Linear Logic and Optimal Reductions

Optimal Reductions

Stefano Guerrini

LIPN - Institut Galilée, Université Paris Nord 13

MPRI - Linear logic and logical paradigms of computation

2009–2010 (v. 04/12/09)

Lambda calculus

- The λ-calculus is the main reference for the semantics and the implementations of functional programming languages.
  - The β-rule $(\lambda x.t)s \to s[t/x]$ being the basic computational mechanism.
  - Computing corresponding to finding the (unique) normal form, if any (in some cases, according to a given reduction strategy)
- In accord to the standard behaviour of programming languages, the application of β-rule may be restricted, introducing a corresponding new definition of normal forms.
  - *call-by-value* reduces a redex only if the argument is a value. Given a set $V \subset \Lambda$ of values

    $$(\lambda x.t)v \to s[v/x] \qquad \text{only if } v \in V$$

    By contrast, the standard unrestricted β-rule is usually referred to as *call-by-name*.
  - *lazy* or *weak* evaluation does not allow to reduce redexes in the scope of a λ-abstraction.

Reduction strategies

A reduction strategy is a rule that fixes the (set or the sequence of) redex(s) to reduce while computing the normal form of a term.

Let Λ be the set of the λ-terms.

Definition

A *reduction strategy* is a function

$$F : \Lambda \to \Lambda \qquad s.t. \qquad t \to^* F(t)$$

- $t \in \Lambda_{nf}$ implies $F(t) = t$, where $\Lambda_{nf} \subset \Lambda$ is the set of the normal forms.
- A reduction strategy F is *one-step*, when $t \to F(t)$ for every $t \notin \Lambda_{nf}$.
- A reduction strategy is *recursive* when F is recursive.
- A recursive reduction strategy is effective when it can be computed in a relatively simple way: constant, linear time, or even polynomial, it may depend on our purposes (theory of computation or effective implementations).

Reduction strategies (cont.)

- Every reduction strategy F induces an F-reduction

$$t \to F(T) \to F^2(t) \to \cdots$$

We shall use $L_F(t)$ to denote the length of the F-reduction of t.

$$L_F(t) = \begin{cases} \min\{n \mid F^n(t) \in \Lambda_{nf}\} & \text{when } F^n(t) \in \Lambda_{nf} \text{ for some } n \\ \infty & \text{otherwise} \end{cases}$$

- A reduction strategy F is *normalizing* when it computes the normal-form, if any, of every term t.

$$t \in \Lambda_{nf} \qquad \Longrightarrow \qquad F^n(t) \in \Lambda_{nf} \quad \text{for some } n$$

- A reduction strategy F is *Church-Rosser* when it reduces β-equivalent terms to the same term. For every $s, t \in \Lambda$

$$s =_\beta t \qquad \Longrightarrow \qquad F^n(s) = F^m(t) \quad \text{for some } n, m$$

Leftmost reduction

- Let us write $t \to_{\mathrm{lm}} s$ when $t \to s$ by reducing the leftmost redex of t.
- The *leftmost* (outermost) reduction is the reduction strategy defined by

$$\mathsf{lm}(t) = \begin{cases} t & \text{if } t \in \Lambda_{nf} \\ s & \text{when } t \notin \Lambda_{nf} \text{ and } t \to_{\mathrm{lm}} s \end{cases}$$

Proposition

The leftmost reduction strategy is effective and normalizing.

- A reduction strategy is *quasi leftmost* if

$$\forall i, \exists j \geq i : \qquad F^j(t) \to_{\mathrm{lm}} F^{j+1}(t)$$

Proposition

Let Q be a quasi leftmost reduction strategy. $L_Q(t) = \infty$ iff t has no normal form.

- Every quasi leftmost reduction strategy is normalizing.

Innermost reduction

- The innermost reduction is the reduction strategy that reduces the (rightmost) innermost redex.
- The innermost reduction $\to_{\mathrm{im}}$ is not normalizing.

$$(\lambda x.s)\omega \to_{\mathrm{im}} (\lambda x.s)\omega \to_{\mathrm{im}} (\lambda x.s)\omega \to_{\mathrm{im}} \cdots$$

  where $\omega = \Delta\Delta$ and $\Delta = \lambda x.xx$
- The innermost reduction corresponds to an evaluation strategy that always tries to evaluate the argument of a function before to replace it into the body of the function.

Perpetual reduction

- A reduction strategy F is *perpetual* when it takes an infinite reduction path for every term that is not strongly normalizing.

  $$t \text{ has an infinite reduction path} \qquad \Longrightarrow \qquad L_F(t) = \infty$$

  (and the F reduction of t does not contain empty reduction steps, i.e., $F^n(t) \to^+ F^{n+1}(t)$ for every n).
- An (effective) perpetual strategy can be defined by a simple modification of leftmost reduction in which we avoid to erase terms that contains redexes.
  - A redex $(\lambda x.u)v$ is an erasing or K-redex when $x \notin \mathrm{FV}(t)$.
  - If $r = (\lambda x.u)v$ is the leftmost redex of t, and $C[]$ is the context s.t. $t = C[r]$, we define

  $$F_\infty(C[(\lambda x.u)v]) = \begin{cases} C[u[v/x]] & \text{when } r \text{ is not an erasing or } v \in \Lambda_{nf} \\ C[(\lambda x.u)F_\infty(v)] & \text{when } r \text{ is an erasing and } u \notin \Lambda_{nf} \end{cases}$$
- The key idea is that, when we erase a term v, not only v does not contain any redex, but no reduction can lead to the creation of a redex in v (indeed, no reduction can modify v).

