Turing Machine: 7 tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

Turing Machines that compute functions

A function $f: \Sigma^* \to \Sigma^*$ is said to be (Turing) total computable if there is a Turing Machine M such that given input x at the start of the computation, the machine halts with the output $f(x)$, for all $x \in \Sigma^*$.

A function $f: \Sigma^* \dashrightarrow \Sigma^*$ is partial computable if $dom(f) \subseteq \Sigma^*$.

Similarly we can define $f: \mathbb{N} \to \mathbb{N}$ as total, & partial computable by picking an encoding:

Standard Enumeration of Binary Strings.

$$\lambda,\ \underbrace{0,\ 1},\ \underbrace{00,\ 01,\ 10,\ 11},\ 000,\ \dots,\ 111,\ 0000,\ \dots$$

$$\begin{array}{ccc} \lambda & 0 & 1 \\ | & | & | \\ 0 & 1 & 2 \end{array}$$

This is a bijection b/w $\mathbb{N}$ and Σ^*; denote it by $\underset{s_0}{\overset{\lambda}{\shortparallel}},\ \underset{s_1}{\overset{0}{\shortparallel}},\ \underset{s_2}{\overset{1}{\shortparallel}},\ \dots$

We say $f: \mathbb{N} \to \mathbb{N}$ is total computable if

$\exists M \quad \forall n \in \mathbb{N}$

$$s^{-1}(M(s_n)) = f(n).$$

$$\begin{array}{ccc} n & & n' \\ \downarrow & & \uparrow \\ s_n & \xrightarrow{M} & s_{n'} \end{array}$$

A language $L \subseteq \Sigma^*$ is said to be computably enumerable if it is either finite or $\exists$ a total computable bijection from $\mathbb{N}$ to L.

Note: Σ^* is countably infinite

Since $\forall$ language L, $L \subseteq \Sigma^*$, L is countable.

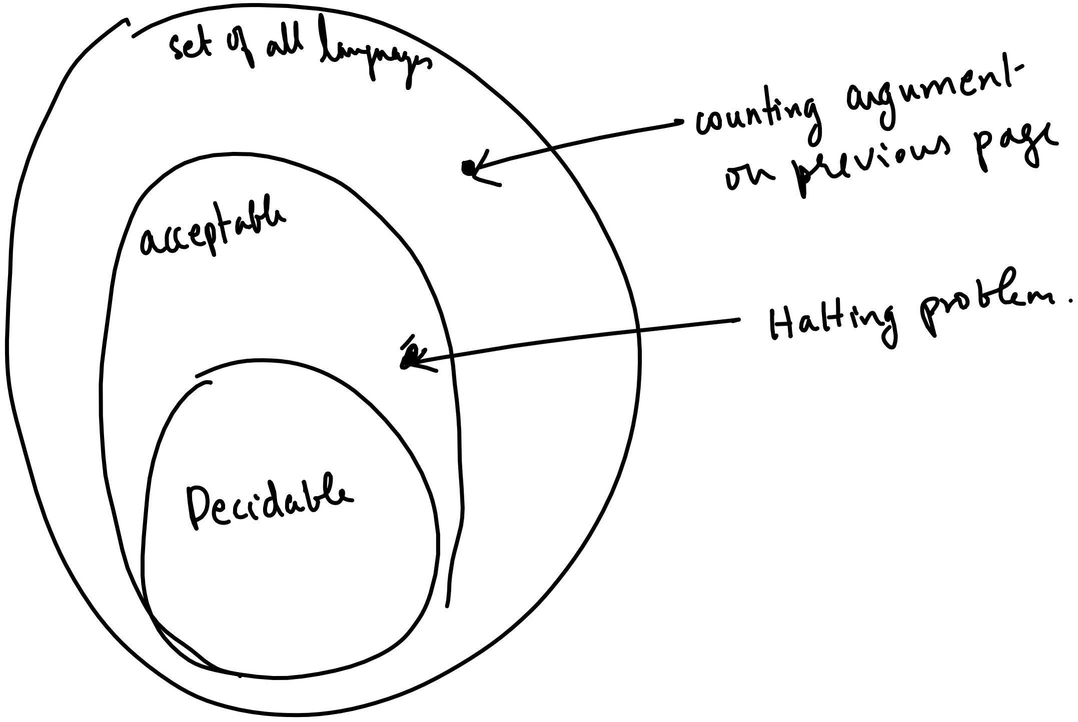
The set of all languages is $\mathcal{P}(\Sigma^*)$.

Hence there uncountably many languages.

But since every TM can be encoded as a finite binary string [see, eg Sipser] there are only countably infinite many Turing Machines.

$\Rightarrow$ $\exists$ countably infinitely many Turing acceptable languages.

$\Rightarrow$ $\exists$ a non Turing acceptable language.



Theorem.

For any language $L \subseteq \Sigma^*$, L is acceptable iff it is computably enumerable.

Proof

L is r.e. $\Rightarrow$ L is acceptable:

L is finite, then L is decidable, hence L is acceptable.

