Abstract Rewriting

ISR 2008, Obergurgl, Austria

Vincent van Oostrom

Theoretical Philosophy
Utrecht University
Netherlands

16:00 – 17:30, Mon/Wednesday July 23, ISR 2008

Reintroduction

Abstract rewriting

- Newman 1942 (confluence, orthogonality)
- Hindley, Rosen, de Bruijn (orthogonality, commutation)
- Klop, Huet, Geser (abstract reduction as framework)
- Jouannaud/Kirchner, Ohlebusch (rewriting modulo)
- Melliès, Khasidashvili (standardisation, neededness)
- Ghani/Lüth (substitution)
- ...

Abstract rewriting

- Newman 1942 (confluence, orthogonality)
- Hindley, Rosen, de Bruijn (orthogonality, commutation)
- Klop, Huet, Geser (abstract reduction as framework)
- Jouannaud/Kirchner, Ohlebusch (rewriting modulo)
- Melliès, Khasidashvili (standardisation, neededness)
- Ghani/Lüth (substitution)
- ...

Standard notions

Newman	modern	notations I use
cell	step	$\to$
path	conversion	$\leftrightarrow^*$
descending path	reduction/rewriting seq.	$\twoheadrightarrow$
lower bound	common reduct	$\downarrow$
upper bound	common ancestor	$\uparrow$
property A	Church–Rosser property	$\leftrightarrow^* \subseteq \twoheadleftarrow \cdot \twoheadrightarrow$
property B	confluence property	$\twoheadleftarrow \cdot \twoheadrightarrow \subseteq \twoheadrightarrow \cdot \twoheadleftarrow$ $(\uparrow \subseteq \downarrow)$
property C	semi-confluence	$\leftarrow \cdot \twoheadrightarrow \subseteq \twoheadrightarrow \cdot \twoheadleftarrow$
property D	local confluence	$\leftarrow \cdot \to \subseteq \twoheadrightarrow \cdot \twoheadleftarrow$
derivate	residual	$/$
conversion calc.	λ-calculus	

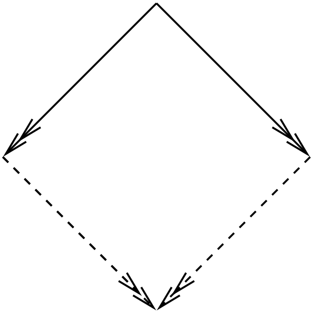
Plan

- Monday
- formalism: abstract rewrite relations (whether, Terese Ch. 1)
- A set of objects
- $\rightarrow \subseteq A \times A$ rewrite relation on A
- confluence property, lower bounds
- proof method: decreasing diagrams (Terese Ch. 14)
- proof method: Z property

Plan

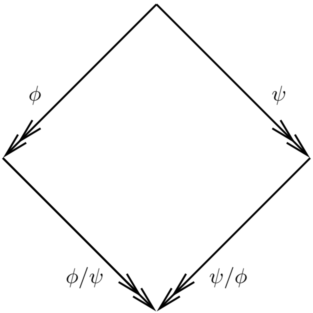
- Wednesday
- formalism: abstract rewrite systems (how, Terese Ch. 8)
- A set of objects
- $\to$ set of rewrite steps with source/target maps
- orthogonality, greatest lower bounds
- axiomatisation: residual systems (Terese Ch. 8.7)
- proof method: confluification into multi-steps

Confluence vs. orthogonality



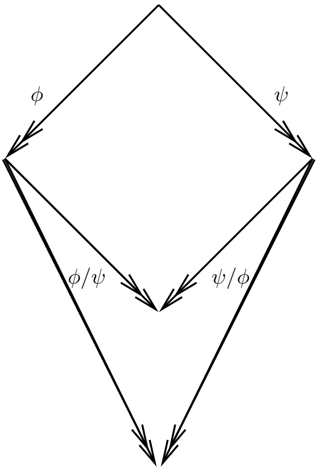
confluence, lower bound

Confluence vs. orthogonality



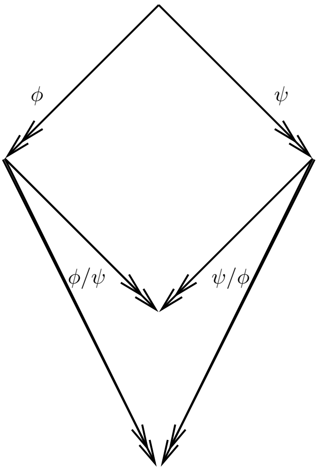
confluence, lower bound via witnessing residual function /

Confluence vs. orthogonality



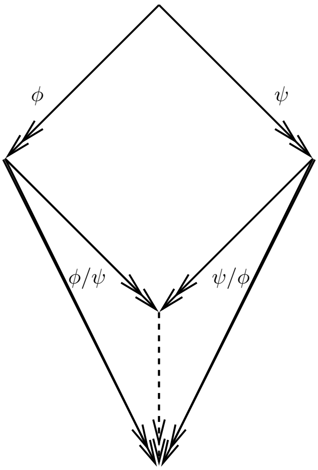
orthogonality, other lower bounds ...

Confluence vs. orthogonality



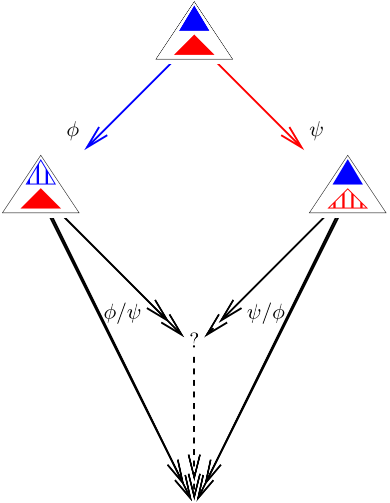
orthogonality, best among lower bounds?

Confluence vs. orthogonality



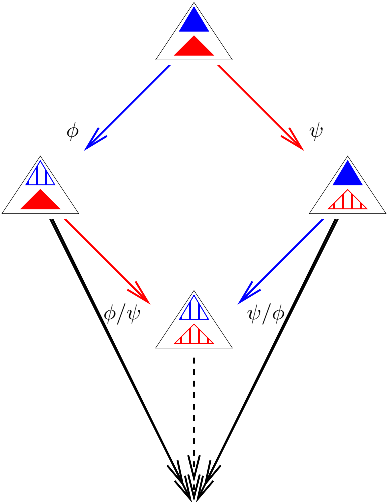
orthogonality, greatest lower bound

Confluence vs. orthogonality



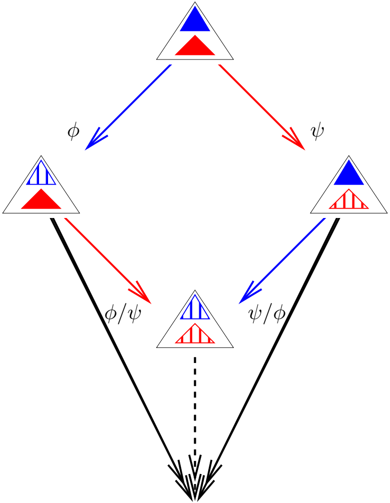
orthogonality, greatest lower bound = doing work of both?

Confluence vs. orthogonality



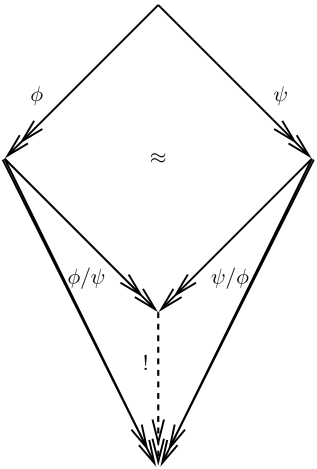
orthogonality, greatest lower bound = doing work of both?

Confluence vs. orthogonality



orthogonality, greatest lower bound $\neq$ doing work of both in $I(IK)$

Confluence vs. orthogonality



orthogonality, greatest lower bound w.r.t. notion of same work $\approx$

How to axiomatise orthogonality?

