Lambda-Calculus (III-3)

jean-jacques.levy@inria.fr
Tsinghua University,
September 10, 2010

http://moscova.inria.fr/~levy/courses/tsinghua/lambda

Reminders

- Redexes may be tracked with **residuals**
- One can define **parallel** reduction $\xrightarrow{\mathcal{F}}$ of a given set $\mathcal{F}$ of redexes
  by considering any of its finite developments.
- Lemma of **parallel moves** (other version of confluency lemma `1111`)
- **Cube lemma** (consistency of residual relation w.r.t. finite developments)
- The labeled calculus was a technical tool to name redexes and prove Curry's **Finite Development Theorem**.

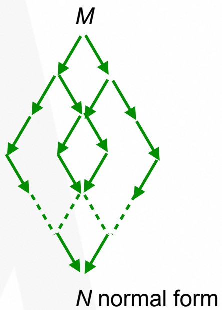
Plan

- Normalization
- Strong normalization
- Standardization theorem
- Normalization strategies

Termination

Strong Normalization

- *M* is **strongly normalizable** iff every reduction from *M* is finite



- **Exercice:** which of following terms is strongly normalizable ?

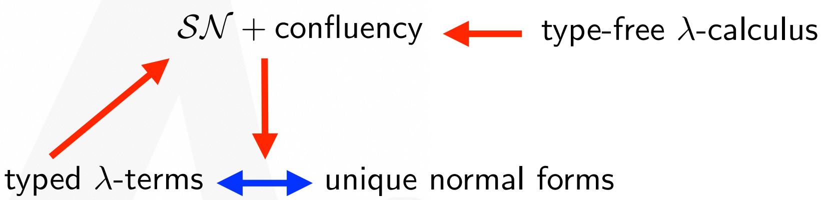
  $I, II, \Delta\Delta, \Delta I, Y, YI, YK, KI(\Delta\Delta)$
  where $I = \lambda x.x,\ \Delta = \lambda x.xx,\ K = \lambda x.\lambda y.x$
  and $Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$.

Non termination

- In a fully expressive language, you have non-termination:
- in PCF + Y operator, in Ocaml, in Haskell, some terms are not in $\mathcal{SN}$
- Confluency ensures deterministic calculations
- but possibly not terminating with a normal form.

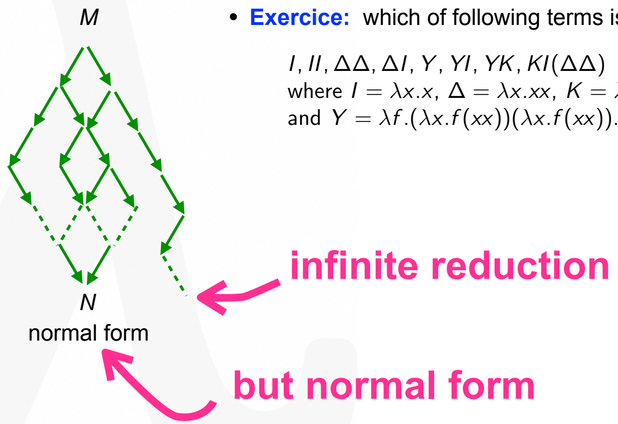
Strong Normalization

- In typed lambda-calculi, all terms are strongly normalizable:
- in 1st-order typed calculus, in system *F* , *F-omega,* terms are in $\mathcal{SN}$
- terms of Coq are also strongly normalizable.



Normalization

- *M* is **normalizable** iff a reduction from *M* leads to a normal form.



- **Exercice:** which of following terms is normalizable ?

  $I, II, \Delta\Delta, \Delta I, Y, YI, YK, KI(\Delta\Delta)$
  where $I = \lambda x.x,\ \Delta = \lambda x.xx,\ K = \lambda x.\lambda y.x$
  and $Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$.

Normalization strategies

- Suppose M is normalizable. Is there a strategy to reach the normal form ? (**normalizing strategy**)
- Conversely, if M has an infinite reduction, is there a strategy to fall in an infinite reduction ? (**perpetual strategies**) `[see Barendregt + Klop]`
- Take: $M = (\lambda x.y)(\Delta\Delta) \xrightarrow{*} y$

  but $(\lambda x.y)(\Delta\Delta) \longrightarrow (\lambda x.y)(\Delta\Delta) \longrightarrow \cdots$
- Take: $M = I(\Delta(KI(\Delta\Delta))) \xrightarrow{*} I$

  but $M = I(\Delta(KI(\Delta\Delta))) \longrightarrow I(\Delta(KI(\Delta\Delta))) \longrightarrow \cdots$
- Take: $M = I(\Delta(K(\Delta\Delta)I)) \xrightarrow{*} \Delta\Delta \longrightarrow \Delta\Delta \longrightarrow \cdots$

  but $M \xrightarrow{*} N$ in normal form ??