One-step optimal strategies

- Given two reduction strategies F and G we can try to compare them in terms of efficiency, by comparing the lengths of their reduction paths.
- F is better than G, if $L_F(t) \leq L_G(t)$ for every term t.
- A one-step strategy is *optimal* if it is better than any other one-step strategy.
- While we can easily define an optimal one-step reduction strategy, any of such optimal strategy is not recursive.

Proposition

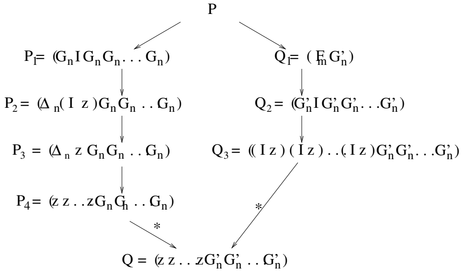
There exists no optimal one-step reduction strategies.

No optimal one-step strategy: an example

$$F_m = \lambda x.(x\, I\, \underbrace{x\, x \ldots x}_{m \text{ times}}) \qquad I = \lambda x.x \qquad \Delta_n = \lambda x.(\underbrace{x\, x \ldots x}_{n \text{ times}})$$

$$G_n = \lambda y.(\Delta_n\,(y\,z)) \qquad G'_n = \lambda y.(\underbrace{(y\,z)\,(y\,z) \ldots (y\,z)}_{n \text{ times}})$$

$$P = F_m G_n = F_m(\lambda y.\Delta_n(yz))$$



Every reduction strategy duplicates work

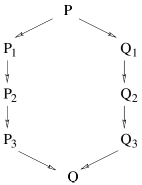
- lhs is lazy innermost: $m + 4$ β-reductions.
- rhs is lazy leftmost: $n + 3$ β-reductions.

No optimal one-step strategy: an example (cont.)

- Informally, we may say that in the previous example there are only four distinct redexes:
  1. $F_m G_n$ and $F_m G'_n$
  2. $\Delta_n(yz)$ and $\Delta_n(Iz)$
  3. Iz
  4. $G_n I$ and $G'_n I$
- These redexes, which appear in distinct terms along the reductions, have the same common origin:
  1. $F_m G'_n$ is a reduct of the original redex $F_m G_n$
  2. $\Delta_n(Iz)$ is obtained from $\Delta_n(yz)$ by replacing I for the free variable y
  3. all the redexes Iz come from the application yz in G_n
  4. $G_n I$ and $G'_n I$ is obtained by replacing G_n and G'_n for x in the application xI of F_m, respectively, and G'_n is a reduct of G_n

An ideal sharing machine

- If we want to avoid useless duplication of work, we must
  - avoid the duplication of redexes (e.g., $\Delta_n(yx)$)
  - avoid the duplication of applications that will be instantiated to redexes (e.g., xI and yz), we may say that such applications are a sort of *virtual redexes*.
- In the example, no one-step reduction strategy can avoid all such duplications at the same time.
- We might hope to have some kind of *sharing* mechanism (implementing some multistep reduction strategy) that keeps a unique representation of some parts of terms (e.g., the m copies of G_n in the leftmost derivation) reducing them just once.
- The reductions of such a sharing machine would correspond to the diagram



Sharable redexes

- In order to find an ideal/practical sharing machine, we need a robust and stable definition of sharable redexes (or more in general of sharable terms, subterms, or contexts).
- We shall give a direct approach based on a detailed analysis of equivalent reduction paths.
- Using the correspondence between λ-terms and nets (and the so-called Geometry of Interaction), we shall be able to reformulate the notion of sharable redexes by means of paths on the starting net.
  - In nets, a β-redex is a direct connection between the lhs of an application and the top of a λ-abstraction.
  - Studying net reductions we shall see that some paths disappear, while other paths are stretched/contracted or duplicated.
  - Two redexes are sharable when they correspond to the same path in the starting net that eventually reduce to a β-redex.
- Using nets we can reestablish the original idea of sharable redexes:
  - two redexes are sharable iff we can trace back to a common origin on the starting net/term.

