Rewriting

Part 1. Abstract Reduction

Temur Kutsia

RISC, JKU Linz

Literature

Franz Baader and Tobias Nipkow. Term Rewriting and All That. Cambridge University Press, 1998.

Book's page: http://www21.in.tum.de/~nipkow/TRaAT/

Resources about rewriting:
http://rewriting.loria.fr/
http://www.jaist.ac.jp/~hirokawa/tool/
http://cl-informatik.uibk.ac.at/users/ami/research/rr/

Motivation

Abstract Reduction Systems

Equational reasoning

Restricted class of languages.

The only predicate symbol is equality $\approx$.

Reasoning with equations:

- derive consequences of given equations,
- find values for variables that satisfy a given equation.

At the heart of many problems in mathematics and computer science.

Example: addition of natural numbers

Equations (identities):

$$x + 0 \approx x$$
$$x + s(y) \approx s(x + y)$$

How to calculate $s(0) + s(s(0))$?

Example: addition of natural numbers

Orient equations, obtaining rewriting rules.

Apply the rules to transform expressions.

Example: addition of natural numbers

Orient equations, obtaining rewriting rules.

Apply the rules to transform expressions. Rewrite rules:

$$x + 0 \to x \qquad (R_1)$$

$$x + s(y) \to s(x + y) \qquad (R_2)$$

Rewriting $s(0) + s(s(0))$:

Example: addition of natural numbers

Orient equations, obtaining rewriting rules.

Apply the rules to transform expressions. Rewrite rules:

$$x + 0 \to x \qquad (R_1)$$

$$x + s(y) \to s(x + y) \qquad (R_2)$$

Rewriting $s(0) + s(s(0))$:

$$s(0) + s(s(0)) \to \qquad \text{(by } R_2\text{, with } x \mapsto s(0), y \mapsto s(0)\text{)}$$

Example: addition of natural numbers

Orient equations, obtaining rewriting rules.

Apply the rules to transform expressions. Rewrite rules:

$$x + 0 \to x \qquad (R_1)$$
$$x + s(y) \to s(x + y) \qquad (R_2)$$

Rewriting $s(0) + s(s(0))$:

$$s(0) + s(s(0)) \to \qquad \text{(by } R_2\text{, with } x \mapsto s(0), y \mapsto s(0))$$
$$s(s(0) + s(0)) \to \qquad \text{(by } R_2\text{, with } x \mapsto s(0), y \mapsto 0)$$

Example: addition of natural numbers

Orient equations, obtaining rewriting rules.

Apply the rules to transform expressions. Rewrite rules:

$$x + 0 \to x \qquad (R_1)$$
$$x + s(y) \to s(x + y) \qquad (R_2)$$

Rewriting $s(0) + s(s(0))$:

$$s(0) + s(s(0)) \to \qquad \text{(by } R_2\text{, with } x \mapsto s(0), y \mapsto s(0))$$
$$s(s(0) + s(0)) \to \qquad \text{(by } R_2\text{, with } x \mapsto s(0), y \mapsto 0)$$
$$s(s(s(0) + 0)) \to \qquad \text{(by } R_1\text{, with } x \mapsto s(0))$$

Example: addition of natural numbers

Orient equations, obtaining rewriting rules.

Apply the rules to transform expressions. Rewrite rules:

$$x + 0 \to x \qquad (R_1)$$
$$x + s(y) \to s(x + y) \qquad (R_2)$$

Rewriting $s(0) + s(s(0))$:

$$s(0) + s(s(0)) \to \qquad \text{(by } R_2\text{, with } x \mapsto s(0), y \mapsto s(0))$$
$$s(s(0) + s(0)) \to \qquad \text{(by } R_2\text{, with } x \mapsto s(0), y \mapsto 0)$$
$$s(s(s(0) + 0)) \to \qquad \text{(by } R_1\text{, with } x \mapsto s(0))$$
$$s(s(s(0)))$$

What is rewriting

Process of transforming one expression into another.

Rules describe how one expression can be rewritten into another.

Identities and rewriting

Rewriting as a computational mechanism:

- Apply given equations in one direction, as rewrite rules.
- Compute normal forms.
- Close relationship with functional programming.
- Example: symbolic differentiation.

Identities and rewriting

Rewriting as a computational mechanism:

- Apply given equations in one direction, as rewrite rules.
- Compute normal forms.
- Close relationship with functional programming.
- Example: symbolic differentiation.

Rewriting as a deduction mechanism:

- Apply given equations in both directions.
- Define equivalence classes of terms.
- Equational reasoning.
- Example: group theory.

Symbolic differentiation

Expressions: Terms built over variables $(u, v, \ldots)$ and the following function symbols:

- constants $0, 1$ (numbers),
- constants X, Y (indeterminates),
- unary symbol D_X (partial derivative with respect to X),
- binary symbols $+, *$.

Symbolic differentiation

Expressions: Terms built over variables $(u, v, \ldots)$ and the following function symbols:

- constants $0, 1$ (numbers),
- constants X, Y (indeterminates),
- unary symbol D_X (partial derivative with respect to X),
- binary symbols $+, *$.

Examples of terms:

- $(X + X) * Y + 1$.
- $D_X(u * v)$.
- $(X + Y) * D_X(X * Y)$.

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$

$$D_X(Y) \to 0 \qquad (R_2)$$

$$D_X(u+v) \to D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u*v) \to (u*D_X(v)) + (D_X(u)*v) \qquad (R_4)$$

Symbolic differentiation

Rewrite rules:

$$
\begin{aligned}
D_X(X) &\to 1 && (R_1)\\
D_X(Y) &\to 0 && (R_2)\\
D_X(u+v) &\to D_X(u) + D_X(v) && (R_3)\\
D_X(u*v) &\to (u * D_X(v)) + (D_X(u) * v) && (R_4)
\end{aligned}
$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$
$$D_X(Y) \to 0 \qquad (R_2)$$
$$D_X(u+v) \to D_X(u) + D_X(v) \qquad (R_3)$$
$$D_X(u*v) \to (u*D_X(v)) + (D_X(u)*v) \qquad (R_4)$$

Differentiate $D_X(X*X)$:

$$D_X(X*X)$$
$$\downarrow (R_4)$$
$$(X*D_X(X)) + (D_X(X)*X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \quad (R_1)$$
$$D_X(Y) \to 0 \quad (R_2)$$
$$D_X(u + v) \to D_X(u) + D_X(v) \quad (R_3)$$
$$D_X(u * v) \to (u * D_X(v)) + (D_X(u) * v) \quad (R_4)$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$
$$\downarrow (R_4)$$
$$(X * D_X(X)) + (D_X(X) * X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$

$$D_X(Y) \to 0 \qquad (R_2)$$

$$D_X(u + v) \to D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u * v) \to (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$

$$\downarrow (R_4)$$

$$(X * D_X(X)) + (D_X(X) * X)$$

$$\swarrow (R_1)$$

$$(X * 1) + (D_X(X) * X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$

$$D_X(Y) \to 0 \qquad (R_2)$$

$$D_X(u + v) \to D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u * v) \to (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$

$$\downarrow (R_4)$$

$$(X * D_X(X)) + (D_X(X) * X)$$

$$\swarrow (R_1)$$

$$(X * 1) + (D_X(X) * X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$
$$D_X(Y) \to 0 \qquad (R_2)$$
$$D_X(u+v) \to D_X(u) + D_X(v) \qquad (R_3)$$
$$D_X(u*v) \to (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$
$$\downarrow (R_4)$$
$$(X * D_X(X)) + (D_X(X) * X)$$
$$\swarrow (R_1)$$
$$(X * 1) + (D_X(X) * X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$

$$D_X(Y) \to 0 \qquad (R_2)$$

$$D_X(u+v) \to D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u*v) \to (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X*X)$:

