A Universal Turing Machine

A limitation of Turing Machines:

$$\text{Turing Machines are } \underbrace{\text{“hardwired”}}_{\substack{\text{they execute} \\ \text{only one program}}}$$

Real Computers are re-programmable

Solution: Universal Turing Machine

Attributes:

- Reprogrammable machine
- Simulates any other Turing Machine

Universal Turing Machine

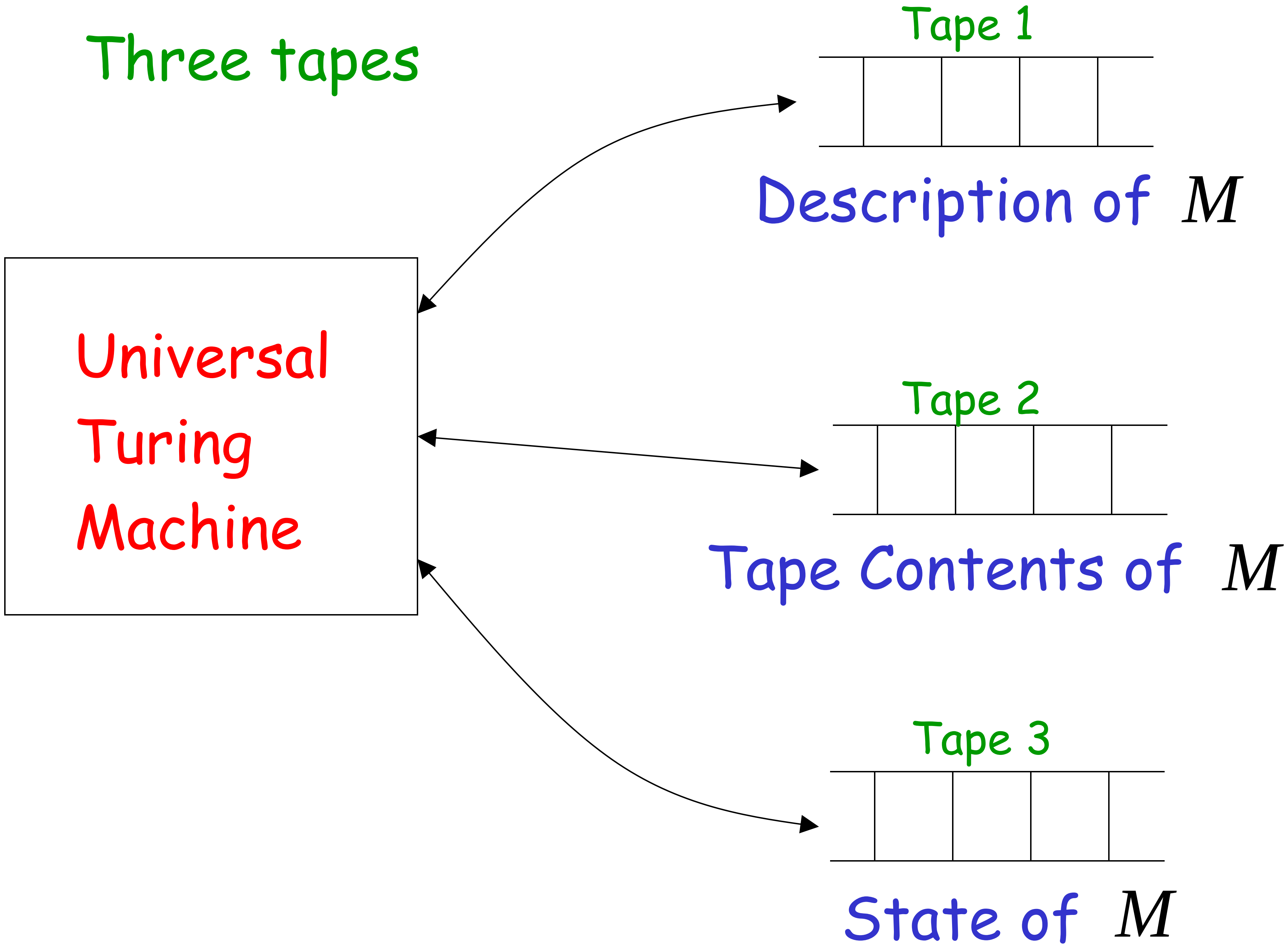
simulates any Turing Machine M

Input of Universal Turing Machine:

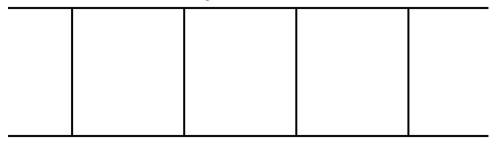
Description of transitions of M

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

$$\begin{tikzcd}
\text{Symbols:} & a \arrow[d] & b \arrow[d] & c \arrow[d] & d \arrow[d] & \cdots \\
\text{Encoding:} & 1 & 11 & 111 & 1111 &
\end{tikzcd}$$

State Encoding

$$\begin{tikzcd}
\text{States:} & q_1 \arrow[d] & q_2 \arrow[d] & q_3 \arrow[d] & q_4 \arrow[d] & \cdots \\
\text{Encoding:} & 1 & 11 & 111 & 1111 &
\end{tikzcd}$$

Head Move Encoding

$$\begin{tikzcd}
\text{Move:} & L \arrow[d] & R \arrow[d] \\
\text{Encoding:} & 1 & 11
\end{tikzcd}$$

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:

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

Encoding:

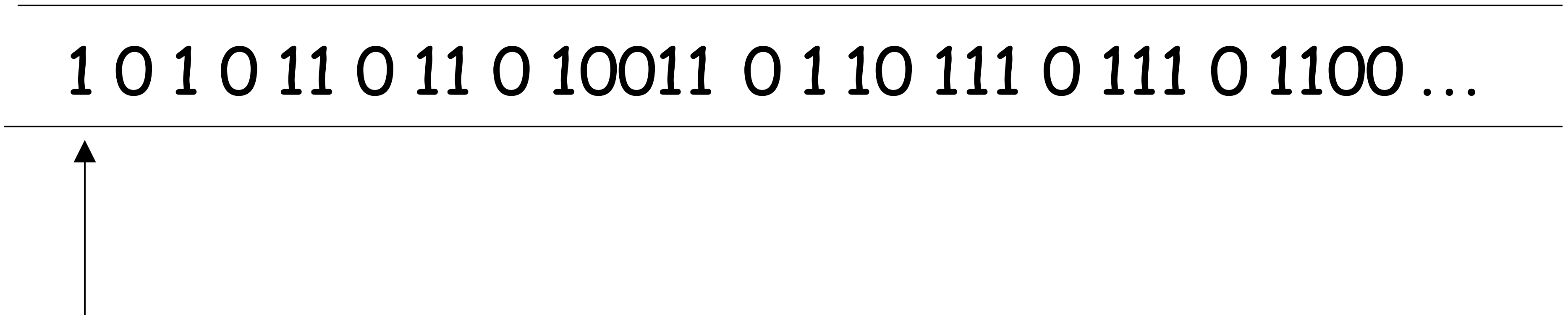
$1\,0\,1\,0\,11\,0\,11\,0\,1 \quad 00 \quad 11\,0\,11\,0\,111\,0\,111\,0\,11$

separator

Tape 1 contents of Universal Turing Machine:

binary encoding
of the simulated machine M

Tape 1



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

$$\begin{tikzcd}
\text{Even integers:} \atop \text{(positive)} & 0, \arrow[dd] & 2, \arrow[dd] & 4, \arrow[dd] & 6, \arrow[dd] & \dots \\
\text{Correspondence:} & & & & & \\
\text{Positive integers:} & 1, & 2, & 3, & 4, & \dots
\end{tikzcd}$$

$2n$ 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

$$\begin{tikzcd}[column sep=small]
\text{Rational numbers:} & \frac{1}{1}, \arrow[dd] & \frac{1}{2}, \arrow[dd] & \frac{1}{3}, \arrow[dd] & \dots \\
\text{Correspondence:} & & & & \\
\text{Positive integers:} & 1, & 2, & 3, & \dots
\end{tikzcd}$$

Doesn't work:

