Decidability

Annihilation reduction

- Left over from last lecture
- Want to solve hamiltonian path using pebbles
- Add one vertex with 2 pebbles, connect to all others
- Put one pebble on all others

Pseudopolynomial time

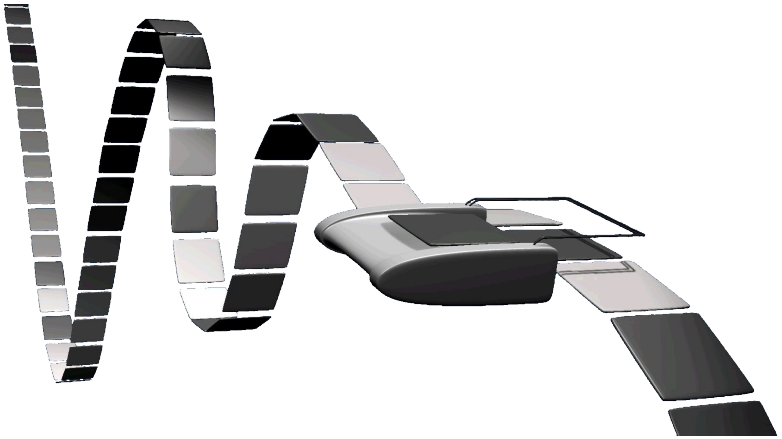
- Leftover from last lecture

Languages

- A language L is a set of strings
- Any sort of rules may be placed on those strings, like “The language of string representing even numbers”
- We phrase questions in terms of a language
- “Is x prime” is rephrased as “Is x in the language representing prime numbers”

Turing Machines

- Abstract machine
- Works on a semi-infinite tape
- Contains a FSM
- Pointer to a location on the tape
- Set of input and tape symbols



How do they work?

- TMs read a symbol and use it to transition between states
- Can end up in accept or reject states
- Example on board: Even unary numbers

Languages revisited

- Often we want to know if a turing machine can determine if a string is inside or outside of a language

- Problem: We dont know how long a turing machine might take

Recognition

- A Turing machine **recognizes** a language if it always correctly accepts after a finite amount of time
- We don’t care about strings that are not in the language yet.
- It can loop forever on a reject, but we only care about behavior on accepts
- If there exists a TM that recognizes the language, the language is recognizable
- Example, strings of finite length
- Related concept: co-recognizable. Recognizes the complement

Decision

- If the TM also rejects in a finite time if the string is not in the language then the language is **decidable**
- Alternate definition: L is decidable if it is recognizable and co-recognizable
- Many, many examples. Basic mathematics, shortest path, many algs we have discussed

Question

- Are all languages recognizable?

No.

Unrecognizability

- Some languages are impossible to recognize
- To prove recognizability it suffices to give a TM that solves the problem
- How to prove a language is undecidable?

Machines working on Machines

- As a TM is just a FSM working on tape, we can encode it as a string!
- For a TM M, we write the description of M as <M>
- Now we have languages like “The language of turing machines that accept the string 100”

ATM

- ATM: Problem of deciding if a TM accepts certain strings
- How do you make a TM that decides this?

ATM

- You can’t.
- There exists a turing machine that causes a paradox

ATM is undecidable

R:

On input <M,x>
Simulate M on x
Return the opposite

Assume we have a machine that decides ATM, A

A(R(R,x))

How to show more things are undecidable

- Similar to showing things are NP-Complete
- Try to solve problems using other problems

The Halting Problem

- Question:
- Is there a program that can detect infinite loops?
- 2 ways to prove
- Directly (on board)

Proof by mapping reduction

- Show that if you can solve the halting problem then you can solve ATM
- Design the TM (on board)
- This is a contradiction!

Summary

- Problems can be decidable, undecidable, recognizable, unrecognizable, co-recognizable, un-co-recognizable
- Proof can be done directly by setting up a paradox
- Or can be done by solving one problem with another