$$D_X(X*X)$$

$$\downarrow (R_4)$$

$$(X * D_X(X)) + (D_X(X) * X)$$

$$\swarrow (R_1)$$

$$(X*1) + (D_X(X) * X)$$

$$\searrow (R_1)$$

$$(X*1) + (1 * X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$

$$D_X(Y) \to 0 \qquad (R_2)$$

$$D_X(u+v) \to D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u * v) \to (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$

$$\downarrow (R_4)$$

$$(X * D_X(X)) + (D_X(X) * X)$$

$$\swarrow (R_1)$$

$$(X * 1) + (D_X(X) * X)$$

$$\searrow (R_1)$$

$$(X * 1) + (1 * X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$

$$D_X(Y) \to 0 \qquad (R_2)$$

$$D_X(u+v) \to D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u*v) \to (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$

$$\downarrow (R_4)$$

$$(X * D_X(X)) + (D_X(X) * X)$$

$$\swarrow (R_1)$$

$$(X * 1) + (D_X(X) * X)$$

$$\searrow (R_1)$$

$$(X * 1) + (1 * X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$

$$D_X(Y) \to 0 \qquad (R_2)$$

$$D_X(u + v) \to D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u * v) \to (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$

$$\downarrow (R_4)$$

$$(X * D_X(X)) + (D_X(X) * X)$$

$$\swarrow (R_1) \qquad \searrow (R_1)$$

$$(X * 1) + (D_X(X) * X) \qquad (X * D_X(X)) + (1 * X)$$

$$\searrow (R_1)$$

$$(X * 1) + (1 * X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \rightarrow 1 \qquad (R_1)$$

$$D_X(Y) \rightarrow 0 \qquad (R_2)$$

$$D_X(u + v) \rightarrow D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u * v) \rightarrow (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$

$$\downarrow (R_4)$$

$$(X * D_X(X)) + (D_X(X) * X)$$

$$\swarrow (R_1) \qquad\qquad \searrow (R_1)$$

$$(X * 1) + (D_X(X) * X) \qquad\qquad (X * D_X(X)) + (1 * X)$$

$$\searrow (R_1)$$

$$(X * 1) + (1 * X)$$

Symbolic differentiation

Rewrite rules:

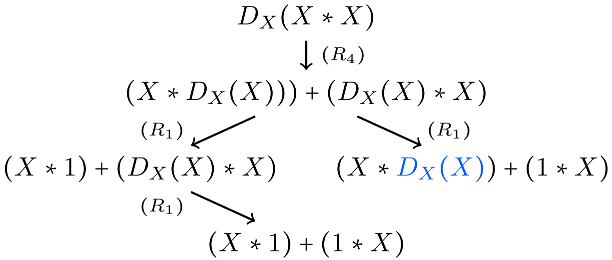
$$D_X(X) \to 1 \quad (R_1)$$

$$D_X(Y) \to 0 \quad (R_2)$$

$$D_X(u + v) \to D_X(u) + D_X(v) \quad (R_3)$$

$$D_X(u * v) \to (u * D_X(v)) + (D_X(u) * v) \quad (R_4)$$

Differentiate $D_X(X * X)$:



Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$

$$D_X(Y) \to 0 \qquad (R_2)$$

$$D_X(u+v) \to D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u*v) \to (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X*X)$:

$$D_X(X*X)$$

$$\downarrow (R_4)$$

$$(X * D_X(X)) + (D_X(X) * X)$$

$$\swarrow (R_1) \qquad (R_1) \searrow$$

$$(X*1) + (D_X(X) * X) \qquad (X * D_X(X)) + (1 * X)$$

$$(R_1) \searrow \qquad \swarrow (R_1)$$

$$(X*1) + (1*X)$$

Symbolic differentiation

Rewrite rules:

$$D_X(X) \to 1 \qquad (R_1)$$

$$D_X(Y) \to 0 \qquad (R_2)$$

$$D_X(u+v) \to D_X(u) + D_X(v) \qquad (R_3)$$

$$D_X(u*v) \to (u * D_X(v)) + (D_X(u) * v) \qquad (R_4)$$

Differentiate $D_X(X * X)$:

$$D_X(X * X)$$

$$\downarrow (R_4)$$

$$(X * D_X(X)) + (D_X(X) * X)$$

$$\swarrow (R_1) \qquad \searrow (R_1)$$

$$(X * 1) + (D_X(X) * X) \qquad (X * D_X(X)) + (1 * X)$$

$$(R_1) \searrow \qquad \swarrow (R_1)$$

$$(X * 1) + (1 * X)$$

Properties of term rewriting systems

The symbolic differentiation example can be used to illustrate two most important properties of TRSs:

Properties of term rewriting systems

The symbolic differentiation example can be used to illustrate two most important properties of TRSs:

1. Termination:
   - Is it always the case that after finitely many rule applications we reach an expression to which no more rules apply (normal form)?
   - For symbolic differentiation rules this is the case.
   - But how to prove it?
   - An example of non-terminating rule: $u + v \to v + u$

Properties of term rewriting systems

The symbolic differentiation example can be used to illustrate two most important properties of TRSs:

2. Confluence:
   - If there are different ways of applying rules to a given term t, leading to different terms t_1 and t_2, can they be reduced by rule applications to a common term?
   - For symbolic differentiation rules this is the case.
   - But how to prove it?

Properties of term rewriting systems

Adding the rule $u + 0 \to u$ (R_5) destroys confluence:

$$D_X(X+0) \xrightarrow{(R_5)} D_X(X) \xrightarrow{(R_1)} 1$$

$$D_X(X+0) \xrightarrow{(R_3)} D_X(X) + D_X(0) \xrightarrow{(R_1)} 1 + D_X(0)$$

Confluence can be regained by adding $D_X(0) \to 0$ (completion).

Group theory

Terms are built over variables and the following function symbols:

- binary $\circ$,
- unary i,
- constant e.

Examples of terms:

- $x \circ (y \circ i(y))$
- $(e \circ x) \circ i(e)$
- $i(x \circ y)$

Identities (aka group axioms), defining groups:

Associativity of $\circ$	$(x \circ y) \circ z \approx x \circ (y \circ z)$	(G_1)
e left unit	$e \circ x \approx x$	(G_2)
i left inverse	$i(x) \circ x \approx e$	(G_3)

Group theory

Identities can be applied in both directions.

Word problem for identities:

- Given a set of identities E and two terms s and t.
- Is it possible to transform s into t, using the identities in E as rewrite rules applied in both directions?

For instance, is it possible to transform e into $x \circ i(x)$,
i.e., is the left inverse also a right-inverse?

Group theory

$$(x \circ y) \circ z \approx x \circ (y \circ z) \qquad (G_1)$$

$$e \circ x \approx x \qquad (G_2)$$

$$i(x) \circ x \approx e \qquad (G_3)$$

Transform e into $x \circ i(x)$:

Group theory

$$
\begin{aligned}
(x \circ y) \circ z &\approx x \circ (y \circ z) && (G_1)\\
e \circ x &\approx x && (G_2)\\
i(x) \circ x &\approx e && (G_3)
\end{aligned}
$$

Transform e into $x \circ i(x)$:

$$
\begin{aligned}
e &\approx_{G_3} i(x \circ i(x)) \circ (x \circ i(x))\\
&\approx_{G_2} i(x \circ i(x)) \circ (x \circ (e \circ i(x)))\\
&\approx_{G_3} i(x \circ i(x)) \circ (x \circ ((i(x) \circ x) \circ i(x)))\\
&\approx_{G_1} i(x \circ i(x)) \circ ((x \circ (i(x) \circ x)) \circ i(x))\\
&\approx_{G_1} i(x \circ i(x)) \circ (((x \circ i(x)) \circ x) \circ i(x))\\
&\approx_{G_1} i(x \circ i(x)) \circ ((x \circ i(x)) \circ (x \circ i(x)))\\
&\approx_{G_1} (i(x \circ i(x)) \circ (x \circ i(x))) \circ (x \circ i(x))\\
&\approx_{G_3} e \circ (x \circ i(x))\\
&\approx_{G_2} x \circ i(x)
\end{aligned}
$$

Solving word problems by rewriting?

