# Transitive closure

From Wikipedia, the free encyclopedia

*For other uses, see Closure (disambiguation).*
*This article is about the transitive closure of a binary relation. For the transitive closure of a set, see transitive set#Transitive closure.*

In mathematics, the **transitive closure** of a binary relation $R$ on a set $X$ is the transitive relation $R^+$ on set $X$ such that $R^+$ contains $R$ and $R^+$ is minimal (Lidl and Pilz 1998:337). If the binary relation itself is transitive, then the transitive closure is that same binary relation; otherwise, the transitive closure is a different relation. For example, if $X$ is a set of airports and $x\,R\,y$ means "there is a direct flight from airport $x$ to airport $y$", then the transitive closure of $R$ on $X$ is the relation $R^+$: "it is possible to fly from $x$ to $y$ in one or more flights."



## Transitive relations and examples [edit]

A relation $R$ on a set $X$ is transitive if, for all $x$, $y$, $z$ in $X$, whenever $x\,R\,y$ and $y\,R\,z$ then $x\,R\,z$. Examples of transitive relations include the equality relation on any set, the "less than or equal" relation on any linearly ordered set, and the relation "$x$ was born before $y$" on the set of all people. Symbolically, this can be denoted as: if $x < y$ and $y < z$ then $x < z$.

One example of a non-transitive relation is "city $x$ can be reached via a direct flight from city $y$" on the set of all cities. Simply because there is a direct flight from one city to a second city, and a direct flight from the second city to the third, does not imply there is a direct flight from the first city to the third. The transitive closure of this relation is a different relation, namely "there is a sequence of direct flights that begins at city $x$ and ends at city $y$". Every relation can be extended in a similar way to a transitive relation.

An example of a non-transitive relation with a less meaningful transitive closure is "$x$ is the day of the week after $y$". The transitive closure of this relation is "some day $x$ comes after a day $y$ on the calendar", which is trivially true for all days of the week $x$ and $y$ (and thus equivalent to the Cartesian square, which is "$x$ and $y$ are both days of the week").

## Existence and description [edit]

For any relation $R$, the transitive closure of $R$ always exists. To see this, note that the intersection of any family of transitive relations is again transitive. Furthermore, there exists at least one transitive relation containing $R$, namely the trivial one: $X \times X$. The transitive closure of $R$ is then given by the intersection of all transitive relations containing $R$.

For finite sets, we can construct the transitive closure step by step, starting from $R$ and adding transitive edges. This gives the intuition for a general construction. For any set $X$, we can prove that transitive closure is given by the following expression

$$R^+ = \bigcup_{i \in \{1,2,3,\ldots\}} R^i.$$

where $R^i$ is the $i$-th power of $R$, defined inductively by

$$R^1 = R$$

Wikipedia
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Čeština
Deutsch
Español
Euskara
Français
Nederlands
日本語
Polski
Português
Русский
Српски / srpski
Українська
中文
Edit links

and, for $i > 0$,

$$R^{i+1} = R \circ R^i$$

where $\circ$ denotes composition of relations.

To show that the above definition of $R^+$ is the least transitive relation containing $R$, we show that it contains $R$, that it is transitive, and that it is the smallest set with both of those characteristics.

- $R \subseteq R^+$: $R^+$ contains all of the $R^i$, so in particular $R^+$ contains $R$.
- $R^+$ is transitive: every element of $R^+$ is in one of the $R^i$, so $R^+$ must be transitive by the following reasoning: if $(s_1, s_2) \in R^j$ and $(s_2, s_3) \in R^k$, then from composition's associativity, $(s_1, s_3) \in R^{j+k}$ (and thus in $R^+$) because of the definition of $R^i$.
- $R^+$ is minimal: Let $G$ be any transitive relation containing $R$, we want to show that $R^+ \subseteq G$. It is sufficient to show that for every $i > 0$, $R^i \subseteq G$. Well, since $G$ contains $R$, $R^1 \subseteq G$. And since $G$ is transitive, whenever $R^i \subseteq G$, $R^{i+1} \subseteq G$ according to the construction of $R^i$ and what it means to be transitive. Therefore, by induction, $G$ contains every $R^i$, and thus also $R^+$.

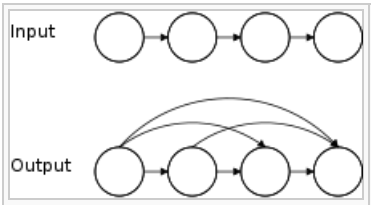
## Properties [edit]

The intersection of two transitive relations is transitive.

The union of two transitive relations need not be transitive. To preserve transitivity, one must take the transitive closure. This occurs, for example, when taking the union of two equivalence relations or two preorders. To obtain a new equivalence relation or preorder one must take the transitive closure (reflexivity and symmetry—in the case of equivalence relations—are automatic).

## In graph theory [edit]

In computer science, the concept of transitive closure can be thought of as constructing a data structure that makes it possible to answer reachability questions. That is, can one get from node *a* to node *d* in one or more hops? A binary relation tells you only that node a is connected to node *b*, and that node *b* is connected to node *c*, etc. After the transitive closure is constructed, as depicted in the following figure, in an O(1) operation one may determine that node *d* is reachable from node *a*. The data structure is typically stored as a matrix, so if matrix[1][4] = 1, then it is the case that node 1 can reach node 4 through one or more hops.