Normalization strategies

- In programming languages, evaluation strategies could be:
  - **call-by-value**: compute value of arguments of functions and pass values to the function parameters (Ocaml, Java)
  - **call-by-name**: pass symbolic expression of arguments to the function parameters and calculate them when needed.
  - **call-by-need**: variation of call-by-name in order to avoid recalculations of arguments (lazy languages -- Haskell)
- there are also CBV, CBN strategies in the lambda-calculus (we don't do it here)
- Call-by-need is more complex [JJL'78, Lamping'90, Gonthier-Abadi-JJL'92]

Normalization strategies

- Take: $M = Y'(KI) \xrightarrow{*} I$

  but $M = Y'(KI) \xrightarrow{*} KI(Y'(KI)) \xrightarrow{*} KI(KI(Y'(KI))) \xrightarrow{*} \cdots$

  where $Y' = (\lambda xy.y(xxy))(\lambda xy.y(xxy))$
- Comparable to evaluation strategies in programming languages:

```
static int f (int x, int y) {
    if (x == 0)
        return 1;
    else
        return f (x-1, f(x, y));
}
```

  what is value of `f (1, 0)` ???
- In PCF, it would be:

  $Y(\lambda f\, x\, y.\, \texttt{ifz}\ x\ \texttt{then}\ 1\ \texttt{else}\ f(x-1)\,(f\ x\ y))\ 1\ 0$

Standard reduction

Redex R is **to the left of** redex S if the λ of R is to the left of the λ of S.

$$M = \cdots \underbrace{(\lambda x.A)B}_{R} \cdots \underbrace{(\lambda y.C)D}_{S} \cdots$$

or

$$M = \cdots \underbrace{(\lambda x. \cdots \underbrace{(\lambda y.C)D}_{S} \cdots)B}_{R} \cdots$$

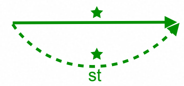
or

$$M = \cdots \underbrace{(\lambda x.A)(\cdots \underbrace{(\lambda y.C)D}_{S} \cdots)}_{R} \cdots$$

The reduction $M = M_0 \xrightarrow{R_1} M_1 \xrightarrow{R_2} M_2 \cdots \xrightarrow{R_n} M_n = N$ is standard iff for all i, j $(0 < i < j \leq n)$, redex R_j is not a residual of redex R'_j to the left of R_i in M_{i-1}.

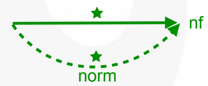
Standardization

- **Theorem [standardization] (Curry)** Any reduction can be standardized.

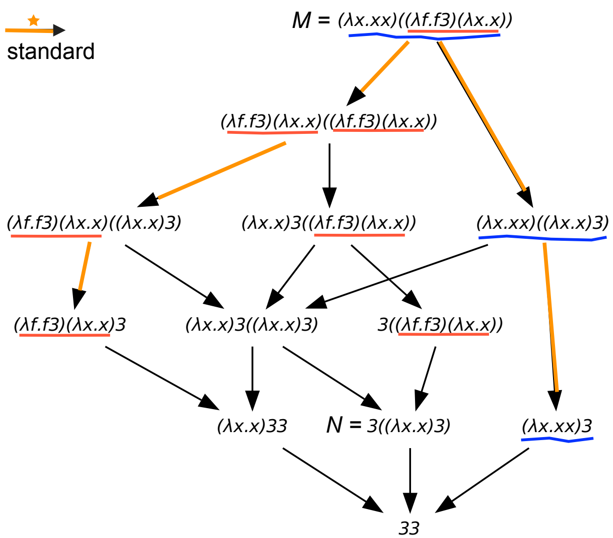


- The **normal reduction** (each step contracts the leftmost-outermost redex) is a standard reduction.

- **Corollary [normalization]** If M has a normal form, the normal reduction reaches the normal form.



Standard reduction



Standardization lemma

- **Notation:** write $R <_\ell S$ if redex R is to the left of redex S.

- **Lemma 1** Let R, S be redexes in M such that $R <_\ell S$. Let $M \xrightarrow{S} N$.

Then $R/S = \{R'\}$. Furthermore, if $T' <_\ell R'$, then $\exists T,\ T <_\ell R, T' \in T/S$.
[one cannot create a redex through another more-to-the-left]

- **Proof of standardization thm:** [Klop] application of the finite developments theorem and previous lemma.

