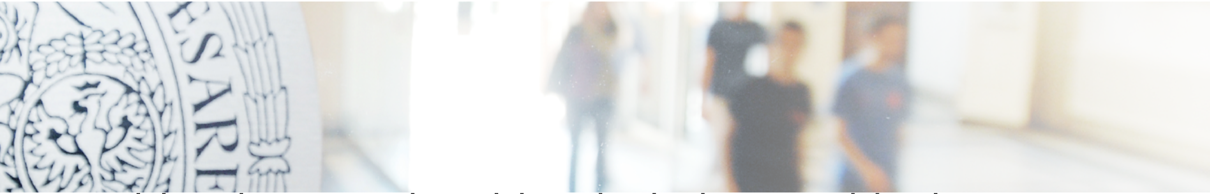


Multi-redexes and multi-treks induce residual systems

least upper bounds and left-cancellation up to homotopy

Vincent van Oostrom

http://cl-informatik.uibk.ac.at

1. Residual systems

2. Multi-redexes

3. Conclusions

Rewrite systems

Definition (Rewrite system [Newman 42])

rewrite system $\to$ comprises:

- a set of objects
- a set of (rewrite) steps
- functions src, tgt mapping a step to its source, target object

Rewrite systems

Definition (Rewrite system)

rewrite system $\to$ comprises:

- a set of objects
- a set of steps
- functions $\mathsf{src}, \mathsf{tgt}$ mapping a step to its source, target object

Remark (Omnipresence of rewrite systems)

- *steps as first-class citizens (theory of computation!)*

Rewrite systems

Definition (Rewrite system)

rewrite system $\to$ comprises:

- a set of objects
- a set of steps
- functions $\mathrm{src}, \mathrm{tgt}$ mapping a step to its source, target object

Remark (Omnipresence of rewrite systems)

- *steps as first-class citizens*
- *rewrite (binary endo)relation R is system: step $\frac{b}{a}$ from a to b if a R b*

Rewrite systems

Definition (Rewrite system)

rewrite system $\to$ comprises:

- a set of objects
- a set of steps
- functions src, tgt mapping a step to its source, target object

Remark (Omnipresence of rewrite systems)

- *steps as first-class citizens*
- *rewrite relation R is system: step $\frac{b}{a}$ from a to b if $a\ R\ b$*
- *formally same as multidigraph and quiver*
  (here the name steps signals interest in transformational properties)

Rewrite systems

Definition (Rewrite system)

rewrite system $\to$ comprises:

- a set of objects
- a set of steps
- functions $\mathsf{src}, \mathsf{tgt}$ mapping a step to its source, target object

Remark (Omnipresence of rewrite systems)

- *steps as first-class citizens*
- *rewrite relation R is system: step $\frac{b}{a}$ from a to b if $a \mathrel{R} b$*
- *formally same as multidigraph and quiver*
  (here the name steps signals interest in transformational properties)

Rewrite systems

Definition (Rewrite system)

rewrite system $\to$ comprises:

- a set of objects
- a set of steps
- functions $\mathrm{src}, \mathrm{tgt}$ mapping a step to its source, target object

Remark (the rewrite method)

derive properties of (empty,finite,infinite) computations from those of steps

$\to$-steps used to generate first compositions $\to^*$ (trees of composable steps), next paths/reductions $\twoheadrightarrow$ (quotienting out composition monoid), and then quasi-orders (quotienting out parallel paths)

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap_\beta$)

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap_\beta$)

- objects are multisteps without βs

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap\!\!\!\to_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi, \phi)$, for x in variables; modulo α
  (i use many- and multi- to signal series resp. parallel quantities)

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap\!\!\to_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi,\phi)$, for x in variables; step $\to_\beta$ if one β
- homomorphic extension mapping $\beta(x.\phi,\psi)$ to lhs $(\lambda x.\phi)\,\psi$ and rhs $\phi[x{:=}\psi]$ $(\mathrm{src}(\beta(x.\phi,\psi)) := (\lambda x.\mathrm{src}(\phi))\,\mathrm{src}(\psi)$ and $\mathrm{tgt}(\beta(x.\phi,\psi)) := \mathrm{tgt}(\phi)[x{:=}\mathrm{tgt}(\psi)])$

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi,\phi)$, for x in variables; step $\to_\beta$ if one β
- homomorphic extension mapping $\beta(x.\phi,\psi)$ to lhs $(\lambda x.\phi)\,\psi$ and rhs $\phi[x{:=}\psi]$

Example

- $\beta(x.x,\beta(y.y,z))$ multistep $(\lambda x.x)\,((\lambda y.y)\,z) \multimap z$

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap\!\!\rightarrow_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi, \phi)$, for x in variables; step $\rightarrow_\beta$ if one β
- homomorphic extension mapping $\beta(x.\phi, \psi)$ to lhs $(\lambda x.\phi)\,\psi$ and rhs $\phi[x{:=}\psi]$

Example

- $\beta(x.x, \beta(y.y, z))$ multistep $I\,(I\,z) \multimap\!\!\rightarrow z$ with $I := \lambda x.x$

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap\!\!\to_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi,\phi)$, for x in variables; step $\to_\beta$ if one β
- homomorphic extension mapping $\beta(x.\phi,\psi)$ to lhs $(\lambda x.\phi)\,\psi$ and rhs $\phi[x:=\psi]$

Example

- $\beta(x.x,\beta(y.y,z))$ multistep $I\,(I\,z) \multimap\!\!\to z$ with $I := \lambda x.x$
- $\beta(x.x, I\,z)$ and $I\,\beta(y.y,z)$ are distinct (single) steps $I\,(I\,z) \to_\beta I\,z$

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi,\phi)$, for x in variables; step $\to_\beta$ if one β
- homomorphic extension mapping $\beta(x.\phi,\psi)$ to lhs $(\lambda x.\phi)\,\psi$ and rhs $\phi[x{:=}\psi]$

Example

- $\beta(x.x,\beta(y.y,z))$ multistep $I\,(I\,z) \multimap z$ with $I := \lambda x.x$
- $\beta(x.x,I\,z)$ and $I\,\beta(y.y,z)$ are outer and inner steps $I\,(I\,z) \to_\beta I\,z$

