A Universal Turing Machine

A limitation of Turing Machines:

Turing Machines are "hardwired"

$\underbrace{\text{"hardwired"}}_{}$

they execute
only one program

Real Computers are re-programmable

Solution: Universal Turing Machine

Attributes:

- Reprogrammable machine
- Simulates any other Turing Machine

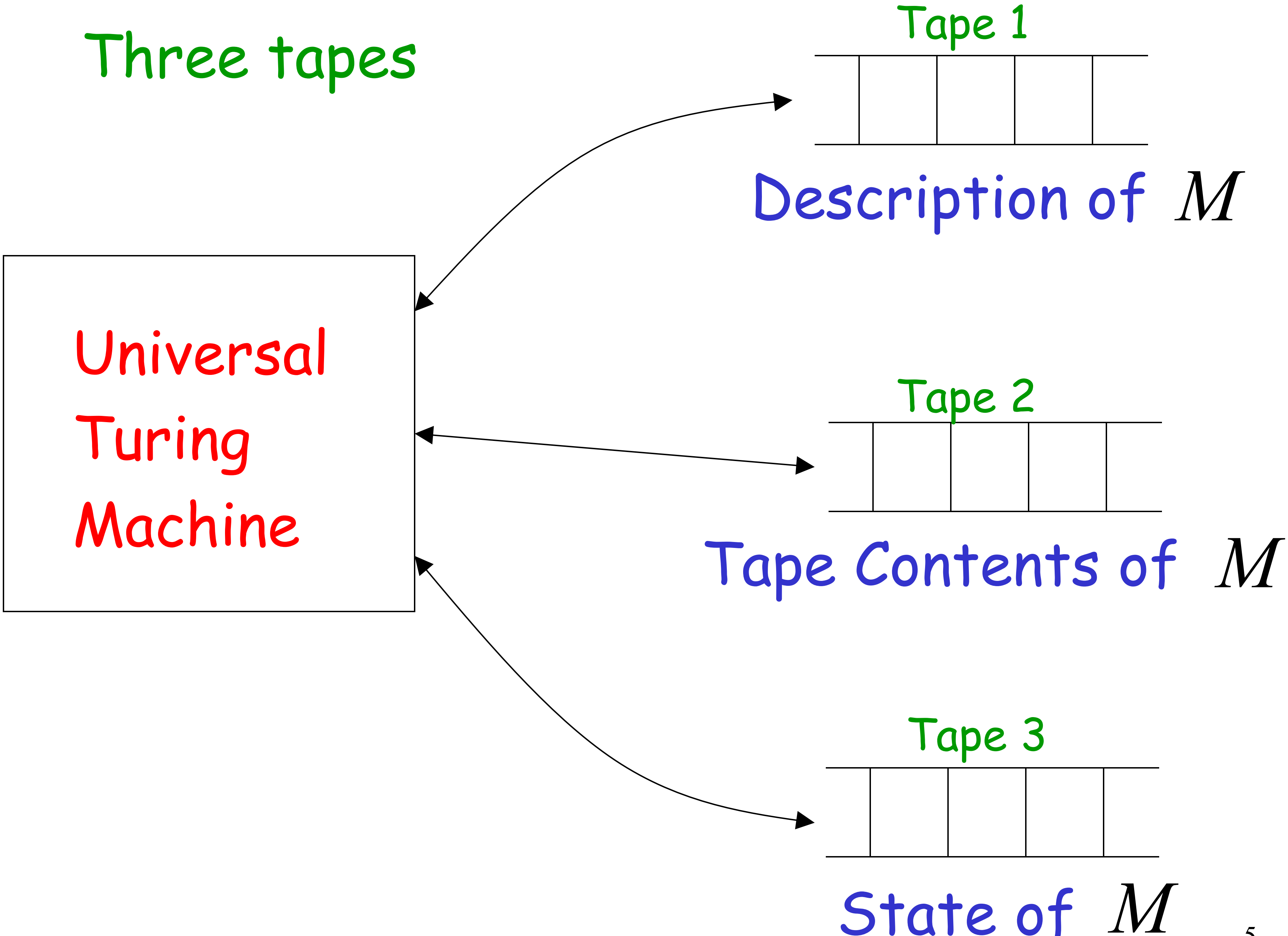
Universal Turing Machine

simulates any Turing Machine M

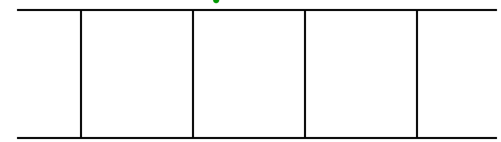
Input of Universal Turing Machine:

Description of transitions of M

Input string of M


Three tapes
Universal
Turing
Machine
Tape 1
Description of M
Tape 2
Tape Contents of M
Tape 3
State of M

Tape 1



Description of M

We describe Turing machine M
as a string of symbols:

We encode M as a string of symbols

Alphabet Encoding

Symbols:	a	b	c	d	$\cdots$
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
Encoding:	1	11	111	1111	

State Encoding

States:	q_1	q_2	q_3	q_4	$\cdots$
Encoding:	1	11	111	1111	

Head Move Encoding

Move:	L	R
Encoding:	1	11

Transition Encoding

Transition: $\delta(q_1, a) = (q_2, b, L)$

Encoding: $1\,0\,1\,0\,11\,0\,11\,0\,1$

separator

Turing Machine Encoding

Transitions:

$$\delta(q_1, a) = (q_2, b, L) \qquad \delta(q_2, b) = (q_3, c, R)$$

Encoding:

$$10101101101 \; 00 \; 1101101110111011$$

separator

Tape 1 contents of Universal Turing Machine:

binary encoding
of the simulated machine M

Tape 1

1 0 1 0 11 0 11 0 10011 0 1 10 111 0 111 0 1100...

