Questions/Review

- Halting and other problems
- Reduction/equivalence

Unit B5: Overview

- Review (based on summary questions)
- Explore the notion of "computation" via the Church-Turing thesis
  - Discuss equivalence of different mechanisms [time permitting]
- Preview Module B Evaluation Workshop

Part 1

Review

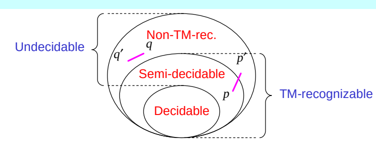
Language

- Formal definition
- TM and language
  - A TM "recognize" a language.
  - A TM "decides" a language.
- Language as a problem
- Countability
  - The set of all the strings
  - The set of all the languages

Examples?

Review

"Co"-X



- Definition
- Each case
- Proving the theorem: Decidable ⇔ TM-recognizable *and* co-TM-recognizable

Review

Halting Problem

- Subproof: $ACCEPT$ is undecidable.
  - General idea
  - Use of L_d
- Alternative: Not using L_d

Liar Paradox, Revisited

- "This sentence is false."
- Intuition behind this problem
  - The set of sentences are countable.
  - A sentence in this set can be indexed.
  - The use of "this" can be interpreted as the use of its own index (self reference).
  - The language must have a means for a sentence to refer to an arbitrary sentence (universality) including itself.

Correction

Reduction

Incorrect

- If $ACCEPT_{\mathrm{TM}}$ is reducible to $HALT_{\mathrm{TM}}$ and $HALT_{\mathrm{TM}}$ is decidable, $ACCEPT_{\mathrm{TM}}$ is also decidable.
- If $HALT_{\mathrm{TM}}$ is reducible $ACCEPT_{\mathrm{TM}}$ to and $HALT_{\mathrm{TM}}$ is undecidable, $ACCEPT_{\mathrm{TM}}$ is also undecidable.

Correct: reduction with respect to computability analysis

- If $ACCEPT_{\mathrm{comp}}$ is reducible to $HALT_{\mathrm{comp}}$ and $HALT_{\mathrm{TM}}$ is decidable, $ACCEPT_{\mathrm{TM}}$ is also decidable.
- If $HALT_{\mathrm{comp}}$ is reducible $ACCEPT_{\mathrm{comp}}$ to and $HALT_{\mathrm{TM}}$ is undecidable, $ACCEPT_{\mathrm{TM}}$ is also undecidable.

Correction

Reduction

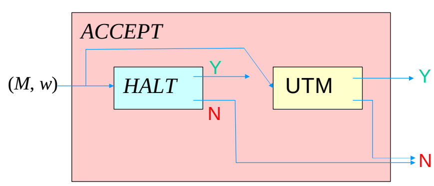
Incorrect

- $ACCEPT_{\mathrm{TM}}$ and $NACCEPT_{\mathrm{TM}}$ are equivalent.
- $ACCEPT_{\mathrm{TM}}$ and $HALT_{\mathrm{TM}}$ are equivalent.

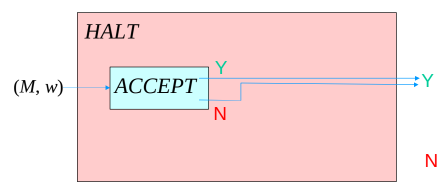
Correct

- $ACCEPT_{\mathrm{comp}}$ and $NACCEPT_{\mathrm{comp}}$ are equivalent.
- $ACCEPT_{\mathrm{comp}}$ and $HALT_{\mathrm{comp}}$ are equivalent.

$ACCEPT_{\mathrm{comp}}$ reduces to $HALT_{\mathrm{comp}}$



$HALT_{\mathrm{comp}}$ reduces to $ACCEPT_{\mathrm{comp}}$



Review

Rice's Theorem

- Non-trivial properties of languages/TMs are undecidable $\Leftarrow$ Uncountability
  - Note: Trivial properties (e.g., emptiness) that do not divide the entire set is trivially decidable.
- Potentially useful for Ex B5 (Comprehensive Ex)

Text Section 9.3.3

Other Questions/Comments

- General understanding
- Big picture
- Exam
- More "concrete" problems

Part 2

Church-Turing Thesis

- Hypothesis as definition
  Computation/effective procedure [informal]
  = TM operation [formal]
- A variety of models of computation are equivalent to TM.
- All sorts of (algorithmic) computation are created equal [and do the same thing, computation].