How to axiomatise orthogonality?

for rewriting (steps not transitive)

How to axiomatise orthogonality?

for rewriting (steps not transitive)
Newman 1942:

> The purpose of this paper is to make a start on a general theory of "sets of moves" by obtaining some conditions under which the answers to both the above questions are favorable. The results are essentially about "partially-ordered" systems, i.e. sets in which there is a transitive relation $>$, and sufficient conditions are given for every two elements to have a lower bound (i.e. for the set to be "directed") if it is known that every two "sufficiently near" elements have a lower bound. What further conditions are required for the existence of a *greatest* lower bound is not relevant to the present purpose, and is reserved for a later discussion.

Abstract rewrite system

Definition

ARS $\rightarrow$ is $\langle A, \Phi, \mathsf{src}, \mathsf{tgt}\rangle$

- A set of objects a, b, c, ...
- Φ set of steps ϕ, ψ, χ, ...
- src, tgt : $\Phi \rightarrow A$
  source and target functions

Abstract rewrite system

Definition

ARS $\rightarrow$ is $\langle A, \Phi, \text{src}, \text{tgt} \rangle$

- A set of objects a, b, c, ...
- Φ set of steps ϕ, ψ, χ, ...
- src, tgt : $\Phi \rightarrow A$
  source and target functions

$\phi : a \rightarrow b$ denotes step ϕ with source a and target b

Abstract rewrite system

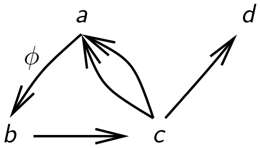
Definition

ARS $\rightarrow$ is $\langle A, \Phi, \mathrm{src}, \mathrm{tgt}\rangle$

- A set of objects a, b, c, ...
- Φ set of steps ϕ, ψ, χ, ...
- src, tgt : $\Phi \rightarrow A$
  source and target functions

$\phi : a \rightarrow b$ denotes step ϕ with source a and target b

ARS is directed graph, e.g.



Newman's axioms for residuals

(J$_1$) *If* $\xi J \eta$, $\xi \mid \eta$ *has precisely one member.*

(J$_2$) *If* $\eta_1 \in \xi_1 \mid \zeta$ *and* $\eta_2 \in \xi_2 \mid \zeta$, *and if* $\xi_1 J \xi_2$ *or* $\xi_1 = \xi_2$, *then* $\eta_1 J \eta_2$ *or* $\eta_1 = \eta_2$.

J represents non-nesting of redexes

Newman's axioms for residuals

(J$_1$) *If* $\xi J \eta$, $\xi \mid \eta$ *has precisely one member.*
(J$_2$) *If* $\eta_1 \in \xi_1 \mid \zeta$ *and* $\eta_2 \in \xi_2 \mid \zeta$, *and if* $\xi_1 J \xi_2$ *or* $\xi_1 = \xi_2$, *then* $\eta_1 J \eta_2$ *or* $\eta_1 = \eta_2$.

J represents non-nesting of redexes

Example (Schroer)

λ-calculus does not satisfy Newman's axioms
$\omega(\lambda y.\omega y) \to (\lambda y.\omega y)\lambda y.\omega y \to \underline{\omega(\lambda y.\omega y)} \to (\lambda y.\omega y)\lambda y.\omega y$
with $\omega = \lambda x.xx$

Newman's axioms for residuals

(J_1) *If* $\xi J \eta$, $\xi \mid \eta$ *has precisely one member.*
(J_2) *If* $\eta_1 \in \xi_1 \mid \zeta$ *and* $\eta_2 \in \xi_2 \mid \zeta$, *and if* $\xi_1 J \xi_2$ *or* $\xi_1 = \xi_2$, *then* $\eta_1 J \eta_2$ *or* $\eta_1 = \eta_2$.

J represents non-nesting of redexes

Example (Schroer)

λ-calculus does not satisfy Newman's axioms
$\omega(\lambda y.\omega y) \to (\lambda y.\omega y)\lambda y.\omega y \to \underline{\omega(\lambda y.\omega y)} \to (\lambda y.\omega y)\lambda y.\omega y$
with $\omega = \lambda x.xx$

- by (J_2) residuals of ωy are (mutually) J-related.

Newman's axioms for residuals

(J_1) *If* $\xi J \eta$, $\xi \mid \eta$ *has precisely one member.*
(J_2) *If* $\eta_1 \in \xi_1 \mid \zeta$ *and* $\eta_2 \in \xi_2 \mid \zeta$, *and if* $\xi_1 J \xi_2$ *or* $\xi_1 = \xi_2$, *then* $\eta_1 J \eta_2$ *or* $\eta_1 = \eta_2$.

J represents non-nesting of redexes

Example (Schroer)

λ-calculus does not satisfy Newman's axioms
$\omega(\lambda y.\omega y) \to (\lambda y.\omega y)\lambda y.\omega y \to \underline{\omega(\lambda y.\omega y)} \to (\lambda y.\omega y)\lambda y.\omega y$
with $\omega = \lambda x.xx$

- by (J_2) residuals of ωy are (mutually) J-related.
- by (J_2) whole term and ωy-redex are (mutually) J-related.

Newman's axioms for residuals

(J_1) *If* $\xi J \eta$, $\xi \mid \eta$ *has precisely one member.*
(J_2) *If* $\eta_1 \in \xi_1 \mid \zeta$ *and* $\eta_2 \in \xi_2 \mid \zeta$, *and if* $\xi_1 J \xi_2$ *or* $\xi_1 = \xi_2$, *then* $\eta_1 J \eta_2$ *or* $\eta_1 = \eta_2$.

J represents non-nesting of redexes

Example (Schroer)

λ-calculus does not satisfy Newman's axioms
$\omega(\lambda y.\omega y) \to (\lambda y.\omega y)\lambda y.\omega y \to \underline{\omega(\lambda y.\omega y)} \to (\lambda y.\omega y)\lambda y.\omega y$
with $\omega = \lambda x.xx$

- by (J_2) residuals of ωy are (mutually) J-related.
- by (J_2) whole term and ωy-redex are (mutually) J-related.
- the ωy-redex is duplicated violating (J_1).

From term rewrite system to ARS

From term rewrite system to ARS

combinatory logic (CL) rules:

$$\begin{array}{rcl} Ix & \rightarrow & x \\ Kxy & \rightarrow & x \\ Sxyz & \rightarrow & xz(yz) \end{array}$$

From term rewrite system to ARS

named combinatory logic (CL) rules:

$$
\begin{aligned}
\iota(x) :& \quad Ix &\to& \quad x \\
\kappa(x,y) :& \quad Kxy &\to& \quad x \\
\varsigma(x,y,z) :& \quad Sxyz &\to& \quad xz(yz)
\end{aligned}
$$

From term rewrite system to ARS

named combinatory logic (CL) rules:

$$\begin{array}{rrcl} \iota(x): & Ix & \to & x \\ \kappa(x,y): & Kxy & \to & x \\ \varsigma(x,y,z): & Sxyz & \to & xz(yz) \end{array}$$

Definition

multi-step ARS $\multimap\!\!\!\to$:

From term rewrite system to ARS

named combinatory logic (CL) rules:

$$\begin{aligned} \iota(x) : &\quad Ix \quad \rightarrow \quad x \\ \kappa(x,y) : &\quad Kxy \quad \rightarrow \quad x \\ \varsigma(x,y,z) : &\quad Sxyz \quad \rightarrow \quad xz(yz) \end{aligned}$$

Definition

multi-step ARS $\multimap\!\!\rightarrow$:

- objects: terms over alphabet

From term rewrite system to ARS

named combinatory logic (CL) rules:

$$\begin{aligned} \iota(x) :&\quad Ix \quad\to\quad x \\ \kappa(x,y) :&\quad Kxy \quad\to\quad x \\ \varsigma(x,y,z) :&\quad Sxyz \quad\to\quad xz(yz) \end{aligned}$$

Definition

multi-step ARS $\multimap\!\!\!\rightarrow$:

- objects: terms over alphabet
- steps: terms over function symbols + rule names

From term rewrite system to ARS

named combinatory logic (CL) rules:

$$\begin{array}{rrcl} \iota(x): & Ix & \to & x \\ \kappa(x,y): & Kxy & \to & x \\ \varsigma(x,y,z): & Sxyz & \to & xz(yz) \end{array}$$

