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

<table>
<thead>
<tr>
<th>Newman</th>
<th>modern</th>
<th>notations I use</th>
</tr>
</thead>
<tbody>
<tr>
<td>cell</td>
<td>step</td>
<td>$\rightarrow$</td>
</tr>
<tr>
<td>path</td>
<td>conversion</td>
<td>$\leftrightarrow^\ast$</td>
</tr>
<tr>
<td>descending path</td>
<td>reduction/rewriting seq.</td>
<td>$\rightarrow$</td>
</tr>
<tr>
<td>lower bound</td>
<td>common reduct</td>
<td>$\downarrow$</td>
</tr>
<tr>
<td>upper bound</td>
<td>common ancestor</td>
<td>$\uparrow$</td>
</tr>
<tr>
<td>property A</td>
<td>Church–Rosser property</td>
<td>$\leftrightarrow^\ast \subseteq \leftarrow \cdot \rightarrow$</td>
</tr>
<tr>
<td>property B</td>
<td>confluence property</td>
<td>$\leftarrow \cdot \rightarrow \subseteq \cdot \leftarrow$ $(\uparrow \subseteq \downarrow)$</td>
</tr>
<tr>
<td>property C</td>
<td>semi-confluence</td>
<td>$\leftarrow \cdot \rightarrow \subseteq \rightarrow \cdot \leftarrow$</td>
</tr>
<tr>
<td>property D</td>
<td>local confluence</td>
<td>$\leftarrow \cdot \rightarrow \subseteq \rightarrow \cdot \leftarrow$</td>
</tr>
<tr>
<td>derivate</td>
<td>residual</td>
<td>$\parallel$</td>
</tr>
<tr>
<td>conversion calc.</td>
<td>λ-calculus</td>
<td></td>
</tr>
</tbody>
</table>
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
- → 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

\[ \psi/\phi \]

\[ \phi/\psi \]

\[ \phi/\psi \]

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?
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, \text{src}, \text{tgt} \rangle

- A set of objects \( a, b, c, \ldots \)
- \( \Phi \) set of steps \( \phi, \psi, \chi, \ldots \)
- \( \text{src}, \text{tgt} : \Phi \rightarrow A \)

source and target functions
Abstract rewrite system

**Definition**

ARS → is \( \langle A, \Phi, \text{src}, \text{tgt} \rangle \)

- \( A \) set of objects \( a, b, c, \ldots \)
- \( \Phi \) set of steps \( \phi, \psi, \chi, \ldots \)
- src, tgt : \( \Phi \rightarrow A \)

source and target functions

\( \phi : a \rightarrow b \) denotes step \( \phi \) with source \( a \) and target \( b \)
Abstract rewrite system

Definition

ARS → is \( \langle A, \Phi, \text{src}, \text{tgt} \rangle \)

- A set of objects \( a, b, c, \ldots \)
- \( \Phi \) set of steps \( \phi, \psi, \chi, \ldots \)
- \( \text{src, tgt} : \Phi \rightarrow A \)
  - source and target functions

\( \phi : a \rightarrow b \) denotes step \( \phi \) with source \( a \) and target \( b \)

ARS is directed graph, e.g.
Newman’s axioms for residuals

(J₁) If $\xi J \eta$, $\xi \mid \eta$ has precisely one member.
(J₂) If $\eta₁ \in \xi₁ \mid \xi$ and $\eta₂ \in \xi₂ \mid \xi$, and if $\xi₁ J \xi₂$ or $\xi₁ = \xi₂$, then $\eta₁ J \eta₂$ or $\eta₁ = \eta₂$.

$J$ represents non-nesting of redexes.
Newman’s axioms for residuals

(J₁) If $\xi J \eta$, $\xi \mid \eta$ has precisely one member.

(J₂) If $\eta_1 \in \xi_1 \mid \xi$ and $\eta_2 \in \xi_2 \mid \xi$, 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)

$\lambda$-calculus does not satisfy Newman’s axioms

$\omega(\lambda y.\omega y) \rightarrow (\lambda y.\omega y)\lambda y.\omega y \rightarrow \omega(\lambda y.\omega y) \rightarrow (\lambda y.\omega y)\lambda y.\omega y$

with $\omega = \lambda x.xx$
Newman’s axioms for residuals

(J₁) If $\xi J \eta$, $\xi \mid \eta$ has precisely one member.

