Rice theorem states that any non-trivial semantic property of a language which is recognized by a Turing machine is undecidable. A property, P, is the language of all Turing machines that satisfy that property.

Formal Definition

If P is a non-trivial property, and the language holding the property, Lp , is recognized by Turing machine M, then Lp = {<M> | L(M) ∈ P} is undecidable.

Description and Properties

* Property of languages, P, is simply a set of languages. If any language belongs to P (L ∈ P), it is said that L satisfies the property P.
* A property is called to be trivial if either it is not satisfied by any recursively enumerable languages, or if it is satisfied by all recursively enumerable languages.
* A non-trivial property is satisfied by some recursively enumerable languages and are not satisfied by others. Formally speaking, in a non-trivial property, where L ∈ P, both the following properties hold:
  + **Property 1** − There exists Turing Machines, M1 and M2 that recognize the same language, i.e. either ( <M1>, <M2> ∈ L ) or ( <M1>,<M2> ∉ L )
  + **Property 2** − There exists Turing Machines M1 and M2, where M1 recognizes the language while M2 does not, i.e. <M1> ∈ L and <M2> ∉ L

Proof

Suppose, a property P is non-trivial and φ ∈ P.

Since, P is non-trivial, at least one language satisfies P, i.e., L(M0) ∈ P , ∋ Turing Machine M0.

Let, w be an input in a particular instant and N is a Turing Machine which follows −

On input x

* Run M on w
* If M does not accept (or doesn't halt), then do not accept x (or do not halt)
* If M accepts w then run M0 on x. If M0 accepts x, then accept x.

A function that maps an instance ATM = {<M,w>| M accepts input w} to a N such that

* If M accepts w and N accepts the same language as M0, Then L(M) = L(M0) ∈ p
* If M does not accept w and N accepts φ, Then L(N) = φ ∉ p

Since ATM is undecidable and it can be reduced to Lp, Lp is also undecidable.