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi,\phi)$, for x in variables; step $\to_\beta$ if one β
- homomorphic extension mapping $\beta(x.\phi,\psi)$ to lhs $(\lambda x.\phi)\,\psi$ and rhs $\phi[x{:=}\psi]$

Example

- $\beta(x.x,\beta(y.y,z))$ multistep $I\,(I\,z) \multimap z$ with $I := \lambda x.x$
- $\beta(x.x,I\,z)$ and $I\,\beta(y.y,z)$ are outer and inner steps $I\,(I\,z) \to_\beta I\,z$

Remark

$\multimap_\beta$ is *Tait–Martin-Löf step (aka parallel reduction [Takahashi 95])*

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi,\phi)$, for x in variables; step $\to_\beta$ if one β
- homomorphic extension mapping $\beta(x.\phi,\psi)$ to lhs $(\lambda x.\phi)\,\psi$ and rhs $\phi[x{:=}\psi]$

Example

- $\beta(x.x,\beta(y.y,z))$ multistep $I\,(I\,z) \multimap z$ with $I := \lambda x.x$
- $\beta(x.x,I\,z)$ and $I\,\beta(y.y,z)$ are outer and inner steps $I\,(I\,z) \to_\beta I\,z$

Remark

multisteps by adjoining β-rule as symbol to signature [vO 97]
(this reification of rules works for string/term/graph/... rewrite systems)

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap\!\!\to_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi,\phi)$, for x in variables; step $\to_\beta$ if one β
- homomorphic extension mapping $\beta(x.\phi,\psi)$ to lhs $(\lambda x.\phi)\,\psi$ and rhs $\phi[x{:=}\psi]$

Example

- $\beta(x.x,\beta(y.y,z))$ multistep $I\,(I\,z) \multimap\!\!\to z$ with $I := \lambda x.x$
- $\beta(x.x,I\,z)$ and $I\,\beta(y.y,z)$ are outer and inner steps $I\,(I\,z) \to_\beta I\,z$

Remark

advantages: compact (multi)step representations; stay in term language
(no disadvantages; no need for inference system; src, tgt *instead)*

Example: Church's $\lambda\beta$-calculus as rewrite system

Definition (Multistep rewrite system $\multimap_\beta$)

- objects are multisteps without βs
- multisteps $\phi ::= x \mid \lambda x.\phi \mid \phi\,\phi \mid \beta(x.\phi,\phi)$, for x in variables; step $\to_\beta$ if one β
- homomorphic extension mapping $\beta(x.\phi,\psi)$ to lhs $(\lambda x.\phi)\,\psi$ and rhs $\phi[x{:=}\psi]$

Example

- $\beta(x.x,\beta(y.y,z))$ multistep $I\,(I\,z) \multimap z$ with $I := \lambda x.x$
- $\beta(x.x,I\,z)$ and $I\,\beta(y.y,z)$ are outer and inner steps $I\,(I\,z) \to_\beta I\,z$

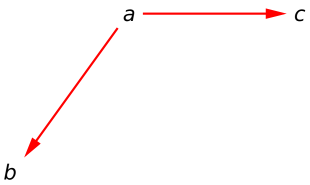
Remark

advantages: compact (multi)step representations; stay in term language
(no disadvantages; all there is to know: no need for annotations of relations)

Residuation as Skolemisation of the diamond property

Definition (Diamond property)

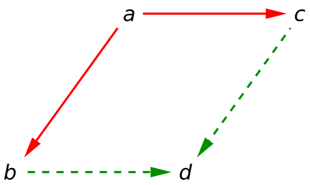
$\to$ has the diamond property if $\forall$ peak $b \leftarrow a \to c$



Residuation as Skolemisation of the diamond property

Definition (Diamond property)

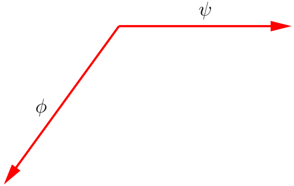
$\to$ has the diamond property if $\forall$ peak $b \leftarrow a \to c$, $\exists$ valley $b \to d \leftarrow c$



Residuation as Skolemisation of the diamond property

Definition (Diamond property)

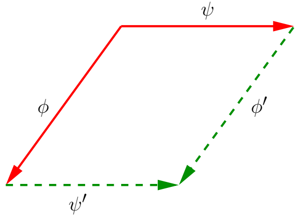
$\to$ has the diamond property if $\forall$ peak $b \;{}_{\phi}{\leftarrow}\; a \to_{\psi} c$



Residuation as Skolemisation of the diamond property

Definition (Diamond property)

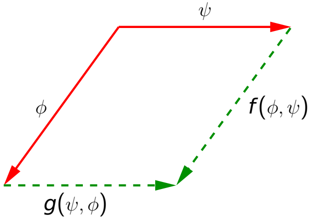
$\to$ has the diamond property if $\forall$ peak $b \;{}_{\phi}{\leftarrow}\; a \to_{\psi} c$, $\exists$ valley $b \to_{\psi'} d \;{}_{\phi'}{\leftarrow}\; c$



Residuation as Skolemisation of the diamond property

Definition (Skolemised diamond property)

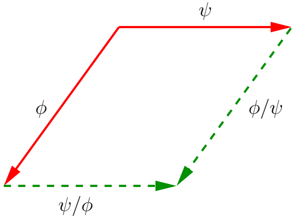
$\rightarrow$ has the diamond property if $\forall$ peak $b \;{}_{\phi}\!\leftarrow a \rightarrow_{\psi} c$, $\; b \rightarrow_{f(\phi,\psi)} d \;{}_{g(\phi,\psi)}\!\leftarrow c$



Residuation as Skolemisation of the diamond property

Definition (Skolemised diamond property)

$\to$ has the diamond property if $\forall$ peak $b \;{}_{\phi}\!\leftarrow a \to_{\psi} c$, $\;b \to_{\phi/\psi} d \;{}_{\psi/\phi}\!\leftarrow c$