(J₂) If $\eta_1 \in \xi_1 \mid \xi$ and $\eta_2 \in \xi_2 \mid \xi$, 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)

$\lambda$-calculus does not satisfy Newman’s axioms

$\omega(\lambda y.\omega y) \rightarrow (\lambda y.\omega y)\lambda y.\omega y \rightarrow \omega(\_\_\_\_\_\_) \rightarrow (\lambda y.\omega y)\lambda y.\omega y$

with $\omega = \lambda x.xx$

- by (J₂) residuals of $\omega y$ are (mutually) $J$-related.
Newman’s axioms for residuals

(J₁) If $\xi J \eta$, $\xi \mid \eta$ has precisely one member.

(J₂) If $\eta_1 \in \xi \mid \xi$ and $\eta_2 \in \xi_2 \mid \xi$, 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)

$\lambda$-calculus does not satisfy Newman’s axioms

$\omega(\lambda y.\omega y) \to (\lambda y.\omega y)\lambda y.\omega y \to \omega(\lambda y.\omega y) \to (\lambda y.\omega y)\lambda y.\omega y$

with $\omega = \lambda x.x x$

- by (J₂) residuals of $\omega y$ are (mutually) $J$-related.
- by (J₂) whole term and $\omega y$-redex are (mutually) $J$-related.
Newman’s axioms for residuals

\((J_1)\) If \(\xi J \eta\), \(\xi \parallel \eta\) has precisely one member.

\((J_2)\) If \(\eta_1 \in \xi_1 \parallel \xi\) and \(\eta_2 \in \xi_2 \parallel \xi\), 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)

\(\lambda\)-calculus does not satisfy Newman’s axioms

\[\omega(\lambda y.\omega y) \rightarrow (\lambda y.\omega y)\lambda y.\omega y \rightarrow \omega(\lambda y.\omega y) \rightarrow (\lambda y.\omega y)\lambda y.\omega y\]

with \(\omega = \lambda x.xx\)

- by \((J_2)\) residuals of \(\omega y\) are (mutually) \(J\)-related.
- by \((J_2)\) whole term and \(\omega y\)-redex are (mutually) \(J\)-related.
- the \(\omega y\)-redex is duplicated violating \((J_1)\).
From term rewrite system to ARS

Definition

multi-step ARS

$\circ \rightarrow$:

objects: terms over alphabet

steps: terms over function symbols + rule names

$\text{src}(f(\vec{s})) = f(\text{src}(\vec{s}))$ with $f$ function symbol

$\text{src}(\varrho(\vec{s})) = l(\text{src}(\vec{s}))$ with $\varrho(\vec{x})$ name of rule

$l(\vec{x}) \rightarrow r(\vec{x})$

step ARS $\rightarrow$: restriction of $\circ \rightarrow$ steps to exactly one rule name
From term rewrite system to ARS

combinatory logic (CL) rules:

\[
\begin{align*}
Ix & \rightarrow x \\
Kxy & \rightarrow x \\
Sxyz & \rightarrow xz(yz)
\end{align*}
\]
From term rewrite system to ARS

named combinatory logic (CL) rules:

\[ \iota(x) : \quad \text{Ix} \rightarrow x \]
\[ \kappa(x, y) : \quad \text{Kxy} \rightarrow x \]
\[ \varsigma(x, y, z) : \quad \text{Sxyz} \rightarrow xz(yz) \]
From term rewrite system to ARS

named combinatory logic (CL) rules:

\[ \iota(x) : Ix \rightarrow x \]
\[ \kappa(x, y) : Kxy \rightarrow x \]
\[ \varsigma(x, y, z) : Sxyz \rightarrow xz(yz) \]

Definition

multi-step ARS \( \rightarrow \rightarrow \):
From term rewrite system to ARS

named combinatory logic (CL) rules:

\[ \iota(x) : \quad Ix \rightarrow x \]
\[ \kappa(x, y) : \quad Kxy \rightarrow x \]
\[ \varsigma(x, y, z) : \quad Sxyz \rightarrow xz(yz) \]

Definition

multi-step ARS \[\rightarrow\rightarrow\]:

- objects: terms over alphabet
From term rewrite system to ARS

named combinatory logic (CL) rules:

\[
\begin{align*}
i(x) & : \quad Ix \rightarrow x \\
k(x, y) & : \quad Kxy \rightarrow x \\
s(x, y, z) & : \quad Sxyz \rightarrow xz(yz)
\end{align*}
\]

Definition

multi-step ARS \(\rightarrow\rightarrow\):

- objects: terms over alphabet
- steps: terms over function symbols + rule names
From term rewrite system to ARS

named combinatory logic (CL) rules:

\[
\begin{align*}
\iota(x) : & \quad Ix \rightarrow x \\
\kappa(x, y) : & \quad Kxy \rightarrow x \\
\varsigma(x, y, z) : & \quad Sxyz \rightarrow xz(yz)
\end{align*}
\]

Definition

multi-step ARS $\rightsquigarrow$:

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

named combinatory logic (CL) rules:

\[
\begin{align*}
i(x) &: \quad Ix \quad \rightarrow \quad x \\
k(x, y) &: \quad Kxy \quad \rightarrow \quad x \\
\varsigma(x, y, z) &: \quad Sxyz \quad \rightarrow \quad xz(yz)
\end{align*}
\]

Definition

multi-step ARS \(\rightsquigarrow\): 

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

src\((f(\underline{s}))\) = \(f(\text{src}(\underline{s}))\) with \(f\) function symbol

src\((\rho(\underline{s}))\) = \(l(\text{src}(\underline{s}))\) with \(\rho(\underline{x})\) name of rule \(l(\underline{x}) \rightarrow r(\underline{x})\)

step ARS \(\rightarrow\): restriction of \(\rightsquigarrow\) steps to exactly one rule name
From term rewrite system to ARS

**named combinatory logic (CL) rules:**

\[ \iota(x) : \ Ix \rightarrow x \]
\[ \kappa(x,y) : \ Kxy \rightarrow x \]
\[ \varsigma(x,y,z) : \ Sxyz \rightarrow xz(yz) \]

**Definition**

**multi-step ARS** \( \rightarrow \): 
- objects: terms over alphabet
- steps: terms over function symbols + rule names
- \( \text{src}(f(\vec{s})) = f(\text{src}(\vec{s})) \) with \( f \) function symbol
- \( \text{src}(\varrho(\vec{s})) = \text{src}(\vec{s}) \) with \( \varrho(\vec{x}) \) name of rule \( l(\vec{x}) \rightarrow r(\vec{x}) \)

**step ARS** \( \rightarrow \): restriction of \( \rightarrow \rightarrow \) steps to exactly one rule name

\[ \iota(IK) : \ I(IK) \rightarrow \rightarrow IK \]
\[ \text{I}(\iota(K)) : \ I(IK) \rightarrow \rightarrow IK \]
\[ \text{I}(IK) : \ I(IK) \rightarrow \rightarrow I(IK) \]
\[ \iota(\text{I}(K)) : \ I(IK) \rightarrow \rightarrow K \]
From term rewrite system to ARS

named combinatory logic (CL) rules:

\[ \iota(x) : \text{Ix } \rightarrow \text{x} \]
\[ \kappa(x, y) : \text{Kxy } \rightarrow \text{x} \]
\[ \varsigma(x, y, z) : \text{Sxyz } \rightarrow \text{xz(yz)} \]

Definition
multi-step ARS \(\rhd\rhd\): 

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

step ARS \(\rightarrow\): restriction of \(\rhd\rhd\) steps to exactly one rule name

\[ \iota(IK) : \text{I}(IK) \rightarrow IK \quad \text{I}(\iota(K)) : \text{I}(IK) \rightarrow IK \]
Steps vs. multi-steps vs. full-developments

step →: contract one redex-pattern
Steps vs. multi-steps vs. full-developments

multi-step $\leftrightarrow$ (development): contract some redex-patterns

$\rightarrow \subseteq \quad \rightarrow \subseteq \rightarrow$
Steps vs. multi-steps vs. full-developments

full-development $\leadsto$: contract all redex-patterns

$\rightarrow\subseteq \quad \leadsto \quad \subseteq$
Residuals

Intuition

residual of step $\phi$ after step $\psi$:
what remains (to be done) of step $\phi$ after doing $\psi$. 
Residuals

Intuition

Residual of step $\phi$ after step $\psi$: what remains (to be done) of step $\phi$ after doing $\psi$.

