



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

Deutsch  
Español  
فارسی  
Italiano  
Polski  
Português  
Українська  
中文

 Edit links

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

# Powerset construction

From Wikipedia, the free encyclopedia

In the [theory of computation](#) and [automata theory](#), the **powerset construction** or **subset construction** is a standard method for [converting](#) a [nondeterministic finite automaton](#) (NFA) into a [deterministic finite automaton](#) (DFA) which recognizes the same [formal language](#). It is important in theory because it establishes that NFAs, despite their additional flexibility, are unable to recognize any language that cannot be recognized by some DFA. It is also important in practice for converting easier-to-construct NFAs into more efficiently executable DFAs. However, if the NFA has  $n$  states, the resulting DFA may have up to  $2^n$  states, an exponentially larger number, which sometimes makes the construction impractical for large NFAs.

The construction, sometimes called the **Rabin–Scott powerset construction** (or subset construction) to distinguish it from similar constructions for other types of automata, was first published by [Michael O. Rabin](#) and [Dana Scott](#) in 1959.<sup>[1]</sup>

## Contents [hide]

- [1](#) [Intuition](#)
- [2](#) [Construction](#)
- [3](#) [Example](#)
- [4](#) [Complexity](#)
- [5](#) [Applications](#)
- [6](#) [References](#)
- [7](#) [Additional reading](#)

## Intuition [\[edit\]](#)

To simulate the operation of a DFA on a given input string, one needs to keep track of a single state at any time: the state that the automaton will reach after seeing a [prefix](#) of the input. In contrast, to simulate an NFA, one needs to keep track of a set of states: all of the states that the automaton could reach after seeing the same prefix of the input, according to the nondeterministic choices made by the automaton. If, after a certain prefix of the input, a set  $S$  of states can be reached, then after the next input symbol  $x$  the set of reachable states is a deterministic function of  $S$  and  $x$ . Therefore, the sets of reachable NFA states play the same role in the NFA simulation as single DFA states play in the DFA simulation, and in fact the sets of NFA states appearing in this simulation may be re-interpreted as being states of a DFA.<sup>[2]</sup>

## Construction [\[edit\]](#)

The powerset construction applies most directly to an NFA that does not allow state transformations without consuming input symbols (aka: "ε-moves"). Such an automaton may be defined as a [5-tuple](#)  $(Q, \Sigma, T, q_0, F)$ , in which  $Q$  is the set of states,  $\Sigma$  is the set of input symbols,  $T$  is the transition function (mapping a state and an input symbol to a set of states),  $q_0$  is the initial state, and  $F$  is the set of accepting states. The corresponding DFA has states corresponding to subsets of  $Q$ . The initial state of the DFA is  $\{q_0\}$ , the (one-element) set of initial states. The transition function of the DFA maps a state  $S$  (representing a subset of  $Q$ ) and an input symbol  $x$  to the set  $T(S, x) = \cup \{T(q, x) \mid q \in S\}$ , the set of all states that can be reached by an  $x$ -transition from a state in  $S$ . A state  $S$  of the DFA is an accepting state if and only if at least one member of  $S$  is an accepting state of the NFA.<sup>[2][3]</sup>

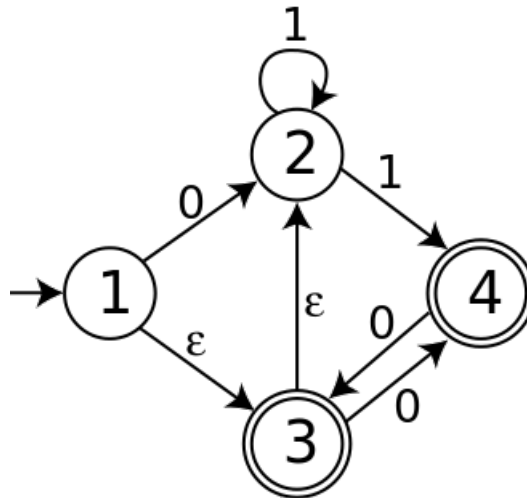
In the simplest version of the powerset construction, the set of all states of the DFA is the [powerset](#) of  $Q$ , the set of all possible subsets of  $Q$ . However, many states of the resulting DFA may be useless as they may be unreachable from the initial state. An alternative version of the construction creates only the states that are actually reachable.<sup>[4]</sup> For an NFA with ε-moves, the construction must be modified somewhat. In this case, the initial state consists of all NFA states reachable by ε-moves from  $q_0$ , and the value  $T(S, x)$  of the transition function is the set of all states reachable by ε-moves from  $\cup \{T(q, x) \mid q \text{ in } S\}$ .<sup>[2]</sup> Alternatively, a similar reachability calculation can be used to convert NFAs with ε-moves to NFAs without ε-moves, after which the simpler construction can be used.<sup>[5]</sup>