Definition

multi-step ARS $\multimap\!\!\to$:

- objects: terms over alphabet
- steps: terms over function symbols + rule names
- $\mathrm{src}(f(\vec{s})) = f(\mathrm{src}(\vec{s}))$ with f function symbol
  $\mathrm{src}(\varrho(\vec{s})) = l(\mathrm{src}(\vec{s}))$ with $\varrho(\vec{x})$ name of rule $l(\vec{x}) \to r(\vec{x})$

From term rewrite system to ARS

named combinatory logic (CL) rules:

$$\begin{array}{rrcl} \iota(x) : & Ix & \to & x \\ \kappa(x,y) : & Kxy & \to & x \\ \varsigma(x,y,z) : & Sxyz & \to & xz(yz) \end{array}$$

Definition

multi-step ARS $\multimap$:

- objects: terms over alphabet
- steps: terms over function symbols + rule names
- $\mathrm{src}(f(\vec{s})) = f(\mathrm{src}(\vec{s}))$ with f function symbol
  $\mathrm{src}(\varrho(\vec{s})) = l(\mathrm{src}(\vec{s}))$ with $\varrho(\vec{x})$ name of rule $l(\vec{x}) \to r(\vec{x})$

step ARS $\to$: restriction of $\multimap$ steps to exactly one rule name

From term rewrite system to ARS

named combinatory logic (CL) rules:

$$\begin{array}{rrcl} \iota(x): & Ix & \to & x \\ \kappa(x,y): & Kxy & \to & x \\ \varsigma(x,y,z): & Sxyz & \to & xz(yz) \end{array}$$

Definition

multi-step ARS $\multimap$:

- objects: terms over alphabet
- steps: terms over function symbols + rule names
- $\mathrm{src}(f(\vec{s})) = f(\mathrm{src}(\vec{s}))$ with f function symbol
  $\mathrm{src}(\varrho(\vec{s})) = l(\mathrm{src}(\vec{s}))$ with $\varrho(\vec{x})$ name of rule $l(\vec{x}) \to r(\vec{x})$

step ARS $\to$: restriction of $\multimap$ steps to exactly one rule name

$\iota(IK): I(IK) \multimap IK$ $\quad$ $I(\iota(K)): I(IK) \multimap IK$

$I(IK): I(IK) \multimap I(IK)$ $\quad$ $\iota(\iota(K)): I(IK) \multimap K$

From term rewrite system to ARS

named combinatory logic (CL) rules:

$$\begin{array}{rrcl} \iota(x): & Ix & \to & x \\ \kappa(x,y): & Kxy & \to & x \\ \varsigma(x,y,z): & Sxyz & \to & xz(yz) \end{array}$$

Definition

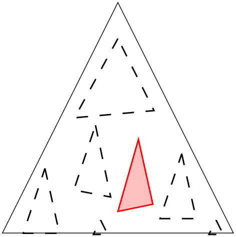
multi-step ARS $\multimap$:

- objects: terms over alphabet
- steps: terms over function symbols + rule names
- $\mathrm{src}(f(\vec{s})) = f(\mathrm{src}(\vec{s}))$ with f function symbol
  $\mathrm{src}(\varrho(\vec{s})) = l(\mathrm{src}(\vec{s}))$ with $\varrho(\vec{x})$ name of rule $l(\vec{x}) \to r(\vec{x})$

step ARS $\to$: restriction of $\multimap$ steps to exactly one rule name

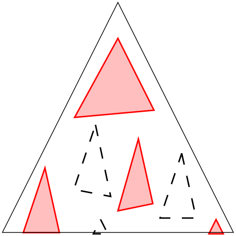
$\iota(IK) : I(IK) \to IK$ $\qquad$ $I(\iota(K)) : I(IK) \to IK$

Steps vs. multi-steps vs. full-developments



step $\rightarrow$: contract one redex-pattern

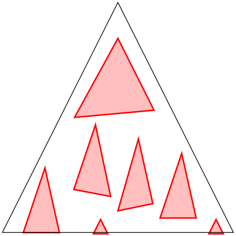
Steps vs. multi-steps vs. full-developments



multi-step $\multimap\!\!\rightarrow$ (development): contract some redex-patterns

$$\rightarrow \;\subseteq\; \multimap\!\!\rightarrow \;\subseteq\; \twoheadrightarrow$$

Steps vs. multi-steps vs. full-developments



full-development $\multimap\!\!\bullet\!\!\to$: contract all redex-patterns

$$\multimap\!\!\bullet\!\!\to \;\subseteq\; \multimap\!\!\circ\!\!\to \;\subseteq\; \twoheadrightarrow$$

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.

Example

residual of $I(\iota(K)) : I(IK) \multimap\!\!\!\to IK$ after
$\iota(IK) : I(IK) \multimap\!\!\!\to IK$?

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.

Example

residual of $I(\iota(K)) : \ I(IK) \multimap\!\!\!\rightarrow IK$ after
$\iota(IK) : \ I(IK) \multimap\!\!\!\rightarrow IK$?
$\iota(K) : \ IK \multimap\!\!\!\rightarrow K$!

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.

Example

residual of $I(\iota(K)) : \; I(IK) \multimap\!\!\!\rightarrow IK$ after
$\iota(IK) : \; I(IK) \multimap\!\!\!\rightarrow IK$?
$\iota(K) : \; IK \multimap\!\!\!\rightarrow K$!
and conversely?

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.

Example

residual of $I(\iota(K)) : I(IK) \multimap\!\!\rightarrow IK$ after
$\iota(IK) : I(IK) \multimap\!\!\rightarrow IK$?
$\iota(K) : IK \multimap\!\!\rightarrow K$!
and conversely?
same (but now residual is blue!)

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.

Example

residual of $SIK(IK) \multimap\!\!\rightarrow SIKK$ after
$SIK(IK) \multimap\!\!\rightarrow I(IK)(K(IK))$?

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.

Example

residual of $SIK(IK) \multimap\!\!\to SIKK$ after
$SIK(IK) \multimap\!\!\to I(IK)(K(IK))$?
$I(IK)(K(IK)) \multimap\!\!\to IK(KK)$!

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.

Example

residual of $SIK(IK) \multimap SIKK$ after
$SIK(IK) \multimap I(IK)(K(IK))$?
$I(IK)(K(IK)) \multimap IK(KK)$!
and conversely?

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.

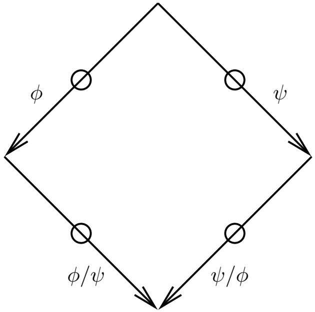
Example

residual of $SIK(IK) \multimap SIKK$ after
$SIK(IK) \multimap I(IK)(K(IK))$?
$I(IK)(K(IK)) \multimap IK(KK)$!
and conversely?
$SIKK \multimap IK(KK)$!

Residuals

Intuition

residual of step ϕ after step ψ:
what remains (to be done) of step ϕ after doing ψ.