Redexes: copying and creation

- During a reduction, redexes may be

  copied:

  $$(\lambda x.sxx)t \to stt$$

  makes two copies of every redex in t.

  or *created*:

  $$(\lambda y.ys)(\lambda x.t) \to (\lambda x.t)s$$
  $$(\lambda y.(\lambda x.t))us \to (\lambda x.t[u/y])s$$
  $$(\lambda y.y)(\lambda x.t)s \to (\lambda x.t)s$$

- In some cases, two redexes must be considered as copies of the same redex even if they are created independently:

  $$(\lambda y.(\lambda x.sxx)(yz))(\lambda x.t) \to (\lambda y.s(yz)(yz))(\lambda x.t) \to s(\underline{(\lambda x.t)z})(\underline{(\lambda x.t)z})$$

  but, if we rearrange the reduction

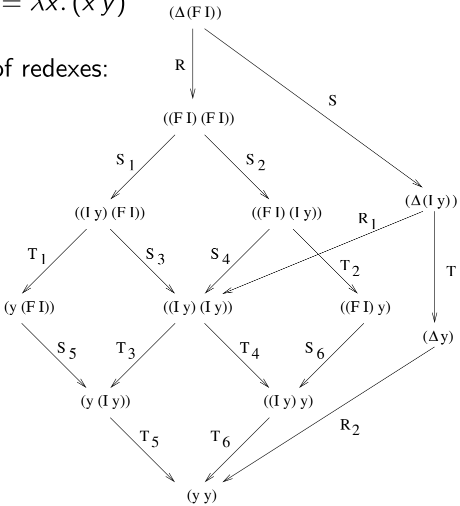
  $$(\lambda y.(\lambda x.sxx)(yz))(\lambda x.t) \to (\lambda x.sxx)(\underline{(\lambda x.t)z}) \to s(\underline{(\lambda x.t)z})(\underline{(\lambda x.t)z})$$

  the two redexes can be obtained as copies of the same redex. In some sense, we can find a common ancestor of the two redexes.

An example

$$\Delta = \lambda x.\,(x\,x) \quad I = \lambda x.\,x \quad F = \lambda x.\,(x\,y)$$

- Intuitively, there are just three kinds of redexes: $R = \Delta(F\,I) \quad S = F\,I \quad T = I\,y$
- all the redexes R_i and S_i are residuals of R and S
- The two redexes T_1 and T_2 clearly look sharable (and they are indeed 'shared' as the unique redex T in the innermost reduction on the left) but are not residuals of any (common) redex.
- The redexes T_3 and T_4 are residuals of the same redex T (after R_1). T_3 (T_4) is also a residual of T_1 (T_2). All these redexes seem to be in a same family.



Residuals

Informally, a *residual* of a redex is what is left of a redex after some other redex has been reduced. Residuals can be identified by:

- Underlining every redex of a term M and associating a distinct name to each underlining

$$\underline{(\Delta\, \underline{(F\, I)}_S)}_R$$

- Assuming that β-reduction preserves marking and naming while copying terms

$$\underline{(\Delta\, \underline{(F\, I)}_S)}_R \to_R (\underline{(F\, I)}_S\, \underline{(F\, I)}_S) \to_{S_1} ((I\, y)\, \underline{(F\, I)}_S) \to_{S_3} ((I\, y)\, (I\, y))$$

- Given a reduction ρ of the term M, the *residual* of a redex R of M under ρ is the set ρ/R of the redexes marked by an underlining with name R.
- The redex R is the *ancestor* of any redex in R/ρ
- Any redex in R/ρ is a *residual redex* of R (under ρ)

Finite developments

- Let $\mathcal{F}$ be a set of redexes of a λ-term M. A reduction $\rho : M \to^* N$ is *relative to* $\mathcal{F}$ when no redex underlined by a name $R \notin \mathcal{F}$ is reduced along the marked reduction corresponding to ρ.
- A reduction $\rho : M \to^* N$ is a (complete) *development* of a set of redexes $\mathcal{F}$ of M when
  - ρ is relative to $\mathcal{F}$ and,
  - after the marked reduction corresponding to ρ, N does not contain any redex underlined by a name $R \in \mathcal{F}$.

Theorem (finite developments)

Let $\mathcal{F}$ be a set of redexes of a λ-term.

1. *There is no infinite reduction relative to $\mathcal{F}$.*
2. *All developments end at the same term.*
3. *For any redex R of M and any pair of developments ρ and σ relative to $\mathcal{F}$, we have that $R/\rho = R/\sigma$.*

Parallel reduction

- $\mathcal{F} : M \to^* N$ denotes a *parallel reduction* in which a whole set of redexes $\mathcal{F}$ is simultaneously reduced in one step.
- The result of the parallel reduction of a set of redexes $\mathcal{F}$ can be obtained by taking any development of $\mathcal{F}$.
- The notion of residual can be directly extended to parallel reductions and to set of redexes

$$R' \in \mathcal{G}/\mathcal{F} \quad \text{iff} \quad R' \in R/\mathcal{F} \text{ for some } R \in \mathcal{G}$$

- Let us define $\mathcal{F} \sqcup \mathcal{G} = \mathcal{F}(\mathcal{G}/\mathcal{F})$

Lemma (parallel moves)

Let $\mathcal{F}$ and $\mathcal{G}$ be set of redexes of a λ-term M. We have that:

1. *$\mathcal{F} \sqcup \mathcal{G}$ and $\mathcal{G} \sqcup \mathcal{F}$ ends at the same expression;*
2. *$\mathcal{H}/(\mathcal{F} \sqcup \mathcal{G}) = \mathcal{H}/(\mathcal{G} \sqcup \mathcal{F})$.*

- The parallel moves lemma suggests that $\mathcal{F} \sqcup \mathcal{G} \equiv \mathcal{G} \sqcup \mathcal{F}$ w.r.t. some equivalence of reductions well-defined w.r.t. the definition of residuals.