Is there a simpler way to solve word problems?

Try to solve it by rewriting (uni-directional application of identities):

$$s \xrightarrow{*} \hat{s} = \hat{t} \xleftarrow{*} t$$

Reduce s and t to normal forms $\hat{s}$ and $\hat{t}$.

Check whether $\hat{s} = \hat{t}$, i.e., syntactically equal.
(= is the meta-equality.)

Solving word problems by rewriting?

Is there a simpler way to solve word problems?

Try to solve it by rewriting (uni-directional application of identities):

$$s \xrightarrow{*} \hat{s} = \hat{t} \xleftarrow{*} t$$

Reduce s and t to normal forms $\hat{s}$ and $\hat{t}$.

Check whether $\hat{s} = \hat{t}$, i.e., syntactically equal.
(= is the meta-equality.)

But... it would only work if normal forms exist and are unique.

Solving word problems by rewriting?

In the group theory example, e and $x \circ i(x)$ are equivalent, but it can not be decided by (left-to-right) rewriting: Both terms are in the normal form.

Uniqueness of normal forms is violated: non-confluence.

Normal forms may not exist: The process of reducing a term may lead to an infinite chain of transformations: non-termination.

Solving word problems by rewriting?

In the group theory example, e and $x \circ i(x)$ are equivalent, but it can not be decided by (left-to-right) rewriting: Both terms are in the normal form.

Uniqueness of normal forms is violated: non-confluence.

Normal forms may not exist: The process of reducing a term may lead to an infinite chain of transformations: non-termination.

Termination and confluence ensure existence and uniqueness of normal forms.

Solving word problems by rewriting?

In the group theory example, e and $x \circ i(x)$ are equivalent, but it can not be decided by (left-to-right) rewriting: Both terms are in the normal form.

Uniqueness of normal forms is violated: non-confluence.

Normal forms may not exist: The process of reducing a term may lead to an infinite chain of transformations: non-termination.

Termination and confluence ensure existence and uniqueness of normal forms.

If a given set of identities leads to non-confluent system, we will try to apply the idea of completion to extend the rewrite system to a confluent one.

Motivation

Abstract Reduction Systems

Abstract vs concrete

Concrete rewrite formalisms:

- string rewriting
- term rewriting
- graph rewriting
- λ calculus
- etc.

Abstract reduction:

- No structure on objects to be rewritten.
- Abstract treatment of reductions.

Abstract reduction systems

Abstract reduction system (ARS): A pair $(A, \to)$, where

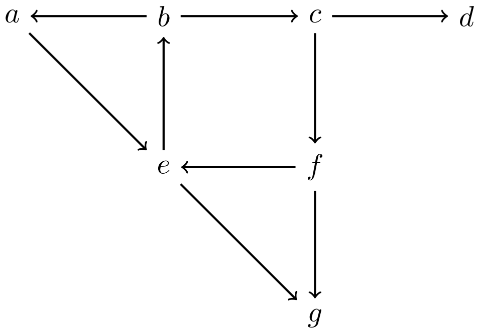
- A is a set,
- the reduction $\to$ is a binary relation on A: $\to \subseteq A \times A$.

Write $a \to b$ for $(a, b) \in \to$.

Abstract reduction system: example

$$A = \{a, b, c, d, e, f, g\}$$

$$\rightarrow = \left\{ \begin{array}{l} (a,e),(b,a),(b,c),(c,d),(c,f) \\ (e,b),(e,g),(f,e),(f,g) \end{array} \right\}$$



Equivalence and reduction

Again, two views at reductions.

1. Directed computation: Follow the reductions, trying to compute a normal form: $a_0 \to a_1 \to \cdots$
2. View $\to$ as description of $\overset{*}{\leftrightarrow}$.
   - $a \overset{*}{\leftrightarrow} b$ means there is a path between a and b, with arrows traversed in both directions: $a \leftarrow c \to d \leftarrow b$
   - Goal: Decide whether $a \overset{*}{\leftrightarrow} b$.
   - Bidirectional rewriting is expensive.
   - Unidirectional rewriting with subsequent comparison of normal form works if the reduction system is confluent and terminating.

Termination, confluence: central topics.

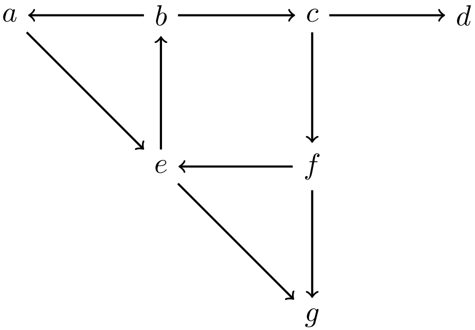
Basic notions

Composition of two relations.

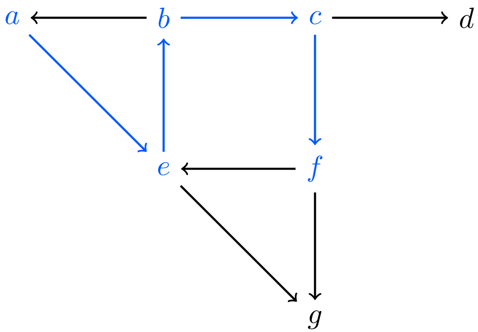
Given two relations $R \subseteq A \times B$ and $S \subseteq B \times C$, their composition is defined as

$$R \circ S := \{(x, z) \mid \exists y \in B.\ (x, y) \in R \wedge (y, z) \in S\}$$

Abstract reduction system: example

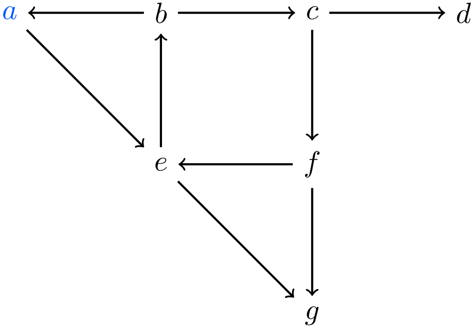


Abstract reduction system: example



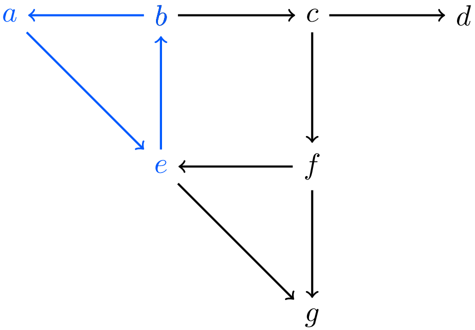
- Finite rewrite sequence: $a \to e \to b \to c \to f$

Abstract reduction system: example



- Finite rewrite sequence: $a \to e \to b \to c \to f$
- Empty rewrite sequence: a

Abstract reduction system: example



- Finite rewrite sequence: $a \to e \to b \to c \to f$
- Empty rewrite sequence: a
- Infinite rewrite sequence: $a \to e \to b \to a \to \cdots$

Relations derived from $\rightarrow$

$\xrightarrow{0} := \{(x,x) \mid x \in A\}$	identity
$\xrightarrow{=} := \rightarrow \cup \xrightarrow{0}$	reflexive closure
$\xrightarrow{i+1} := \xrightarrow{i} \circ \rightarrow$	$(i+1)$-fold composition, $i \geq 0$
$\xrightarrow{+} := \cup_{i>0} \xrightarrow{i}$	transitive closure
$\xrightarrow{*} := \xrightarrow{+} \cup \xrightarrow{0}$	reflexive transitive closure
$\xrightarrow{-1} := \{(y,x) \mid (x,y) \in \rightarrow\}$	inverse
$\leftarrow := \xrightarrow{-1}$	inverse
$\leftrightarrow := \rightarrow \cup \leftarrow$	symmetric closure
$\overset{+}{\leftrightarrow} := (\leftrightarrow)^{+}$	transitive symmetric closure
$\overset{*}{\leftrightarrow} := (\leftrightarrow)^{*}$	reflexive transitive symmetric closure

Terminology

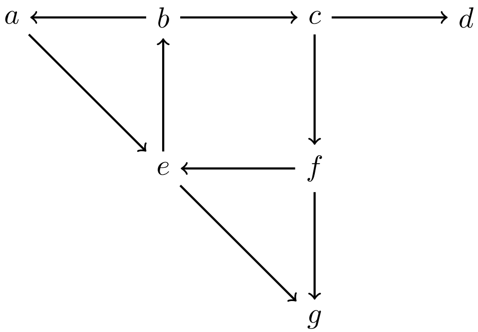
If $x \xrightarrow{*} y$ then we say:

- x rewrites to y, or
- there is some finite path from x to y, or
- y is a reduct of x.