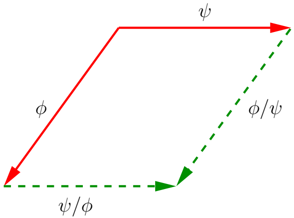
Remark (Symmetrisation)

may totally order objects $\Longrightarrow$ may assume f, g same residuation function $/$

Residuation as Skolemisation of the diamond property

Definition (Skolemised diamond property)

$\to$ has the diamond property if $\forall$ peak $b\ {}_{\phi}\!\leftarrow a \to_{\psi} c$, $b \to_{\phi/\psi} d\ {}_{\psi/\phi}\!\leftarrow c$



Remark (Diamond from steps $\implies$ reductions $\implies$ quasi-orders)

confluence of $\to$ is diamond property of $\twoheadrightarrow$; upper bound (d of b, c) in quasi-order

Example: no residuation for $\rightarrow_\beta$

Example (Failure of diamond for $\rightarrow_\beta$)

- for peak $z\ {}_{\beta(x.x,z)}\!\leftarrow I\, z \rightarrow_{\beta(x.x,z)} z$, only empty valley z; no diamond

Example: no residuation for $\to_\beta$

Example (Failure of diamond for $\to_\beta$)

- for peak $z \;{}_{\beta(x.x,z)}\!\!\leftarrow I\,z \to_{\beta(x.x,z)} z$, only empty valley z
- for peak $\delta\, z \;{}_{\delta\,\beta(x.x,z)}\!\!\leftarrow \delta\,(I\,z) \to_{\beta(x.x\,x,I\,z)} I\,z\,(I\,z)$ with $\delta := \lambda x.x\,x$
  only duplicating valley $\delta\, z \to z\,z \leftarrow (I\,z)\,z \leftarrow I\,z\,(I\,z)$; no diamond

Example: no residuation for $\to_\beta$

Example (Failure of diamond for $\to_\beta$)

- for peak $z \;{}_{\beta(x.x,z)}\!\leftarrow I\,z \to_{\beta(x.x,z)} z$, only empty valley z
- for peak $\delta\,z \;{}_{\delta\,\beta(x.x,z)}\!\leftarrow \delta\,(I\,z) \to_{\beta(x.x\,x,I\,z)} I\,z\,(I\,z)$ with $\delta := \lambda x.x\,x$
  only duplicating valley $\delta\,z \to z\,z \leftarrow (I\,z)\,z \leftarrow I\,z\,(I\,z)$

Faceting to the rescue

Idea: adjoin reduction (many-step) in valley as (single) step
(here: adjoin z : z and $I\,z\,\beta(x.x,z)\cdot\beta(x.x,z)\,z$: $I\,z\,(I\,z) \to (I\,z)\,z \to z\,z$ as steps)

Example: no residuation for $\to_\beta$

Example (Failure of diamond for $\to_\beta$)

- for peak $z \;{}_{\beta(x.x,z)}\!\leftarrow I\,z \to_{\beta(x.x,z)} z$, only empty valley z
- for peak $\delta\,z \;{}_{\delta\,\beta(x.x,z)}\!\leftarrow \delta\,(I\,z) \to_{\beta(x.x\,x,I\,z)} I\,z\,(I\,z)$ with $\delta := \lambda x.x\,x$
  only duplicating valley $\delta\,z \to z\,z \leftarrow (I\,z)\,z \leftarrow I\,z\,(I\,z)$

$$
\begin{tikzcd}
\delta\,(I\,z) \arrow[rrr] \arrow[dd] & & & I\,z\,(I\,z) \arrow[dl, "I\,z\,\beta(x.x,z)"] \\
& & I\,z\,z \arrow[dl, "\beta(x.x,z)\,z"] & \\
\delta\,z \arrow[r] & z\,z & &
\end{tikzcd}
$$

Example: no residuation for $\to_\beta$

Example (Failure of diamond for $\to_\beta$)

- for peak $z \;{}_{\beta(x.x,z)}\!\leftarrow I\,z \to_{\beta(x.x,z)} z$, only empty valley z
- for peak $\delta\,z \;{}_{\delta\,\beta(x.x,z)}\!\leftarrow \delta\,(I\,z) \to_{\beta(x.x\,x,I\,z)} I\,z\,(I\,z)$ with $\delta := \lambda x.x\,x$
  only duplicating valley $\delta\,z \to z\,z \leftarrow (I\,z)\,z \leftarrow I\,z\,(I\,z)$

$$
\begin{tikzcd}
\delta\,(I\,z) \arrow[r] \arrow[dd] & I\,z\,(I\,z) \arrow[d] \arrow[dd, bend left, "\begin{array}{l}\text{adjoin reduction as step}\\ I\,z\,\beta(x.x,z)\cdot\beta(x.x,z)\,I\end{array}"] \\
 & I\,z\,z \arrow[d] \\
\delta\,z \arrow[r] & z\,z
\end{tikzcd}
$$

Example: residuation for $\mathrel{-\!\!\circ\!\!\to}_\beta$

Remark

multisteps $\mathrel{-\!\!\circ\!\!\to}_\beta$ are (notation for) repeated faceting for $\to_\beta$
(like completion but goal now to get beautiful diamonds, not complete system)

$$
\begin{tikzcd}
\delta\,(I\,z) \arrow[r] \arrow[dd] & I\,z\,(I\,z) \arrow[d] \arrow[dd, bend left=60, "\text{is multistep } \beta(x.x,z)\,\beta(x.x,z)"] \\
& I\,z\,z \arrow[d] \\
\delta\,z \arrow[r] & z\,z
\end{tikzcd}
$$

Example: residuation for $\multimap_\beta$

Lemma

$\multimap_\beta$ *has the diamond property*

Proof idea.

define residuation $/$ and join $\vee$ such that if ϕ, ψ co-initial $\Longrightarrow$
$\phi \vee \psi$ and $\phi \cdot (\psi/\phi)$ have same source,target, join $\vee$ commutative $\square$

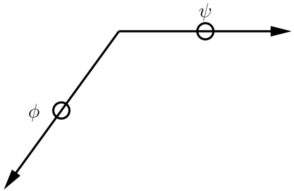
Example: residuation for $\mathrel{-\!\!\circ\!\!\to}_\beta$