Permutation equivalence

We want to equate reductions in which redexes are permuted preserving their grouping, as in $\mathcal{F} \sqcup \mathcal{G}$ and $\mathcal{G} \sqcup \mathcal{F}$.

- The parallel reduction of the empty set of redexes $\emptyset : M \to M$ is distinct from the empty reduction ϵ, but it seems reasonable to ask $\emptyset \equiv \epsilon$.
- We expect that $\mathcal{F} \sqcup \emptyset \equiv \mathcal{F} \equiv \emptyset \sqcup \mathcal{F}$
- and that $\mathcal{F} \cup \mathcal{G} \equiv \mathcal{F} \sqcup \mathcal{G}$, even if $\mathcal{F} \cup \mathcal{G} \neq \mathcal{F} \sqcup \mathcal{G}$, since they denote different sets of reduction sequences.

Definition (permutation equivalence)

$\equiv$ is the smallest equivalence such that:

1. $\mathcal{F} \sqcup \mathcal{G} \equiv \mathcal{G} \sqcup \mathcal{F}$, when $\mathcal{F}$ and $\mathcal{G}$ are set of redexes of the same λ-term;
2. $\emptyset = \epsilon$;
3. $\rho\sigma\tau \equiv \rho\sigma'\tau$, when $\sigma \equiv \sigma'$.

In other words, the *permutation equivalence* of reductions $\equiv$ is the smallest congruence with respect to composition of reductions satisfying the parallel moves lemma and elimination of empty steps (ϵ reductions).

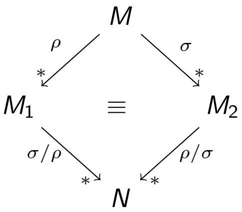
Diamond property

- The notion of residual extends to reductions.

$$
\begin{aligned}
\epsilon/\rho &= \epsilon \\
(\sigma\mathcal{F})/\rho &= (\sigma/\rho)(\mathcal{F}/(\rho/\sigma))
\end{aligned}
$$

Theorem (diamond property)

For any pair of reductions ρ and σ starting from the same term, we have that $\rho\,(\sigma/\rho) \equiv \sigma\,(\rho/\sigma)$.



Equivalence of reductions by permutation

- The diamond property is the base of the *equivalence by permutation*:

  two reductions ρ and σ are permutation equivalent if, by composing instances of the previous diagram, we can get a commuting diagram in which the composition of the external reductions yields ρ and σ.

- The diamond property is a strong version of the Church-Rosser (confluence) property:

  it proves the confluence of the calculus, showing at the same time how to complete any pair of reductions ρ and σ in order to get the same result.

Families of redexes

Definition (copy)

A redex S with history σ is a *copy* of a redex R with history ρ, written $\rho R \leq \sigma S$, if and only if there is a derivation τ such that $\rho\tau$ is permutation equivalent to σ ($\rho\tau \equiv \sigma$) and S is a residual of R with respect to τ ($S \in R/\tau$).

Definition (family)

The symmetric and transitive closure of the copy relation is called the *family* relation, and will be denoted with $\simeq$.

Zig-zag relation

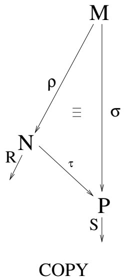
$$\rho R \simeq \sigma S$$

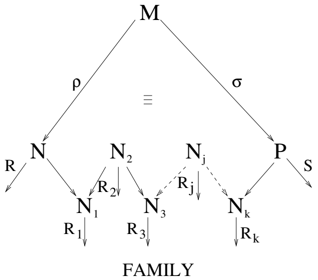
iff there is a finite sequence

$$\rho R = \tau_0 R_0, \quad \tau_1 R_1, \quad \dots, \quad \tau_k R_k \leq \sigma S$$

such that, for $i = 1, \dots, k$

$$\tau_{i-1} R_{i-1} \leq \tau_i R_i \quad \text{or} \quad \tau_i R_i \leq \tau_{i-1} R_{i-1}$$