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

Better Approach

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

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

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

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

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

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

$$\begin{tikzcd}
\text{Rational Numbers:} & \frac{1}{1}, \arrow[dd] & \frac{1}{2}, \arrow[dd] & \frac{2}{1}, \arrow[dd] & \frac{1}{3}, \arrow[dd] & \frac{2}{2}, \arrow[dd] & \ldots \\
\text{Correspondence:} & & & & & & \\
\text{Positive Integers:} & 1, & 2, & 3, & 4, & 5, & \ldots
\end{tikzcd}$$

We proved:

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

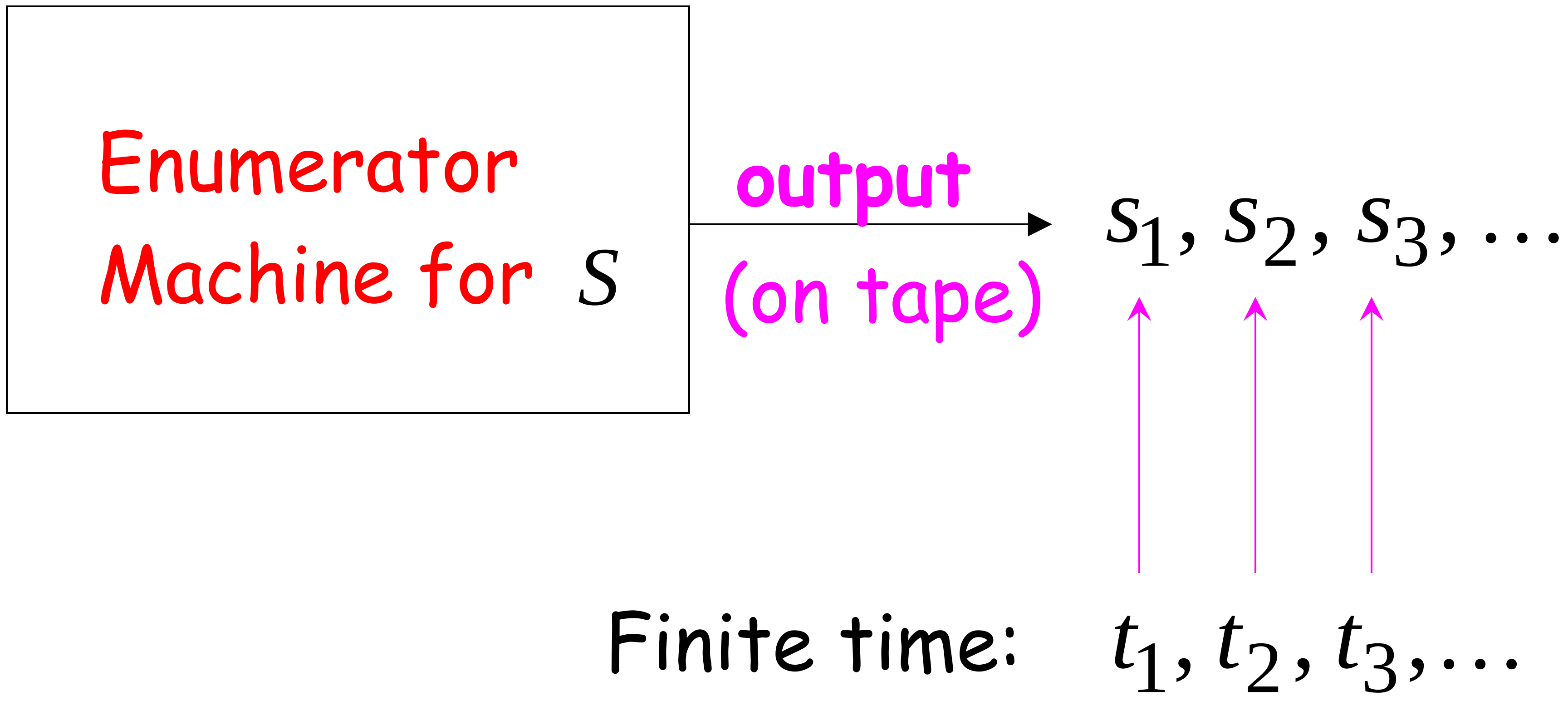
Definition

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

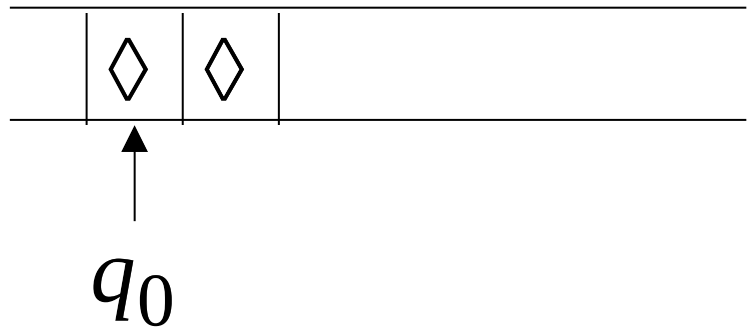
strings $s_1, s_2, s_3, \ldots \in S$



Enumerator Machine

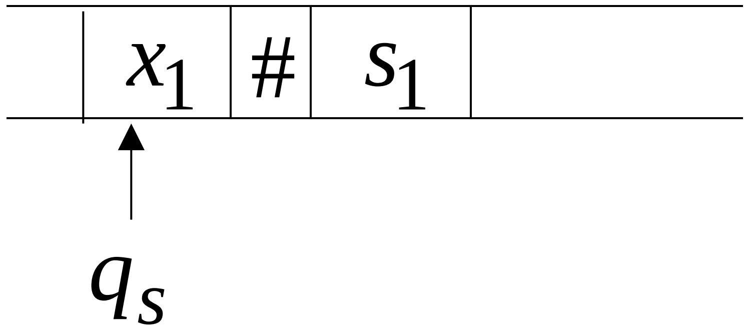
Configuration

Time 0

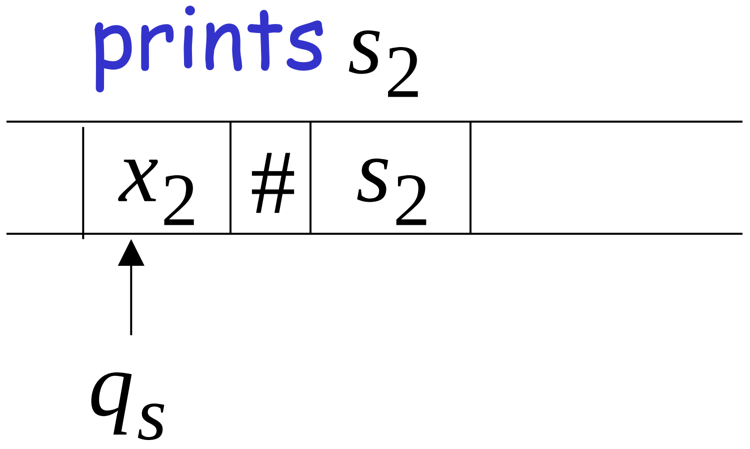


prints s_1

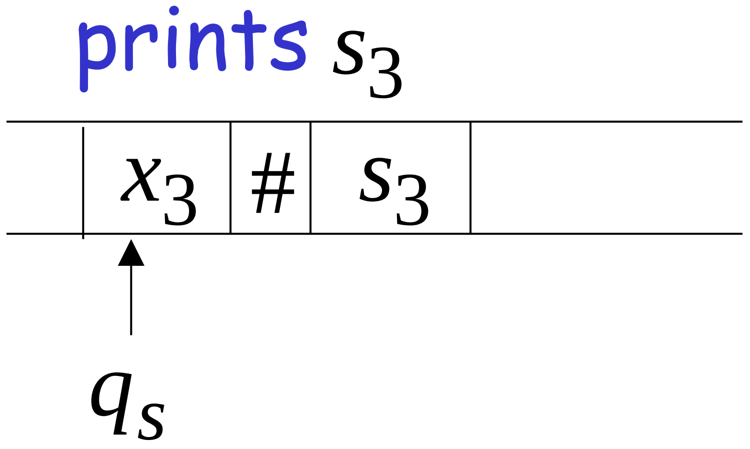
Time t_1



Time t_2



Time t_3



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:

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

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:

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

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

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

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

Enumerator:

Repeat

1. Generate the next binary string of 0's and 1's in proper order

2. Check if the string describes a Turing Machine

    if **YES:** print string on output tape

    if **NO:** ignore string

Binary strings Turing Machines

0

1

00

01

⋮

1 0 1 0 11 0 11 0 0

1 0 1 0 11 0 11 0 1 $\xrightarrow{s_1}$ 1 0 1 0 11 0 11 0 1

⋮

1 0 11 0 1010010101 101 $\xrightarrow{s_2}$ 1 0 11 0 1010010101 101

⋮

End of Proof

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)

Proof:

Since S is countable, we can write

$$S = \{\underset{\underset{\text{Elements of } S}{\uparrow}}{s_1}, s_2, s_3, \ldots\}$$

Elements of the powerset 2^S have the form:

$$\emptyset$$

$$\{s_1, s_3\}$$

$$\{s_5, s_7, s_9, s_{10}\}$$

……

We encode each element of the powerset with a binary string of 0's and 1's

Powerset element (in arbitrary order)	Binary encoding				
	s_1	s_2	s_3	s_4	$\cdots$
$\{s_1\}$	1	0	0	0	$\cdots$
$\{s_2, s_3\}$	0	1	1	0	$\cdots$
$\{s_1, s_3, s_4\}$	1	0	1	1	$\cdots$

Observation:

Every infinite binary string corresponds to an element of the powerset:

Example: $1\,0\,0\,1\,1\,1\,0 \quad \cdots$

Corresponds to: $\{s_1, s_4, s_5, s_6, \ldots\} \in 2^S$

Let's assume (for contradiction) that the powerset 2^S is countable

Then: we can enumerate the elements of the powerset

$$2^S = \{t_1, t_2, t_3, \ldots\}$$

Powerset element	suppose that this is the respective **Binary encoding**					
t_1	1	0	0	0	0	$\cdots$
t_2	1	1	0	0	0	$\cdots$
t_3	1	1	0	1	0	$\cdots$
t_4	1	1	0	0	1	$\cdots$
$\cdots$			$\cdots$			

Take the binary string whose bits are the complement of the diagonal

$$\begin{array}{lcccccc}
t_1 & \enclose{circle}{1} & 0 & 0 & 0 & 0 & \cdots \\
t_2 & 1 & \enclose{circle}{1} & 0 & 0 & 0 & \cdots \\
t_3 & 1 & 1 & \enclose{circle}{0} & 1 & 0 & \cdots \\
t_4 & 1 & 1 & 0 & \enclose{circle}{0} & 1 & \cdots
\end{array}$$

Binary string: $\boldsymbol{t} = 0011\ldots$

(binary complement of diagonal)

The binary string $$t = 0011\ldots$$

corresponds
to an element of
the powerset 2^S: $$t = \{s_3, s_4, \ldots\} \in 2^S$$

Thus, t must be equal to some t_i

$$t = t_i$$

However,

the i-th bit in the encoding of t is the complement of the i-th bit of t_i , thus:

$$t \neq t_i$$

Contradiction!!!

Since we have a contradiction:

The powerset 2^S of S is uncountable

End of proof

An Application: Languages

Consider Alphabet : $A = \{a, b\}$

The set of all Strings:

$$S = \{a, b\}^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

infinite and countable

(we can enumerate the strings in proper order)

Consider Alphabet : $A = \{a, b\}$

The set of all Strings:

$$S = \{a, b\}^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

infinite and countable

Any language is a subset of S :

$$L = \{aa, ab, aab\}$$

Consider Alphabet : $A = \{a, b\}$

The set of all Strings:

$$S = A^{*} = \{a, b\}^{*} = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$$

infinite and countable

The powerset of S contains all languages:

$$2^{S} = \{\varnothing, \{\lambda\}, \{a\}, \{a, b\}, \{aa, b\}, \ldots, \{aa, ab, aab\}, \ldots\}$$

uncountable

Consider Alphabet : $A = \{a, b\}$

countable

Turing machines:

Languages accepted By Turing Machines:

$$\begin{tikzcd} M_1 \arrow[d, "\text{accepts}"'] & M_2 \arrow[d] & M_3 \arrow[d] & \cdots \\ L_1 & L_2 & L_3 & \cdots \end{tikzcd}$$

countable

Denote: $X = \{L_1, L_2, L_3, \ldots\}$

countable

Note: $X \subseteq 2^S$

$(S = \{a, b\}^*)$

Languages accepted by Turing machines: X countable

All possible languages: 2^S uncountable

Therefore: $X \neq 2^S$

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

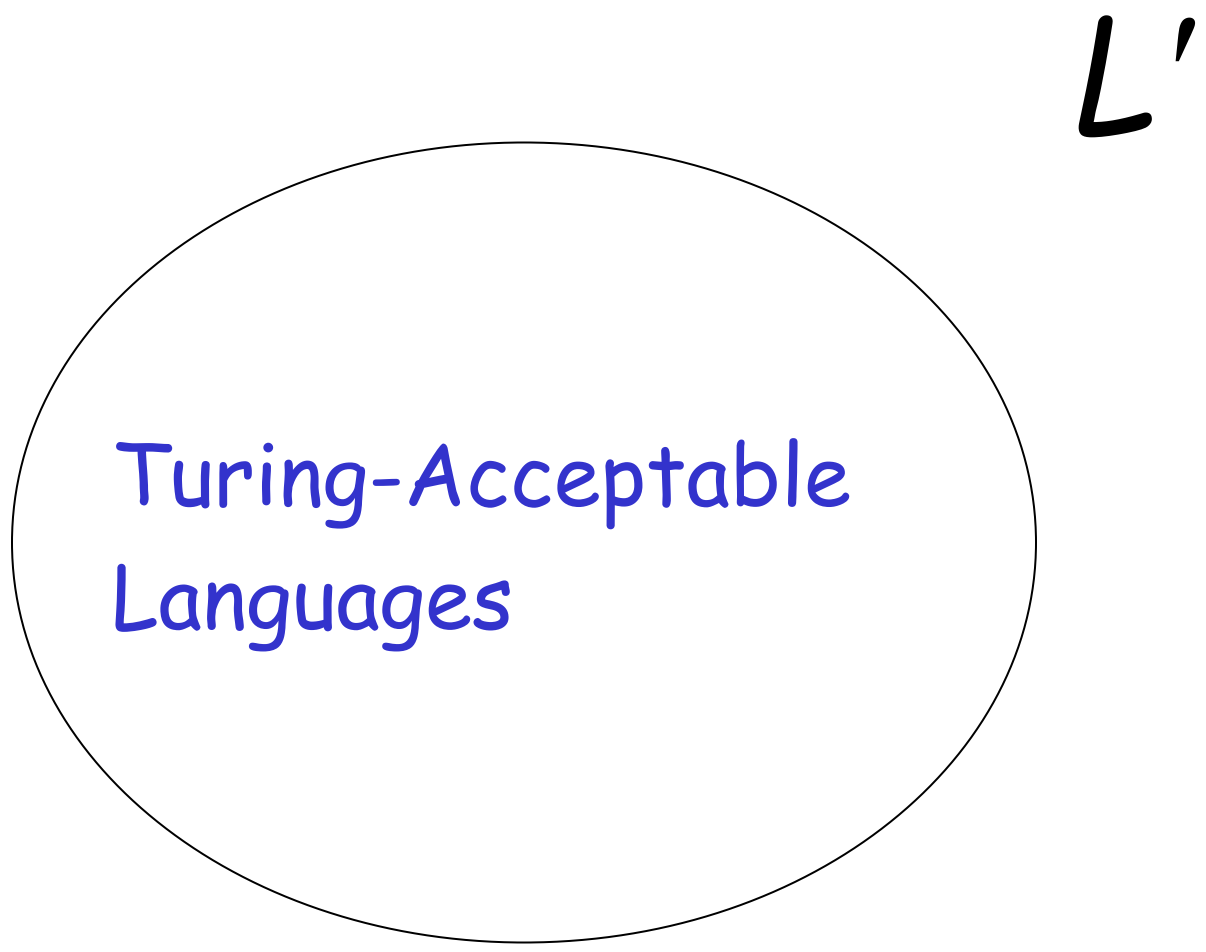
Conclusion:

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



Note that: $X = \{L_1, L_2, L_3, \ldots\}$

is a *multi-set* (elements may repeat)
since a language may be accepted
by more than one Turing machine

However, if we remove the repeated elements,
the resulting set is again countable since every element
still corresponds to a positive integer