Theorem: For every (known) high-level language (HLL) \( L \), and every program \( P \) written in \( L \), we can build a TM \( M_P \) such that for all (streamed) inputs \( x \), \( N_P(x) \) emulates \( P(x) \).

Proof: Compile \( P \) to the mini-assembler for the object code inside
the single TM \( M_P \) to get \( M_P \). Moreover, if \( P \) runs in \( O(n^k) \) time
then so does \( M_P \) (with exponent \( 2k + 3 \times \text{greater or so} \)).

Church Turing Thesis: 1. This will hold for any HLL that humans will ever devise. \( \checkmark \)
2. For every human or alien \( A \) that makes consistent decisions \( D(x) \) on inputs \( x \), there is a TM \( M_A \) that on input \( x \) outputs \( D(x) \).

Poly-Time CT Thesis
(Alan Cobham, Ted juda, 1965)

Moreover, no HLL will deliver super-polynomial speedup of TMs.

May be false if we allow quantum computing hardware.

<table>
<thead>
<tr>
<th>Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rec. def.</td>
<td>A language ( L ) is recursively enumerable (r.e.) if there is a TM ( M ) such that ( L(x) = A ) and ( \forall x \in M(x) )</td>
</tr>
</tbody>
</table>
| R. def. | A function \( f: \Sigma^* \to \Sigma^* \) is computable if there is a \( \text{TM} M \) that on any \( x \in \Sigma^* \) with \( \text{output} f(x) \). If \( M \) halts on input \( x \), then \( f(x) \) is defined; otherwise, \( f(x) \) is undefined. If on the other hand, \( f(x) \) is undefined, \( M \) is allowed to bubble it 

The class of all decidable languages is denoted by \( \text{REC} \) or \( \text{DEC} \).

The class of all c.e. languages is denoted (only) by \( \text{RE} \).

Missing part of definition at top is “Turing acceptable” as a synonym for the other three terms.
Theorem: A language $L$ is decidable if its characteristic function $L(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases}$ is total computable.

$L$ is r.e. $\iff$ its partial characteristic function $T_L(x) = \begin{cases} 1 & \text{if } x \in L \text{ is partial} \\ \text{undefined} & \text{if } x \notin L \text{ computable} \end{cases}$ is total computable.

Friday: Define $\frac{2}{3} \notin L$: $M$ does not accept $\langle M \rangle$ if $M$ is not Turing acceptable.

Church-Turing Thesis: 1. This will hold for any $M$ that humans will ever design.
   2. For every human or alien $A$ that makes consistent decisions in $\mathbb{N}$, $A$ will halt on input $\langle M \rangle$. $A$ is a universal Turing machine.

Polynomial Time (P) Thesis:

(Alan Cobham, 1965) 1. Any $M$ is total, which will allow super-polynomial work on $\mathbb{N}$.

Theorem: $\mathbb{N}$ is not $\mathbb{N}$-computable, which is an open problem. More generally, we can allow quantum computing hardware.