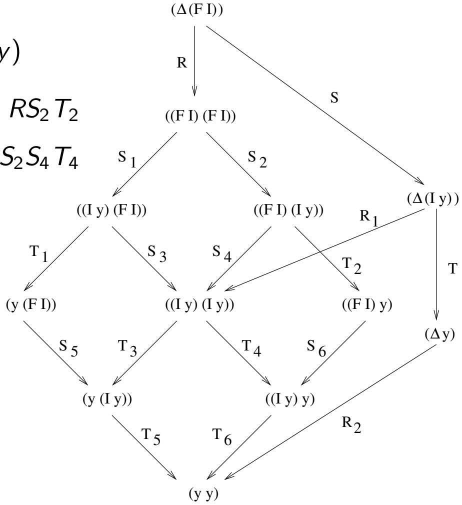


An example

$\Delta = \lambda x.(x\,x) \quad I = \lambda x.x \quad F = \lambda x.(x\,y)$

$RS_1T_1 \leq RS_1S_3T_3 \geq ST \leq RS_2S_4T_4 \geq RS_2T_2$

$ST \simeq RS_1T_1 \simeq RS_2T_2 \simeq RS_1S_3T_3 \simeq RS_2S_4T_4$



Equivalence of histories

Lemma

Let $\rho \equiv \rho'$ *and* $\sigma \equiv \sigma'$. *Then:*

1. $\rho R \leq \sigma S$ *iff* $\rho' R \leq \sigma' S$
2. $\rho R \simeq \sigma S$ *iff* $\rho' R \simeq \sigma' S$

- In order to check that two redexes are in the same family, we can consider any pair of histories of the redexes, provided that they are permutation equivalent to the given ones. In particular, we can restrict our analysis to *standard derivations*.

Standard derivations

Definition (standard derivation)

A derivation $R_1 R_2 \cdots$ is *standard* when R_i is not a residual of any redex at the left of R_j, for any i and any $j < i$.

- In the case of parallel reductions, the derivation $\mathcal{F}_1 \mathcal{F}_2 \cdots$ is standard when the previous proviso holds assuming that R_i and R_j are the leftmost redexes of the respective sets $\mathcal{F}_i$ and $\mathcal{F}_j$.
- The *standardization theorem* ensures that, for any reduction, there exists an equivalent one which is standard.

Some properties of the copy relation

Let us define $\rho \sqsubseteq \sigma$, when $\rho\tau \equiv \sigma$ for some reduction τ.

Lemma

$\rho R \leq \sigma S$ iff $\rho \sqsubseteq \sigma$ and $S \in R/(\sigma/\rho)$.

Lemma (interpolation)

For any $\rho \sqsubseteq \sigma \sqsubseteq \tau$ and $\rho R \leq \tau T$, there exists a redex S such that $\rho R \leq \sigma S \leq \tau T$.

Lemma (uniqueness)

If $\rho R_i \leq \sigma S$, for $i = 1, 2$, then $R_1 = R_2$.

Therefore, the copy relation is decidable, since for any σS, the minimal redex ρR such that $\rho R \leq \sigma S$ is unique.

Causal history

- Let us assume a simplification process (extraction) that, for any redex ρR, throws away all the redexes in the history ρ that are not relevant to the "creation" of R.
- At the end of this process, we essentially obtain the "causal history" of R (with respect to ρ).
- In order to aovid the porbems caused by the permutation equivalence of reductions, we assume to work with standard derivations.

Parallelization

- Let $x \notin \mathrm{FV}(N)$ and $M[\_]^i_x$ be the context in which the ith occurrence of $x \in \mathrm{FV}(M)$ is replaced by a hole.

$$R = (\lambda x.\, M)\, N \to_R M[N] = M[N]^i_x[N/x] \to^*_{\rho_i} M[N]^i_x[N/x]$$

  the reduction ρ_i reduces redexes in the ith copy of N only.

- For every, such a reduction ρ_i, there is

$$R = (\lambda x.\, M)\, N \to^*_{\rho'} (\lambda x.\, M)\, N' = R/\rho'$$

- and k reductions

$$M[N/x] = M[N]^j_x[N/x] \to^*_{\rho_j} M[N']^j_x[N/x]$$

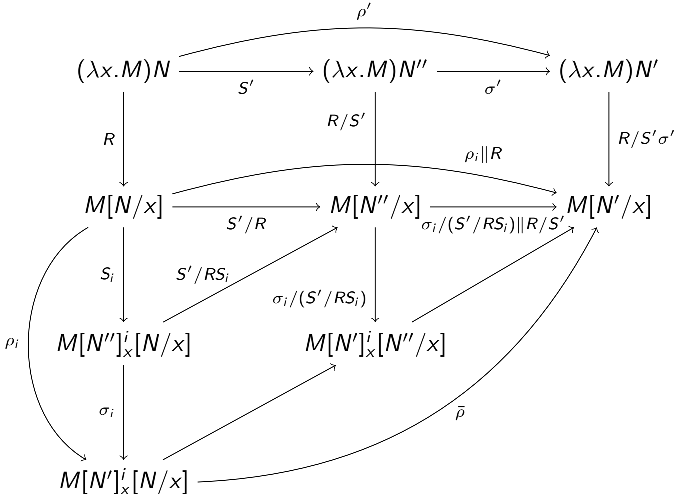
- moreover, let $\bar{\rho} = \rho_1 \sqcup \cdots \sqcup \rho_k : M[N/x] \to^* M[N'/x]$

$$R \to_R M[N/x] \to^*_{\bar{\rho}} M[N'/x] \;{}_{R/\rho'}{}^{*}\!\leftarrow (\lambda x.M)N' \;{}_{\rho'}{}^{*}\!\leftarrow R$$

$$R\bar{\rho} \equiv \rho'(R/\rho')$$

Parallelization (cont.)