A Turing Machine is described with a binary string of 0's and 1's

Therefore:

The set of Turing machines forms a language:

each string of this language is the binary encoding of a Turing Machine

Language of Turing Machines

L = { 010100101, (Turing Machine 1)

00100100101111, (Turing Machine 2)

111010011110010101, ……

…… }

Countable Sets

Infinite sets are either:

Countable

or

Uncountable

Countable set:

There is a one to one correspondence

of

elements of the set

to

Natural numbers (Positive Integers)

(every element of the set is mapped to a number
such that no two elements are mapped to same number)

Example: The set of even integers is countable

Even integers: (positive) $0, \ 2, \ 4, \ 6, \ \ldots$

Correspondence:

$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

Positive integers: $1, \ 2, \ 3, \ 4, \ \ldots$

$2n$ corresponds to $n+1$

Example: The set of rational numbers is countable

Rational numbers: $\frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \ldots$

Naïve Approach

Nominator 1

Rational numbers: $\frac{1}{1}, \frac{1}{2}, \frac{1}{3}, \ldots$

Correspondence: $\downarrow \quad \downarrow \quad \downarrow$

Positive integers: $1, 2, 3, \ldots$

Doesn't work:

we will never count numbers with nominator 2: $\frac{2}{1}, \frac{2}{2}, \frac{2}{3}, \ldots$

Better Approach

$$\frac{1}{1} \quad \frac{1}{2} \quad \frac{1}{3} \quad \frac{1}{4} \quad \cdots$$

