Lecture 19: Undecidability in Theory and Practice

PS5, Problem 5
Consider a one-tape Turing Machine that is identical to a regular Turing machine except the input may not be overwritten. That is, the symbol in any square that is non-blank in the initial configuration must never change. Otherwise, the machine may read and write to the rest of the tape with no constraints (beyond those that apply to a regular Turing Machine).

a. What is the set of languages that can be recognized by an unmodifiable-input TM?
b. Is $\text{HALT}_{\text{UTM}}$ decidable?

Nevertheless, $\text{HALT}_{\text{UTM}}$ is Undecidable
Prove by reducing $\text{HALT}_{\text{TM}}$ to $\text{HALT}_{\text{UTM}}$:

$\text{HALT}_{\text{TM}}(<M, w>) = \text{HALT}_{\text{UTM}}(<\text{MU}w, \varepsilon>)$ where $\text{MU}w$ = an unmodifiable-TM that ignores the input, writes a # on the tape, followed by $w$, then, simulates $M$ on the tape starting at the first square after the #, treating the # as if it is the left edge of the tape.

Impossibility of Copying

input (unmodifiable)

How can the TM keep track of which input square to copy next?

Option 1: Use the writable part of the tape.
Problem: can’t read it without losing head position
Option 2: Use the FSM states.
Problem: there is a finite number of them!

Hence, it is equivalent to a DFA $\Rightarrow$ regular languages

Computability in Theory and Practice

(Intellectual Computability Discussion on TV Video)
Lecture 19: Undecidability in Theory and Practice

Ali G Problem

- **Input:** a list of numbers (mostly 9s)
- **Output:** the product of the numbers

$L_{\text{ALG}} = \{ <k_0, k_1, \ldots, k_n, p> | \text{each } k_i \text{ represents a number and } p \text{ represents a number that is the product of all the } k_i s.\}$

Is $L_{\text{ALG}}$ decidable? Yes. It is easy to see a simple algorithm (e.g., elementary school multiplication) that solves it.

Can real computers solve it?

Ali G was Right!

- Theory assumes ideal computers:
  - Unlimited, perfect memory
  - Unlimited (finite) time
- Real computers have:
  - Limited memory, time, power outages, flaky programming languages, etc.
  - There are many decidable problems we cannot solve with real computer: the actual inputs do matter (in practice, but not in theory!)

The “Busy Beaver” Game

- Design a Turing Machine that:
  - Uses $k$ symbols (e.g., “0” and “1”)
  - Starts with a tape of all “0”s
  - Eventually halts (can’t run forever)
  - Has $n$ states (not counting $q_{\text{Accept}}$ and $q_{\text{Reject}}$)
- Goal is to run for as many steps as possible (before halting)
- 2-way infinite tape TM

Tibor Radó, 1962
Lecture 19: Undecidability in Theory and Practice

Busy Beaver: \( N = 1 \)

\( BB(1, 2) = 1 \)

Most steps a 1-state machine that halts can make

\[ \text{Input: 0} \quad \text{Write: 1} \quad \text{Move: \rightarrow} \]

\[ \text{Input: 1} \quad \text{Write: 1} \quad \text{Move: \rightarrow} \]

\[ \text{Input: 1} \quad \text{Write: 1} \quad \text{Move: \rightarrow} \]

\[ \text{Input: 1} \quad \text{Write: 1} \quad \text{Move: \rightarrow} \]

\[ \text{Input: 1} \quad \text{Write: 1} \quad \text{Move: \rightarrow} \]

\[ \text{Input: 1} \quad \text{Write: 1} \quad \text{Move: \rightarrow} \]

BB(2, 2) = ?

Step 2

Step 3

Step 4

Step 5
Lecture 19: Undecidability in Theory and Practice

A

Input: 0
Write: 1
Move: →

Input: 0
Write: 1
Move: ←

H

Start

BB(2, 2) ≥ 6

Halted

Input: 1
Write: 1
Move: /

Input: 1
Write: 1
Move: ←

Step 6

What is BB(6, 2)?

Busy Beaver Numbers

- BB(1) = 1
- BB(2) = 6
- BB(3) = 21
- BB(4) = 107
- BB(5) = Unknown!
- BB(6) > 10^{2879}

Best found before 2001, only 925 digits!

http://drb9.drb.insel.de/~heiner/BB/index.html

6-state machine found by Buntrock and Marxen, 2001

(1730 digits)

Best found before 2001, only 925 digits!
Is there a language problem?

\[ L_{BB} = \{ \langle n, k, s \rangle \mid \text{where } n \text{ and } k \text{ represent integers and } s \text{ is the maximum number of steps a TM with } n \text{ non-final states and } k \text{ tape symbols can run before halting} \} \]

Is \( L_{BB} \) Decidable?

\[ L_{BB} \text{ is Undecidable} \]

**Proof by reduction:**
Assume \( M_{BB} \) exists that decides \( L_{BB} \)

\[ HALT_{TM}(\langle M, w \rangle) = \]

\[ n = \text{number of states in } M \]
\[ k = \text{number of symbols in } M's \text{ tape alphabet} \]

find \( s \) by trying \( s = 1, 2, ... \) until \( M_{BB} \) accepts \( \langle n, k, s \rangle \)

simulate \( M \) on \( w \) for up to \( s \) steps

if it halts, accept

if it doesn’t complete, reject

Challenges

- The standard Busy Beaver problem is defined for a doubly-infinite tape TM. For the one-way infinite tape TM, what is \( BB(4, 2) \)?
- Find a new record BB number

Exam 2: one week from today.