- The reductions $\bar{\rho}$ is a sort of parallelization of ρ_i w.r.t. R, say $\bar{\rho} = \rho_i \| R$.
- $\rho_i \| R$ can be defined inductively. Let $\rho_i = S\sigma$



Parallelization (cont.)

Definition (parallelization)

Let $R = (\lambda x.M)\,N$. Let $R\rho_i$ be a derivation such that ρ_i is internal to only one instance of N, say the ith one, in the contractum $M[N/x]$ of R.

The reduction $\rho_i \| R$ (ρ_i *parallelized* by R) is inductively defined as follows:

$$\begin{array}{rcll} \epsilon \,\|\, R & = & \epsilon & \\ S_i\sigma_i \,\|\, R & = & (S'/R)\,((\sigma/\mathcal{F})\|(R/S')) & \quad \text{where } S_i \in S'/R,\ \mathcal{F} = S'/RS_i \end{array}$$

Let $R\rho_i : (\lambda x.M)N \to^* M[N']^i_x[N/x]$, we can prove that:

- $\rho_i \| R : M[N/x] \to^* M[N'/x]$
- $\rho_i \| R \equiv \bar{\rho} = \rho_1 \sqcup \cdots \sqcup \rho_k$
- there is a unique ρ' s.t. $R(\rho_i \| R) \equiv \rho'(R/\rho')$

Parallelization (cont.)

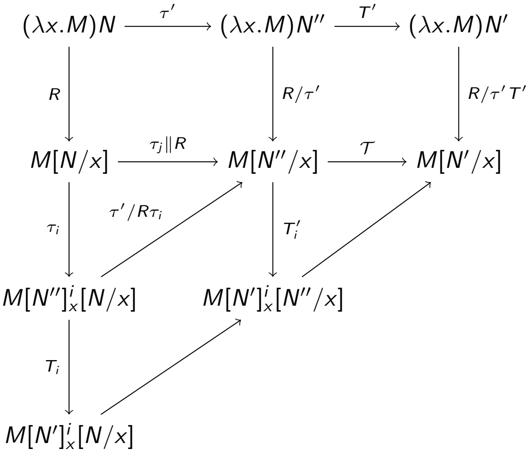
Let $\rho_i = \tau_i T_i$ be the reduction internal to the ith copy of the argument, and $\rho_j = \tau_j T_j$ the isomorphic reduction in the jth copy.

- In general, there is no redex T' in R s.t. $R\tau_j T_j \leq T'$
- However, let $\rho' = \tau' T'$

$$\mathcal{T} = T'/(R/\tau') = \{T'_j\}_{j=1}^{k}$$

$$\tau' T' \leq R(\tau_i \| R) T'_i = R(\tau_i \| R) T'_i \geq R\tau_i T_i$$

$$\tau' T' \simeq R\tau_i T_i$$



Extraction

- We aim at finding a relation that allows to remove the part of a reduction that are not relevant for the creation of a redex.
- Given a redex with history ρR, we want to find a "simpler" reduction ρ' and a redex R' s.t. $\rho' R' \simeq \rho R$, a sort of "causal history" of R.
- The easiest case is when we can find $\rho' R' \leq \rho R$.
  - We can take just $\rho' R'$.
  - The rest of $\rho \equiv \rho' \sqcup \tau$ (i.e., τ) is not related to the creation of R).
- In some case, we have to use parallelization.
  - Given $(\lambda x.M)N \to_R M[N/x] \to^*_{\tau_i} M[N']^i_x[N/x] \to_{T_i}$, where $\tau_i T_i$ is internal to the ith copy of N.
  - We can think that R is not a cause of T_i.
  - R can be removed by taking the previously constructed $\tau' T'$.
  - For instance, let $R = \Delta(\underline{(\lambda x.xM)I}_S)$
    - ★ $R \to_R (\underline{(\lambda x.xM)I}_{S_1})(\underline{(\lambda x.xM)I}_{S_2}) \to_{S_1} (\underline{IM}_{T_1})(\underline{(\lambda x.xM)I}_{S_2'})$
    - ★ $R \to_S \Delta(\underline{IM}_{T'}) \to_R (\underline{IM}_{T_1'})(\underline{IM}_{T_2'})$
    - ★ From $RS_1 T_1$ we want to obtain ST'
    - ★ Let us reamrk that $ST' \not\leq NRS_1 T_1$,
    - ★ but $ST' \leq SRT_1' \geq RS_1 T_1$, thus $ST' \simeq RS_1 T_1$

Extraction (cont.)

