Lecture #15: Universal Turing Machines and First Hard Problems

What To Do Before the Lecture

1. Complete the preparatory viewing or reading for Lecture #15 — nothing that each video will probably be understandable if you play it at double speed.
2. ***Print*** and read through the rest of this document and — if you have time — try to solve the problems!
3. The supplemental document is for interest only. It includes an outline of a proof that the language L_{TM} and $L_{\text{TM+I}}$ are decidable.

Problems To Be Solved

Proving That A_{TM} is Undecidable

Claim. *The language A_{TM} is undecidable.*

How To Prove This:

On input $\mu \in \Sigma_{\mathsf{TM}}^{\star}$:

1. `if` $(\mu \in L_{\mathsf{TM}})$ `{`

   Let $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ be the Turing machine that is encoded by μ.

2. `if` $(\Sigma = \Sigma_{\mathsf{TM}})$ `{`

3. `if` (M accepts μ) `{` // *Does M accepts its own encoding?*

4. `reject` μ

   `} else {`

5. `accept` μ

   `}`

   `} else }`

6. `reject` μ

   `}`

   `} else {`

7. `reject` μ

   `}`

Figure 1: Algorithm Used to Get a Contradiction

Details of Proof:

Suppose, to obtain a contradiction, that the language A_{TM} is decidable. Consider the algorithm shown in Figure 1.

Subclaim: There exists a Turing machine M_D that implements the algorithm in Figure 1.

Explanation:

It follows that there exists a string $\mu_D \in \Sigma_{\mathsf{TM}}^{\star}$ that encodes the Turing machine M_D.

What Happens When M_D is Executed on its Encoding, μ_D?

Proving that A_{TM}^C is Unrecognizable — If Time Permits

Claim. *The language A_{TM}^C is unrecognizable.*

How To Prove This:

Details of Proof:

Suppose to obtain a contradiction, that A_{TM}^C is recognizable.

- Since A_{TM} is recognizable (because it is the language of a universal Turing machine) there exists a (standard) Turing machine M_{Yes} such that $L(M) = A_{\mathsf{TM}}$.
- Since A_{TM}^C is also recognizable (by the above assumption) there is also a (standard) Turing machine M_{No} such that $L(M) = A_{\mathsf{TM}}^C$.

On input $\mu \in \Sigma_{\mathsf{TM}}^{\star}$:

1. `integer` $i := 1$
2. `while (true) {`
3.   `if` (M_{Yes} accepts μ after taking at most i steps) `{`
4.   `accept` μ
5.   `} else if` (M_{Yes} rejects μ after taking at most i steps) `{`
6.   `reject` μ

  `}`

7.   `if` (M_{No} accepts μ after taking at most i steps) `{`
8.   `reject` μ
9.   `} else if` (M_{No} rejects μ after taking at most i steps) `{`
10.   `accept` μ

  `}`

11. $i := i + 1$

`}`

Figure 2: Another Algorithm Used to Get a Contradiction

Consider the algorithm shown in Figure 2.

What Happens When This Algorithm is Executed on μ, when $\mu \in A_{\mathsf{TM}}$?

What Happens When This Algorithm is Executed on μ, when $\mu \notin A_{\textsf{TM}}$?

What Can We Conclude From This?