Lemma

$\mathrel{-\!\!\circ\!\!\to}_\beta$ *has the diamond property*

Proof idea.

define residuation $/$ and join $\vee$ such that if ϕ, ψ co-initial $\implies$
$\phi \vee \psi$ and $\phi \cdot (\psi/\phi)$ have same source,target, join $\vee$ commutative $\square$



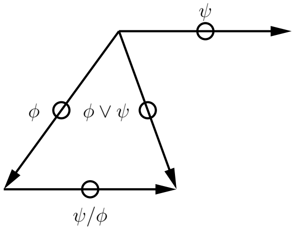
Example: residuation for $\multimap_\beta$

Lemma

$\multimap_\beta$ *has the diamond property*

Proof idea.

define residuation $/$ and join $\vee$ such that if ϕ, ψ co-initial $\implies$
$\phi \vee \psi$ and $\phi \cdot (\psi/\phi)$ have same source,target, join $\vee$ commutative $\square$



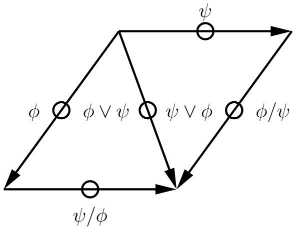
Example: residuation for $\mathrel{-\!\!\circ\!\!\rightarrow}_\beta$

Lemma

$\mathrel{-\!\!\circ\!\!\rightarrow}_\beta$ has the diamond property

Proof idea.

define residuation $/$ and join $\vee$ such that if ϕ, ψ co-initial $\Longrightarrow$
$\phi \vee \psi$ and $\phi \cdot (\psi/\phi)$ have same source,target, join $\vee$ commutative $\square$



Example: residuation for $\multimap\!\!\to_\beta$

Proof.

defining join $\vee$ and residuation $/$ as follows works (induction on multisteps):

ϕ	ψ	$\phi \vee \psi$	ϕ / ψ
$\beta(x.\phi', \phi'')$	$(\lambda x.\psi')\,\psi''$	$\beta(x.\phi' \vee \psi', \phi'' \vee \psi'')$	$\beta(x.\phi'/\psi', \phi''/\psi'')$
$(\lambda x.\phi')\,\phi''$	$\beta(x.\psi', \psi'')$	,,	$(\phi'/\psi')[x{:=}\phi''/\psi'']$
$\beta(x.\phi', \phi'')$	$\beta(x.\psi', \psi'')$	,,	,,
x	x	x	x
$\lambda x.\phi'$	$\lambda x.\psi'$	$\lambda x.\phi' \vee \psi'$	$\lambda x.\phi'/\psi'$
$\phi'\phi''$	$\psi'\psi''$	$(\phi' \vee \psi')(\phi'' \vee \psi'')$	$(\phi'/\psi')(\phi''/\psi'')$ $\square$

Example: residuation for $\multimap_\beta$

Proof.

defining join $\vee$ and residuation $/$ as follows works (induction on multisteps):

ϕ	ψ	$\phi \vee \psi$	ϕ/ψ
$\beta(x.\phi', \phi'')$	$(\lambda x.\psi')\,\psi''$	$\beta(x.\phi' \vee \psi', \phi'' \vee \psi'')$	$\beta(x.\phi'/\psi', \phi''/\psi'')$
$(\lambda x.\phi')\,\phi''$	$\beta(x.\psi', \psi'')$	,,	$(\phi'/\psi')[x{:=}\phi''/\psi'']$
$\beta(x.\phi', \phi'')$	$\beta(x.\psi', \psi'')$	,,	,,
x	x	x	x
$\lambda x.\phi'$	$\lambda x.\psi'$	$\lambda x.\phi' \vee \psi'$	$\lambda x.\phi'/\psi'$
$\phi'\phi''$	$\psi'\psi''$	$(\phi' \vee \psi')(\phi'' \vee \psi'')$	$(\phi'/\psi')(\phi''/\psi'')$ $\Box$

is Tait–Martin-Löf proof: short by multistep notation, commutation of join

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development
- [Newman 42]: axiomatic residuation
  (should but did not apply to β (Schroer review); first α-error in literature?)

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development
- [Newman 42]: axiomatic residuation
- Hindley 60s and 70s: axiomatic residuation for β

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development
- [Newman 42]: axiomatic residuation
- Hindley 60s and 70s: axiomatic residuation for β
- [Lévy 78]: $\lambda\beta$-calculus permutation equivalence, optimality, cube, ...

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development
- [Newman 42]: axiomatic residuation
- Hindley 60s and 70s: axiomatic residuation for β
- [Lévy 78]: $\lambda\beta$-calculus permutation equivalence, optimality, cube, ...
- Huet & Lévy 79: first-order TRS residuation, neededness

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development
- [Newman 42]: axiomatic residuation
- Hindley 60s and 70s: axiomatic residuation for β
- [Lévy 78]: $\lambda\beta$-calculus permutation equivalence, optimality, cube, ...
- Huet & Lévy 79: first-order TRS residuation, neededness
- [Huet 86]: prism, FSCD

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development
- [Newman 42]: axiomatic residuation
- Hindley 60s and 70s: axiomatic residuation for β
- [Lévy 78]: $\lambda\beta$-calculus permutation equivalence, optimality, cube, ...
- Huet & Lévy 79: first-order TRS residuation, neededness
- [Huet 86]: prism, FSCD
- Khasidashvili & Glauert 90s: axiomatic neededness, optimality, ...