Standardization axioms

- 3 axioms are sufficient to get lemma 1
- **Axiom 1 [linearity]** $S \not\leq_\ell R$ implies $\exists! R',\ R' \in R/S$
- **Axiom 2 [context-freeness]** $S \not\leq_\ell R$ and $R' \in R/S$ and $T' \in T/S$ implies $T \Re R$ iff $T' \Re R'$ where $\Re$ is $<_\ell$ or $>_\ell$
- **Axiom 3 [left barrier creation]** $(R <_\ell S$ and $\nexists T',\ T \in T'/S)$ implies $R' <_\ell T$ where $R/S = \{R'\}$

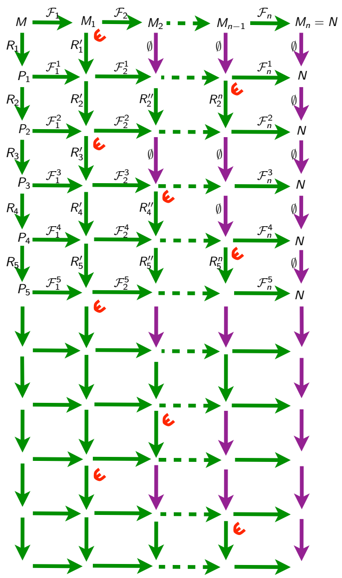
Standardization proof

- **Proof:**

Each square is an application of the lemma of parallel moves. Let ρ_i be the horizontal reductions and σ_j the vertical ones. Each horizontal step is a parallel step, vertical steps are either elementary or empty.

We start with reduction ρ_0 from M to N. Let R_1 be the leftmost redex in M with residual contracted in ρ_0. By lemma 1, it has a single residual R_1' in M_1, M_2, ...until it belongs to some $\mathcal{F}_k$. Here $R_1' \in \mathcal{F}_2$. There are no more residuals of R_1 in M_{k+1}, M_{k+2}, ...

Let R_2 be leftmost redex in P_1 with residual contracted in ρ_1. Here the unique residual is contracted at step n. Again with R_3 leftmost with residual contracted in ρ_2. Etc.

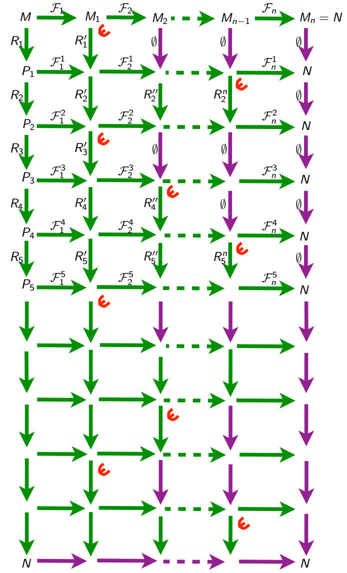


Standardization proof

- **Proof (cont'd):**

Now reduction σ_0 starting from M cannot be infinite and stops for some p. If not, there is a rightmost column σ_k with infinitely non-empty steps. After a while, this reduction is a reduction relative to a set $\mathcal{F}_i^j$, which cannot be infinite by the Finite Development theorem.

Then ρ_p is an empty reduction and therefore the final term of σ_0 is N.



Standardization proof

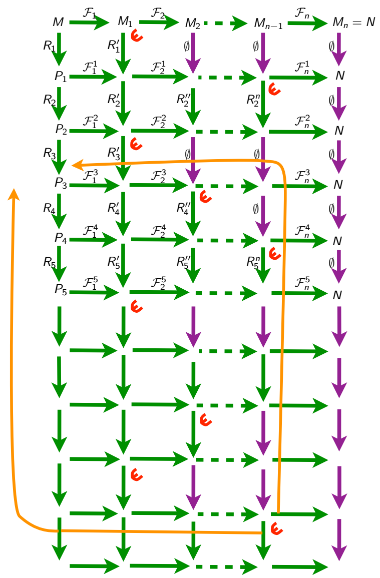
- **Proof (cont'd):**

We claim σ_0 is a standard reduction. Suppose R_k $(k > i)$ is residual of S_i to the left of R_i in P_{i-1}.

By construction R_k has residual S_k^j along ρ_{i-1} contracted at some j step. So S_k^j is residual of S_i.

By the cube lemma, it is also residual of some S_i^j along σ_{j-1}. Therefore there is S_i^j in $\mathcal{F}_i^j$ residual of S_i leftmore or outer than R_i.

Contradiction.



Homeworks

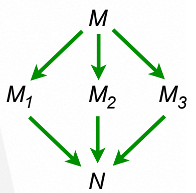
CENTRE DE RECHERCHE COMMUN
INRIA MICROSOFT RESEARCH

Exercices

1- Show that $\Delta\Delta(II)$ has no normal form when $I = \lambda x.x$ and $\Delta = \lambda x.xx$.

2- Show that $\Delta\Delta M_1 M_2 \cdots M_n$ has no normal form for any $M_1, M_2, \ldots M_n$ $(n \geq 0)$.

3- Show there is no M whose reduction graph is exactly the following:



4- Show that rightmost-outermost reduction may miss normal forms.

5- Show that if $M \xrightarrow{*} \lambda x.N$, there is a minimal N_0 such that for all P, such that if $M \xrightarrow{*} \lambda x.P$, then $N_0 \xrightarrow{*} P$.