Transitive closure constructs the output graph from the input graph.

The transitive closure of a directed acyclic graph (DAG) is the reachability relation of the DAG and a strict partial order.

## In logic and computational complexity [edit]

The transitive closure of a binary relation cannot, in general, be expressed in first-order logic (FO). This means that one cannot write a formula using predicate symbols *R* and *T* that will be satisfied in any model if and only if *T* is the transitive closure of *R*. In finite model theory, first-order logic (FO) extended with a transitive closure operator is usually called **transitive closure logic**, and abbreviated FO(TC) or just TC. TC is a sub-type of fixpoint logics. The fact that FO(TC) is strictly more expressive than FO was discovered by Ronald Fagin in 1974; the result was then rediscovered by Alfred Aho and Jeffrey Ullman in 1979, who proposed to use fixpoint logic as a database query language (Libkin 2004:vii). With more recent concepts of finite model theory, proof that FO(TC) is strictly more expressive than FO follows immediately from the fact that FO(TC) is not Gaifman-local (Libkin 2004:49).

In computational complexity theory, the complexity class NL corresponds precisely to the set of logical sentences expressible in TC. This is because the transitive closure property has a close relationship with the NL-complete problem STCON for finding directed paths in a graph. Similarly, the class L is first-order logic with the commutative, transitive closure. When transitive closure is added to second-order logic instead, we obtain PSPACE.

## In database query languages [edit]

*Further information: Hierarchical and recursive queries in SQL*

Since the 1980s Oracle Database has implemented a proprietary SQL extension CONNECT BY... START WITH that allows the computation of a transitive closure as part of a declarative query. The SQL 3 (1999) standard added a more general WITH RECURSIVE construct also allowing transitive closures to be computed inside the query processor; as of 2011 the latter is implemented in IBM DB2, Microsoft SQL Server, and PostgreSQL, although not in MySQL (Benedikt and Senellart 2011:189).

Datalog also implements transitive closure computations (Silberschatz et al. 2010:C.3.6).

## Algorithms [edit]

Efficient algorithms for computing the transitive closure of a graph can be found in Nuutila (1995). The fastest worst-case methods, which are not practical, reduce the problem to matrix multiplication. The problem can also be solved by the Floyd–Warshall algorithm, or by repeated breadth-first search or depth-first search starting from each node of the graph.

More recent research has explored efficient ways of computing transitive closure on distributed systems based on the MapReduce paradigm (Afrati et al. 2011).

## See also [edit]

- Deductive closure
- Transitive reduction (a smallest relation having the transitive closure of *R* as its transitive closure)
- Symmetric closure
- Reflexive closure
- Ancestral relation

## References [edit]


- Lidl, R. and Pilz, G., 1998, *Applied abstract algebra*, 2nd edition, Undergraduate Texts in Mathematics, Springer, ISBN 0-387-98290-6
- Keller, U., 2004, *Some Remarks on the Definability of Transitive Closure in First-order Logic and Datalog* (unpublished manuscript)
- Erich Grädel; Phokion G. Kolaitis; Leonid Libkin; Maarten Marx; Joel Spencer; Moshe Y. Vardi; Yde Venema; Scott Weinstein (2007). *Finite Model Theory and Its Applications*. Springer. pp. 151–152. ISBN 978-3-540-68804-4.
- Libkin, Leonid (2004), *Elements of Finite Model Theory*, Springer, ISBN 978-3-540-21202-7
- Heinz-Dieter Ebbinghaus; Jörg Flum (1999). *Finite Model Theory* (2nd ed.). Springer. pp. 123–124, 151–161, 220–235. ISBN 978-3-540-28787-2.
- Aho, A. V.; Ullman, J. D. (1979). "Universality of data retrieval languages". *Proceedings of the 6th ACM SIGACT-SIGPLAN Symposium on Principles of programming languages - POPL '79*. p. 110. doi:10.1145/567752.567763.
- Benedikt, M.; Senellart, P. (2011). "Databases". In Blum, Edward K.; Aho, Alfred V. *Computer Science. The Hardware, Software and Heart of It*. pp. 169–229. doi:10.1007/978-1-4614-1168-0_10. ISBN 978-1-4614-1167-3.
- Nuutila, E., Efficient Transitive Closure Computation in Large Digraphs. Acta Polytechnica Scandinavica, Mathematics and Computing in Engineering Series No. 74, Helsinki 1995, 124 pages. Published by the Finnish Academy of Technology. ISBN 951-666-451-2, ISSN 1237-2404, UDC 681.3.
- Abraham Silberschatz; Henry Korth; S. Sudarshan (2010). *Database System Concepts* (6th ed.). McGraw-Hill. ISBN 978-0-07-352332-3. Appendix C (online only)
- Foto N. Afrati, Vinayak Borkar, Michael Carey, Neoklis Polyzotis, Jeffrey D. Ullman, Map-Reduce Extensions and Recursive Queries, EDBT 2011, March 22–24, 2011, Uppsala, Sweden, ISBN 978-1-4503-0528-0


## External links [edit]

- "Transitive closure and reduction", The Stony Brook Algorithm Repository, Steven Skiena .
- "Apti Algoritmi", An example and some C++ implementations of algorithms that calculate the transitive closure of a given binary relation, Vreda Pieterse.

Categories: Mathematical relations | Closure operators | Graph algorithms

This page was last modified on 28 August 2015, at 17:14.