Harvard CS 121 and CSCI E-207

Lecture 15: Undecidability

Harry Lewis

November 3, 2009

- Reading: Sipser §4.2, §5.1.

Motivation

- Goal: to find an explicit undecidable language

- By the Church–Turing thesis, such a language has a membership problem that cannot be solved by any kind of algorithm

- We know such languages exist, by a counting argument.
  - Every decidable language is decided by a TM
  - There are only countably many TMs
  - There are uncountably many languages

$\therefore$ Most languages are not decidable (or even Turing-recognizable)

Is every Turing-recognizable set decidable?

This would be true if there were an algorithm to solve

The Acceptance Problem:

Given a TM M and an input w, does M accept input w?

Formally, $\mathrm{A}_{\mathrm{TM}} = \{\langle M, w\rangle : M \text{ accepts } w\}$.

Proposition: If A_{TM} is decidable, then every Turing-recognizable language is decidable.

"A_{TM} is the hardest Turing-recognizable language."

A simplifying detail: every string represents some TM

- Let Σ be the alphabet over which TMs are represented (that is, $\langle M \rangle \in \Sigma^*$ for any TM M)

- Let $w \in \Sigma^*$

- if $w = \langle M \rangle$ for some TM M then w represents M

- Otherwise w represents some fixed TM M_0 (say the simplest possible TM).

Thm: A_{TM} is not decidable

- Look at A_{TM} as a table answering every question:

	w_0	w_1	w_2	w_3	
M_0	Y	N	N	Y	
M_1	Y	Y	N	N	(WLOG assume
M_2	N	N	N	N	every string w_i
M_3	Y	Y	Y	Y	encodes a TM M_i)

- Entry matching (M_i, w_j) is Y iff M_i accepts w_j
- If A_{TM} were decidable, then so would be the diagonal D and its complement.
  - $D = \{w_i : M_i$ accepts $w_i\}$.
  - $\overline{D} = \{w_i : M_i$ does not accept $w_i\}$.
- But $\overline{D}$ differs from every row, i.e. it differs from every Turing-recognizable language. $\Rightarrow\Leftarrow$.

Unfolding the Diagonalization

- Suppose for contradiction that A_{TM} were decidable.

- Then there is a TM M^* that decides $\overline{D} = \{\langle M \rangle : M$ does not accept $\langle M \rangle\}$.
  - $M^*(\langle N \rangle)$ runs the decider for A_{TM} on $\langle N, \langle N \rangle\rangle$ and does the opposite.

- Run M^* on its own description $\langle M^* \rangle$.

- Does it accept?

  M^* accepts $\langle M^* \rangle$

  $\Leftrightarrow \langle M^* \rangle \in \overline{D}$

  $\Leftrightarrow M^*$ does not accept $\langle M^* \rangle$.

- Contradiction!

Alan Mathison Turing (1912-1954)

24 Years Old when he published *On computable numbers . . .*

Some More Undecidable Problems About TMs

- The Halting Problem: Given M and w, does M halt on input w?

  Proof:

  Suppose $\text{HALT}_{\text{TM}} = \{\langle M, w\rangle : M$ halts on $w\}$ were decided by some TM H.

  Then we could use H to decide A_{TM} as follows.

  On input $\langle M, w\rangle$,

  - Modify M so that whenever it is about to go into q_{reject}, it instead goes into an infinite loop. Call the resulting TM M'.
  - Run $H(\langle M', w\rangle)$ and do the same.

  Note that M' halts on w iff M accepts w, so this is indeed a decider for A_{TM}. $\Rightarrow\Leftarrow$.

Undecidable Problems, Continued

- For a certain fixed M_0:

  Given w, does M_0 halt on input w?

Undecidable Problems, Continued

- For a certain fixed M_0:

  Given w, does M_0 halt on input w?

What about:

- For a fixed M_0 *and* a fixed w_0, does M_0 halt on input w_0?

Further Undecidable Problems

- Given M, does M halt on the empty string?

  Proof by reduction:

  Suppose M_1 decided $\{ \langle M \rangle$: M halts on $\varepsilon \}$.
  Then M_1 could be used to decide HALT_{TM}:

  Given $\langle M, w \rangle$,

  Construct $\langle M_w \rangle$, where M_w is a TM that writes

  w on the empty tape and then runs M.

  Then run M_1 on input $\langle M_w \rangle$

  M_1 halts on $\langle M_w \rangle \Leftrightarrow M_w$ halts on $\varepsilon \Leftrightarrow M$ halts on w

  But HALT_{TM} is undecidable. $\Rightarrow\Leftarrow$

"Co-X"

- For any property X that a set might have, a set S is **co-X** iff $\overline{S}$ has property X.
- For example, a co-finite set of natural numbers is a set that is missing only a finite number of elements.
- A co-regular language is . . . ?
- A co-recursive language is . . . ?
- What about a co-CF language?
- Proved last time:
  - A language is recursive if and only if it is both r.e. and co-r.e.

Non-r.e. Languages

Theorem: The following languages are not r.e.:

- $\overline{\mathrm{A_{TM}}} = \{\langle M, w\rangle : M \text{ does not accept } w\}$
- $\overline{\mathrm{HALT_{TM}}} = \{\langle M, w\rangle : M \text{ does not halt on } w\}$
- $\overline{\mathrm{HALT}^{\varepsilon}_{\mathrm{TM}}} = \{\langle M\rangle : M \text{ does not halt on } \varepsilon\}$

Proof: If these languages were r.e., then $\mathrm{A_{TM}}$, $\mathrm{HALT_{TM}}$, and $\mathrm{HALT}^{\varepsilon}_{\mathrm{TM}}$ would be both r.e. and co-r.e. and hence recursive.

Is it possible to determine, given a TM M, whether M accepts a finite or infinite set?

- Let $A_{\text{finite}} = \{\langle M \rangle : L(M)$ is finite$\}$. Is A_{finite} recursive?

Is it possible to determine, given a TM M, whether M accepts a finite or infinite set?

- Let $A_{\text{finite}} = \{\langle M \rangle : L(M) \text{ is finite}\}$. Is A_{finite} recursive?

- Suppose M_2 decides A_{finite}. To decide A_{TM}, given $\langle M \rangle$ and $\langle w \rangle$, construct $\langle M_w^* \rangle$ so that
  - If M accepts w then M_w^* accepts its input, regardless of what it is, and
  - If M does not accept w then M_w^* runs forever.

- Then run M_2 on input $\langle M_w^* \rangle$.

- $L(M_w^*)$ is either Σ^* (and therefore infinite) or $\emptyset$ (and therefore finite) depending on whether or not M accepts w.