ϕ/ψ and ψ/ϕ: multi-steps ending in same object

Residual system

Definition

residual system is ARS $\rightarrow$ extended with

- 1 the empty step for each object (doing nothing)

Residual system

Definition

residual system is ARS $\to$ extended with

- 1 the empty step for each object (doing nothing)
- $/$ the residual map from pairs of (co-initial) steps to steps

Residual system

Definition

residual system is ARS $\rightarrow$ extended with

- 1 the empty step for each object (doing nothing)
- $/$ the residual map from pairs of (co-initial) steps to steps
- satisfying axioms

$$\begin{array}{rcll} \phi/\phi & \approx & 1 & \\ \phi/1 & \approx & \phi & \\ 1/\phi & \approx & 1 & \\ (\phi/\psi)/(\chi/\psi) & \approx & (\phi/\chi)/(\psi/\chi) & \text{(cube)} \end{array}$$

Residual system

Definition

residual system is ARS $\rightarrow$ extended with

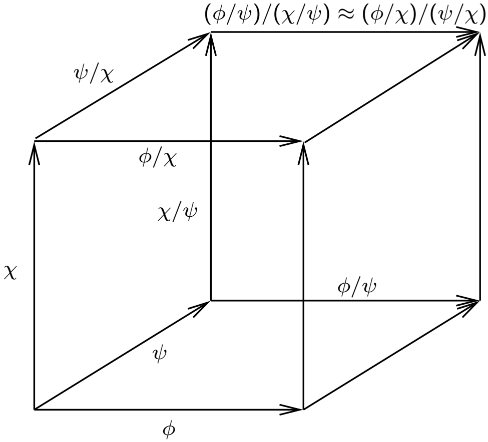
- 1 the empty step for each object (doing nothing)
- $/$ the residual map from pairs of (co-initial) steps to steps
- satisfying axioms

$$\begin{array}{rcll} \phi/\phi & \approx & 1 & \\ \phi/1 & \approx & \phi & \\ 1/\phi & \approx & 1 & \\ (\phi/\psi)/(\chi/\psi) & \approx & (\phi/\chi)/(\psi/\chi) & \text{(cube)} \end{array}$$

Exercise

show that third axiom is derivable

Cube axiom


$(\phi/\psi)/(\chi/\psi) \approx (\phi/\chi)/(\psi/\chi)$
ψ/χ
ϕ/χ
χ/ψ
χ
ϕ/ψ
ψ
ϕ

Residual system for orthogonal term rewrite systems

Definition

TRS is orthogonal if left-linear and non-overlapping

Residual system for orthogonal term rewrite systems

Definition

TRS is orthogonal if left-linear and non-overlapping

- multi-steps as steps

Residual system for orthogonal term rewrite systems

Definition

TRS is orthogonal if left-linear and non-overlapping

- multi-steps as steps
- residual operation defined by induction on multi-steps

$$
\begin{aligned}
f(\phi_1,\ldots,\phi_n)/f(\psi_1,\ldots,\psi_n) &= f(\phi_1/\psi_1,\ldots,\phi_n/\psi_n)\\
\varrho(\phi_1,\ldots,\phi_n)/l(\psi_1,\ldots,\psi_n) &= \varrho(\phi_1/\psi_1,\ldots,\phi_n/\psi_n)\\
l(\phi_1,\ldots,\phi_n)/\varrho(\psi_1,\ldots,\psi_n) &= r(\phi_1/\psi_1,\ldots,\phi_n/\psi_n)\\
\varrho(\phi_1,\ldots,\phi_n)/\varrho(\psi_1,\ldots,\psi_n) &= r(\phi_1/\psi_1,\ldots,\phi_n/\psi_n)
\end{aligned}
$$

for every rule $\varrho(x_1,\ldots,x_n) : l(x_1,\ldots,x_n) \to r(x_1,\ldots,x_n)$

Residual system for orthogonal term rewrite systems

Definition

TRS is orthogonal if left-linear and non-overlapping

- multi-steps as steps
- residual operation defined by induction on multi-steps

$$\begin{array}{rcl} f(\phi_1,\ldots,\phi_n)/f(\psi_1,\ldots,\psi_n) & = & f(\phi_1/\psi_1,\ldots,\phi_n/\psi_n) \\ \varrho(\phi_1,\ldots,\phi_n)/l(\psi_1,\ldots,\psi_n) & = & \varrho(\phi_1/\psi_1,\ldots,\phi_n/\psi_n) \\ l(\phi_1,\ldots,\phi_n)/\varrho(\psi_1,\ldots,\psi_n) & = & r(\phi_1/\psi_1,\ldots,\phi_n/\psi_n) \\ \varrho(\phi_1,\ldots,\phi_n)/\varrho(\psi_1,\ldots,\psi_n) & = & r(\phi_1/\psi_1,\ldots,\phi_n/\psi_n) \end{array}$$

for every rule $\varrho(x_1,\ldots,x_n) : \ l(x_1,\ldots,x_n) \to r(x_1,\ldots,x_n)$

Example

- $I(\iota(K))/\iota(IK) = \iota(K)$
- $SIK(\iota(K))/\varsigma(I,K,IK) = I(\iota(K))(K(\iota(K)))$

Residual order

Definition

$\phi \lesssim \psi$ if $\phi/\psi \approx 1$ (nothing remains)

Residual order

Definition

$\phi \lesssim \psi$ if $\phi/\psi \approx 1$ (nothing remains)

Theorem

$\lesssim$ *is a quasi-order*

Proof.

- reflexivity: $\phi/\phi \approx 1$
- transitivity: if $\phi/\psi \approx 1$ and $\psi/\chi \approx 1$ then $\phi/\chi \approx 1$

$\square$

Residual order

Definition

$\phi \lesssim \psi$ if $\phi/\psi \approx 1$ (nothing remains)

Theorem

$\lesssim$ *is a quasi-order*

Proof.

- reflexivity: $\phi/\phi \approx 1$
- transitivity: if $\phi/\psi \approx 1$ and $\psi/\chi \approx 1$ then $\phi/\chi \approx 1$

$\square$

Exercise

$\lesssim$ *is not necessarily a partial order (anti-symmetric)*

Residual order

Definition

$\phi \lesssim \psi$ if $\phi/\psi \approx 1$ (nothing remains)

Theorem

$\lesssim$ is a quasi-order

Proof.

- reflexivity: $\phi/\phi \approx 1$
- transitivity: if $\phi/\psi \approx 1$ and $\psi/\chi \approx 1$ then $\phi/\chi \approx 1$

□

Exercise

$\lesssim$ is not necessarily a partial order (anti-symmetric)

Theorem

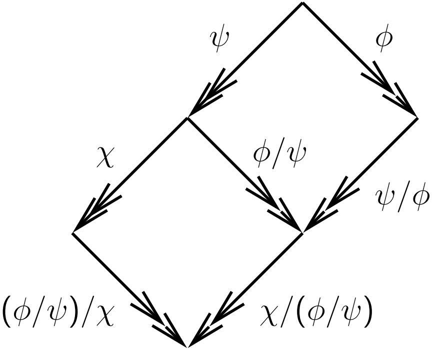
residual systems preserved by quotienting by $\lesssim \cap \gtrsim$.
yields a system having a residual order which is partial order.

From (multi-steps) to sequences

how to define residual system for sequences of (multi-)steps?

From (multi-steps) to sequences

how to define residual system for sequences of (multi-)steps?



$$\phi/(\psi \circ \chi) \approx (\phi/\psi)/\chi$$

$$(\psi \circ \chi)/\phi \approx (\psi/\phi) \circ (\chi/(\phi/\psi))$$

take residuals (multi-)stepwise

Residual system with composition

extending residual operation to sequences generates:

Definition

Residual system with composition

- 1 the empty reduction
- $/$ the residual map from pairs of (co-initial) reductions to reductions
- $\circ$ the composition map on composable reductions
- $$\begin{array}{rcl} \phi/\phi & \approx & 1 \\ \phi/1 & \approx & \phi \\ 1/\phi & \approx & 1 \\ (\phi/\psi)/(\chi/\psi) & \approx & (\phi/\chi)/(\psi/\chi) \\ 1 \circ 1 & \approx & 1 \\ \chi/(\phi \circ \psi) & \approx & (\chi/\phi)/\psi \\ (\phi \circ \psi)/\chi & \approx & (\phi/\chi) \circ (\psi/(\chi/\phi)) \end{array}$$

Residual order gives greatest lower bound

Theorem

residual systems with composition preserved by quotienting by $\lesssim \cap \gtrsim$.
yields a system having a residual order which is partial order.
$\phi \circ \psi / \phi$ *is greatest lower bound of* ϕ, ψ

Residual order gives greatest lower bound

Theorem

residual systems with composition preserved by quotienting by $\lesssim \cap \gtrsim$.
yields a system having a residual order which is partial order.
$\phi \circ \psi/\phi$ *is greatest lower bound of* ϕ, ψ

Example

Residual order gives greatest lower bound

Theorem

residual systems with composition preserved by quotienting by $\lesssim \cap \gtrsim$.
yields a system having a residual order which is partial order.
$\phi \circ \psi / \phi$ *is greatest lower bound of* ϕ, ψ

