1 The Halting Problem and Unsolvability

Here is a way to present unsolvability that differs from the text. See handout 9 also for this subject.

Let Δ be $\{encode(M) : M \text{ loops on input } encode(M)\}$. Thus Δ involves Turing machines running on their own descriptions.

Theorem 1.1 (Unsolvability) *Δ is not recursively enumerable.*

Proof: Suppose Δ were recursively enumerable (partially decidable).

Then there is a Turing machine T that partially decides Δ.

Then for all M, T halts on input $encode(M)$ iff $encode(M) \in \Delta$.

Thus T halts on $encode(M)$ iff M loops on $encode(M)$.

Question: What does T do on input $encode(T)$?

T halts on $encode(T)$ iff T loops on $encode(T)$.

Contradiction.

Thus T does not exist, and Δ is not recursively enumerable.

This argument is related to the paradox,

> In a certain village, the barber shaves everyone who does not shave himself. Then who shaves the barber?

Points to remember about this proof:

1. Which Turing machine is T? What properties does it have? (It partially decides Δ.)
2. What is the input to T? (It's own description)
3. Derive a contradiction

What if Turing machine M cannot run on input $encode(M)$ because M has too few symbols? We can say then by convention that M just halts right away if it encounters an unknown symbol. Of course, the encoding can be converted to a binary string, so that any Turing machine with at least two non-blank tape symbols can read it.

1.1 The Halting Problem

Consider $\overline{\Delta} = \{encode(M) : M \text{ halts on input } encode(M)\}$.

Then $\overline{\Delta}$ is not decidable, else Δ would also be decidable, but Δ is not even partially decidable.

Let H be $\{encode(M)encode(w) : M \text{ halts on input } w\}$. H is more general than $\overline{\Delta}$, so if H were decidable, $\overline{\Delta}$ would be also.

Thus H is not decidable. This is the *unsolvability of the Halting problem.*

Because the halting problem is not solvable on a Turing machine, it is not solvable on any computer, or by any algorithm, given the Church-Turing thesis. Many other unsolvability results are derived starting from the ones given here.

1.2 Limitations on our ability to reason

The unsolvabilty of the halting problem also implies limitations on our ability to reason; particularly, to prove the non-halting of Turing machines. See handouts 10 and 11 for a discussion of this subject. The idea is that in any reasonable system of logic, there will be Turing machines T and inputs x such that T loops on input x, but this fact cannot be proven in the logic. If the Turing machine gets into a tight loop, this can be detected, but Turing machines can loop in very subtle ways that our most advanced logics cannot detect. In fact, for any reasonable system of logic, there is a systematic way to construct a specific Turing machine T and a specific input x to T such that T loops on input x but this cannot be proved in the logic. So no matter how advanced our logics become, there will always be Turing machines that are too subtle for them to fully understand.