$$\frac{2}{1} \quad \frac{2}{2} \quad \frac{2}{3} \quad \cdots$$

$$\frac{3}{1} \quad \frac{3}{2} \quad \cdots$$

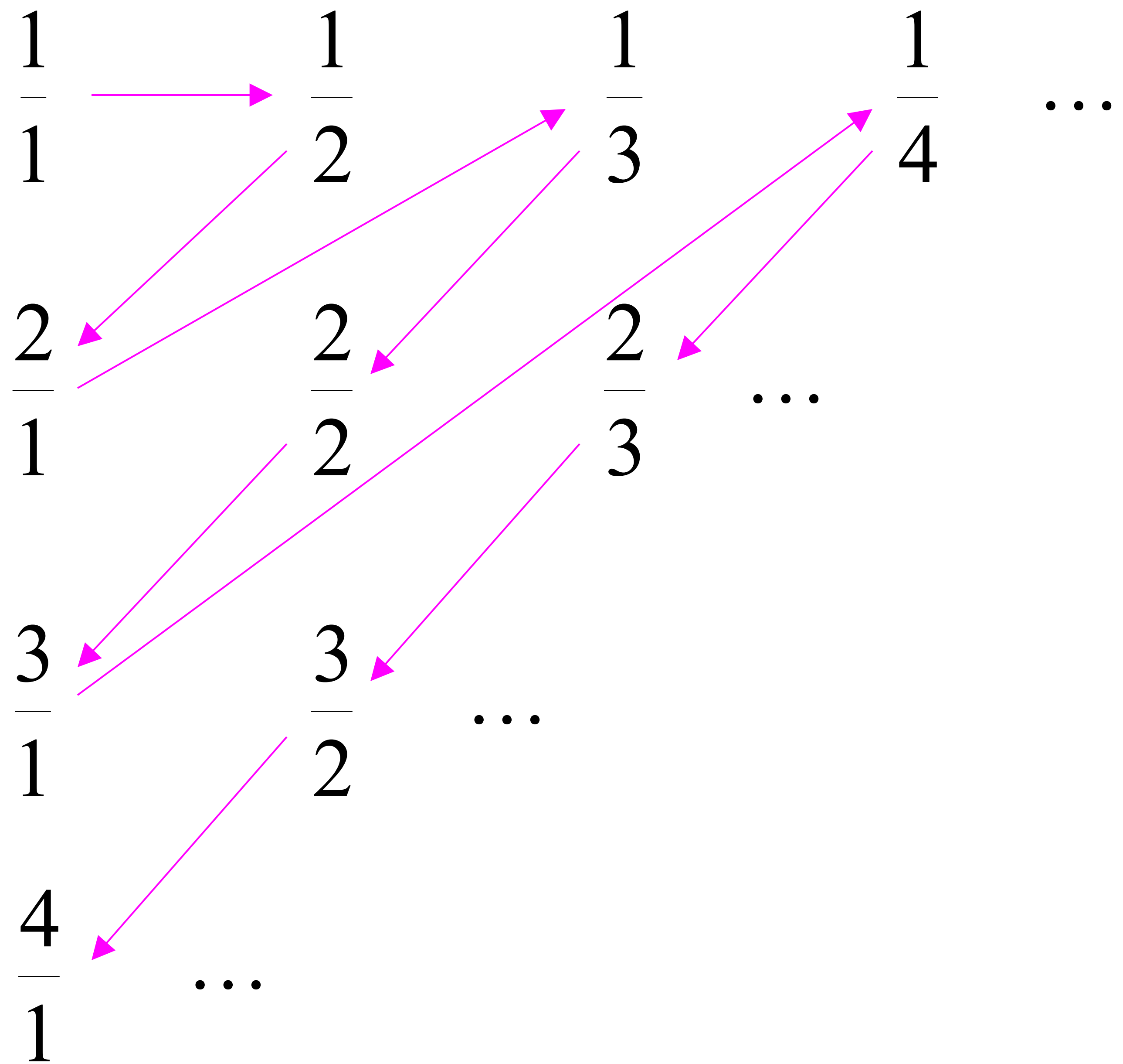
$$\frac{4}{1} \quad \cdots$$

$$
\begin{array}{ccccc}
\frac{1}{1} & \rightarrow \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots \\
\frac{2}{1} & \frac{2}{2} & \frac{2}{3} & \cdots & \\
\frac{3}{1} & \frac{3}{2} & \cdots & & \\
\frac{4}{1} & \cdots & & &
\end{array}
$$