It is also possible for the NFA to have more than one initial state.<sup>[6]</sup> In this case, the initial state of the

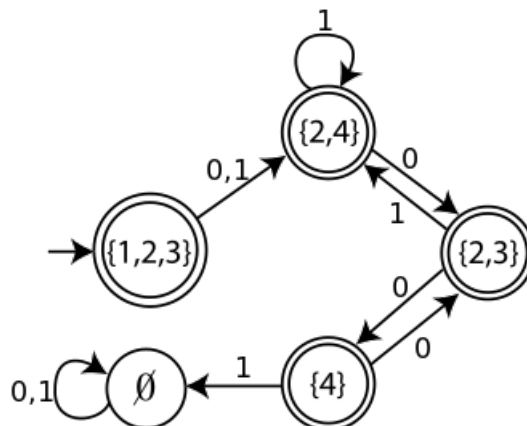
corresponding DFA is the set of all initial states of the NFA, or (if the NFA also has  $\epsilon$ -moves) the set of all states reachable from initial states by  $\epsilon$ -moves.

## Example [\[edit\]](#)

The NFA below has four states; state 1 is initial, and states 3 and 4 are accepting. Its alphabet consists of the two symbols 0 and 1, and it has  $\epsilon$ -moves.



The initial state of the DFA constructed from this NFA is the set of all NFA states that are reachable from state 1 by  $\epsilon$ -moves; that is, it is the set  $\{1,2,3\}$ . A transition from  $\{1,2,3\}$  by input symbol 0 must follow either the arrow from state 1 to state 2, or the arrow from state 3 to state 4. Additionally, neither state 2 nor state 4 have outgoing  $\epsilon$ -moves. Therefore,  $T(\{1,2,3\},0) = \{2,4\}$ , and by the same reasoning the full DFA constructed from the NFA is as shown below.



As can be seen in this example, there are five states reachable from the start state of the DFA; the remaining 11 sets in the powerset of the set of NFA states are not reachable.

## Complexity [\[edit\]](#)

Because the DFA states consist of sets of NFA states, an  $n$ -state NFA may be converted to a DFA with at most  $2^n$  states.<sup>[2]</sup> For every  $n$ , there exist  $n$ -state NFAs such that every subset of states is reachable from the initial subset, so that the converted DFA has exactly  $2^n$  states, giving  $\Theta(2^n)$  **worst-case** time complexity.<sup>[7]</sup> A simple example requiring nearly this many states is the language of strings over the alphabet  $\{0,1\}$  in which there are at least  $n$  characters, the  $n$ th from last of which is 1. It can be represented by an  $(n+1)$ -state NFA, but it requires  $2^n$  DFA states, one for each  $n$ -character suffix of the input; cf. picture for  $n=4$ .<sup>[4]</sup>

## Applications [\[edit\]](#)

**Brzowski's** algorithm for **DFA minimization** uses the powerset construction, twice. It converts the input DFA into an NFA for the reverse language, by reversing all its arrows and exchanging the roles of initial and accepting states, converts the NFA back into a DFA using the powerset construction, and then repeats its process. Its worst-case complexity is exponential, unlike some other known DFA minimization algorithms, but in many examples it performs more quickly than its worst-case complexity would suggest.<sup>[8]</sup>

**Safra's construction**, which converts a non-deterministic **Büchi automaton** with  $n$  states into a deterministic

Muller automaton or into a deterministic Rabin automaton with  $2^{O(n \log n)}$  states, uses the powerset construction as part of its machinery.<sup>[9]</sup>