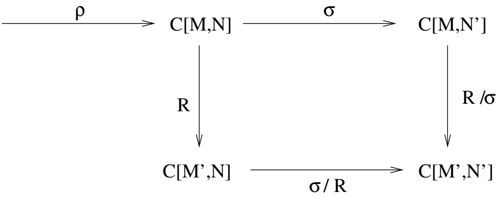
In order to define the extraction relation we distinguish four cases.

1. The base case is when the last redex in the reduction is the residual of some other redex that appears along the reduction

$$\rho R S \rhd_1 \rho S' \qquad \text{if } S \in S'/R$$

2. When two disjoint reductions are executed in parallel, we may remove the one that is not relevant

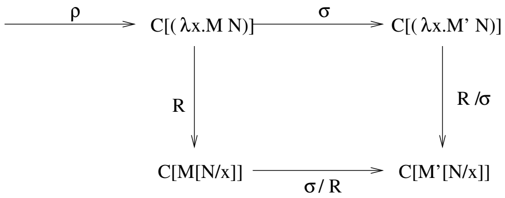
$$\rho\,(R \sqcup \sigma) \rhd_2 \rho\sigma \qquad \text{if } \sigma \neq \epsilon \text{ and } R,\, \sigma \text{ are disjoint}$$



Extraction (cont.)

3. If in a redex $R = (\lambda x.M)N$ we have a reduction σ internal to the functional part of R, then in the reduction $R \sqcup \sigma \equiv R(\sigma/R)$ we can avoid to reduce R

$$\rho(R \sqcup \sigma) \rhd_3 \rho\sigma \qquad \text{if } \sigma \neq \epsilon \text{ and } \sigma \text{ is internal to the function part of } R$$



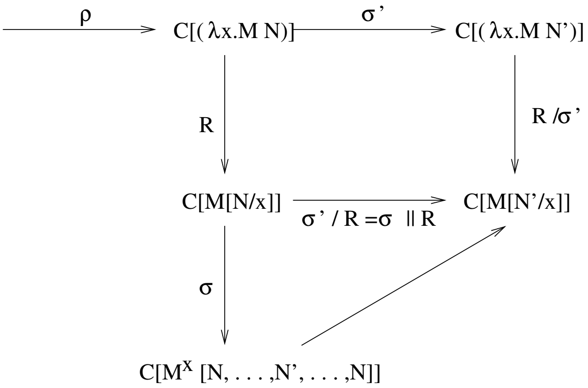
Extraction (cont.)

4. The last case is when we need parallelization.

$\rho R \sigma \rhd_4^i \rho\sigma'$

$\sigma \neq \epsilon$

1. σ is internal to the ith instance of the argument of R in its contractum
2. $\sigma'/R = \sigma \| R$



Extraction: the definition

Definition (extraction)

The contraction by *extraction* $\triangleright$ is the union of the following four relations:

1. $\rho R S \triangleright_1 \rho S'$, if $S \in S'/R$;
2. $\rho\,(R \sqcup \sigma) \triangleright_2 \rho\sigma$, if $\sigma \neq \epsilon$ and R, σ are disjoint reductions;
3. $\rho(R \sqcup \sigma) \triangleright_3 \rho\sigma$, if $\sigma \neq \epsilon$ and σ is internal to the function part of R;
4. $\rho R\sigma \triangleright_4^i \rho\sigma'$, if $|\sigma| \geq 1$, σ is internal to the ith instance of the argument of R in its contractum, and $\sigma'/R = \sigma \| R$.

The *extraction* relation $\trianglerighteq$ is the transitive and reflexive closure of $\triangleright$.

Theorem

$\trianglerighteq$ *is confluent and strongly normalizing.*

Extraction and families

- The extraction procedure relates redexes in the same family.
- In particular, for the four cases, we have already seen that, given

$$R = (\lambda x.M)N \to M[N/x] \to^*_{\tau_i} M[N']^i_x[N] \to_{T_i} M[N'']^i_x[N]$$

$$R = (\lambda x.M)N \to^*_{\tau'_i} (\lambda x.M)N' \to_{T'_i} (\lambda x.M)N''$$

s.t. $\tau' T' \sqcup R \equiv R(\tau_i T_i \| R)$

then $R\tau_i T_i \trianglerighteq_4 \tau' T'$ and $R\tau_i T_i \simeq \tau' T'$

- In the other cases instead, we directly have

$$\rho R \trianglerighteq_{1,2,3} \rho' R' \implies \rho' R' \leq \rho R \implies \rho R \simeq \rho' R'$$

Proposition

If there exists a reduction τT such that $\rho R \trianglerighteq \tau T \trianglelefteq \sigma S$, then $\rho R \simeq \sigma S$.

Families and extraction

- The converse of the previous proposition is also true: given two redexes in the same family, extraction reduce them to a common redex.

Lemma

If $R \leq \sigma S$, then $\sigma S \trianglerighteq R$.

Lemma

Let $\rho R \leq \sigma S$. There is a reduction τT such that $\rho R \trianglerighteq \tau T \trianglelefteq \sigma S$.