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development
- [Newman 42]: axiomatic residuation
- Hindley 60s and 70s: axiomatic residuation for β
- [Lévy 78]: $\lambda\beta$-calculus permutation equivalence, optimality, cube, ...
- Huet & Lévy 79: first-order TRS residuation, neededness
- [Huet 86]: prism, FSCD
- Khasidashvili & Glauert 90s: axiomatic neededness, optimality, ...
- Terese 03: residual systems (presented next), equivalence of equivalences

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development
- [Newman 42]: axiomatic residuation
- Hindley 60s and 70s: axiomatic residuation for β
- [Lévy 78]: $\lambda\beta$-calculus permutation equivalence, optimality, cube, ...
- Huet & Lévy 79: first-order TRS residuation, neededness
- [Huet 86]: prism, FSCD
- Khasidashvili & Glauert 90s: axiomatic neededness, optimality, ...
- Terese 03: residual systems (presented next), equivalence of equivalences
- [Melliès 02]: axiomatic multi-redexes/treks (presented after)

Ubiquity of residuation

Example (Residuation in rewriting (replication))

- [Church–Rosser 36]: $\lambda\beta$-development
- [Newman 42]: axiomatic residuation
- Hindley 60s and 70s: axiomatic residuation for β
- [Lévy 78]: $\lambda\beta$-calculus permutation equivalence, optimality, cube, ...
- Huet & Lévy 79: first-order TRS residuation, neededness
- [Huet 86]: prism, FSCD
- Khasidashvili & Glauert 90s: axiomatic neededness, optimality, ...
- Terese 03: residual systems (presented next), equivalence of equivalences
- [Melliès 02]: axiomatic multi-redexes/treks (presented after)

in concurrency (linear): [Mazurkiewicz 70s], [Stark 89], [Winskel 89], Wolfram,

Residual systems

Definition (Residual system, Terese 03)

residual system $\langle \to, 1, / \rangle$ has for co-initial ϕ, ψ, χ in rewrite system $\to$:

$$\phi/1 = \phi \tag{1}$$

$$\phi/\phi = 1 \tag{2}$$

$$1/\phi = 1 \tag{3}$$

$$(\phi/\psi)/(\chi/\psi) = (\phi/\chi)/(\psi/\chi) \tag{4}$$

Residual systems

Definition (Residual system, Terese 03)

residual system $\langle \to, 1, / \rangle$ has for co-initial ϕ, ψ, χ in rewrite system $\to$:

$$\phi/1 = \phi \tag{1}$$

$$\phi/\phi = 1 \tag{2}$$

$$1/\phi = 1 \tag{3}$$

$$(\phi/\psi)/(\chi/\psi) = (\phi/\chi)/(\psi/\chi) \tag{4}$$

1 is loop (one for each object)

Residual systems

Definition (Residual system, Terese 03)

residual system $\langle \to, 1, / \rangle$ has for co-initial ϕ, ψ, χ in rewrite system $\to$:

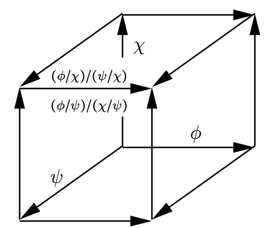
$$\phi/1 = \phi \tag{1}$$

$$\phi/\phi = 1 \tag{2}$$

$$1/\phi = 1 \tag{3}$$

$$(\phi/\psi)/(\chi/\psi) = (\phi/\chi)/(\psi/\chi) \tag{4}$$

(4) is Lévy's cube:



Residual systems

Definition (Residual system, Terese 03)

residual system $\langle \to, 1, / \rangle$ has for co-initial ϕ, ψ, χ in rewrite system $\to$:

$$\phi/1 = \phi \tag{1}$$

$$\phi/\phi = 1 \tag{2}$$

$$1/\phi = 1 \tag{3}$$

$$(\phi/\psi)/(\chi/\psi) = (\phi/\chi)/(\psi/\chi) \tag{4}$$

Residual systems

Definition (Residual system, Terese 03)

residual system $\langle \rightarrow, 1, / \rangle$ has for co-initial ϕ, ψ, χ in rewrite system $\rightarrow$:

$$\phi/1 = \phi \tag{1}$$

$$\phi/\phi = 1 \tag{2}$$

$$1/\phi = 1 \tag{3}$$

$$(\phi/\psi)/(\chi/\psi) = (\phi/\chi)/(\psi/\chi) \tag{4}$$

Intuition: residuation makes join semi-lattice $\Longrightarrow$ least upper bounds

- residuation diamond $\Longrightarrow$ commutativity of join (seen above)
- unit law (2) $\Longrightarrow$ idempotence of join
- cube law (4) $\Longrightarrow$ associativity of join

Residual systems

Definition (Residual system, Terese 03)

residual system $\langle \to, 1, / \rangle$ has for co-initial ϕ, ψ, χ in rewrite system $\to$:

$$\phi/1 = \phi \tag{1}$$

$$\phi/\phi = 1 \tag{2}$$

$$1/\phi = 1 \tag{3}$$

$$(\phi/\psi)/(\chi/\psi) = (\phi/\chi)/(\psi/\chi) \tag{4}$$

Remark (only intuition)

in general no order (steps need not compose), join need not exist ($\not\!\!\to\!\!\!\!\!\to$ in OTRSs)

Residual systems

Definition (Residual system, Terese 03)

residual system $\langle \to, 1, / \rangle$ has for co-initial ϕ, ψ, χ in rewrite system $\to$:

$$\phi/1 = \phi \tag{1}$$

$$\phi/\phi = 1 \tag{2}$$

$$1/\phi = 1 \tag{3}$$

$$(\phi/\psi)/(\chi/\psi) = (\phi/\chi)/(\psi/\chi) \tag{4}$$

Example

$\multimap_\beta$ is residual system for $/$ having joins with terms (trivial multisteps) as 1

Ubiquity of residual systems/algebras (single object)

Example

- residual systems: combinatory logic, $\lambda\beta$, orthogonal (first- and higher-order) term rewrite systems, positive braids, associativity, self-distributivity, ..., any confluent countable rewrite system (for contrived notion of lub)

Ubiquity of residual systems/algebras

Example

- residual systems: combinatory logic, $\lambda\beta$, orthogonal (first- and higher-order) term rewrite systems, positive braids, associativity, self-distributivity, ..., any confluent countable rewrite system
- commutative residual algebras: numbers with monus, (measurable) (multi)sets with difference, positive natural numbers with dovision, ...