Terminology

If $x \xrightarrow{*} y$ then we say:

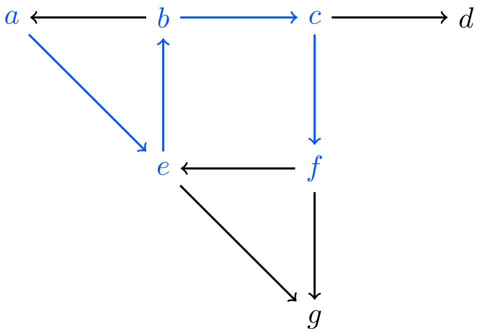
- x rewrites to y, or
- there is some finite path from x to y, or
- y is a reduct of x.



Terminology

If $x \xrightarrow{*} y$ then we say:

- x rewrites to y, or
- there is some finite path from x to y, or
- y is a reduct of x.



$a \xrightarrow{*} f$

Terminology

x is reducible iff there exists y such that $x \to y$.

Terminology

x is reducible iff there exists y such that $x \to y$.

x is in normal form (irreducible) iff x is not reducible.

Terminology

x is reducible iff there exists y such that $x \to y$.

x is in normal form (irreducible) iff x is not reducible.

y is a normal form of x iff $x \xrightarrow{*} y$ and y is in normal form.

Terminology

x is reducible iff there exists y such that $x \to y$.

x is in normal form (irreducible) iff x is not reducible.

y is a normal form of x iff $x \xrightarrow{*} y$ and y is in normal form.

We write $x \xrightarrow{!} y$ if y is a normal form of x.

If x has a unique normal form, it is denoted by $x\downarrow$.

Terminology

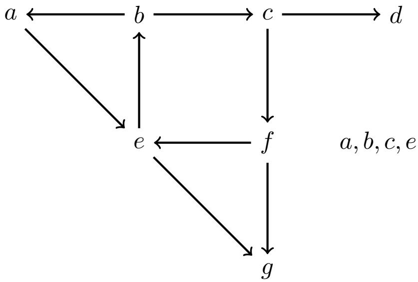
x is reducible iff there exists y such that $x \to y$.

x is in normal form (irreducible) iff x is not reducible.

y is a normal form of x iff $x \xrightarrow{*} y$ and y is in normal form.

We write $x \xrightarrow{!} y$ if y is a normal form of x.

If x has a unique normal form, it is denoted by $x\downarrow$.



a, b, c, e, f are reducible

Terminology

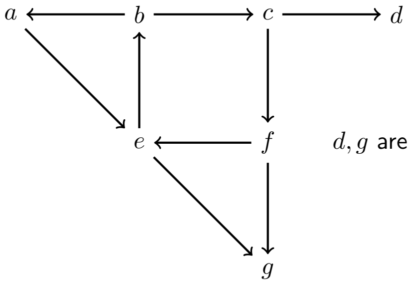
x is reducible iff there exists y such that $x \to y$.

x is in normal form (irreducible) iff x is not reducible.

y is a normal form of x iff $x \xrightarrow{*} y$ and y is in normal form.

We write $x \xrightarrow{!} y$ if y is a normal form of x.

If x has a unique normal form, it is denoted by $x\downarrow$.



d, g are in a normal form

Terminology

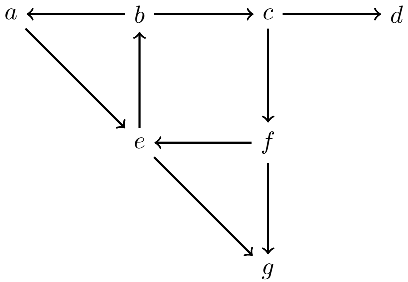
x is reducible iff there exists y such that $x \to y$.

x is in normal form (irreducible) iff x is not reducible.

y is a normal form of x iff $x \xrightarrow{*} y$ and y is in normal form.

We write $x \xrightarrow{!} y$ if y is a normal form of x.

If x has a unique normal form, it is denoted by $x\downarrow$.



$b \xrightarrow{!} d$
$b \xrightarrow{!} g$
$g \xrightarrow{!} g$

Terminology

y is direct successor of x iff $x \to y$.

Terminology

y is direct successor of x iff $x \to y$.

y is successor of x iff $x \xrightarrow{+} y$.

Terminology

y is direct successor of x iff $x \rightarrow y$.

y is successor of x iff $x \xrightarrow{+} y$.

x and y are convertible iff $x \overset{*}{\leftrightarrow} y$.

Terminology

y is direct successor of x iff $x \to y$.

y is successor of x iff $x \xrightarrow{+} y$.

x and y are convertible iff $x \overset{*}{\leftrightarrow} y$.

x and y are joinable iff there exists z such that $x \xrightarrow{*} z \xleftarrow{*} y$.

Terminology

y is direct successor of x iff $x \to y$.

y is successor of x iff $x \xrightarrow{+} y$.

x and y are convertible iff $x \xleftrightarrow{*} y$.

x and y are joinable iff there exists z such that $x \xrightarrow{*} z \xleftarrow{*} y$.

We write $x \downarrow y$ iff x and y are joinable.

Terminology

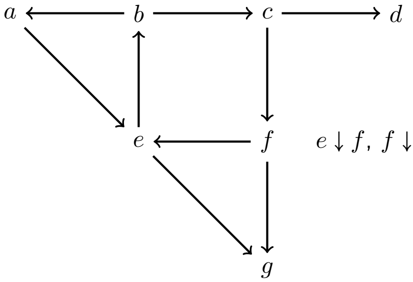
y is direct successor of x iff $x \to y$.

y is successor of x iff $x \xrightarrow{+} y$.

x and y are convertible iff $x \xleftrightarrow{*} y$.

x and y are joinable iff there exists z such that $x \xrightarrow{*} z \xleftarrow{*} y$.

We write $x \downarrow y$ iff x and y are joinable.



$e \downarrow f$, $f \downarrow d$, $a \downarrow f$, not $g \downarrow d$

Terminology

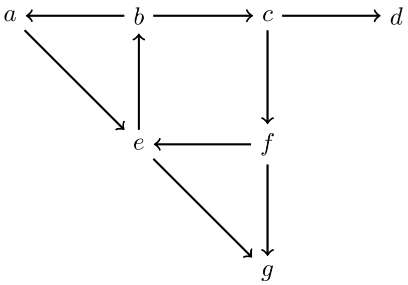
y is direct successor of x iff $x \to y$.

y is successor of x iff $x \xrightarrow{+} y$.

x and y are convertible iff $x \xleftrightarrow{*} y$.

x and y are joinable iff there exists z such that $x \xrightarrow{*} z \xleftarrow{*} y$.

We write $x \downarrow y$ iff x and y are joinable.



$g \xleftrightarrow{*} d$

Example

1. Let $A := \mathbb{N} - \{0, 1\}$ and $\rightarrow := \{(m, n) \mid m > n \text{ and } n \text{ divides } m\}$. Then
   (a) m is in normal form iff m is prime.
   (b) p is a normal form of m iff p is a prime factor of m.
   (c) $m \downarrow n$ iff m and n are not relatively prime.
   (d) $\xrightarrow{+} = \rightarrow$ because $>$ and "divides" are already transitive.
   (e) $\xleftrightarrow{*} = A \times A$.
2. Let $A := \{a, b\}^*$ (the set of words over the alphabet $\{a, b\}$) and $\rightarrow :=$ $\{(ubav, uabv) \mid u, v \in A\}$. Then
   (a) w is in normal form iff w is sorted, i.e. of the form a^*b^*.
   (b) Every w has a unique normal form $w\downarrow$, the result of sorting w.
   (c) $w_1 \downarrow w_2$ iff $w_1 \xleftrightarrow{*} w_2$ iff w_1 and w_2 contain the same number of as and bs.

