

Confluence (abstract rewriting)

In computer science, **confluence** is a property of rewriting systems, describing which terms in such a system can be rewritten in more than one way, to yield the same result. This article describes the properties in the most abstract setting of an abstract rewriting system.

Motivating examples

The usual rules of elementary arithmetic form an abstract rewriting system. For example, the expression $(11 + 9) \times (2 + 4)$ can be evaluated starting either at the left or at the right parentheses; however, in both cases the same result is eventually obtained. If every arithmetic expression evaluates to the same result regardless of reduction strategy, the arithmetic rewriting system is said to be ground-confluent. Arithmetic rewriting systems may be confluent or only ground-confluent depending on details of the rewriting system.^[1]

A second, more abstract example is obtained from the following proof of each group element equalling the inverse of its inverse:^[2]

Group axioms

- | | | |
|-----------|-----------------------|-------------------------|
| A1 | $1 \cdot a$ | $= a$ |
| A2 | $a^{-1} \cdot a$ | $= 1$ |
| A3 | $(a \cdot b) \cdot c$ | $= a \cdot (b \cdot c)$ |

Proof of **R4**: $a^{-1} \cdot (a \cdot b) = b$

$$\begin{aligned} & a^{-1} \cdot (a \cdot b) \\ &= (a^{-1} \cdot a) \cdot b \text{ by A3(r)} \\ &= 1 \cdot b \quad \text{by A2} \\ &= b \quad \text{by A1} \end{aligned}$$

Proof of **R6**: $(a^{-1})^{-1} \cdot 1 = a$

$$\begin{aligned} & (a^{-1})^{-1} \cdot 1 \\ &= (a^{-1})^{-1} \cdot (a^{-1} \cdot a) \text{ by A2(r)} \\ &= a \quad \text{by R4} \end{aligned}$$

Proof of **R10**: $(a^{-1})^{-1} \cdot b = a \cdot b$

$$\begin{aligned} & (a^{-1})^{-1} \cdot b \\ &= (a^{-1})^{-1} \cdot (a^{-1} \cdot (a \cdot b)) \text{ by R4(r)} \\ &= a \cdot b \quad \text{by R4} \end{aligned}$$

Proof of **R11**: $a \cdot 1 = a$

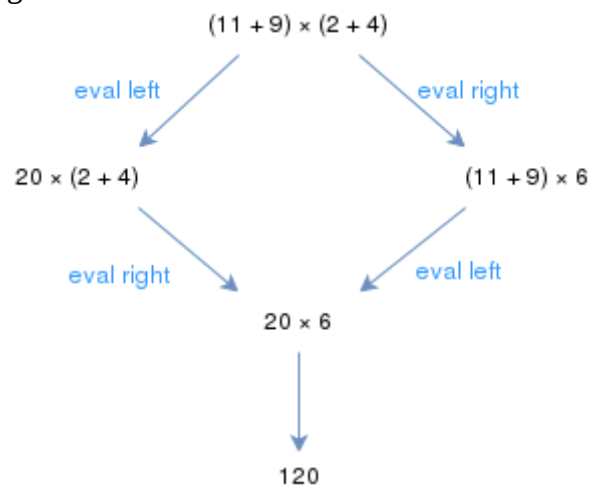
$$\begin{aligned} & a \cdot 1 \\ &= (a^{-1})^{-1} \cdot 1 \text{ by R10(r)} \\ &= a \quad \text{by R6} \end{aligned}$$

Proof of **R12**: $(a^{-1})^{-1} = a$

$$\begin{aligned} & (a^{-1})^{-1} \\ &= (a^{-1})^{-1} \cdot 1 \text{ by R11(r)} \\ &= a \quad \text{by R6} \end{aligned}$$



Pic.1: The name *confluence* is inspired from geography, meaning there the meeting of two bodies of water.



This proof starts from the given group axioms A1–A3, and establishes five propositions R4, R6, R10, R11, and R12, each of them using some earlier ones, and R12 being the main theorem. Some of the proofs require non-obvious, or even creative, steps, like applying axiom A2 in reverse, thereby rewriting "1" to

" $a^{-1} \cdot a$ " in the first step of R6's proof. One of the historical motivations to develop the *theory of term rewriting* was to avoid the need for such steps, which are difficult to find by an inexperienced human, let alone by a computer program.

If a term rewriting system is **confluent** and terminating, a straightforward method exists to prove equality between two expressions (also known as terms) s and t : Starting with s , apply equalities^[note 1] from left to right as long as possible, eventually obtaining a term s' . Obtain from t a term t' in a similar way. If both terms s' and t' literally agree, then s and t are (not surprisingly) proven equal. More importantly, if they disagree, then s and t cannot be equal. That is, any two terms s and t that can be proven equal at all can be done so by that method.

