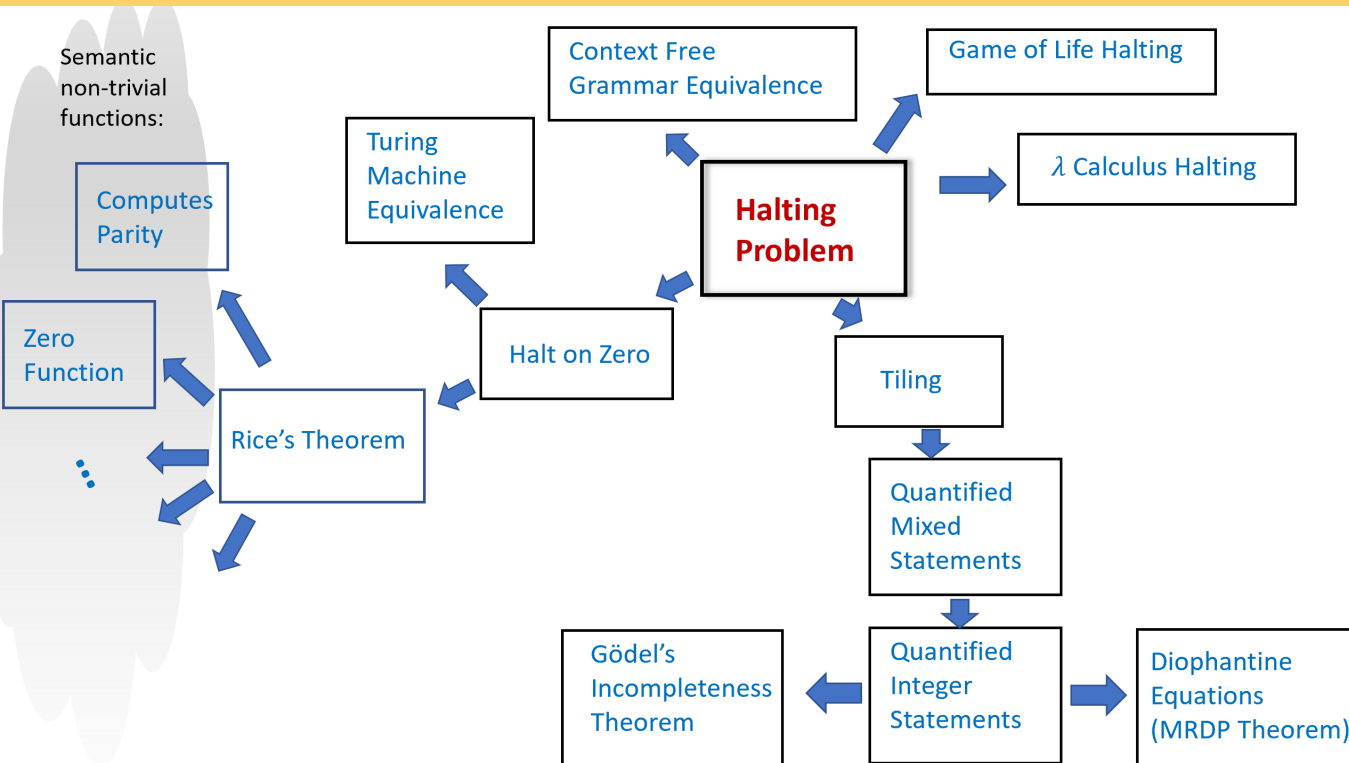


HW 7 due this Friday, Apr 3.

Quiz 10 due next Monday.



Class 21: Running Time

University of Virginia
CS3120: DMT2

<https://weikailin.github.io/cs3120-toc>

Wei-Kai Lin

Plan

Computability

Rice's Theorem

Busy Beaver

Complexity

Running time / space

Textbook [TCS] Section 9

https://introtcs.org/public/lec_08_uncomputability.html

Universality and uncomputability

Recap

Recap: Computable Functions

Definition (Definition 7.2):

For any Boolean function $F: \{0,1\}^* \rightarrow \{0,1\}^*$ and any Turing Machine M , we say M **computes** F iff for all $x \in \{0,1\}^*$, $M(x) = F(x)$.

We say F is **computable** if and only if there exists a Turing machine M such that computes F .

When the function F is Boolean (output 1 bit), we sometimes say F is **decidable**, which stresses both computable and Boolean output.

Recap: Two Uncomputable Functions

$$NSA(x) = \begin{cases} 0, & \text{if } TM_x(x) = 1 \\ 1, & \text{otherwise} \end{cases}$$

$$HALT(w, x) = \begin{cases} 1, & \text{if } TM_w \text{ terminates on } x \\ 0, & \text{otherwise} \end{cases}$$

$$HOZ(w) = \begin{cases} 1, & \text{if } TM_w \text{ terminates on ""} \\ 0, & \text{otherwise} \end{cases}$$

Theorem (9.5): NSA is uncomputable.

