PDF version of the entry
Turing Machines
http://plato.stanford.edu/archives/spr2016/entries/turing-machine/

from the SPRING 2016 EDITION of the

STANFORD ENCYCLOPEDIA
OF PHILOSOPHY



Library of Congress Catalog Data
ISSN: 1095-5054





# Turing Machines

*First published Thu Sep 14, 1995; substantive revision Tue Jun 26, 2012*

Turing machines, first described by Alan Turing in (Turing 1937a), are simple abstract computational devices intended to help investigate the extent and limitations of what can be computed.

Turing was interested in the question of what it means for a task to be computable, which is one of the foundational questions in the philosophy of computer science. Intuitively a task is computable if it is possible to specify a sequence of instructions which will result in the completion of the task when they are carried out by some machine. Such a set of instructions is called an *effective procedure*, or *algorithm*, for the task. The problem with this intuition is that what counts as an effective procedure may depend on the capabilities of the machine used to carry out the instructions. In principle, devices with different capabilities may be able to complete different instruction sets, and therefore may result in different classes of computable tasks (see the entry on computability and complexity).

Turing proposed a class of devices that came to be known as Turing machines. These devices lead to a formal notion of computation that we will call *Turing-computability*. A task is Turing computable if it can be carried out by some Turing machine.

The proposition that Turing's notion captures exactly the intuitive idea of effective procedure is called the Church-Turing thesis. This proposition is not provable, since it is a claim about the relationship between a formal concept and intuition. The thesis would be refuted by an intuitively acceptable algorithm for a task that is not Turing-computable, and no such counterexample has been found. Other independently defined notions of