$$
\begin{tikzcd}
\frac{1}{1} \arrow[r] & \frac{1}{2} \arrow[dl] & \frac{1}{3} & \frac{1}{4} & \cdots \\
\frac{2}{1} & \frac{2}{2} & \frac{2}{3} & \cdots & \\
\frac{3}{1} & \frac{3}{2} & \cdots & & \\
\frac{4}{1} & \cdots & & &
\end{tikzcd}
$$

$$
\begin{array}{ccccc}
\frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots \\
\frac{2}{1} & \frac{2}{2} & \frac{2}{3} & \cdots & \\
\frac{3}{1} & \frac{3}{2} & \cdots & & \\
\frac{4}{1} & \cdots & & &
\end{array}
$$

$$
\begin{array}{ccccc}
\frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots \\
\frac{2}{1} & \frac{2}{2} & \frac{2}{3} & \cdots & \\
\frac{3}{1} & \frac{3}{2} & \cdots & & \\
\frac{4}{1} & \cdots & & &
\end{array}
$$


1/1 1/2 1/3 1/4 ...
2/1 2/2 2/3 ...
3/1 3/2 ...
4/1 ...

Rational Numbers: $\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{1}{3}, \frac{2}{2}, \ldots$

Correspondence:

Positive Integers: $1, \quad 2, \quad 3, \quad 4, \quad 5, \quad \ldots$

We proved:

the set of rational numbers is countable
by describing an enumeration procedure
(enumerator)
for the correspondence to natural numbers