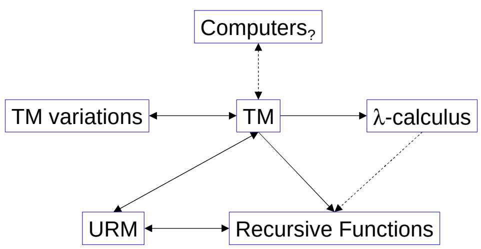
Take-home exercises

Connections

- TMs (including variants)
- Computers
  - Different programming paradigms
- Unlimited Register Machine (URM)
- Recursive functions
- λ-calculus [behind LISP]

Preview

Equivalence Summary



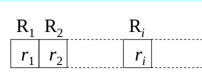
Proof of Equivalence

- Equivalence of models *M* and *N*
  - *N* simulates *M* and *M* simulates *N*
- Set identity, e.g., $A = B$
  - $A \subseteq B$ and $B \subseteq A$
- Equivalence of propositions, e.g., $p \leftrightarrow q$
  - $p \rightarrow q$ and $q \rightarrow p$
- Equality of values, e.g., $x = y$
  - $x \leq y$ and $y \leq x$

TM Variations

- Both accept and reject states (as well as other dying states)
- Semi-infinite tape
- Multiple tapes
- Nondeterministic transition
  - I.e., δ as a relation, not function

URM (1)



- Instructions:
  - Zero: $Z(i) \Rightarrow r_i \leftarrow 0$
  - Successor: $S(i) \Rightarrow r_i \leftarrow r_i + 1$
  - Transfer: $T(i, j) \Rightarrow R_j \leftarrow r_i$
  - Jump: $J(i, j, n) \Rightarrow$
    if $r_i = r_j$: proceed to the nth instruction
    otherwise: proceed to the next instruction

URM (2)

- Program: A finite sequence of instructions. Sequential execution of instructions except for possible jumps
- Initial values: Specific values can be assumed in the registers.
- Convention: The output in R_1.

Simulation by a TM?

Recursive Functions (1)

- Built up from the basic functions by a finite number of operations of substitution, recursion, and minimalization:
- Basic functions
  - Zero: 0
  - Successor: $x + 1$
  - Projection: $U^n_i\ (x_1, x_2, \ldots, x_n) = x_i$

Recursive Functions (2)

- Operations
  - Substitution (composition): Given computable functions $f(y1, \ldots, yk)$ and $g_1(x), \ldots, g_k(x)$, define $h(x) = f(g_1(x), \ldots, g_k(x))$
  - Recursion: Given computable functions $f(x)$ and $g(x, y, z)$, define
    - $h(x, 0) = f(x)$
    - $h(x, y + 1) = g(x, y, h(x, y))$
  - Minimalization: Given computable function $f(x, y)$, define $h(x)$ = the least y such that $f(x, y) = 0$

Simulation by a TM?

TM as Function

A TM can simulate a function. When accepts

- Function on strings
  - Input: A list of strings on the tape
  - Output: A string on the tape
- Function on natural numbers
  - Input: A list of natural numbers on the tape
  - Output: A natural number on the tape

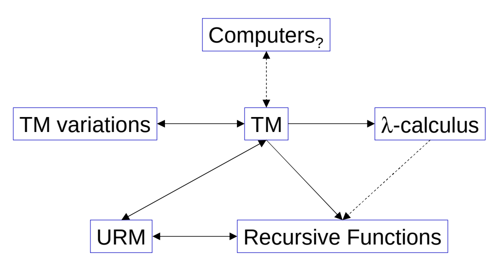
λ-calculus

Formal representation of functions

- Abstraction: $\lambda x.exp$
- Application: $exp_1\ exp_2$
- β-reduction: $(\lambda x.exp)\ y$
  - The result is exp where the instances of x is replaced with y.
  - E.g., $(\lambda x.f(f(x)))\ 2 = f(f(2))$ Is 2 a function?

In real life?

Equivalence Summary



TM vs. Computers

Equivalent?

Philosophy of Mind

- Hypothesis: Mind is a computer [Fodor: specifically uses TMs as the model].

Implications (if correct)?
Correctness?

Unit Summary

- Review
- Church-Turing thesis
  - Defines the informal notion of computation
  - Leads to the equivalence of various algorithmic computational approaches

Computability Summary

- Main goal: To be able to analyze the computability class and identify what we can do with algorithms [Comprehensive Ex]
- Topics
  - Problem as a set
  - Turing machines and (un)decidability
  - Mathematical tools: diagonalization, power set
  - Halting and other main problems; Rice's theorem
  - Reduction
  - Church-Turing thesis

Preview

Module B Evaluation Workshop

- Required materials (hardcopy) **print in advance**
  - Module B evaluation form
  - Supporting notes Significance of supp. notes?
  - Exercises
- Activities
  - Module review exercise
  - Peer discussion
  - Reflection and self-evaluation

Summary Question

- How far have you been thinking Exercise B5 (Comprehensive Exercise)? Explain.
- Questions/Comments/Suggestions