Example

- orthogonal TRSs

Residual order gives greatest lower bound

Theorem

residual systems with composition preserved by quotienting by $\lesssim \cap \gtrsim$*.*
yields a system having a residual order which is partial order.
$\phi \circ \psi/\phi$ *is greatest lower bound of* ϕ*,* ψ

Example

- orthogonal TRSs
- interaction nets

Residual order gives greatest lower bound

Theorem

residual systems with composition preserved by quotienting by $\lesssim \cap \gtrsim$.
yields a system having a residual order which is partial order.
$\phi \circ \psi / \phi$ *is greatest lower bound of* ϕ, ψ

Example

- orthogonal TRSs
- interaction nets
- λ-calculus

Residual order gives greatest lower bound

Theorem

residual systems with composition preserved by quotienting by $\lesssim \cap \gtrsim$.
yields a system having a residual order which is partial order.
$\phi \circ \psi / \phi$ *is greatest lower bound of* ϕ, ψ

Example

- orthogonal TRSs
- interaction nets
- λ-calculus
- orthogonal higher-order term rewriting systems

Residual order gives greatest lower bound

Theorem

residual systems with composition preserved by quotienting by $\lesssim \cap \gtrsim$.
yields a system having a residual order which is partial order.
$\phi \circ \psi / \phi$ *is greatest lower bound of* ϕ, ψ

Example

- orthogonal TRSs
- interaction nets
- λ-calculus
- orthogonal higher-order term rewriting systems
- ...

Non-standard examples of residual systems

Non-standard examples of residual systems

- sorting

Non-standard examples of residual systems

- sorting
- braids

Non-standard examples of residual systems

- sorting
- braids
- self-distributivity

Non-standard examples of residual systems

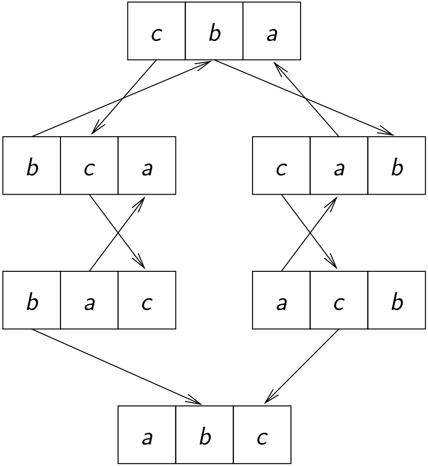
- sorting
- braids
- self-distributivity
- associativity

Non-standard examples of residual systems

- sorting
- braids
- self-distributivity
- associativity
- ...

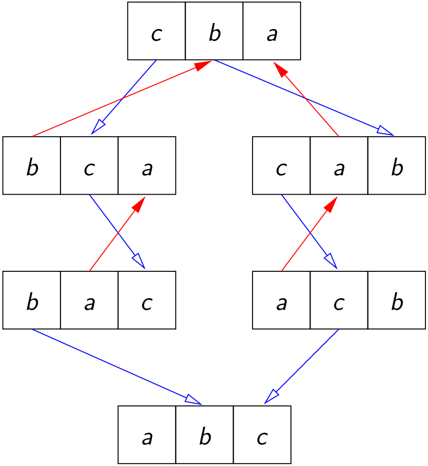
Sorting by swapping adjacent elements

Sorting by swapping adjacent elements



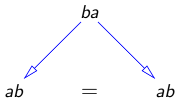
Reduction steps: arrows start at first element of swapped pair

Sorting by swapping adjacent elements

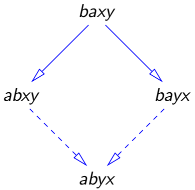


reduction steps: inversions in blue, anti-inversions in red

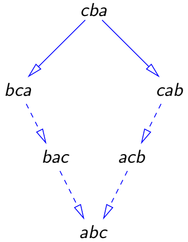
Inversion sort local confluence diagrams



same



independent



self-overlap

Residual system for inversion sort

- 1 the empty step
- / the residual map from pairs of steps to steps
- $$\begin{array}{rcl} \phi/\phi & \approx & 1 \\ \phi/1 & \approx & \phi \\ 1/\phi & \approx & 1 \\ (\phi/\psi)/(\chi/\psi) & \approx & (\phi/\chi)/(\psi/\chi) \end{array}$$

Residual system for inversion sort

Theorem

inversion sorting gives a residual system

Proof.

step ϕ from list ℓ is multi-inversion: relation $\widehat{\ }$ s.t. if $\widehat{ij}$

- out-of-order: $\ell = \ldots i \ldots j \ldots$ but $i > j$;
- transitive: if $\widehat{jk}$, then $\widehat{ik}$;
- scopic: if $\ell = \ldots i \ldots k \ldots j \ldots$, then either $\widehat{ik}$ or $\widehat{jk}$

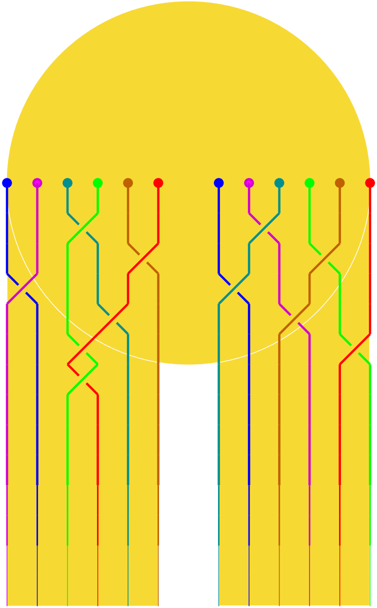
define 1 to be the empty relation,
define ϕ/ψ as $(\phi \cup \psi)^+ - \psi$. □

Example

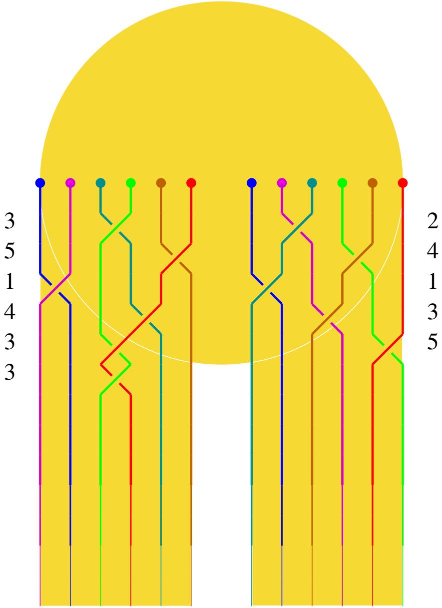
$$(cba \to_{\widehat{c}ba} bca)/(cba \to_{\widehat{cba}} cab) = (cab \to_{\widehat{cab}} abc)$$

Braid problem

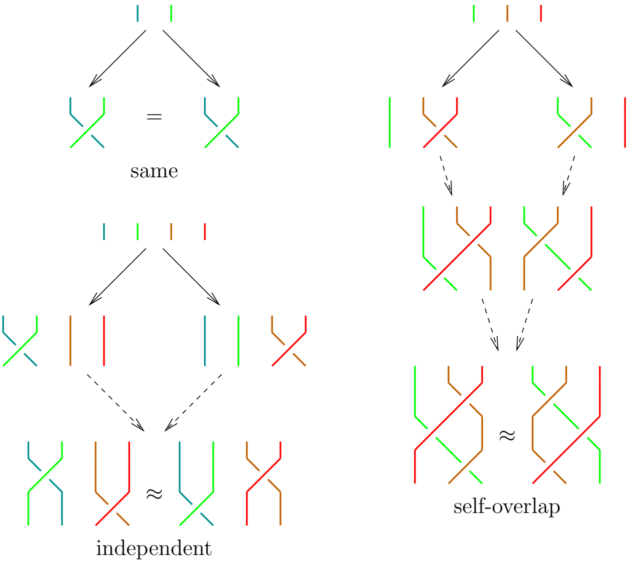
Braid problem



Braid problem

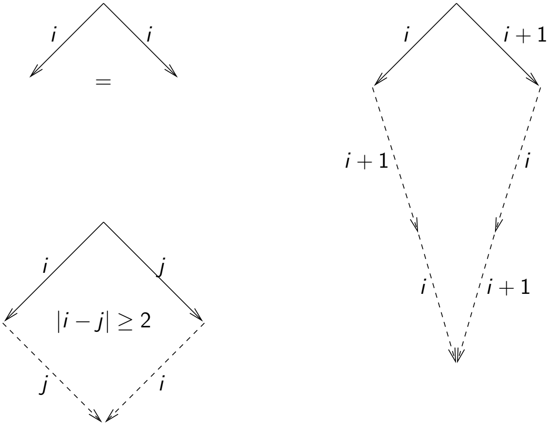


Braid confluence diagrams



reductions end in topologically equivalent (≈) braids

Braid confluence diagrams



reduction steps labelled by gap# of crossing
$ij \approx ji$ if $|i - j| \geq 2$ and $i(i+1)i \approx (i+1)i(i+1)$