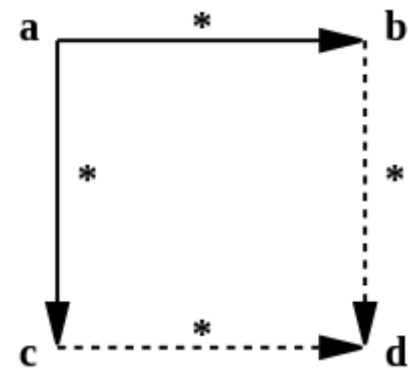
The success of that method does not depend on a certain sophisticated order in which to apply rewrite rules, as **confluence** ensures that any sequence of rule applications will eventually lead to the same result (while the *termination* property ensures that any sequence will eventually reach an end at all). Therefore, if a confluent and terminating term rewriting system can be provided for some equational theory,^[note 2] not a tinge of creativity is required to perform proofs of term equality; that task hence becomes amenable to computer programs. Modern approaches handle more general *abstract rewriting systems* rather than *term rewriting systems*; the latter are a special case of the former.

General case and theory

A rewriting system can be expressed as a directed graph in which nodes represent expressions and edges represent rewrites. So, for example, if the expression a can be rewritten into b , then we say that b is a *reduct* of a (alternatively, a *reduces to* b , or a is an *expansion* of b). This is represented using arrow notation; $a \rightarrow b$ indicates that a reduces to b . Intuitively, this means that the corresponding graph has a directed edge from a to b .

If there is a path between two graph nodes c and d , then it forms a *reduction sequence*. So, for instance, if $c \rightarrow c' \rightarrow c'' \rightarrow \dots \rightarrow d' \rightarrow d$, then we can write $c \xrightarrow{*} d$, indicating the existence of a reduction sequence from c to d . Formally, $\xrightarrow{*}$ is the reflexive-transitive closure of \rightarrow . Using the example from the previous paragraph, we have $(11+9) \times (2+4) \rightarrow 20 \times (2+4)$ and $20 \times (2+4) \rightarrow 20 \times 6$, so $(11+9) \times (2+4) \xrightarrow{*} 20 \times 6$.

With this established, confluence can be defined as follows. $a \in S$ is deemed confluent if for all pairs $b, c \in S$ such that $a \xrightarrow{*} b$ and $a \xrightarrow{*} c$, there exists a $d \in S$ with $b \xrightarrow{*} d$ and $c \xrightarrow{*} d$ (denoted $b \downarrow c$). If every $a \in S$ is confluent, we say that \rightarrow is confluent. This property is also sometimes called the *diamond property*, after the shape of the diagram shown on the right. Some authors reserve the term *diamond property* for a variant of the diagram with single reductions everywhere; that is, whenever $a \rightarrow b$ and $a \rightarrow c$, there must exist a d such that $b \rightarrow d$ and $c \rightarrow d$. The single-reduction variant is strictly stronger than the multi-reduction one.



Pic.2: In this diagram, a reduces to both b or c in zero or more rewrite steps (denoted by the asterisk). In order for the rewrite relation to be confluent, both reducts must in turn reduce to some common d .

Ground confluence

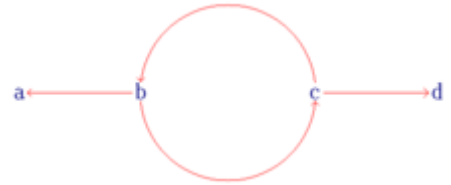
A term rewriting system is **ground confluent** if every ground term is confluent, that is, every term without variables.^[3]

Local confluence

An element $a \in S$ is said to be locally (or weakly) confluent if for all $b, c \in S$ with $a \rightarrow b$ and $a \rightarrow c$ there exists $d \in S$ with $b \xrightarrow{*} d$ and $c \xrightarrow{*} d$. If every $a \in S$ is locally confluent, then \rightarrow is called locally (or weakly) confluent, or having the *weak Church–Rosser property*. This is different from confluence in that b and c must be reduced from a in one step. In analogy with this, confluence is sometimes referred to as *global confluence*.

The relation $\xrightarrow{*}$, introduced as a notation for reduction sequences, may be viewed as a rewriting system in its own right, whose relation is the reflexive-transitive closure of \rightarrow . Since a sequence of reduction sequences is again a reduction sequence (or, equivalently, since forming the reflexive-transitive closure is idempotent), $\xrightarrow{*} = \xrightarrow{*}$. It follows that \rightarrow is confluent if and only if $\xrightarrow{*}$ is locally confluent.

A rewriting system may be locally confluent without being (globally) confluent. Examples are shown in picture 3 and 4. However, Newman's lemma states that if a locally confluent rewriting system has no infinite reduction sequences (in which case it is said to be *terminating* or *strongly normalizing*), then it is globally confluent.