Example

Residual of $I(\iota(K))$: $I(IK) \leadsto IK$ after $\iota(IK): I(IK) \mapsto IK$?
Residuals

Intuition

*residual of step* \( \phi \) *after step* \( \psi \): *what remains (to be done) of step* \( \phi \) *after doing* \( \psi \).

Example

residual of \( I(\iota(K)) \): \( I(IK) \stackrel{\iota}{\rightarrow} IK \) after
\( \iota(IK) \): \( I(IK) \hspace{1pt} \iota \rightarrow IK \)?
\( \iota(K) \): \( IK \iota \rightarrow K \)!
Residuals

Intuition

*residual of step* $\phi$ *after step* $\psi$:
*what remains (to be done) of step* $\phi$ *after doing* $\psi$.

Example

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

Intuition

residual of step \( \phi \) after step \( \psi \):
what remains (to be done) of step \( \phi \) after doing \( \psi \).

Example

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

Intuition

residual of step $\phi$ after step $\psi$: what remains (to be done) of step $\phi$ after doing $\psi$.

Example

residual of $SIK(IK) \rightleftharpoons SIKK$ after $SIK(IK) \rightleftharpoons I(IK)(K(IK))$?
Residuals

Intuition

*residual of step* \( \phi \) *after step* \( \psi \):
what remains (to be done) of step \( \phi \) after doing \( \psi \).

Example

residual of \( SIK(IK) \) \( \rightarrow \) \( SIKK \) after
\( SIK(IK) \) \( \leftrightarrow \) \( I(IK)(K(IK)) \)?
\( I(IK)(K(IK)) \) \( \leftrightarrow \) \( IK(KK) \)!
Residuals

Intuition

*residual of step* \( \phi \) *after step* \( \psi \):
*what remains (to be done) of step* \( \phi \) *after doing* \( \psi \).

Example

residual of \( SIK(IK) \xhookrightarrow{} SIKK \) after
\( SIK(IK) \xhookrightarrow{} I(IK)(K(IK)) \)?
\( I(IK)(K(IK)) \xhookleftarrow{} IK(KK) \)?
and conversely?
Intuition

*residual* of step $\phi$ after step $\psi$: what remains (to be done) of step $\phi$ after doing $\psi$.

Example

residual of $SIK(IK) \leftrightarrow SIKK$ after

$SIK(IK) \leftrightarrow I(IK)(K(IK))$?

$I(IK)(K(IK)) \leftrightarrow IK(KK)$!

and conversely?

$SIKK \leftrightarrow IK(KK)$!
**Residuals**

**Intuition**

*residual of step \( \phi \) after step \( \psi \): what remains (to be done) of step \( \phi \) after doing \( \psi \).*

\[
\phi / \psi \quad \psi / \phi
\]

\( \phi / \psi \) and \( \psi / \phi \): 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)

Exercise show that third axiom is derivable
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

Exercise
show that third axiom is derivable
Residual system

Definition

residual system is ARS $\rightarrow$ extended with

- 1 the empty step for each object (doing nothing)
- $\phi/1 \approx \phi$
- $1/\phi \approx 1$
- $(\phi/\psi)/(\chi/\psi) \approx (\phi/\chi)/(\psi/\chi)$ (cube)

Exercise: show that third axiom is derivable.
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

\[
\begin{align*}
\phi/\phi & \approx 1 \\
\phi/1 & \approx \phi \\
1/\phi & \approx 1 \\
(\phi/\psi)/(\chi/\psi) & \approx (\phi/\chi)/(\psi/\chi) \quad \text{(cube)}
\end{align*}
\]

**Exercise**

show that third axiom is derivable
Cube axiom

\[
\frac{\phi/\psi}{\chi/\psi} \approx \frac{\phi/\chi}{\psi/\chi}
\]
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

\[ f(\phi_1, \ldots, \phi_n)/f(\psi_1, \ldots, \psi_n) = f(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \]
\[ \varrho(\phi_1, \ldots, \phi_n)/l(\psi_1, \ldots, \psi_n) = \varrho(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \]
\[ l(\phi_1, \ldots, \phi_n)/\varrho(\psi_1, \ldots, \psi_n) = r(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \]
\[ \varrho(\phi_1, \ldots, \phi_n)/\varrho(\psi_1, \ldots, \psi_n) = r(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \]