Central notions: Church-Rosser

Definition 1.1

A relation $\rightarrow$ is called Church-Rosser (CR) iff

$$x \overset{*}{\leftrightarrow} y \text{ implies } x \downarrow y.$$

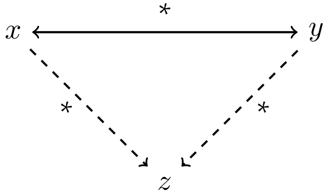
Central notions: Church-Rosser

Definition 1.1

A relation → is called Church-Rosser (CR) iff

$$x \overset{*}{\leftrightarrow} y \text{ implies } x \downarrow y.$$

Graphically:



Solid arrows represent universal and dashed arrows existential quantification: $\forall x, y.\ x \overset{*}{\leftrightarrow} y \Rightarrow \exists z.\ x \overset{*}{\rightarrow} z \wedge y \overset{*}{\rightarrow} z.$

Central notions: confluence

Definition 1.2

A relation $\to$ is called confluent (C) iff

$$y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2 \text{ implies } y_1 \downarrow y_2.$$

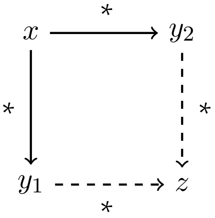
Central notions: confluence

Definition 1.2

A relation $\rightarrow$ is called confluent (C) iff

$$y_1 \xleftarrow{*} x \xrightarrow{*} y_2 \text{ implies } y_1 \downarrow y_2.$$

Graphically:



Solid arrows represent universal and dashed arrows existential quantification: $\forall x, y_1, y_2.\ y_1 \xleftarrow{*} x \xrightarrow{*} y_2 \Rightarrow \exists z.\ y_1 \xrightarrow{*} z \xleftarrow{*} y_2$.

Central notions: local confluence

Definition 1.3

A relation $\rightarrow$ is called locally confluent (LC) iff

$$y_1 \leftarrow x \rightarrow y_2 \text{ implies } y_1 \downarrow y_2.$$

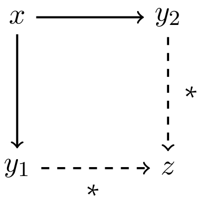
Central notions: local confluence

Definition 1.3

A relation $\to$ is called locally confluent (LC) iff

$$y_1 \leftarrow x \to y_2 \text{ implies } y_1 \downarrow y_2.$$

Graphically:



Solid arrows represent universal and dashed arrows existential quantification: $\forall x, y_1, y_2.\ y_1 \leftarrow x \to y_2 \Rightarrow \exists z.\ y_1 \xrightarrow{*} z \xleftarrow{*} y_2$.

Central notions: T, N, UN, convergence

Definition 1.4

A relation $\to$ is called

- terminating (T) iff there is no infinite descending chain $a_0 \to a_1 \to \cdots$.
- normalizing (N) iff every element has a normal form.
- uniquely normalizing (UN) iff every element has at most one normal form.
- convergent iff it is both confluent and terminating.

Central notions: T, N, UN, convergence

Definition 1.4

A relation $\rightarrow$ is called

- terminating (T) iff there is no infinite descending chain $a_0 \rightarrow a_1 \rightarrow \cdots$.
- normalizing (N) iff every element has a normal form.
- uniquely normalizing (UN) iff every element has at most one normal form.
- convergent iff it is both confluent and terminating.

Alternative terminology:

- Strongly normalizing: terminating.
- Weakly normalizing: normalizing.

Central notions: CR reformulated

Obviously, $x \downarrow y$ implies $x \overset{*}{\leftrightarrow} y$.

Therefore, the Church-Rosser property can be formulated as the equivalence:

$\to$ is called Church-Rosser iff

$$x \overset{*}{\leftrightarrow} y \text{ iff } x \downarrow y.$$

Properties

1. $\mathsf{T} \implies \mathsf{N}$

Properties

1. $T \implies N$
2. $T \not\Longleftarrow N$

Properties

1. T $\Longrightarrow$ N
2. T $\not\Longleftarrow$ N $\qquad \circlearrowleft\ a \longrightarrow b$

Properties

1. T $\Longrightarrow$ N
2. T $\not\Longleftarrow$ N $\circlearrowright a \longrightarrow b$
3. CR $\Longleftrightarrow$ $\overset{*}{\leftrightarrow} = \downarrow$

Properties

1. T $\Longrightarrow$ N
2. T $\not\Longleftarrow$ N $\qquad \circlearrowleft a \longrightarrow b$
3. CR $\Longleftrightarrow$ $\overset{*}{\leftrightarrow} = \downarrow$
4. CR $\Longrightarrow$ UN

Properties

1. T $\Longrightarrow$ N
2. T $\not\Longleftarrow$ N $\circlearrowright a \longrightarrow b$
3. CR $\Longleftrightarrow$ $\overset{*}{\leftrightarrow} = \downarrow$
4. CR $\Longrightarrow$ UN
5. CR $\not\Longleftarrow$ UN

Properties

1. T $\Longrightarrow$ N
2. T $\not\Longleftarrow$ N $\circlearrowleft a \longrightarrow b$
3. CR $\Longleftrightarrow$ $\overset{*}{\leftrightarrow} = \downarrow$
4. CR $\Longrightarrow$ UN
5. CR $\not\Longleftarrow$ UN $\circlearrowleft a \longleftarrow b \longrightarrow c$

Properties

1. $\mathsf{T} \implies \mathsf{N}$
2. $\mathsf{T} \not\Longleftarrow \mathsf{N}$ $\qquad \circlearrowright a \longrightarrow b$
3. $\mathsf{CR} \iff \overset{*}{\leftrightarrow} = \downarrow$
4. $\mathsf{CR} \implies \mathsf{UN}$
5. $\mathsf{CR} \not\Longleftarrow \mathsf{UN}$ $\qquad \circlearrowright a \longleftarrow b \longrightarrow c$
6. $\mathsf{N} \wedge \mathsf{UN} \implies \mathsf{C}$

Properties

1. $\mathsf{T} \implies \mathsf{N}$
2. $\mathsf{T} \not\Longleftarrow \mathsf{N}$ $\qquad \circlearrowright a \longrightarrow b$
3. $\mathsf{CR} \iff \overset{*}{\leftrightarrow} = \downarrow$
4. $\mathsf{CR} \implies \mathsf{UN}$
5. $\mathsf{CR} \not\Longleftarrow \mathsf{UN}$ $\qquad \circlearrowright a \longleftarrow b \longrightarrow c$
6. $\mathsf{N} \wedge \mathsf{UN} \implies \mathsf{C}$
7. $\mathsf{C} \implies \mathsf{LC}$

Properties

1. $\mathsf{T} \implies \mathsf{N}$
2. $\mathsf{T} \not\Longleftarrow \mathsf{N}$ $\qquad \circlearrowright a \longrightarrow b$
3. $\mathsf{CR} \iff \overset{*}{\leftrightarrow} = \downarrow$
4. $\mathsf{CR} \implies \mathsf{UN}$
5. $\mathsf{CR} \not\Longleftarrow \mathsf{UN}$ $\qquad \circlearrowright a \longleftarrow b \longrightarrow c$
6. $\mathsf{N} \wedge \mathsf{UN} \implies \mathsf{C}$
7. $\mathsf{C} \implies \mathsf{LC}$
8. $\mathsf{C} \not\Longleftarrow \mathsf{LC}$

Properties

1. $\mathsf{T} \Longrightarrow \mathsf{N}$
2. $\mathsf{T} \not\Longleftarrow \mathsf{N}$ $\qquad \circlearrowleft a \longrightarrow b$
3. $\mathsf{CR} \Longleftrightarrow \overset{*}{\leftrightarrow} = \downarrow$
4. $\mathsf{CR} \Longrightarrow \mathsf{UN}$
5. $\mathsf{CR} \not\Longleftarrow \mathsf{UN}$ $\qquad \circlearrowleft a \longleftarrow b \longrightarrow c$
6. $\mathsf{N} \wedge \mathsf{UN} \Longrightarrow \mathsf{C}$
7. $\mathsf{C} \Longrightarrow \mathsf{LC}$
8. $\mathsf{C} \not\Longleftarrow \mathsf{LC}$ $\qquad a \longleftarrow b \rightleftarrows c \longrightarrow d$

Properties

Recall what we were looking for.