Pic.3: Cyclic, locally-confluent, but not globally confluent rewrite system^[4]



Pic.4: Infinite non-cyclic, locally-confluent, but not globally confluent rewrite system^[4]

Church–Rosser property

A rewriting system is said to possess the **Church–Rosser property** if and only if $x \xleftrightarrow{*} y$ implies $x \downarrow y$ for all objects x, y . Alonzo Church and J. Barkley Rosser proved in 1936 that lambda calculus has this property;^[5] hence the name of the property.^[6] (The fact that lambda calculus has this property is also known as the Church–Rosser theorem.) In a rewriting system with the Church–Rosser property the word problem may be reduced to the search for a common successor. In a Church–Rosser system, an object has *at most one normal form*; that is the normal form of an object is unique if it exists, but it may well not exist. In lambda calculus for instance, the expression $(\lambda x.xx)(\lambda x.xx)$ does not have a normal form because there exists an infinite sequence of β -reductions $(\lambda x.xx)(\lambda x.xx) \rightarrow (\lambda x.xx)(\lambda x.xx) \rightarrow \dots$ ^[7]

A rewriting system possesses the Church–Rosser property if and only if it is confluent.^[8] Because of this equivalence, a fair bit of variation in definitions is encountered in the literature. For instance, in "Terese" the Church–Rosser property and confluence are defined to be synonymous and identical to the definition of confluence presented here; Church–Rosser as defined here remains unnamed, but is given as an equivalent property; this departure from other texts is deliberate.^[9]

Semi-confluence

The definition of local confluence differs from that of global confluence in that only elements reached from a given element in a single rewriting step are considered. By considering one element reached in a single step and another element reached by an arbitrary sequence, we arrive at the intermediate concept of semi-confluence: $a \in S$ is said to be semi-confluent if for all $b, c \in S$ with $a \rightarrow b$ and $a \xrightarrow{*} c$ there exists $d \in S$ with $b \xrightarrow{*} d$ and $c \xrightarrow{*} d$; if every $a \in S$ is semi-confluent, we say that \rightarrow is semi-confluent.

A semi-confluent element need not be confluent, but a semi-confluent rewriting system is necessarily confluent, and a confluent system is trivially semi-confluent.

Strong confluence

Strong confluence is another variation on local confluence that allows us to conclude that a rewriting system is globally confluent. An element $a \in S$ is said to be strongly confluent if for all $b, c \in S$ with $a \rightarrow b$ and $a \rightarrow c$ there exists $d \in S$ with $b \xrightarrow{*} d$ and either $c \rightarrow d$ or $c = d$; if every $a \in S$ is strongly confluent, we say that \rightarrow is strongly confluent.

A confluent element need not be strongly confluent, but a strongly confluent rewriting system is necessarily confluent.

Examples of confluent systems

- Reduction of polynomials modulo an ideal is a confluent rewrite system provided one works with a Gröbner basis.
- Matsumoto's theorem follows from confluence of the braid relations.
- β -reduction of λ -terms is confluent by the Church–Rosser theorem.

See also

- Convergence (logic)
- Critical pair (logic)
- Normal form (abstract rewriting)

Notes

1. then called *rewrite rules* to emphasize their left-to-right orientation
2. The Knuth–Bendix completion algorithm can be used to compute such a system from a given set of equations. Such a system e.g. for groups is shown here, with its propositions consistently numbered. Using it, a proof of e.g. R6 consists in applying R11 and R12 in any order to the term $(a^{-1})^{-1} \cdot 1$ to obtain the term a ; no other rules are applicable.

References

- *Term Rewriting Systems*, Terese, Cambridge Tracts in Theoretical Computer Science, 2003.
 - *Term Rewriting and All That* (<https://books.google.com/books?id=N7BvXVUCQk8C&q=confluence+OR+confluent>), Franz Baader and Tobias Nipkow, Cambridge University Press, 1998
1. Walters, H.R.; Zantema, H. (Oct 1994). "Rewrite systems for integer arithmetic" (<https://pure.tue.nl/ws/files/4421201/433758.pdf>) (PDF). Utrecht University.

2. K. H. Bläsius and H.-J. Bürckert, ed. (1992). *Deduktionssysteme*. Oldenbourg. p. 291. Here: p.134; axiom and proposition names follow the original text
3. Robinson, Alan J. A.; Voronkov, Andrei (5 July 2001). *Handbook of Automated Reasoning* (<https://books.google.com/books?id=X3z8ujBRgmEC&dq=ground%20confluent&pg=PA560>). Gulf Professional Publishing. p. 560. ISBN 978-0-444-82949-8.
4. N. Dershowitz and J.-P. Jouannaud (1990). "Rewrite Systems". In Jan van Leeuwen (ed.). *Formal Models and Semantics*. Handbook of Theoretical Computer Science. Vol. B. Elsevier. pp. 243–320. ISBN 0-444-88074-7. Here: p.268, Fig.2a+b.
5. Alonzo Church and J. Barkley Rosser. Some properties of conversion. Trans. AMS, 39:472-482, 1936
6. Baader and Nipkow, p. 9
7. Cooper, S. B. (2004). *Computability theory*. Boca Raton: Chapman & Hall/CRC. p. 184. ISBN 1584882379.
8. Baader and Nipkow, p. 11
9. Marc Bezem, Jan Willem Klop, Roel de Vrijer ("Terese"), *Term rewriting systems*, Cambridge University Press, 2003, ISBN 0-521-39115-6, Here: p.11

External links

- Weisstein, Eric W. "Confluent" (<https://mathworld.wolfram.com/Confluent.html>). *MathWorld*.
-

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Confluence_\(abstract_rewriting\)&oldid=1150626661](https://en.wikipedia.org/w/index.php?title=Confluence_(abstract_rewriting)&oldid=1150626661)"