parametric parameter passing lambda-calculus. Information and Computation, 189(1):87-106, February 2004.
- Simona Ronchi Della Rocca and Luca Paolini. The Parametric Lambda-Calculus. A Metamodel for Computation. Texts in Theoretical Computer Science: An EATCS Series. Springer-Verlag, Berlin, 2004.
- Luca Paolini. Lambda-theories: some investigations. Università degli Studi di Genova and Université de la Méditerranée (Aix-Marseille II), 2004.

Downloads are available at http://www.di.unito.it/~paolini/ in "publications".