Theorem (9.6): HALT is uncomputable.

Theorem: HOZ is uncomputable.

Reduction:

If HALT is computable,
then NSA is computable.

Reduction:

If HOZ is computable, then
HALT is computable.

**Uncomputability:
How often does it happen?**

What can we ask about TMs?

Does the machine **behave** in a certain way?

- Does it write "3" during its execution?
- Does it run for 15 steps on input x?

Questions that could have different answers for different machines which compute the same function/language

Does the machine **compute** a certain kind of function/language?

- Does the machine accept only finitely many inputs?
- Could the function of this machine also be computed by a finite state automaton?

Questions that will always have the same answer for machines which compute the same function/language

Call these **Semantic Properties**

Definition: Semantic Property

Definition 9.14:

Two machines M_1, M_2 are **functionally equivalent** (denoted $M_1 \equiv M_2$) if $\forall x \in \{0,1\}^*, M_1(x) = M_2(x)$

A function $F: \{0,1\}^* \rightarrow \{0,1\}$, defined on Turing machines, is **semantic** if for every pair of functionally equivalent TMs (M_1, M_2) , $F(M_1) = F(M_2)$.

Namely, this is a property of the **function/language** of the machine, not of the **behavior** of the machines

Examples of Semantic properties

What's $F(M)$?

Is F a semantic property?

Does the machine **behave** in a certain way?

- Does it write "3" during its execution?
- Does it run for 15 steps on input x ?
- (questions that could have different answers for different machines which compute the same function/language)

Does the machine **compute** a certain kind of function/language?

- Does the machine accept only finitely many inputs?
- Could the function of this machine also be computed by a finite state automaton?
- (questions that will always have the same answer for machines which compute the same **function**/language)

Rice's Theorem (9.15)

If F is a semantic property, then either F is **uncomputable**, or F is trivial.

By definition, F is trivial iff for all Turing machine M ,

$$F(M) = 0$$

$$F(M) = 1$$

easy

Does Rice's theorem apply?

- Machine M decides prime numbers

$F(M) = \begin{cases} 1 & \text{if } M(n)=1 \text{ iff } n \text{ prime} \\ 0 & \text{o.w.} \end{cases}$

Yes

*any $M_1 \equiv M_2$
 $F(M_1) = 1 \iff F(M_2) = 1$*
- " M on input 0 reads more than 10 memory cells on its tape."

$F(M) = \begin{cases} 1 & \text{if "10 cells"---} \\ 0 & \text{o.w.} \end{cases}$

No

*$M_1(0)$ read 1 cell output same
 $M_2(0)$ 11 cells :*
- M always "outputs a TM N that has property 2."

$F(M) = \begin{cases} 1 & \text{if "..." } \boxed{\text{output}} \text{ TM } N \dots \\ 0 & \text{o.w.} \end{cases}$

Yes

$M_1 \equiv M_2$
- M accepts TMs that have property 2

Yes

Does Rice's theorem apply, in real world?

1. Python program P that halts on all input. Yes
2. P sends "hi" to a network server Yes if net msg is defn as output
3. P erases the first file Yes
4. P has no loop No

Full Proof of Rice's Theorem

Suppose F is a non-trivial property of TMs:

$$M_1 \equiv M_2 \rightarrow F(M_1) = F(M_2) \in \{0,1\}$$

Exists P, Q such that $F(P) = 0$ and $F(Q) = 1$.

Assume for contradiction, M_F computes F . Let P be the machine that does not halt for all inputs. Let Q be a machine such that $F(Q) \neq F(P)$; Q exists by non-trivial F . Assume w.t.o.g. $F(P) = 0$ so $F(Q) = 1$ (if not, swap P and Q).

We construct M_H that computes HALT.

Why M_H computes HALT?

$M_H(w, x)$:

1. Construct the following machine M'

$M'(z)$:

a. Run universal TM $U(w, x)$

b. Run and output $Q(z)$

2. Let w' be the string representing M'

3. Run and output $M_F(w')$

Full Proof of Rice's Theorem

Suppose F is a non-trivial property of TMs:

$$M_1 \equiv M_2 \rightarrow F(M_1) = F(M_2) \in \{0,1\}$$

Exists P, Q such that $F(P) = 0$ and $F(Q) = 1$

Assume for contradiction, M_F computes F . Let P be the machine that does not halt for all inputs. Let Q be a machine such that $F(Q) \neq F(P)$; Q exists by non-trivial F . **Assume w.l.o.g. $F(P) = 0$ so $F(Q) = 1$ (if not, swap P and Q).**

We construct M_H that computes HALT.