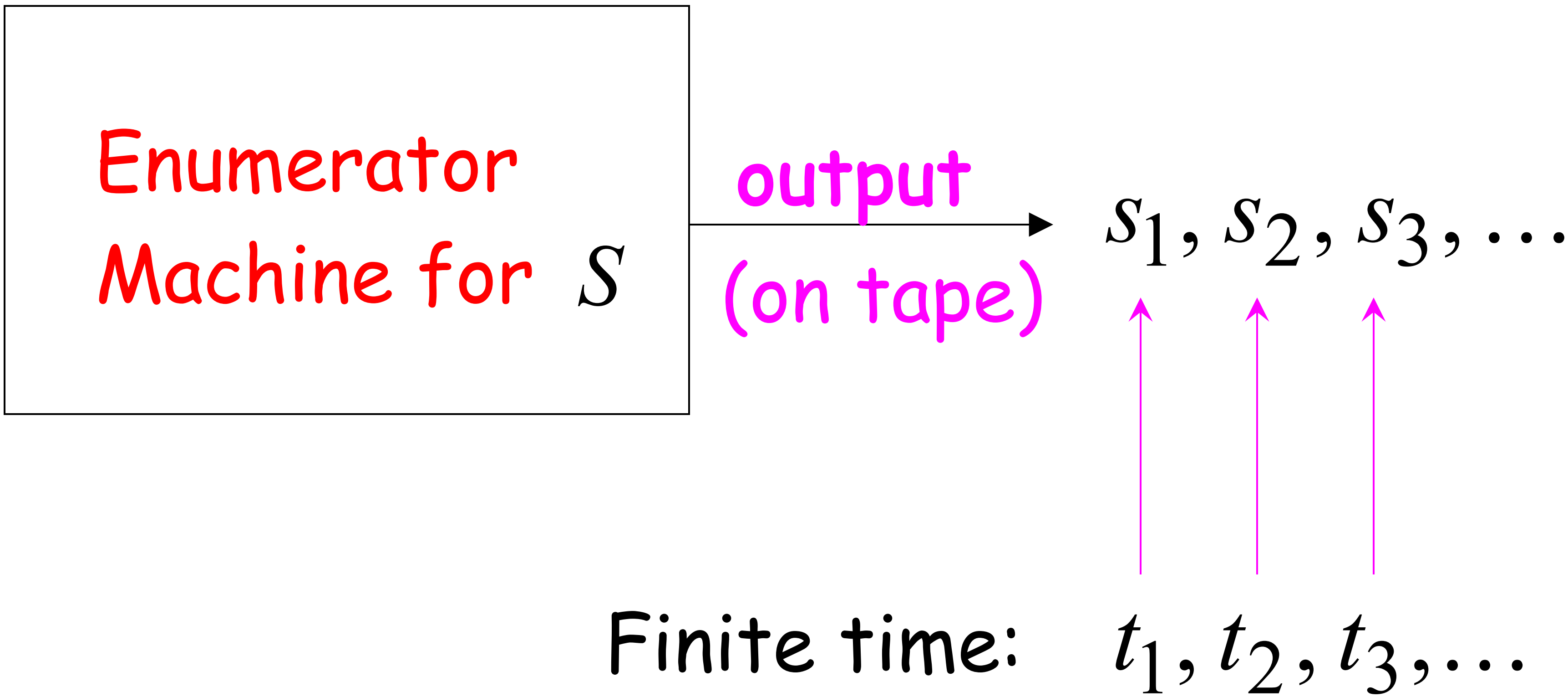
Definition

Let S be a set of strings (Language)

An **enumerator** for S is a Turing Machine
that generates (prints on tape)
all the strings of S one by one

and
each string is generated in finite time

strings $s_1, s_2, s_3, \ldots \in S$



Observation:

If for a set S there is an enumerator, then the set is countable

The enumerator describes the correspondence of S to natural numbers

Example: The set of strings $S = \{a, b, c\}^+$ is countable

Approach:

We will describe an enumerator for S

Naive enumerator:

Produce the strings in lexicographic order:

$$
\begin{aligned}
s_1 &= a \\
s_2 &= aa \\
\vdots \quad & \quad aaa \\
& \quad aaaa \\
& \quad \ldots\ldots
\end{aligned}
$$

Doesn't work:

strings starting with b
will never be produced

Better procedure: Proper Order (Canonical Order)

1. Produce all strings of length 1

2. Produce all strings of length 2

3. Produce all strings of length 3

4. Produce all strings of length 4

....

Produce strings in Proper Order:

$$
\left.
\begin{aligned}
s_1 &= a \\
s_2 &= b \\
\vdots\quad & c
\end{aligned}
\right\} \text{length 1}
$$

$$
\left.
\begin{array}{l}
aa \\ ab \\ ac \\ ba \\ bb \\ bc \\ ca \\ cb \\ cc
\end{array}
\right\} \text{length 2}
$$

$$
\left.
\begin{array}{l}
aaa \\ aab \\ aac \\ \ldots\ldots
\end{array}
\right\} \text{length 3}
$$

Theorem: The set of all Turing Machines is countable

Proof: Any Turing Machine can be encoded with a binary string of 0's and 1's

Find an enumeration procedure for the set of Turing Machine strings

Uncountable Sets

We will prove that there is a language L' which is not accepted by any Turing machine

Technique:

Turing machines are countable

Languages are uncountable

(there are more languages than Turing Machines)

Definition: A set is uncountable if it is not countable

We will prove that there is a language which is not accepted by any Turing machine

Theorem:

If S is an infinite countable set, then

the powerset 2^S of S is uncountable.

(the powerset 2^S is the set whose elements
are all possible sets made from the elements of S)

Languages accepted by Turing machines: X countable

All possible languages: 2^S uncountable

Therefore: $X \neq 2^S$

$$\left(\text{since } X \subseteq 2^S, \text{ we have } X \subset 2^S\right)$$

Conclusion:

There is a language L' not accepted by any Turing Machine:

$$X \subset 2^S \Longrightarrow \exists L' \in 2^S \text{ and } L' \notin X$$

(Language L' cannot be described by any algorithm)

Non Turing-Acceptable Languages