The Church - Turing Thesis

• What is it?

• Why should you care?
Computable Functions

Let $f : \mathbb{N}^k \rightarrow \mathbb{N}$ be a numeric function. A $\lambda$-term $M_f$ $\lambda$-defines $F$ if for every $(n_1, \ldots, n_k) \in \mathbb{N}^k$, with $f(n_1, \ldots, n_k) = m$,

$$M_f n_1 \ldots n_k \rightarrow m$$

The function $f$ is $\lambda$-definable if there exists a $\lambda$-term $M_f$ which defines $f$.

Here $\rightarrow$ denotes reduction.
Primitive Recursion

A set $E$ of numeric functions is closed under primitive recursion if for every $h, g \in E$, if $f$ is defined by

$$f(0, n) = g(n)$$

$$f(m + 1, n) = h(f(m, n), m, n)$$

then $f \in E$
Total minimization

A set $E$ of numeric functions is closed under total minimization if for every $g \in E$ such that for each $n$ there exists $p$ such that $g(n,p) = 0$, then if $f$ is defined by

$$f(n) = \min\{p \in N; g(n,p) = 0\}$$

then $f \in E$. 
Recursive Functions

The *recursive functions* are the elements of the least set that contains the *base functions*:

- The constant function \( \lambda : \mathbb{N} \rightarrow \mathbb{N} \);

- The successor \( s : \mathbb{N} \rightarrow \mathbb{N} \); and

- The projections \( pr^i_k : \mathbb{N} \rightarrow \mathbb{N} \), for \( 1 \leq i \leq k \).

and are closed under, composition, primitive recursion, and total minimization.
A Theorem!

For $f : \mathbb{N}^p \rightarrow \mathbb{N}$ the following properties are equivalent:

1. $f$ is $\lambda$-definable

2. $f$ is recursive
Turing Machines

A Turing Machine is specified by three finite sets $\Gamma, Q$ and $R$:

- $\Gamma$ is the tape alphabet, where $\Box$ (blank) $\notin \Gamma$;

- $Q$ is a set of states;

- $R$ is the set of rules, which are 5-tuples $(q, s, q', s', d)$ where $q, q' \in Q$, $s, s' \in \Gamma \cup \{\Box\}$, and $d \in [-,0,+]$.
Turing Machines 2

’Physically’ a TM consists of:

- An infinite tape divided into cells, each capable of holding a symbol from $\Gamma$ or a blank symbol

- A read/write head which moves at most one cell to the left or right at each step;

- A control unit which contains the ’program’ $R$ and which, at each step, can be found in any one of the states of $Q$.

A TM is deterministic (a DTM) if for every $(q, s) \in Q \times \Gamma \cup \{\square\}$, there exists at most one rule $(q, s, \ldots)$ in $R$. 
Transitions in A TM

In a Turing machine \((\Gamma, Q, R)\): if in a state \(q\), the head points to the symbol \(s \in \Gamma \cup \{\square\}\) and if \(\rho = (q, s, q', s', d) \in R\), then \(\rho\) triggers the following transition:

- the state becomes \(q'\);

- the head writes \(s'\) to replace \(s\) and then moves one cell to the left if \(d = -\) or to the right is \(d = +\).

A *halting* configuration is a configuration for which no rule applies.
A *computation* is a sequence of transitions which is either, infinite or finite and terminates in a halting configuration.
An example

The successor function \( s : \mathbb{N} \rightarrow \mathbb{N} \) is computed by the following DTM:

\[
\begin{align*}
\Gamma &= \{1\} \\
Q &= \{q_0, q_1\} \\
R &= \{(q_0, 1, q_0, 1, +), (q_0, \square, q_1, 0)\}
\end{align*}
\]
Another theorem

All recursive functions are Turing computable.
The Church-Turing Theses

Every effectively computable function is recursive.

If a function $f$ is computable by an algorithm, then this algorithm can be effectively transformed into a Turing machine computing $f$. 