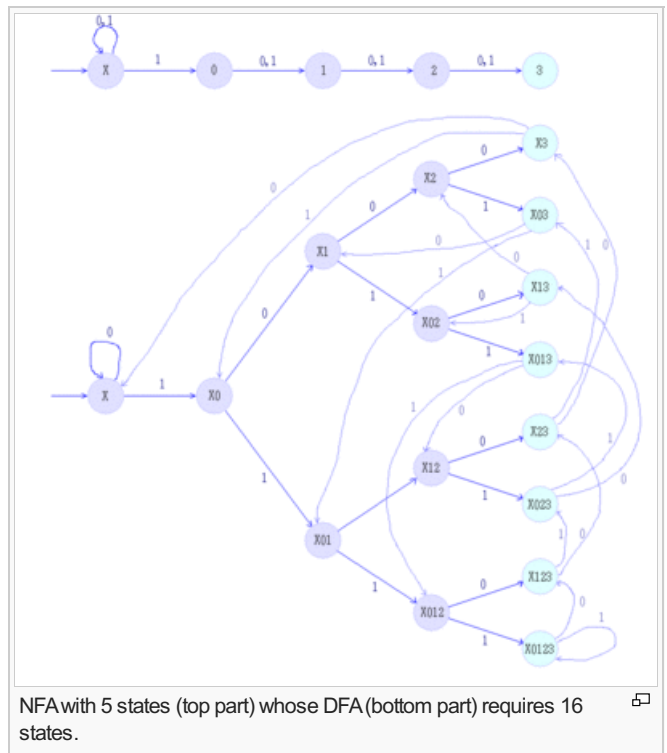
## References [\[edit\]](#)

- ↑ Rabin, M. O.; Scott, D. (1959). "Finite automata and their decision problems". *IBM Journal of Research and Development* **3** (2): 114–125. doi:10.1147/rd.32.0114. ISSN 0018-8646.
- ↑  <sup>*a*</sup>  <sup>*b*</sup>  <sup>*c*</sup>  <sup>*d*</sup> Sipser, Michael. "Theorem 1.19". *Introduction to the Theory of Computation*. pp. 55–56. ISBN 0-534-94728-X.
- ↑ Hopcroft, John E.; Ullman, Jeffrey D. (1979). "The equivalence of DFA's and NFA's". *Introduction to Automata Theory, Languages, and Computation*. Reading Massachusetts: Addison-Wesley. pp. 22–23. ISBN 0-201-02988-X.
- ↑  <sup>*a*</sup>  <sup>*b*</sup> Schneider, Klaus (2004). *Verification of reactive systems: formal methods and algorithms*. Springer. pp. 210–212. ISBN 978-3-540-00296-3.
- ↑ Hopcroft & Ullman (1979), pp. 26–27.
- ↑ Rothe, Jörg (2006), *Complexity Theory and Cryptology: An Introduction to Cryptocomplexity*, Texts in Theoretical Computer Science, Springer, pp. 21–22, ISBN 9783540285205.
- ↑ Moore, Frank R. (1971), "On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata", *IEEE Transactions on Computers* **C-20** (10): 1211–1214, doi:10.1109/T-C.1971.223108.
- ↑ Brzozowski, J. A. (1963). "Canonical regular expressions and minimal state graphs for definite events". *Proc. Sympos. Math. Theory of Automata (New York, 1962)*. Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y. pp. 529–561. MR 0175719.
- ↑ Safra, S. (1988), "On the complexity of  $\omega$ -automata", *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS '88)*, Washington, DC, USA: IEEE Computer Society, pp. 319–327, doi:10.1109/SFCS.1988.21948.

## Additional reading [\[edit\]](#)

- Anderson, James Andrew (2006). *Automata theory with modern applications*. Cambridge University Press. pp. 43–49. ISBN 978-0-521-84887-9.

Categories:  Theory of computation |  Automata theory



This page was last modified on 18 July 2015, at 04:56.

Text is available under the  Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the  Terms of Use and  Privacy Policy. Wikipedia® is a registered trademark of the  Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy  About Wikipedia  Disclaimers  Contact Wikipedia  Developers  Mobile view