Sorting vs. braiding

- sorting is braiding without crossing strands (inverting) twice

Sorting vs. braiding

- sorting is braiding without crossing strands (inverting) twice
- model braids as ‘repeated sorting’

Sorting vs. braiding

- sorting is braiding without crossing strands (inverting) twice
- model braids as ‘repeated sorting’
- model braids as reduction sequences of multi-inversions

Orthogonality of braids

Theorem

braiding gives a residual system with composition

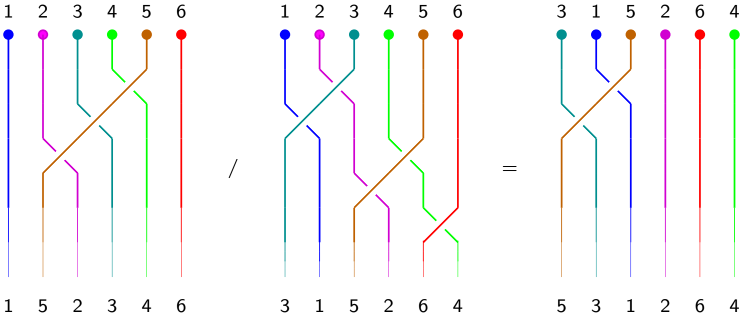
Proof.

- steps are sequences of multi-inversions
- without out-of-order restriction
- define $\circ$ to be formal composition
- $/$ on sequences defined via composition laws

$\square$

Orthogonality of braids

Example



Self-distributivity: $(x \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$

Self-distributivity: $(x \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$

Interpret as first projection

Self-distributivity: $(x \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$

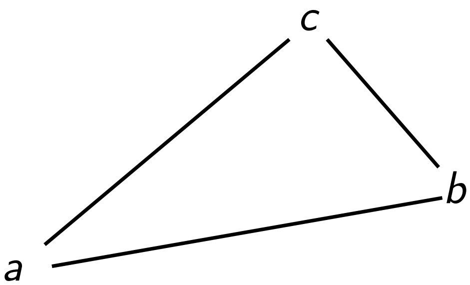
Interpret as an ACI-operation

$$
\begin{aligned}
(x \cdot y) \cdot z &=_A x \cdot (y \cdot z) \\
&=_I x \cdot (y \cdot (z \cdot z)) \\
&=_A x \cdot ((y \cdot z) \cdot z) \\
&=_C x \cdot (z \cdot (y \cdot z)) \\
&=_A (x \cdot z) \cdot (y \cdot z)
\end{aligned}
$$

Examples: disjunction/union, conjunction/intersection

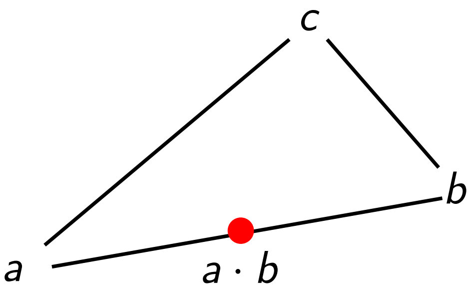
Self-distributivity: $(x \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$

Interpret as 'middle'



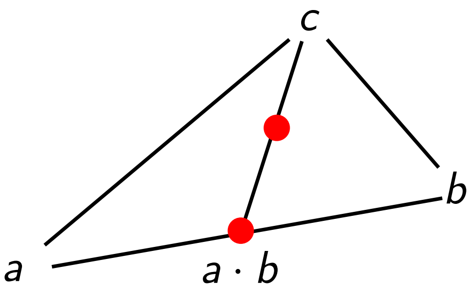
Self-distributivity: $(x \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$

Interpret as 'middle'



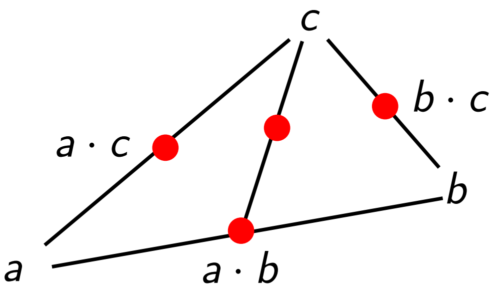
Self-distributivity: $(x \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$

Interpret as 'middle'



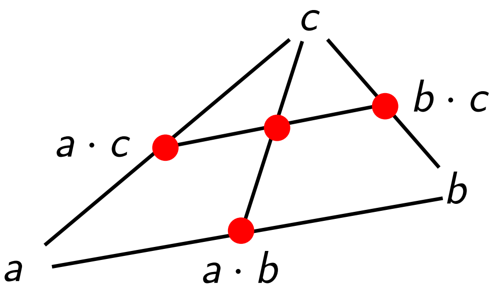
Self-distributivity: $(x \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$

Interpret as 'middle'



Self-distributivity: $(x \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z)$

Interpret as 'middle'



Self-distributivity rule: $xyz \longrightarrow xz(yz)$ critical pair

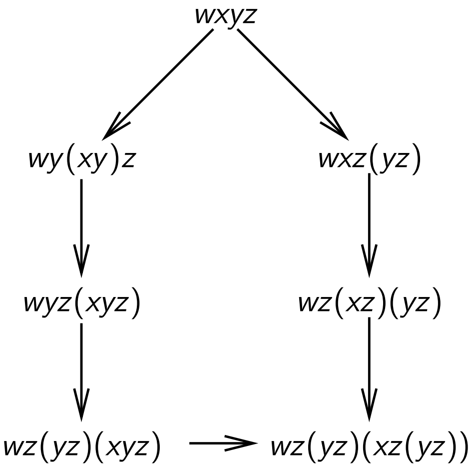
- applicative notation: $\cdot$ infix, associating to left

Self-distributivity rule: $xyz \longrightarrow xz(yz)$ critical pair

- applicative notation: $\cdot$ infix, associating to left
- as expansion rule better behaved than as reduction rule

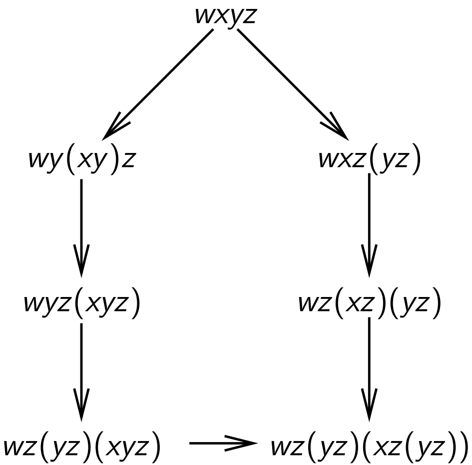
Self-distributivity rule: $xyz \to xz(yz)$ critical pair

- applicative notation: $\cdot$ infix, associating to left
- as expansion rule better behaved than as reduction rule
- a single critical pair:



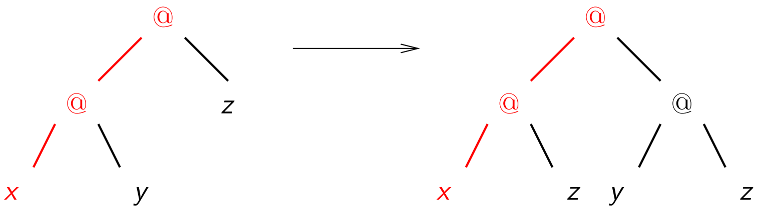
Self-distributivity rule: $xyz \rightarrow xz(yz)$ critical pair

- applicative notation: $\cdot$ infix, associating to left
- as expansion rule better behaved than as reduction rule
- a single critical pair:



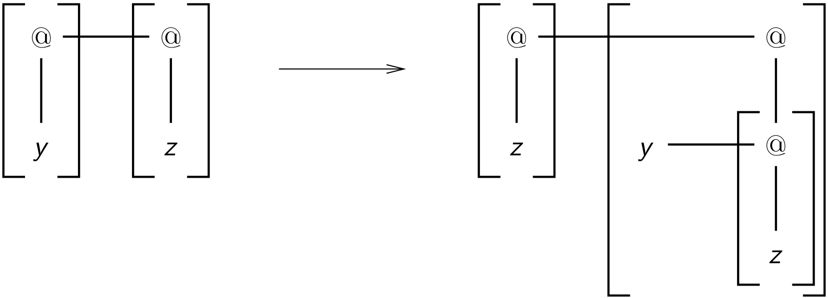
- w represents spine ...

