2011-(09)sep-06
kplx

Last time

Church's thesis "right notion"

M_x TM corresponding to x

Universal TM

run M_x on y for t steps

Halting problem

Does M_x halt on y?

Noncomputable } same
Nonrecursive }

$$D_x = \begin{cases} 1 & M_x \text{ halts on } x \text{ with output } 0. \\ 0 & \end{cases}$$

M_{x_0} makes mistake on x_0

(and infinitly many more inputs).

Gödel:
Given any (nice) axiomatisation of number theory there are true theorems that are not provable.

Proof system

Axioms
Derivation rules $\frac{A \text{ true} \quad A \Rightarrow B}{B \text{ true}}$

Nice: Mechanically decidable if something is an axiom.
A derivation rule is correct.

Proof systems

Complete — A or $\neg A$ provable
Sound — Only true statements prov.

The complete, sound, nice
⇒ The set of true theorems is recursive (computable).

Given A = statement in number theory generate all proofs of increasing length until you find a proof of A or ¬A.

If the axiom is strong enough to express M_x halts on y we get a contradiction

---

Efficient computability

Computable in time 2^{1000} or not who cares?

Computable in time $T(n)$, if computed by TM that runs in $T(n)$ steps on inputs of length n.

Given a graph with m nodes and n edges is it connected.
$O(n+m)$ Depth first search.

Given a bipartite graph with m nodes an n edges
Is there a perfect matching?
$O(m^3)$ $O(nm)$

Question: Does more time allow you to solve more problems?

Can we have arbitrarily complicated problems?

Theorem: Given a function T there is a problem that requires $\geq$ time T to solve

Proof $\quad |x| =$ length of x

$$D_T(x) = \begin{cases} 1 & \text{if } M_x \text{ halts on } x \text{ within } T(|x|) \text{ steps with output } 0. \\ 0 & \text{otherwise} \end{cases}$$

Claim: You cannot compute D_T within time $T(n)$ on all inputs of length n

Proof of claim: Suppose M_{x_0} computes D_T in time $T(|X|)$ on all x look what happens on input x_0.

Is D_T computable in time T,

on input x

1. compute $T(|x|)=t$

2. Run M_x on input x for t steps (and do the obvious).

As long as T is computable so is D_T.

Assume further that $T(n)$ can be computed in $T(n)$ steps)

certainly true for n^2, 2^n, 2^{2^n}

... "time constructible".

Suppose T is time constructable then D_T is computable in $O(\quad)$ steps.

In the best of all worlds one could hope to do 2 in time $O(t)$.

How efficient is the universal machine.

If we have two-tape TMs.

D_T can be computed in time

$O(T \log T)$.

Cor:

Given time constructable T

$\exists$ function computable in time $O(T \log T)$ but not time T.