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

\[
\begin{align*}
    f(\phi_1, \ldots, \phi_n)/f(\psi_1, \ldots, \psi_n) &= f(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \\
    \varrho(\phi_1, \ldots, \phi_n)/l(\psi_1, \ldots, \psi_n) &= \varrho(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \\
    l(\phi_1, \ldots, \phi_n)/\varrho(\psi_1, \ldots, \psi_n) &= r(\phi_1/\psi_1, \ldots, \phi_n/\psi_n) \\
    \varrho(\phi_1, \ldots, \phi_n)/\varrho(\psi_1, \ldots, \psi_n) &= r(\phi_1/\psi_1, \ldots, \phi_n/\psi_n)
\end{align*}
\]
for every rule \( \varrho(x_1, \ldots, x_n) : l(x_1, \ldots, x_n) \to r(x_1, \ldots, x_n) \)

Example

- \( l(\iota(K))/\iota(\iota K) = \iota(K) \)
- \( SIK(\iota(K))/\varsigma(\iota, K, \iota K) = l(\iota(K))(K(\iota(K))) \)
Residual order

Definition
\( \phi \preceq \psi \) if \( \phi / \psi \approx 1 \) (nothing remains)
Residual order

Definition
\( \phi \preceq \psi \) if \( \phi/\psi \approx 1 \) (nothing remains)

Theorem
\( \preceq \) is a quasi-order

Proof.

► reflexivity: \( \phi/\phi \approx 1 \)
► transitivity: if \( \phi/\psi \approx 1 \) and \( \psi/\chi \approx 1 \) then \( \phi/\chi \approx 1 \)
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)
Residual order

Definition
\( \phi \preceq \psi \) if \( \phi / \psi \approx 1 \) (nothing remains)

Theorem
\( \preceq \) 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
\( \preceq \) is not necessarily a partial order (anti-symmetric)

Theorem
residual systems preserved by quotienting by \( \preceq \cap \succeq \).
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?

\[
\begin{align*}
(\phi/\psi)/\chi & \approx (\phi/\psi)/\chi \\
(\psi \circ \chi)/\phi & \approx (\psi/\phi) \circ (\chi/(\phi/\psi))
\end{align*}
\]

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

\[
\begin{align*}
\phi/\phi & \approx 1 \\
\phi/1 & \approx \phi \\
1/\phi & \approx 1 \\
(\phi/\psi)/(\chi/\psi) & \approx (\phi/\chi)/(\psi/\chi) \\
1 \circ 1 & \approx 1 \\
\chi/(\phi \circ \psi) & \approx (\chi/\phi)/\psi \\
(\phi \circ \psi)/\chi & \approx (\phi/\chi) \circ (\psi/(\chi/\phi))
\end{align*}
\]
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* \( \phi \), \( \psi \)
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* \( \phi, \psi \)

Example
Residual order gives greatest lower bound

**Theorem**

*residual systems with composition preserved by quotienting by* \( \sim \cap \succeq \).

*yields a system having a residual order which is partial order.*

\( \phi \circ \psi / \phi \) *is greatest lower bound of* \( \phi, \psi \)

**Example**

▶ orthogonal TRSs
Residual order gives greatest lower bound

**Theorem**

*residual systems with composition preserved by quotienting by* \(\preceq \cap \succsim\).

*yields a system having a residual order which is partial order.*

\(\phi \circ \psi / \phi\) is greatest lower bound of \(\phi, \psi\)

**Example**

- orthogonal TRSs
- interaction nets
Residual order gives greatest lower bound

Theorem

*residual systems with composition preserved by quotienting by* \( \preceq \cap \succeq \).

*yields a system having a residual order which is partial order.*

\( \phi \circ \psi / \phi \) *is greatest lower bound of* \( \phi, \psi \)

Example

- orthogonal TRSs
- interaction nets
- \( \lambda \)-calculus
Residual order gives greatest lower bound

**Theorem**

*residual systems with composition preserved by quotienting by* \( \preceq \cap \succeq \).