Spine rectification



Spine is stable!

Spine rectification



If you don't have a spine, they can't break you

Self-distributivity rule: $[y][z] \to [z][y[z]]$

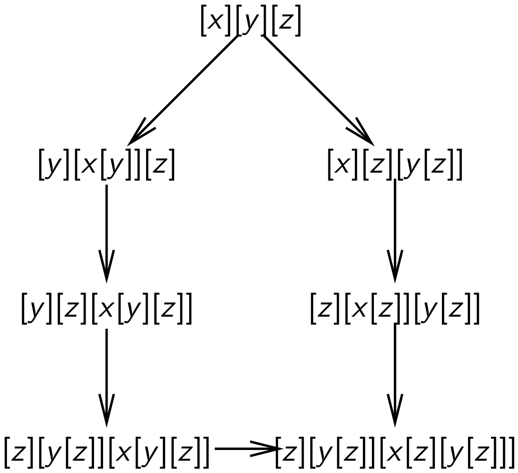
- elements on spine juxtaposed

Self-distributivity rule: $[y][z] \to [z][y[z]]$

- elements on spine juxtaposed
- rule to be applied modulo associativity

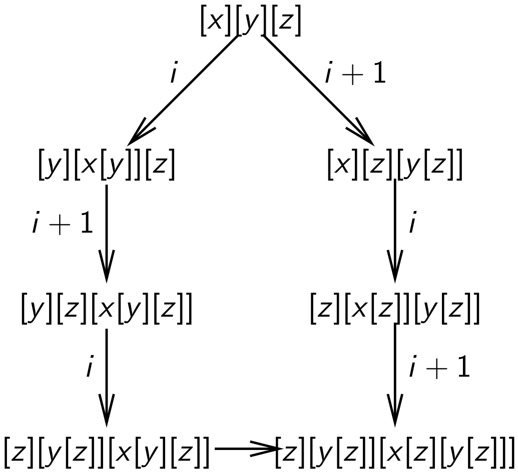
Self-distributivity rule: $[y][z] \to [z][y[z]]$

- elements on spine juxtaposed
- rule to be applied modulo associativity
- the critical pair becomes:



Self-distributivity rule: $[y][z] \to [z][y[z]]$

- elements on spine juxtaposed
- rule to be applied modulo associativity
- the critical pair becomes:



- almost braiding, but one extra step ...

Braiding vs. self-distributivity

- $[y][z] \to [z][y[z]]$ swaps z and y, remembering y crossed z...

Braiding vs. self-distributivity

- $[y][z] \to [z][y[z]]$ swaps z and y, remembering y crossed z...
- braids.

Braiding vs. self-distributivity

- $[y][z] \to [z][y[z]]$ swaps z and y, remembering y crossed z...
- braids.
- self-distributivity braids inside memory...

Braiding vs. self-distributivity

- $[y][z] \to [z][y[z]]$ swaps z and y, remembering y crossed z...
- braids.
- self-distributivity braids inside memory...
- extra step.

Orthogonality of self-distributivity

Theorem

self-distributivity gives a residual system

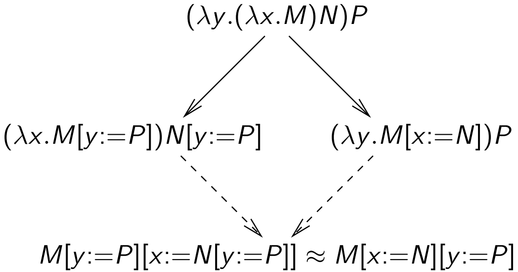
Idea.

Multi-distribution defined similar to multi-conversions, but

- relates positions in the (rectified) term
- may relate only to right-wing uncles; $(\widehat{piq)(pj})$ with $i < j$
- must be left-convex; $(\widehat{piq_1 q_2)(pj})$ implies $(\widehat{piq_1)(pj})$

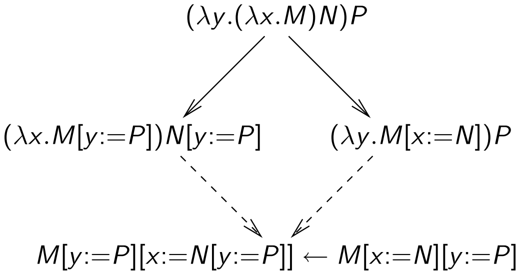
/ as before; constructed by using standard residual theory to relate positions before and after the (non-linear) term rewrite step □

Substitution lemma of λ-calculus as self-distributivity



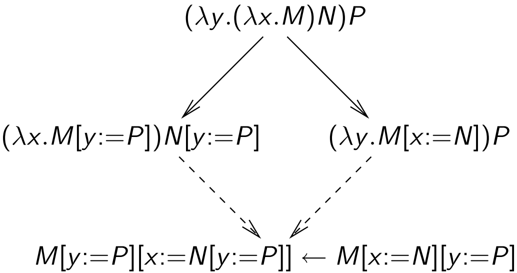
Substitution Lemma of the λ-calculus

Substitution lemma of λ-calculus as self-distributivity



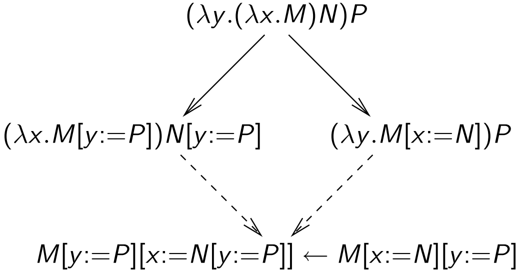
Critical pair for λ-calculus with explicit substitutions

Substitution lemma of λ-calculus as self-distributivity



Critical pair for λ-calculus with explicit substitutions
Is this rule in itself confluent? (left-to-right no)

Substitution lemma of λ-calculus as self-distributivity



Critical pair for λ-calculus with explicit substitutions
This is self-distributivity, so even orthogonal!

Confluification

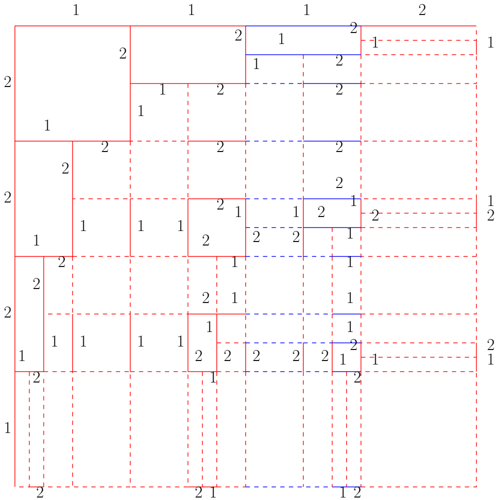
Definition

confluification if local confluence completed by sequences, adjoin these to steps.

Confluification

Definition

confluification if local confluence completed by sequences, adjoin these to steps.



Confluification

Definition

confluification if local confluence completed by sequences, adjoin these to steps.