Ability to check equivalence by the search of a common reduct.

This is exactly the Church-Rosser property.

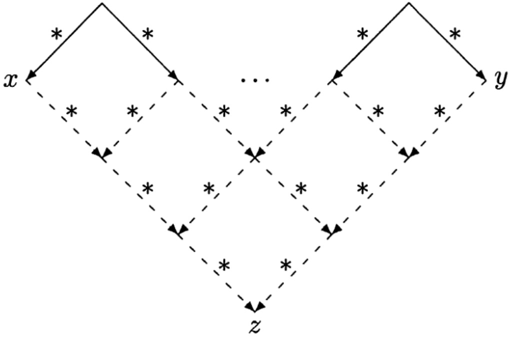
How does it relate to confluence and termination?

Church-Rosser and confluence

The Church-Rosser property and confluence coincide.

CR $\Longrightarrow$ C is immediate.

CR $\Longleftarrow$ C has a nice diagrammatic proof:



Central notions: semi-confluence

Definition 1.5

A relation $\to$ is called semi-confluent (SC) iff

$$y_1 \leftarrow x \xrightarrow{*} y_2 \text{ implies } y_1 \downarrow y_2.$$

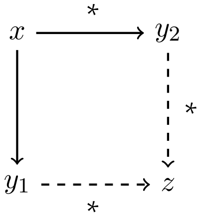
Central notions: semi-confluence

Definition 1.5

A relation $\to$ is called semi-confluent (SC) iff

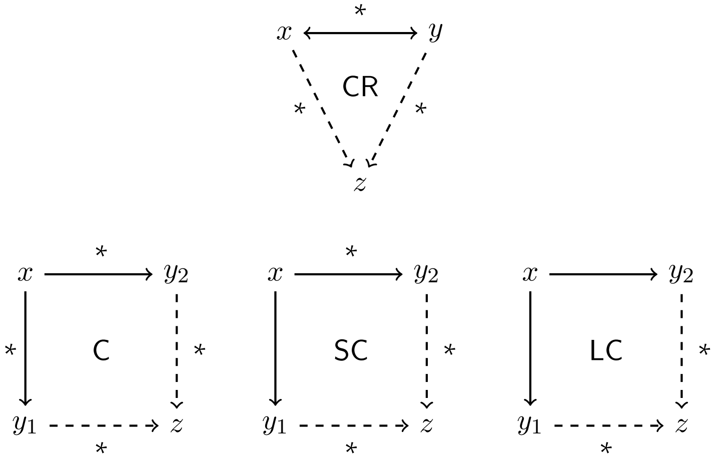
$$y_1 \leftarrow x \xrightarrow{*} y_2 \text{ implies } y_1 \downarrow y_2.$$

Graphically:



Solid arrows represent universal and dashed arrows existential quantification: $\forall x, y_1, y_2.\ y_1 \leftarrow x \xrightarrow{*} y_2 \Rightarrow \exists z.\ y_1 \xrightarrow{*} z \xleftarrow{*} y_2$.

CR, C, SC, LC



Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(1 \Rightarrow 2)$

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(1 \Rightarrow 2)$

- Assume $\rightarrow$ is CR and $y_1 \xleftarrow{*} x \xrightarrow{*} y_2$. Show $y_1 \downarrow y_2$.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(1 \Rightarrow 2)$

- Assume $\rightarrow$ is CR and $y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2$. Show $y_1 \downarrow y_2$.
- $y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2$ implies $y_1 \overset{*}{\leftrightarrow} y_2$.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(1 \Rightarrow 2)$

- Assume $\rightarrow$ is CR and $y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2$. Show $y_1 \downarrow y_2$.
- $y_1 \overset{*}{\leftarrow} x \overset{*}{\rightarrow} y_2$ implies $y_1 \overset{*}{\leftrightarrow} y_2$.
- CR implies $y_1 \downarrow y_2$.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(2 \Rightarrow 3)$

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(2 \Rightarrow 3)$

- Semi-confluence is a special case of confluence.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\to$ *has the Church-Rosser property.*
2. $\to$ *is confluent.*
3. $\to$ *is semi-confluent.*

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$

- Assume $\rightarrow$ is SC and $x \overset{*}{\leftrightarrow} y$. Show $x \downarrow y$.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$

- Assume $\rightarrow$ is SC and $x \overset{*}{\leftrightarrow} y$. Show $x \downarrow y$.
- Induction on the length of the chain $x \overset{*}{\leftrightarrow} y$.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$

- Assume $\rightarrow$ is SC and $x \overset{*}{\leftrightarrow} y$. Show $x \downarrow y$.
- Induction on the length of the chain $x \overset{*}{\leftrightarrow} y$.
- Base case: $x = y$. Trivial.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\to$ *has the Church-Rosser property.*
2. $\to$ *is confluent.*
3. $\to$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$

- Assume $\to$ is SC and $x \overset{*}{\leftrightarrow} y$. Show $x \downarrow y$.
- Induction on the length of the chain $x \overset{*}{\leftrightarrow} y$.
- Base case: $x = y$. Trivial.
- Assume $x \overset{*}{\leftrightarrow} y' \leftrightarrow y$. Show $x \downarrow y$.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\to$ *has the Church-Rosser property.*
2. $\to$ *is confluent.*
3. $\to$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$

- Assume $\to$ is SC and $x \overset{*}{\leftrightarrow} y$. Show $x \downarrow y$.
- Induction on the length of the chain $x \overset{*}{\leftrightarrow} y$.
- Base case: $x = y$. Trivial.
- Assume $x \overset{*}{\leftrightarrow} y' \leftrightarrow y$. Show $x \downarrow y$.
- By IH, $x \downarrow y'$, i.e. $x \overset{*}{\to} z \overset{*}{\leftarrow} y'$ for some z.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$ (Cont.)

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$ (Cont.)

- Show $x \downarrow y$ by case distinction on $y' \leftrightarrow y$.

□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

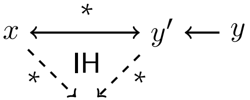
The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$ (Cont.)

- Show $x \downarrow y$ by case distinction on $y' \leftrightarrow y$.
- $y' \leftarrow y$: $x \downarrow y$ follows directly from $x \downarrow y'$:



□

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$ (Cont.)

- Show $x \downarrow y$ by case distinction on $y' \leftrightarrow y$.

$\square$

Church-Rosser, confluence, and semi-confluence

Theorem 1.1

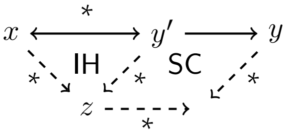
The following conditions are equivalent:

1. $\rightarrow$ *has the Church-Rosser property.*
2. $\rightarrow$ *is confluent.*
3. $\rightarrow$ *is semi-confluent.*

Proof.

$(3 \Rightarrow 1)$ (Cont.)

- Show $x \downarrow y$ by case distinction on $y' \leftrightarrow y$.
- $y' \rightarrow y$: Semi-confluence implies $z \downarrow y$ and, hence $x \downarrow y$:



□

Corollaries

If $\to$ is confluent and $x \overset{*}{\leftrightarrow} y$ then

1. $x \overset{*}{\to} y$ if y is in a normal form, and
2. $x = y$ if both x and y are in a normal form.