Ubiquity of residual systems/algebras

Example

- residual systems: combinatory logic, $\lambda\beta$, orthogonal (first- and higher-order) term rewrite systems, positive braids, associativity, self-distributivity, ..., any confluent countable rewrite system
- commutative residual algebras: numbers with monus, (measurable) (multi)sets with difference, positive natural numbers with dovision, ...
- semi-lattices induce residual systems, categories having push-outs induce residual systems (for epis)

Ubiquity of residual systems/algebras

Example

- residual systems: combinatory logic, $\lambda\beta$, orthogonal (first- and higher-order) term rewrite systems, positive braids, associativity, self-distributivity, ..., any confluent countable rewrite system
- commutative residual algebras: numbers with monus, (measurable) (multi)sets with difference, positive natural numbers with dovision, ...
- semi-lattices induce residual systems, categories having push-outs induce residual systems (for epis)
- commutative residual algebras have multiset representation theorem, are equivalent to commutative BCK algebras with relative cancellation, induce lattice-ordered groups (groupoids for residual systems; with provisos), ...

Ubiquity of residual systems/algebras

Example

- residual systems: combinatory logic, $\lambda\beta$, orthogonal (first- and higher-order) term rewrite systems, positive braids, associativity, self-distributivity, . . . , any confluent countable rewrite system
- commutative residual algebras: numbers with monus, (measurable) (multi)sets with difference, positive natural numbers with dovision, . . .
- semi-lattices induce residual systems, categories having push-outs induce residual systems (for epis)
- commutative residual algebras have multiset representation theorem, are equivalent to commutative BCK algebras with relative cancellation, induce lattice-ordered groups (groupoids for residual systems; with provisos), . . .
- inclusion–exclusion principle, [EWD 1313], Bayes’ Theorem,. . .

Residual systems with composition

Definition (Residual system with composition, Terese 03)

residual system $\langle \rightarrow, 1, /, \cdot \rangle$ with composition $\cdot$ and for coinitial ϕ, ψ, χ in ARS $\rightarrow$:

$$\phi/1 = \phi \tag{1}$$

$$\phi/\phi = 1 \tag{2}$$

$$1/\phi = 1 \tag{3}$$

$$(\phi/\psi)/(\chi/\psi) = (\phi/\chi)/(\psi/\chi) \tag{4}$$

$$\chi/(\phi \cdot \psi) = (\chi/\phi)/\psi \tag{7}$$

$$(\phi \cdot \psi)/\chi = (\phi/\chi) \cdot (\psi/(\chi/\phi)) \tag{8}$$

Residual systems with composition

Definition (Residual system with composition, Terese 03)

residual system $\langle \to, 1, /, \cdot \rangle$ with composition $\cdot$ and for coinitial ϕ, ψ, χ in ARS $\to$:

$$\phi/1 = \phi \tag{1}$$

$$\phi/\phi = 1 \tag{2}$$

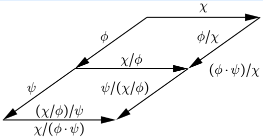
$$1/\phi = 1 \tag{3}$$

$$(\phi/\psi)/(\chi/\psi) = (\phi/\chi)/(\psi/\chi) \tag{4}$$

$$\chi/(\phi \cdot \psi) = (\chi/\phi)/\psi \tag{7}$$

$$(\phi \cdot \psi)/\chi = (\phi/\chi) \cdot (\psi/(\chi/\phi)) \tag{8}$$

composite identities (7) and (8):



Facts on residual systems (with composition)

Lemma (Terese 03)

- $\langle \to, 1, / \rangle$ *generates reduction system w/ composition on* $\to^*$ *up to* $1 \cdot 1 = 1$ *(by tiling with diamonds and cubes)*

Facts on residual systems (with composition)

Lemma (Terese 03)

- $\langle \to, 1, / \rangle$ *generates reduction system w/ composition on* $\to^*$ *up to* $1 \cdot 1 = 1$
- $\preccurlyeq$ *is quasi-order with* $\phi \preccurlyeq \psi := \phi/\psi = 1$ *(natural or projection order)*

Facts on residual systems (with composition)

Lemma (Terese 03)

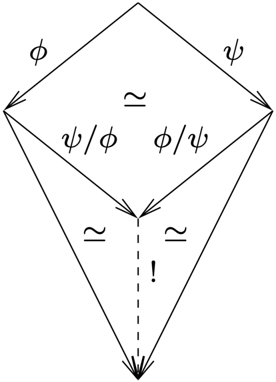
- $\langle \rightarrow, 1, / \rangle$ *generates reduction system w/ composition on* $\rightarrow^*$ *up to* $1 \cdot 1 = 1$
- $\preccurlyeq$ *is quasi-order with* $\phi \preccurlyeq \psi := \phi/\psi = 1$
- $\simeq$ *is congruence for operations with* $\simeq := \preccurlyeq \cap \succcurlyeq \implies$ *may be quotiented out*

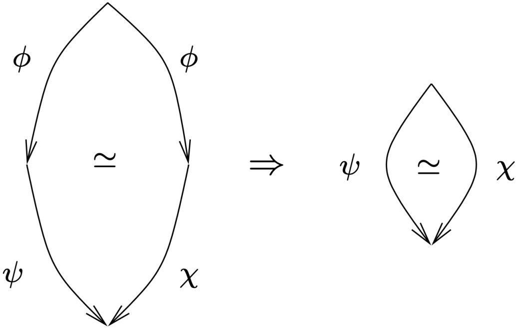
Facts on residual systems (with composition)

Lemma (Terese 03)

- $\langle \to, 1, / \rangle$ *generates reduction system w/ composition on* $\to^*$ *up to* $1 \cdot 1 = 1$
- $\precsim$ *is quasi-order with* $\phi \precsim \psi := \phi/\psi = 1$
- $\simeq$ *is congruence for operations with* $\simeq := \precsim \cap \succsim$
- $\simeq$*-quotient of* $\to^*$ *gives category for* $\twoheadrightarrow$ *with push-outs, epis*
  (identify multistep, development: $\beta(x.x, z)\, \beta(x.x, z) \simeq I\, z\, \beta(x.x, z) \cdot \beta(x.x, z)\, I$*)*

Facts on residual systems (with composition)





Facts on residual systems (with composition)

Lemma (recent)