*yields a system having a residual order which is partial order.*

\( \phi \circ \psi / \phi \) *is greatest lower bound of* \( \phi \), \( \psi \)

**Example**

- orthogonal TRSs
- interaction nets
- \( \lambda \)-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 \( \phi, \psi \)

Example

- orthogonal TRSs
- interaction nets
- \( \lambda \)-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

- $ab = ab$ (same)
- $baxy = bayx$ (independent)
- $baxy = abyx$ (independent)
- $baxy = abxy$ (self-overlap)

- $cba = cab$ (self-overlap)
- $bca = bac$ (self-overlap)
- $bca = acb$ (self-overlap)
Residual system for inversion sort

1 the empty step

/ the residual map from pairs of steps to steps

\[ \frac{\phi}{\phi} \approx 1 \]
\[ \frac{\phi}{1} \approx \phi \]
\[ \frac{1}{\phi} \approx 1 \]
\[ \frac{\phi}{\psi}/\frac{\chi}{\psi} \approx \frac{\phi}{\chi}/\frac{\psi}{\chi} \]
Residual system for inversion sort

Theorem
inversion sorting gives a residual system

Proof.
step \( \phi \) from list \( \ell \) is multi-inversion: relation \( \sim \) s.t. if \( \hat{ij} \)
  - out-of-order: \( \ell = \ldots i \ldots j \ldots \) but \( i > j \);
  - transitive: if \( \hat{jk} \), then \( \hat{ik} \);
  - scopic: if \( \ell = \ldots i \ldots k \ldots j \ldots \), then either \( \hat{ik} \) or \( \hat{jk} \)
define 1 to be the empty relation,
define \( \phi/\psi \) as \((\phi \cup \psi)^+ - \psi\).

Example
\((cba \rightarrow_{cba} bca)/(cba \rightarrow_{cba} cab) = (cab \rightarrow_{cab} abc)\)
Braid problem
Braid problem
Braid confluence diagrams

reductions end in topologically equivalent ($\approx$) braids
Braid confluence diagrams

\[ i \rightarrow i = \begin{cases} \vdots \\ \vdots \end{cases} \]

\[ i \rightarrow i \]

\[ i \rightarrow j \rightarrow i \quad \text{if} \quad |i - j| \geq 2 \]

\[ j \rightarrow i \rightarrow j \]

\[ i \rightarrow i+1 \]

\[ i+1 \rightarrow i \]

\[ i+1 \rightarrow i+1 \]

reduction steps labelled by gap\# of crossing

\[ ij \approx ji \text{ if } |i - j| \geq 2 \text{ and } i(i + 1)i \approx (i + 1)i(i + 1) \]
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
Orthogonality of braids

Example

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
\end{array}
\]

\[
\begin{array}{cccccc}
1 & 3 & 5 & 2 & 6 & 4 \\
5 & 3 & 1 & 4 & 2 & \\
\end{array}
\]

\[
\begin{array}{cccccc}
3 & 1 & 5 & 2 & 6 & 4 \\
5 & 3 & 1 & 2 & 6 & 4 \\
\end{array}
\]
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

\[
(x \cdot y) \cdot z =_A x \cdot (y \cdot z) \\
= I x \cdot (y \cdot (z \cdot z)) \\
= A x \cdot ((y \cdot z) \cdot z) \\
= C x \cdot (z \cdot (y \cdot z)) \\
= A (x \cdot z) \cdot (y \cdot z)
\]

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: $xyz \rightarrow xz(yz)$ critical pair

- applicative notation: $\cdot$ infix, associating to left
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
Self-distributivity rule: $xyz \rightarrow xz(yz)$ critical pair

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

```
\[ wxyz \]
```

```
\[ wy(xy)z \]
```

```
\[ wz(yz)(xyz) \]
```

```
\[ wz(yz)(xz(yz)) \]
```
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:

\[
\begin{align*}
&\text{wxyz} \\
\rightarrow &\quad \text{wy}(xy)z \\
\rightarrow &\quad \text{wyz}(xyz) \\
\rightarrow &\quad \text{wz}(yz)(xyz) \\
\rightarrow &\quad \text{wz}(yz)(xz(yz))
\end{align*}
\]

- \( w \) represents spine \ldots
Spine rectification

Spine is stable!
Spine rectification

If you don’t have a spine, they can’t break you
Self-distributivity rule: \([y][z] \rightarrow [z][y[z]]\)

- elements on spine juxtaposed
Self-distributivity rule: 

\[ y[z] \rightarrow [z][y[z]] \]

- elements on spine juxtaposed
- rule to be applied modulo associativity
Self-distributivity rule: \([y][z] \rightarrow [z][y[z]]\)