Hence, for confluent relations, convertibility is equivalent to joinability.

Without termination, joinability can not be decided.

Corollaries

If $\to$ is confluent, then every element has at most one normal form (C $\Longrightarrow$ UN).

If $\to$ is normalizing and confluent, then every element has exactly one normal form.

Hence, for confluent and normalizing reductions the notation $x\downarrow$ is well-defined.

Goal-directed equivalence test

Theorem 1.2

If $\rightarrow$ is confluent and normalizing, then

- *every element x has a unique normal form $x\downarrow$,*
- *$x \stackrel{*}{\leftrightarrow} y$ iff $x\downarrow = y\downarrow$.*

Normalization requires breadth-first search for normal forms.

Goal-directed equivalence test

Theorem 1.2
If $\to$ is confluent and normalizing, then

- *every element* x *has a unique normal form* $x\downarrow$,
- $x \overset{*}{\leftrightarrow} y$ *iff* $x\downarrow = y\downarrow$.

Normalization requires breadth-first search for normal forms.

Theorem 1.3
If $\to$ is confluent and terminating, then

- *every element* x *has a unique normal form* $x\downarrow$,
- $x \overset{*}{\leftrightarrow} y$ *iff* $x\downarrow = y\downarrow$.

Termination permits depth-first search for normal forms.

Confluence and termination

How to show confluence and termination of an ARS?

Showing termination

Idea: Embedding the reduction into a well-founded order.

Showing termination

Idea: Embedding the reduction into a well-founded order.

Well-founded order $(B, >)$: No infinite descending chain $b_0 > b_1 > b_2 > \cdots$ in B.

Showing termination

Examples of well-founded orders:

- $(\mathbb{N}, >)$: The set of natural numbers with the standard ordering.
- $(\mathbb{N} \setminus \{0\}, >)$: The set of positive integers where $a > b$ iff $b \mid a$ and $b \neq a$.
- $(\{a, b, c\}^*, >)$: The set of finite words over a fixed alphabet, where $w_1 > w_2$ iff w_2 is a proper substring of w_1.

Showing termination

Examples of well-founded orders:

- $(\mathbb{N}, >)$: The set of natural numbers with the standard ordering.
- $(\mathbb{N} \setminus \{0\}, >)$: The set of positive integers where $a > b$ iff $b \mid a$ and $b \neq a$.
- $(\{a, b, c\}^*, >)$: The set of finite words over a fixed alphabet, where $w_1 > w_2$ iff w_2 is a proper substring of w_1.

Examples of non-well-founded orders:

- $(\mathbb{Z}, >)$: The set of integers with the standard ordering.
- $(\mathbb{Q}_0^+, >)$: The set of non-negative rationals with the standard ordering.
- $(\{a, b, c\}^*, >)$: The set of finite words over a fixed alphabet, where $>$ is the lexicographic ordering, e.g. $a > ab > abb > \cdots$.

Showing termination

Theorem 1.4

Let $(A, \to)$ *be an ARS. Then* $\to$ *is terminating iff there exists a well-founded order* $(B, >)$ *and a mapping* $\varphi : A \to B$ *such that*

$$a_1 \to a_2 \text{ implies } \varphi(a_1) > \varphi(a_2).$$

Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\rightarrow$ is terminating and locally confluent, then it is confluent.

Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\rightarrow$ is terminating and locally confluent, then it is confluent.

Proof.

- Use well-founded induction. Let $(A, \rightarrow)$ be an ARS. Then WFI is the inference rule:

$$\frac{\forall x \in A.(\forall y \in A.(x \xrightarrow{+} y \Rightarrow P(y)) \Rightarrow P(x))}{\forall x \in A.P(x)} \text{ (WFI)}$$

  where P is some property of elements of A.
- Reads: To prove $P(x)$ for all $x \in A$, try to prove $P(x)$ under the assumption that $P(y)$ holds for all successors y of x.
- Holds when $\rightarrow$ is terminating.

Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\rightarrow$ is terminating and locally confluent, then it is confluent.

Proof. (Cont.)

Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\rightarrow$ is terminating and locally confluent, then it is confluent.

Proof. (Cont.)

- Let P be

$$P(x) = \forall y, z.\ y \overset{*}{\leftarrow} x \overset{*}{\rightarrow} z \Rightarrow y \downarrow z.$$

Obviously, $\rightarrow$ is confluent if $P(x)$ holds for all $x \in A$.

Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\to$ is terminating and locally confluent, then it is confluent.

Proof. (Cont.)

- Let P be

$$P(x) = \forall y, z.\ y \xleftarrow{*} x \xrightarrow{*} z \Rightarrow y \downarrow z.$$

  Obviously, $\to$ is confluent if $P(x)$ holds for all $x \in A$.

- Show $P(x)$ under the assumption $P(t)$ for all $x \xrightarrow{+} t$.

Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\rightarrow$ is terminating and locally confluent, then it is confluent.

Proof. (Cont.)

- Let P be

$$P(x) = \forall y, z.\ y \overset{*}{\leftarrow} x \overset{*}{\rightarrow} z \Rightarrow y \downarrow z.$$

  Obviously, $\rightarrow$ is confluent if $P(x)$ holds for all $x \in A$.
- Show $P(x)$ under the assumption $P(t)$ for all $x \overset{+}{\rightarrow} t$.
- Fix x, y, z arbitrarily. Assume $y \overset{*}{\leftarrow} x \overset{*}{\rightarrow} z$. Prove $y \downarrow z$.

Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\to$ is terminating and locally confluent, then it is confluent.

Proof. (Cont.)

- Let P be

$$P(x) = \forall y, z.\ y \xleftarrow{*} x \xrightarrow{*} z \Rightarrow y \downarrow z.$$

  Obviously, $\to$ is confluent if $P(x)$ holds for all $x \in A$.
- Show $P(x)$ under the assumption $P(t)$ for all $x \xrightarrow{+} t$.
- Fix x, y, z arbitrarily. Assume $y \xleftarrow{*} x \xrightarrow{*} z$. Prove $y \downarrow z$.
- Case 1: $x = y$ or $x = z$. Trivial.

Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\to$ is terminating and locally confluent, then it is confluent.

Proof. (Cont.)

□

Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\to$ is terminating and locally confluent, then it is confluent.

Proof. (Cont.)

- Case 2: $x \to y_1 \xrightarrow{*} y$ and $x \to z_1 \xrightarrow{*} z$.

□

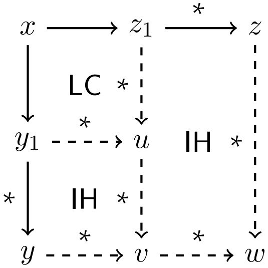
Showing confluence (for a terminating relation)

Lemma 1.1 (Newman's Lemma)

If $\rightarrow$ is terminating and locally confluent, then it is confluent.

Proof. (Cont.)

- Case 2: $x \rightarrow y_1 \overset{*}{\rightarrow} y$ and $x \rightarrow z_1 \overset{*}{\rightarrow} z$.



□

Showing confluence (termination not required)

Definition 1.6

A relation $\rightarrow$ is called strongly confluent (StC) iff

$$\forall x, y_1, y_2.\;\; y_1 \leftarrow x \rightarrow y_2 \Rightarrow \exists z.\; y_1 \xrightarrow{*} z \xleftarrow{=} y_2.$$

Remark: The definition is symmetric: $y_1 \leftarrow x \rightarrow y_2$ must imply both $y_1 \xrightarrow{*} z_1 \xleftarrow{=} y_2$ and $y_1 \xrightarrow{=} z_2 \xleftarrow{*} y_2$ for suitably chosen z_1 and z_2.