- $\langle\to, 1, /\rangle$ *generates reduction system w/ composition on* $\to^*$ *up to* $1 \cdot 1 = 1$
- $\precsim$ *is quasi-order with* $\phi \precsim \psi := \phi/\psi = 1$
- $\simeq$ *is congruence for operations with* $\simeq := \precsim \cap \succsim$
- $\simeq$*-quotient of* $\to^*$ *gives category for* $\twoheadrightarrow$ *with push-outs, epis*
- *natural order on reductions partial order with lubs and* $\cdot$ *left-cancellation* ($\phi \vee \psi$ *definable by* $\phi \cdot (\psi/\phi)$; *is Bayes' Theorem* $P(A \cap B) = P(A) \cdot P(B \mid A)$)

Example: category with pushouts/epis from $\to_\beta$

Construction stages:

1. facet $\to_\beta$-steps $\Longrightarrow$ multisteps $\multimap_\beta$ $(\to_\beta \subseteq \multimap_\beta \subseteq \twoheadrightarrow_\beta)$

Example: category with pushouts/epis from $\to_\beta$

Construction stages:

1. facet $\to_\beta$-steps $\Longrightarrow$ multisteps $\multimap_\beta$
2. $\multimap_\beta$ has diamond property $\Longrightarrow$ residuation $/$

Example: category with pushouts/epis from $\to_\beta$

Construction stages:

1. facet $\to_\beta$-steps $\implies$ multisteps $\multimap_\beta$
2. $\multimap_\beta$ has diamond property $\implies$ residuation $/$
3. check residual laws (1)–(4) for $/$ $\implies$ residual system $\langle \multimap_\beta, 1, / \rangle$

Example: category with pushouts/epis from $\to_\beta$

Construction stages:

1. facet $\to_\beta$-steps $\Longrightarrow$ multisteps $\multimap_\beta$
2. $\multimap_\beta$ has diamond property $\Longrightarrow$ residuation $/$
3. check residual laws (1)–(4) for $/$ $\Longrightarrow$ residual system $\langle \multimap_\beta, 1, / \rangle$
4. generate rs with composition $\langle \multimap_\beta^*, 1^*, /^*, \cdot \rangle$ on multistep reductions

Example: category with pushouts/epis from $\to_\beta$

Construction stages:

1. facet $\to_\beta$-steps $\implies$ multisteps $\multimap_\beta$
2. $\multimap_\beta$ has diamond property $\implies$ residuation $/$
3. check residual laws (1)–(4) for $/$ $\implies$ residual system $\langle \multimap_\beta, 1, / \rangle$
4. generate rs with composition $\langle \multimap_\beta^*, 1^*, /^*, \cdot \rangle$ on multistep reductions
5. quotient out $\simeq$ $\implies$ category w/ pushouts,epis $\langle \twoheadrightarrow_\beta, 1^*, /^*, \cdot \rangle$ on reductions

Example: category with pushouts/epis from $\to_\beta$

Construction stages:

1. facet $\to_\beta$-steps $\implies$ multisteps $\multimap_\beta$
2. $\multimap_\beta$ has diamond property $\implies$ residuation $/$
3. check residual laws (1)–(4) for $/$ $\implies$ residual system $\langle \multimap_\beta, 1, / \rangle$
4. generate rs with composition $\langle \multimap_\beta^*, 1^*, /^*, \cdot \rangle$ on multistep reductions
5. quotient out $\simeq$ $\implies$ category w/ pushouts,epis $\langle \twoheadrightarrow_\beta, 1^*, /^*, \cdot \rangle$ on reductions

Remark

faceting also works for positive/generalised braids, associativity, ortho TRSs or HRSs, ... but first step can also be done by magic (as long as the rest works)

Axioms on multi-redexes of [Melliès 02]

Axioms

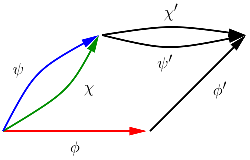
(*self-destruction*, SD) no redex has a residual after itself (as step)

(*finiteness*, F) every redex has finitely many residuals after a step

(*finite developments*, FD) developments of multi-redexes are finite

(*permutation*, PERM) every peak ϕ, ψ of steps can be completed by a valley of complete developments of the residuals of ψ after ϕ, respectively the residuals of ϕ after ψ, such that both legs of the resulting local confluence diagram induce the same redex-trace relation

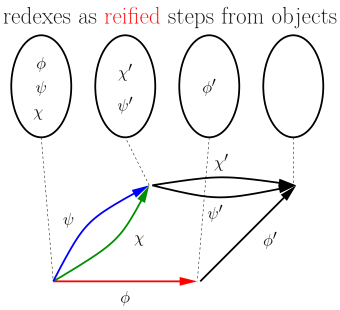
Visualisation and formalisation of multi-redex axioms



Remark

some rewrite system

Visualisation and formalisation of multi-redex axioms

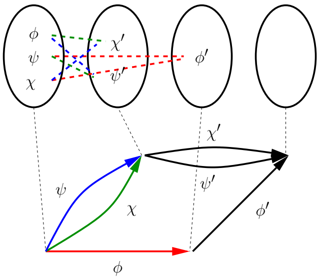


Remark

redex is reified step from a given object; multi-redex is set of such

Visualisation and formalisation of multi-redex axioms

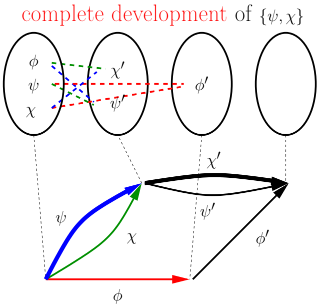
redex-tracing relating redexes in source,target of step



Remark

redex-tracing of step relating redexes in its source and target (residuals) pointwise extended to multi-redexes

Visualisation and formalisation of multi-redex axioms



Remark

development of multi-redex as reduction only contracting residuals

complete if no residuals remaining

Visualisation and formalisation of multi-redex axioms

Axioms

(SD) $(\phi\ [\![\phi\rangle\!\rangle) = \emptyset$ where $[\![\phi\rangle\!\rangle$ is redex-trace relation of ϕ

(F) $(\psi\ [\![\phi\rangle\!\rangle)$ is finite for co-initial ϕ, ψ

(FD) $\phi_1 \cdot \ldots \cdot \phi_n$ development of multi-redex Φ if $\phi_{i+1} \in (\Phi\ [\![\phi_1 \cdot \ldots \cdot \phi_i\rangle\!\rangle)$ for all i
complete if no residuals remaing