- elements on spine juxtaposed
- rule to be applied modulo associativity
- the critical pair becomes:
Self-distributivity rule: \([y][z] \rightarrow [z][y[z]]\)

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

\[
\begin{align*}
[x][y][z] & \\
i & i + 1 \\
[y][x[y]][z] & [x][z][y[z]] \\
i + 1 & i \\
[y][z][x[y]][z] & [z][x[z]][y[z]] \\
i & i + 1 \\
[z][y[z]][x[y]][z] & [z][y[z]][x[z][y[z]]]
\end{align*}
\]

- almost braiding, but one extra step ...
Braiding vs. self-distributivity

- $[y][z] \rightarrow [z][y[z]]$ swaps $z$ and $y$, remembering $y$ crossed $z$...
Braiding vs. self-distributivity

- $[y][z] \rightarrow [z][y[z]]$ swaps $z$ and $y$, remembering $y$ crossed $z$...
- braids.
Braiding vs. self-distributivity

- \([y][z] \rightarrow [z][y[z]]\) swaps \(z\) and \(y\), remembering \(y\) crossed \(z\)...
- braids.
- self-distributivity braids inside memory...
Braiding vs. self-distributivity

- $[y][z] \rightarrow [z][y[z]]$ 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; \((\pi q)(pj)\) with \(i < j\)
- must be left-convex; \((\pi q_1 q_2)(pj)\) implies \((\pi q_1)(pj)\)

/ as before; constructed by using standard residual theory to relate positions before and after the (non-linear) term rewrite step
Substitution Lemma of the \( \lambda \)-calculus as self-distributivity

\[
(\lambda y.(\lambda x.M)N)P \\
(\lambda x.M[y:=P])N[y:=P] \\
(\lambda y.M[x:=N])P \\
M[y:=P][x:=N[y:=P]] \approx M[x:=N][y:=P]
\]
Substitution lemma of \(\lambda\)-calculus as self-distributivity

\[
(\lambda y.(\lambda x.M)N)P
\]

\[
(\lambda x.M[y:=P])N[y:=P] \quad (\lambda y.M[x:=N])P
\]

\[
M[y:=P][x:=N[y:=P]] \leftarrow M[x:=N][y:=P]
\]

Critical pair for \(\lambda\)-calculus with explicit substitutions
Substitution lemma of λ-calculus as self-distributivity

\[(\lambda y. (\lambda x. M) N) P\]

\[\rightarrow\]

\[(\lambda x. M[y:=P]) N[y:=P]\]

\[(\lambda y. M[x:=N]) P\]

\[\rightarrow\]

\[M[y:=P][x:=N[y:=P]] \leftarrow M[x:=N[y:=P]\]

Critical pair for λ-calculus with explicit substitutions
Is this rule in itself confluent? (left-to-right no)
Substitution lemma of $\lambda$-calculus as self-distributivity

$$(\lambda y. (\lambda x. M) N) P$$

$$(\lambda x. M[y:=P]) N[y:=P] \quad (\lambda y. M[x:=N]) P$$

$$M[y:=P][x:=N[y:=P]] \leftarrow M[x:=N][y:=P]$$

**Critical pair** for $\lambda$-calculus with explicit substitutions
This is self-distributivity, so even orthogonal!
Confluification

**Definition**

Confluication 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 (\(\beta\)-reduction, CL)
- associativity
From residual systems with composition to algebras

Example

- multi-inversions in sorting
- braids
- self-distributivity
- orthogonal term rewriting systems ($\beta$-reduction, CL)
- associativity
- ...
From residual systems with composition to algebras

Example

- multi-inversions in sorting
- braids
- self-distributivity
- orthogonal term rewriting systems ($\beta$-reduction, CL)
- associativity
- ...
- also many residual algebras (singleton carrier) ...
Residual algebras (with composition)

- natural numbers (as steps from object to itself)
- ÷ (cut-off subtraction), 0 (zero), + (addition);

\[
\begin{align*}
n \div n & \approx 0 \\
n \div 0 & \approx n \\
0 \div n & \approx 0 \\
(n \div m) \div (k \div m) & \approx (n \div k) \div (m \div k) \\
0 + 0 & \approx 0 \\
k \div (n + m) & \approx (k \div n) \div m \\
(n + m) \div k & \approx (n \div k) + (m \div (k \div n))
\end{align*}
\]