- for orthogonal term rewriting systems: parallel reductions
- for λ-calculus: developments

From residual systems with composition to algebras

Example

- multi-inversions in sorting

From residual systems with composition to algebras

Example

- multi-inversions in sorting
- braids

From residual systems with composition to algebras

Example

- multi-inversions in sorting
- braids
- self-distributivity

From residual systems with composition to algebras

Example

- multi-inversions in sorting
- braids
- self-distributivity
- orthogonal term rewriting systems (β-reduction, CL)

From residual systems with composition to algebras

Example

- multi-inversions in sorting
- braids
- self-distributivity
- orthogonal term rewriting systems (β-reduction, CL)
- associativity

From residual systems with composition to algebras

Example

- multi-inversions in sorting
- braids
- self-distributivity
- orthogonal term rewriting systems (β-reduction, CL)
- associativity
- ...

From residual systems with composition to algebras

Example

- multi-inversions in sorting
- braids
- self-distributivity
- orthogonal term rewriting systems (β-reduction, CL)
- associativity
- ...
- also many residual algebras (singleton carrier) ...

Residual algebras (with composition)

- natural numbers (as steps from object to itself)
- $\dot{-}$ (cut-off subtraction), 0 (zero), $+$ (addition);

$$
\begin{aligned}
n \dot{-} n &\approx 0 \\
n \dot{-} 0 &\approx n \\
0 \dot{-} n &\approx 0 \\
(n \dot{-} m) \dot{-} (k \dot{-} m) &\approx (n \dot{-} k) \dot{-} (m \dot{-} k) \\
0 + 0 &\approx 0 \\
k \dot{-} (n + m) &\approx (k \dot{-} n) \dot{-} m \\
(n + m) \dot{-} k &\approx (n \dot{-} k) + (m \dot{-} (k \dot{-} n))
\end{aligned}
$$

Generated from its

Residual algebras (with composition)

- natural numbers (as steps from object to itself)
- $\dot{-}$ (cut-off subtraction), 0 (zero), $+$ (addition);

$$
\begin{aligned}
n \dot{-} n &\approx 0 \\
n \dot{-} 0 &\approx n \\
0 \dot{-} n &\approx 0 \\
(n \dot{-} m) \dot{-} (k \dot{-} m) &\approx (n \dot{-} k) \dot{-} (m \dot{-} k) \\
0 + 0 &\approx 0 \\
k \dot{-} (n + m) &\approx (k \dot{-} n) \dot{-} m \\
(n + m) \dot{-} k &\approx (n \dot{-} k) + (m \dot{-} (k \dot{-} n))
\end{aligned}
$$

Truth-values with reverse implication, false (no composition)

Positive natural numbers with cut-off division, 1, multiplication

Residual algebras (with composition)

- multisets over some set (as steps from object to itself)
- $-$ (multiset difference), $\emptyset$ (empty multiset), $\uplus$ (multiset sum);

$$
\begin{array}{rcl}
M - M & \approx & \emptyset \\
M - \emptyset & \approx & M \\
\emptyset - M & \approx & \emptyset \\
(M - N) - (K - N) & \approx & (M - K) - (N - K) \\
\emptyset \uplus \emptyset & \approx & \emptyset \\
K - (M \uplus N) & \approx & (K - M) - N \\
(M \uplus N) - K & \approx & (M - K) \uplus (N - (K - M))
\end{array}
$$

Residual algebras (with composition)

- multisets over some set (as steps from object to itself)
- $-$ (multiset difference), $\emptyset$ (empty multiset), $\uplus$ (multiset sum);

$$
\begin{array}{rcl}
M - M & \approx & \emptyset \\
M - \emptyset & \approx & M \\
\emptyset - M & \approx & \emptyset \\
(M - N) - (K - N) & \approx & (M - K) - (N - K) \\
\emptyset \uplus \emptyset & \approx & \emptyset \\
K - (M \uplus N) & \approx & (K - M) - N \\
(M \uplus N) - K & \approx & (M - K) \uplus (N - (K - M))
\end{array}
$$

Sets with set-difference, $\emptyset$, disjoint union.

Residual algebras (with composition)

- multisets over some set (as steps from object to itself)
- $-$ (multiset difference), $\emptyset$ (empty multiset), $\uplus$ (multiset sum);

$$
\begin{aligned}
M - M &\approx \emptyset \\
M - \emptyset &\approx M \\
\emptyset - M &\approx \emptyset \\
(M - N) - (K - N) &\approx (M - K) - (N - K) \\
\emptyset \uplus \emptyset &\approx \emptyset \\
K - (M \uplus N) &\approx (K - M) - N \\
(M \uplus N) - K &\approx (M - K) \uplus (N - (K - M))
\end{aligned}
$$

all compositions are commutative

commutative residual algebras

Definition

commutative residual algebra with composition (CRAC) satisfies

$$\begin{array}{rcl} (\phi/\psi)/\phi & \approx & 1 \\ \phi/(\phi/\psi) & \approx & \psi/(\psi/\phi) \end{array}$$

(follows from computing $(\phi \circ \psi)/(\psi \circ \phi) \approx 1$!)

commutative residual algebras

Definition

commutative residual algebra with composition (CRAC) satisfies

$$\begin{aligned}(\phi/\psi)/\phi &\approx 1\\ \phi/(\phi/\psi) &\approx \psi/(\psi/\phi)\end{aligned}$$

- 2nd equation states commutativity of intersection $\phi/(\phi/\psi)$

commutative residual algebras

Definition

commutative residual algebra with composition (CRAC) satisfies

$$
\begin{aligned}
(\phi/\psi)/\phi &\approx 1 \\
\phi/(\phi/\psi) &\approx \psi/(\psi/\phi)
\end{aligned}
$$

- 2nd equation states commutativity of intersection $\phi/(\phi/\psi)$
- Very useful for equational reasoning about multisets in Coq.

commutative residual algebras

Definition

commutative residual algebra with composition (CRAC) satisfies

$$
\begin{aligned}
(\phi/\psi)/\phi &\approx 1 \\
\phi/(\phi/\psi) &\approx \psi/(\psi/\phi)
\end{aligned}
$$

- 2nd equation states commutativity of intersection $\phi/(\phi/\psi)$
- Very useful for equational reasoning about multisets in Coq.
- Iso to commutative BCK algebras with relative cancellation

commutative residual algebras

Definition

commutative residual algebra with composition (CRAC) satisfies

$$\begin{array}{rcl}(\phi/\psi)/\phi & \approx & 1 \\ \phi/(\phi/\psi) & \approx & \psi/(\psi/\phi)\end{array}$$

- 2nd equation states commutativity of intersection $\phi/(\phi/\psi)$
- Very useful for equational reasoning about multisets in Coq.
- Iso to commutative BCK algebras with relative cancellation
- In above examples $\preceq$ well-founded; $a \preceq b$ if $a/b \approx 1$.

commutative residual algebras

Definition

commutative residual algebra with composition (CRAC) satisfies

$$\begin{aligned} (\phi/\psi)/\phi &\approx 1 \\ \phi/(\phi/\psi) &\approx \psi/(\psi/\phi) \end{aligned}$$

- 2nd equation states commutativity of intersection $\phi/(\phi/\psi)$
- Very useful for equational reasoning about multisets in Coq.
- Iso to commutative BCK algebras with relative cancellation
- In above examples $\preceq$ well-founded; $a \preceq b$ if $a/b \approx 1$.
- Other interesting CRACs?

commutative residual algebras

Definition

commutative residual algebra with composition (CRAC) satisfies

$$\begin{array}{rcl} (\phi/\psi)/\phi & \approx & 1 \\ \phi/(\phi/\psi) & \approx & \psi/(\psi/\phi) \end{array}$$

- 2nd equation states commutativity of intersection $\phi/(\phi/\psi)$
- Very useful for equational reasoning about multisets in Coq.
- Iso to commutative BCK algebras with relative cancellation
- In above examples $\preceq$ well-founded; $a \preceq b$ if $a/b \approx 1$.
- Other interesting CRACs?
- every well-founded CRAC iso to multiset CRAC

Conclusion

- decreasing diagrams: well-founded indexing
- Z-property: bullet-function
- orthogonal systems: axiomatised residual operation