If (w, x) halts: $M' \equiv Q$

$$M_H(w, x) = M_F(w') = 1$$

Otherwise: $M' \equiv P$

$$M_H(w, x) = M_F(w') = 0$$

$M_H(w, x)$:

1. Construct the following machine M'

$M'(z)$:

a. Run universal TM $U(w, x)$

b. Run and output $Q(z)$

2. Let w' be the string representing M'

3. Run and output $M_F(w')$

This concludes the proof of Rice's Theorem. (The textbook does Not include this proof.)

Running Time and Busy Beavers

Tape

1 10001000

Head

Current state

halt

Halted.

Steps

215

Turing machine program

```

1 ; Binary addition - adds two binary numbers
2 ; Input: two binary numbers, separated by a single space, eg '100 1110'
3
4 0 _ _ r 1
5 0 * * r 0
6 1 _ _ l 2
7 1 * * r 1
8 2 0 _ l 3x
9 2 1 _ l 3y
10 2 _ _ l 7
11 3x _ _ l 4x
12 3x * * l 3x
13 3y _ _ l 4y
14 3y * * l 3y
15 4x 0 x r 0
16 4x 1 y r 0
17 4x _ x r 0
18 4x * * l 4x ; skip the x/y's
19 4y 0 1 * 5
20 4y 1 0 1 4y
21 4y _ 1 * 5
22 4y * * l 4y ; skip the x/y's
23 5 x x l 6
24 5 y y l 6

```

Controls

Run Run at full speed

Pause

Step

Reset

Undo

Initial input: 11011001 10101111

[Advanced options](#)

Running Time of a TM M on input x:
The number of steps such that M(x) reaches
the halt state.

Busy Beaver

$BB(n)$: the TM such that among all n -state Turing machines that halt on no input, $BB(n)$ runs longest

A Turing Machine, is defined by (Σ, C, δ) :

$C \in \mathbb{N}$: a finite number of states

Σ : alphabet – finite set of symbols

$\Sigma \supseteq \{0, 1, \triangleright, \emptyset\}$

δ : transition function

$\delta: [C] \times \Sigma \rightarrow [C] \times \Sigma \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}, \mathbf{H}\}$

Any Turing Machine (Σ, n, δ)
consists of n state.

How many TMs consists of n states?

A Turing Machine, is defined by (Σ, C, δ) :

$C \in \mathbb{N}$: a finite number of states

Σ : alphabet – finite set of symbols

$\Sigma \supseteq \{0, 1, \triangleright, \emptyset\}$

δ : transition function

$\delta: [C] \times \Sigma \rightarrow [C] \times \Sigma \times \{L, R, S, H\}$

Any Turing Machine (Σ, n, δ)
consists of n state.

$4 \times 15 \times 4$ $\leftarrow 4 \times 15$
4 \leftarrow e.g. $n=15$

Busy Beaver

$BB(n)$: the TM such that among all n -state Turing machines that halt on no input, $BB(n)$ runs longest

$n = 15$

finite in \mathbb{N}

Huh?

Finitely many

Is $BB(n)$ computable?

$BB:\mathbb{N} \rightarrow \mathbb{N}$, a function. Alternative, $BB:\{0,1\}^* \rightarrow \{0,1\}^*$.

Machine loaded and ready

Current state: 0 Steps: 0

Turing machine program

| | | | | | | |
|------|----|---|---|---|---|------|
| Next | 1 | 0 | _ | 1 | r | B |
| | 2 | 0 | 1 | 1 | l | C |
| | 3 | B | _ | 1 | r | C |
| | 4 | B | 1 | 1 | r | B |
| | 5 | C | _ | 1 | r | D |
| | 6 | C | 1 | _ | l | E |
| | 7 | D | _ | 1 | l | 0 |
| | 8 | D | 1 | 1 | l | D |
| | 9 | E | _ | 1 | r | halt |
| | 10 | E | 1 | _ | l | 0 |
| | 11 | | | | | |

Controls: Run Run at full speed, Pause, Step, Reset, Undo, Initial input: _____, [Advanced options](#), [Load an example program](#), [Save to the cloud](#)

<http://morphett.info/turing/turing.html?94a288fb9c40906d7e6face4bc422ece>
<https://bbchallenge.org/1RB1LC> 1RC1RB 1RD0LE 1LA1LD 1RZOLA



Today, the team declared victory. They've finally verified the true value of a number called $BB(5)$, which quantifies just how busy that fifth beaver is. They obtained the result — **47,176,870** — using a piece of software called the Coq proof assistant, which certifies that mathematical proofs are free of errors.

Logistics

- Another Mead food/drink?
- Next HW?

HW 7 due this Friday.

Quiz 10 due next Monday.

Plan

Complexity

Problems not in P

Textbook [TCS] Section 12

https://introtcs.org/public/lec_10_efficient_alg.html

Efficient computation