S_{tm} and A_{tm}

Self-Denying Machines

Define language S_{tm} as: set of $\langle M \rangle$ where TM M does NOT accept $\langle M \rangle$.

Theorem. S_{tm} is not r.e.

Proof by contradiction or diagonalization. (see below)

Acceptance Language

Define language A_{tm} as set of $\langle M, w \rangle$ where TM M accepts string w.

Fact: A_{tm} is r.e. (Direct Simulation)

Theorem. A_{tm} is not recursive

Proof by contradiction. (see below)

Proving S_{tm} not r.e.

METHOD 1. proof by contradiction: A machine that allegedly accepts exactly S_{tm} faces an impossibility when it gets itself as input

METHOD 2. diagonalization proof: build a language that is the language of no TM

Proving A_{tm} *is not recursive*

METHOD 1. build machine N that uses A_{tm}-machine to do the opposite of A_{tm}; so that when A_{tm}-machine opines on $\langle N \rangle$ we get a contradiction

METHOD 2. If A_{tm} were recursive, then it could immediately be used as a subroutine in a machine for S_{tm}