Generated from its
Residual algebras (with composition)

- natural numbers (as steps from object to itself)
- ÷ (cut-off subtraction), 0 (zero), + (addition);

\[
\begin{align*}
    n \div n & \approx 0 \\
    n \div 0 & \approx n \\
    0 \div n & \approx 0 \\
    (n \div m) \div (k \div m) & \approx (n \div k) \div (m \div k) \\
    0 + 0 & \approx 0 \\
    k \div (n + m) & \approx (k \div n) \div m \\
    (n + m) \div k & \approx (n \div k) + (m \div (k \div n))
\end{align*}
\]

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);

\[
\begin{align*}
  M - M & \approx \emptyset \\
  M - \emptyset & \approx M \\
  \emptyset - M & \approx \emptyset \\
  (M - N) - (K - N) & \approx (M - K) - (N - K) \\
  \emptyset \uplus \emptyset & \approx \emptyset \\
  K - (M \uplus N) & \approx (K - M) - N \\
  (M \uplus N) - K & \approx (M - K) \uplus (N - (K - M))
\end{align*}
\]
Residual algebras (with composition)

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

\[
\begin{align*}
M - M & \approx \emptyset \\
M - \emptyset & \approx M \\
\emptyset - M & \approx \emptyset \\
(M - N) - (K - N) & \approx (M - K) - (N - K) \\
\emptyset \uplus \emptyset & \approx \emptyset \\
K - (M \uplus N) & \approx (K - M) - N \\
(M \uplus N) - K & \approx (M - K) \uplus (N - (K - M))
\end{align*}
\]

Sets with set-difference, \( \emptyset \), disjoint union.
Residual algebras (with composition)

- multisets over some set (as steps from object to itself)
- \( M - M \approx \emptyset \)
- \( M - \emptyset \approx M \)
- \( \emptyset - M \approx \emptyset \)
- \((M - N) - (K - N) \approx (M - K) - (N - K)\)
- \( \emptyset \uplus \emptyset \approx \emptyset \)
- \( K - (M \uplus N) \approx (K - M) - N \)
- \((M \uplus N) - K \approx (M - K) \uplus (N - (K - M))\)

all compositions are commutative
Definition

commutative residual algebra with composition (CRAC) satisfies

\[ \frac{\phi}{\psi} \approx 1 \]
\[ \frac{\phi}{\frac{\phi}{\psi}} \approx \frac{\psi}{\frac{\psi}{\phi}} \]

(follows from computing \( \frac{\phi \circ \psi}{\psi \circ \phi} \approx 1 \! \))
Definition

A commutative residual algebra with composition (CRAC) satisfies

\[(\phi/\psi)/\phi \approx 1\]
\[\phi/(\phi/\psi) \approx \psi/(\psi/\phi)\]

The 2nd equation states commutativity of intersection \(\phi/(\phi/\psi)\).
Definition

**commutative residual algebra with composition (CRAC)** satisfies

\[
\begin{align*}
(\phi/\psi)/\phi & \approx 1 \\
\phi/(\phi/\psi) & \approx \psi/(\psi/\phi)
\end{align*}
\]

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

Definition

A commutative residual algebra with composition (CRAC) satisfies

\[
\frac{\phi}{\psi}/\phi \approx 1 \\
\phi/(\phi/\psi) \approx \psi/(\psi/\phi)
\]

- 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
Definition

commutative residual algebra with composition (CRAC) satisfies

\[
\frac{\phi}{\psi} / \phi \approx 1
\]

\[
\phi / (\phi / \psi) \approx \psi / (\psi / \phi)
\]

- 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 \( a/b \approx 1 \).
Definition

commutative residual algebra with composition (CRAC) satisfies

\[
\frac{(\phi/\psi)}{\phi} \approx 1 \\
\frac{\phi}{(\phi/\psi)} \approx \frac{\psi}{(\psi/\phi)}
\]

- 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 \( \leq \) well-founded; \( a \leq b \) if \( a/b \approx 1 \).
- Other interesting CRACs?
Definition

A commutative residual algebra with composition (CRAC) satisfies

\[
\frac{\phi/\psi}{\phi} \approx 1 \\
\frac{\phi/(\phi/\psi)}{\psi/(\psi/\phi)}
\]

- 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