(PERM) each peak ϕ, ψ of steps is completed by valley γ, δ of complete developments of $(\psi\ [\![\phi\rangle\!\rangle), (\phi\ [\![\psi\rangle\!\rangle)$ with $[\![\phi \cdot \gamma\rangle\!\rangle = [\![\psi \cdot \delta\rangle\!\rangle$

Visualisation and formalisation of multi-redex axioms

Axioms

(SD) $(\phi \llbracket \phi \rangle\!\rangle) = \emptyset$ where $\llbracket \phi \rangle\!\rangle$ is redex-trace relation of ϕ

(F) $(\psi \llbracket \phi \rangle\!\rangle)$ is finite for co-initial ϕ, ψ

(FD) $\phi_1 \cdot \ldots \cdot \phi_n$ development of multi-redex Φ if $\phi_{i+1} \in (\Phi \llbracket \phi_1 \cdot \ldots \cdot \phi_i \rangle\!\rangle)$ for all i
complete if no residuals remaing

(PERM) each peak ϕ, ψ of steps is completed by valley γ, δ of complete developments of $(\psi \llbracket \phi \rangle\!\rangle), (\phi \llbracket \psi \rangle\!\rangle)$ with $\llbracket \phi \cdot \gamma \rangle\!\rangle = \llbracket \psi \cdot \delta \rangle\!\rangle$

Example ($\to_\beta$)

developments and redex-tracing as in [Church–Rosser 36]; axioms hold

From multi-redexes to multisteps

Lemma

$\langle \multimap\!\!\to, 1, / \rangle$ *is a residual system having joins, for* $\multimap\!\!\to$ *the rewrite system having as objects the objects of* $\to$*, and as steps a multi-redex* $a^{\Phi} : a \multimap\!\!\to b$ *if there is a complete development of* Φ *from a to b;* 1_a *defined as* $\emptyset$*; and residual* Φ / Ψ *defined as* $(\Phi [\![\Psi\rangle\!\rangle)$ *(for* Ψ *any complete development of* Ψ*).*

From multi-redexes to multisteps

Lemma

$\langle \multimap, 1, / \rangle$ is a residual system having joins, for $\multimap$ the rewrite system having as objects the objects of $\to$, and as steps a multi-redex $a^{\Phi} : a \multimap b$ if there is a complete development of Φ from a to b; 1_a defined as $\emptyset$; and residual Φ/Ψ defined as $(\Phi \llbracket \Psi \rangle\!\rangle)$ (for Ψ any complete development of Ψ).

Proof intuition.

all complete developments of Φ same redex-tracing by (PERM), by induction (FD) guarantees no ∞ interaction of redexes in $\Phi \implies$ induction measure $\Box$

From multi-redexes to multisteps

Lemma

$\langle \multimap\!\!\rightarrow, 1, / \rangle$ is a residual system having joins, for $\multimap\!\!\rightarrow$ the rewrite system having as objects the objects of $\rightarrow$, and as steps a multi-redex $a^{\Phi} : a \multimap\!\!\rightarrow b$ if there is a complete development of Φ from a to b; 1_a defined as $\emptyset$; and residual Φ/Ψ defined as $(\Phi [\![\Psi\rangle\!\rangle)$ (for Ψ any complete development of Ψ).

Definition

local homotopy $\equiv_l$ on reductions with the same sources/targets obtained by identifying legs of (PERM) diagrams
formally: equivalence generated by closing $\phi \cdot \gamma \equiv_l \psi \cdot \delta$ for peaks ϕ, ψ and valleys γ, δ given by (PERM) under composition: if $\gamma \equiv_l \gamma'$ then $\delta' \cdot \gamma \cdot \epsilon' \equiv_l \delta' \cdot \gamma' \cdot \epsilon'$.

From multi-redexes to multisteps

Lemma

$\langle \multimap, 1, / \rangle$ is a residual system having joins, for $\multimap$ the rewrite system having as objects the objects of $\to$, and as steps a multi-redex $a^\Phi : a \multimap b$ if there is a complete development of Φ from a to b; 1_a defined as $\emptyset$; and residual Φ/Ψ defined as $(\Phi \llbracket \Psi \rangle\rangle)$ (for Ψ any complete development of Ψ).

Lemma

$\simeq = \equiv_l$

Proof.

by showing $\simeq = \equiv = \equiv_l$ where $\equiv$ is square homotopy obtained by identifying legs of diamonds of multisteps

embeddings needed to mediate between $\to$-reductions and $\multimap$-reductions $\square$

From multi-redexes to multisteps

Lemma

$\langle \multimap, 1, / \rangle$ is a residual system having joins, for $\multimap$ the rewrite system having as objects the objects of $\to$, and as steps a multi-redex $a^{\Phi} : a \multimap b$ if there is a complete development of Φ from a to b; 1_a defined as $\emptyset$; and residual Φ / Ψ defined as $(\Phi \llbracket \Psi \rangle\rangle)$ (for Ψ any complete development of Ψ).

Lemma

$\simeq \; = \; \equiv_l$

Corollary

reductions up to local homotopy have push-outs and are epis.

Conclusions

- residuation $\implies$ upper bounds (of pairs of co-initial steps)
- residual system $\implies$ least upper bounds (of finite co-initial steps)
- multi-redexes $\implies$ sufficient to construct residual system

Reflections

- no light between residuation and confluence
  (papers stating to prove confluence not using residuals: empty statement)
- residuation breaks primacy of composition
  (residuation total but composition only partial)
- residuation a perspective on causality (cf. Winskel 89, Terese 03, Wolfram)
  (does causality involve FD? philosophical/ysics question; cf. proceedings)
- FFD (finite family developments) corresponds to FD of 2-rewriting.
  important but subtle (see proceedings): suggest to formalise FFD (for HRSs)
- residuation in founding papers of: λ-calculus (Church & Rosser, TLCA), rewriting (Newman, RTA), and in FSCD book (Huet)
  (FSCD PC/SC does not respect this: suggest to remove RTA/TLCA/FSCD book from FSCD page and from CfP)