Proposition

If $\rho R \simeq \sigma S$, then $\rho R \trianglerighteq \tau T \trianglelefteq \sigma S$ for some τT.

- Since we have already shown that extraction is sound w.r.t. to families, $\rho R \simeq \tau T \simeq \sigma S$.

Decidability of extraction

Theorem (decidability of extraction)

Let σ and ρ be two standard *reductions. Then $\rho R \simeq \sigma S$ if and only if $\rho R \trianglerighteq \tau T \trianglelefteq \sigma S$ for some τT.*

- The previous result not only establishes a full correspondence between zig-zag and extraction, but also gives an *effective* procedure for deciding the family relation.
  - Since $\trianglerighteq$ is confluent and strongly normalizing, extraction has the unique normal form property.
  - Therefore, two reexes are in the same family iff they have the same normal form w.r.t. extraction.
- The unique normal form of every redex in a family can be taken as the canonical representative of the family.
- We recall that these results can be achieved because we are assuming to work with standard reductions only.

Canonical derivation

Definition (canonical derivation)

Every standard derivation ρR in normal form with respect to $\trianglerighteq$ will be called *canonical*.

Corollary (canonical representative)

The canonical representative derived from each member ρR of the family is the unique canonical derivation $\rho_c R_c$ such that $\rho_s R \trianglerighteq \rho_c R_c$, where ρ_s is the standard derivation equivalent to ρ.

Each family of redexes has a unique canonical representative.

Reductions by families

- We introduce a strategy of derivation by families: a parallel reduction in which at each step several redexes in the same family can be reduced in parallel.
- The idea is that by systematically reducing in parallel *all* the redexes in a given family, another member of that family cannot appear later on during the computation.

First of all, we define the class involved in a (parallel) reduction

Definition

Let $[\rho R] = \{\sigma S \mid \sigma S \simeq \rho R\}$ be the *family class* of ρR. Let $\rho = \mathcal{F}_1 \cdots \mathcal{F}_n$. Then,

$$\mathrm{FAM}(\rho) = \{[\mathcal{F}_1 \cdots \mathcal{F}_i R] \mid R \in \mathcal{F}_{i+1}, i = 0, 1, \ldots, n-1\}$$

is the set of family classes contained in ρ.

Generalized finite developments

We can then generalize the finite development theorem.

Definition (development of family classes)

Let $\mathcal{X}$ be a set of family classes. A derivation ρ is *relative* to $\mathcal{X}$ if $\mathrm{FAM}(\rho) \subseteq \mathcal{X}$.
A derivation ρ is a *development* of $\mathcal{X}$ if there is no redex R such that $[\rho R] \in \mathcal{X}$.

Theorem (generalized finite developments)

Let $\mathcal{X}$ be a (finite) set of family classes. Then:

1. *There is no infinite derivation relative to $\mathcal{X}$.*
2. *If ρ and σ are two developments of $\mathcal{X}$, then $\rho \equiv \sigma$.*

Developments of families

First of all a technical property.

Lemma

Let ρR be the canonical derivation of σS. Then $\rho \sqsubseteq \sigma$ if and only if $\rho R \leq \sigma S$.

We can now prove that, after the development of a set of families $\mathcal{X}$, no redex in a family contained in $\mathcal{X}$ can be created along the reduction.

Lemma

Let ρ be a development of $\mathcal{X}$.

1. *Let σS be the canonical derivation of ρR. Then $\sigma S \leq \rho R$.*
2. *For every σS such that $\rho \sqsubseteq \sigma$, $[\sigma S] \notin \mathrm{FAM}(\rho)$.*

Complete derivations

Definition (complete derivation)

We say that the derivation $\mathcal{F}_1 \cdots \mathcal{F}_n$ is *complete* if and only if $\mathcal{F}_i \neq \emptyset$ and $\mathcal{F}_i$ is a maximal set of redexes such that

$$\forall R, S \in \mathcal{F}_i. \qquad \mathcal{F}_1 \cdots \mathcal{F}_{i-1} R \simeq \mathcal{F}_1 \cdots \mathcal{F}_{i-1} S$$

for $i = 1, 2, \ldots, n$.

Lemma

Every complete derivation ρ *is a development of* $\mathrm{FAM}(\rho)$.

Optimal reductions

Lemma

A derivation $\rho = \mathcal{F}_1 \cdots \mathcal{F}_n$ is complete if and only if, for $i = 1, \dots, n$, $\mathcal{F}_i$ is a maximal set of copies*. Namely, for any i, there exist $\sigma_i S_i$ and τ_i such that $\sigma_i \tau_i \equiv \rho_i$ and $\mathcal{F}_i = S/\tau_i$.*

Proposition

Let ρ be a complete derivation. We have that $|\rho| = \sharp(\mathsf{FAM}(\rho))$, where $\sharp(\mathsf{FAM}(\rho))$ is the cardinality of $\mathsf{FAM}(\rho)$.