O/W let $M : \mathbb{N} \xrightarrow[\text{onto}]{1\text{-}1} L$ be a Turing machine computing a bijection from $\mathbb{N}$ to L.

We need to show L is acceptable. Consider: $M(0)$, $M(1)$, $M(2)$, ...

1. Input x.

2. for $i = 0$ to ...

   2.1 if $M(i) == x$, then accept x & halt

L is acceptable $\Rightarrow$ L is c.e.

If L is finite, then L is c.e.

So let L be infinite. We want a total computable bijection $f: \mathbb{N} \xrightarrow[\text{onto}]{1\text{-}1} L$.

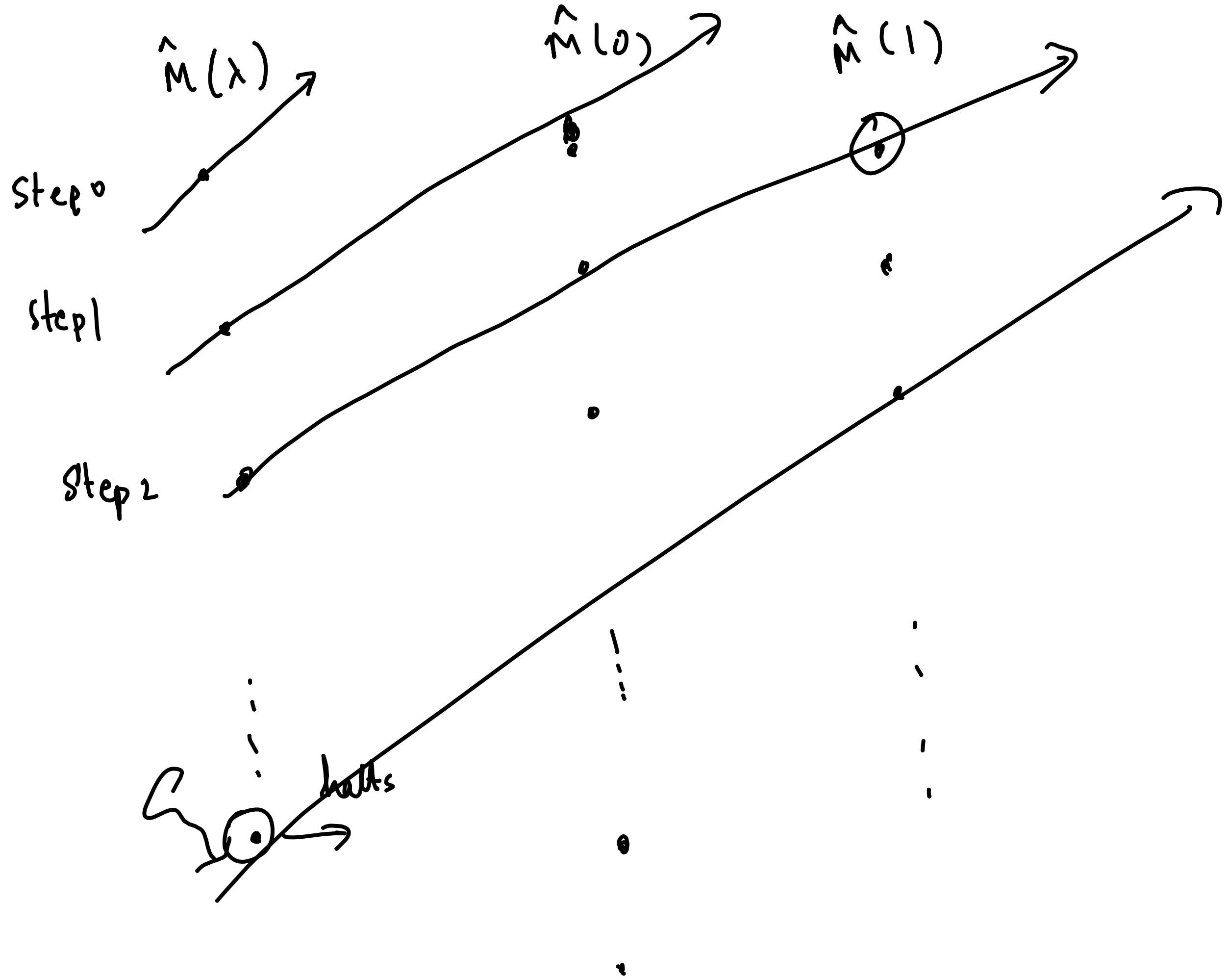
Let $\hat{M}$ be a TM that accepts L.

1^{st} idea: ask $\lambda \overset{?}{\in} L$ to $\hat{M}$

$0 \overset{?}{\in} L$ to $\hat{M}$

$\vdots$

Assume that we can simulate $\hat{M}$ for a single step at a time.


$\hat{M}(\lambda)$
$\hat{M}(0)$
$\hat{M}(1)$
Step 0
Step 1
Step 2
halts