Showing confluence (termination not required)

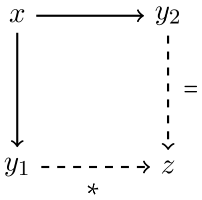
Definition 1.6

A relation $\to$ is called strongly confluent (StC) iff

$$\forall x, y_1, y_2.\ \ y_1 \leftarrow x \to y_2 \Rightarrow \exists z.\ y_1 \xrightarrow{*} z \xleftarrow{=} y_2.$$

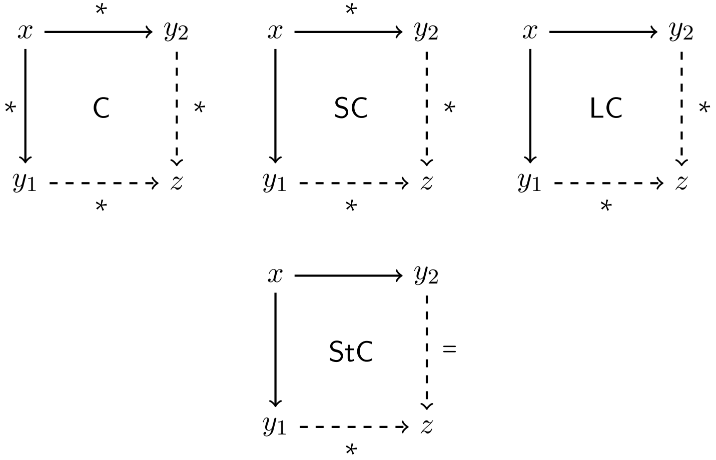
Remark: The definition is symmetric: $y_1 \leftarrow x \to y_2$ must imply both $y_1 \xrightarrow{*} z_1 \xleftarrow{=} y_2$ and $y_1 \xrightarrow{=} z_2 \xleftarrow{*} y_2$ for suitably chosen z_1 and z_2.

Graphically:



Solid arrows represent universal and dashed arrows existential quantification.

C, SC, LC, StC

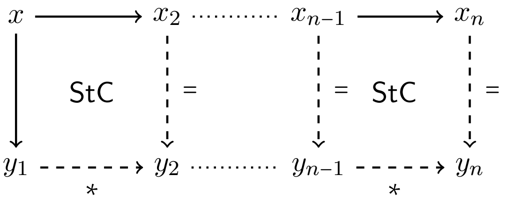


Showing confluence (termination not required)

Theorem 1.5

Any strongly confluent relation is semi-confluent (and, thus, confluent).

Proof.



□

Showing confluence (termination not required)

StC is a pretty strong property.

Trying to show strong confluence of $\rightarrow$ would not be practical.

Showing confluence (termination not required)

StC is a pretty strong property.

Trying to show strong confluence of $\rightarrow$ would not be practical.

The trick to show confluence of $\rightarrow$ is not to prove its strong confluence, but to define a StC relation $\rightarrow_s$ such that $\xrightarrow{*}_s = \xrightarrow{*}$.

Showing confluence (termination not required)

StC is a pretty strong property.

Trying to show strong confluence of $\rightarrow$ would not be practical.

The trick to show confluence of $\rightarrow$ is not to prove its strong confluence, but to define a StC relation $\rightarrow_s$ such that $\xrightarrow{*}_s = \xrightarrow{*}$.

If $\xrightarrow{*}_1 = \xrightarrow{*}_2$, then $\rightarrow_1$ is confluent iff $\rightarrow_2$ is confluent.

Hence, if $\xrightarrow{*}_s = \xrightarrow{*}$, then $\mathsf{StC}(\rightarrow_s) \Rightarrow \mathsf{C}(\rightarrow_s) \Leftrightarrow \mathsf{C}(\rightarrow)$.

Showing confluence (termination not required)

StC is a pretty strong property.

Trying to show strong confluence of $\to$ would not be practical.

The trick to show confluence of $\to$ is not to prove its strong confluence, but to define a StC relation $\to_s$ such that $\xrightarrow{*}_s = \xrightarrow{*}$.

If $\xrightarrow{*}_1 = \xrightarrow{*}_2$, then $\to_1$ is confluent iff $\to_2$ is confluent.

Hence, if $\xrightarrow{*}_s = \xrightarrow{*}$, then $\mathsf{StC}(\to_s) \Rightarrow \mathsf{C}(\to_s) \Leftrightarrow \mathsf{C}(\to)$.

To simplify the search of $\to_s$, the condition can be weakened due to following easy lemma:

If $\to_1 \subseteq \to_2 \subseteq \xrightarrow{*}_1$, then $\xrightarrow{*}_1 = \xrightarrow{*}_2$.

Showing confluence (termination not required)

Summarizing the ideas from the previous slide:

Theorem 1.6

If $\to \subseteq \to_s \subseteq \overset{*}{\to}$ *and* $\to_s$ *is strongly confluent, then* $\to$ *is confluent.*

Showing confluence (termination not required)

The theorem can be made stronger, considering the diamond property:

Definition 1.7

A relation $\to$ has the diamond property iff

$$\forall x, y_1, y_2.\ \ y_1 \leftarrow x \to y_2 \Rightarrow \exists z.\ y_1 \to z \leftarrow y_2.$$

Showing confluence (termination not required)

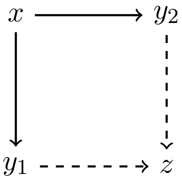
The theorem can be made stronger, considering the diamond property:

Definition 1.7

A relation $\to$ has the diamond property iff

$$\forall x, y_1, y_2.\;\; y_1 \leftarrow x \to y_2 \Rightarrow \exists z.\; y_1 \to z \leftarrow y_2.$$

Graphically:



Showing confluence (termination not required)

The diamond property implies strong confluence, therefore:

Theorem 1.7

If $\rightarrow \subseteq \rightarrow_d \subseteq \xrightarrow{*}$ *and* $\rightarrow_d$ *has the diamond property, then* $\rightarrow$ *is confluent.*

Confluence by commutation

Confluence proofs can be localized by splitting a reduction up into several smaller reductions and showing their confluence separately.

An additional property, commuting, should be satisfied.

Confluence by commutation

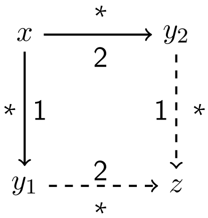
Confluence proofs can be localized by splitting a reduction up into several smaller reductions and showing their confluence separately.

An additional property, commuting, should be satisfied.

Definition 1.8

Two relations $\rightarrow_1$ and $\rightarrow_2$ commute iff

$$\forall x, y_1, y_2.\ \ y_1 \overset{*}{\leftarrow}_1 x \overset{*}{\rightarrow}_2 y_2 \Rightarrow \exists z.\ y_1 \overset{*}{\rightarrow}_2 z \overset{*}{\leftarrow}_1 y_2.$$



Confluence by commutation

Lemma 1.2 (Commutative Union Lemma)

If $\to_1$ and $\to_2$ are confluent and commute, then $\to_1 \cup \to_2$ is also confluent.

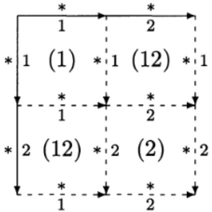
Confluence by commutation

Lemma 1.2 (Commutative Union Lemma)

If $\rightarrow_1$ and $\rightarrow_2$ are confluent and commute, then $\rightarrow_1 \cup \rightarrow_2$ is also confluent.

Proof.

- $\xrightarrow{*}_1 \circ \xrightarrow{*}_2$ has the diamond property:



(1) Confluence of $\rightarrow_1$
(2) Confluence of $\rightarrow_2$
(12) Commutation of $\rightarrow_1$ and $\rightarrow_2$

Confluence by commutation

Lemma 1.3 (Commutative Union Lemma)

If $\to_1$ and $\to_2$ are confluent and commute, then $\to_1 \cup \to_2$ is also confluent.

Proof. (Cont.)

- The following inclusions hold:

$$\to_1 \cup \to_2 \subseteq \xrightarrow{*}_1 \circ \xrightarrow{*}_2 \subseteq (\to_1 \cup \to_2)^*.$$

- By Theorem 1.7, $\to_1 \cup \to_2$ is confluent.

□