More undecidable problems

Meenakshi D'Souza

International Institute of Information Technology
Bangalore

Term II 2022-23

More decidable/undecidable problems

Problem (a)

Is it decidable whether a given Turing machine has at least 481 states?

More decidable/undecidable problems

Problem (a)

Is it decidable whether a given Turing machine has at least 481 states?

Yes, it is.

We can give a TM N which given $enc(M)$

- Counts the number of states in M upto 481.
- Accepts if it reaches 481, rejects otherwise.

More decidable/undecidable problems

Problem (b)

Is it decidable whether a given Turing machine takes more than 481 steps on input ϵ without halting?

More decidable/undecidable problems

Problem (b)

Is it decidable whether a given Turing machine takes more than 481 steps on input ϵ without halting?

Yes, it is.

We can give a TM N which given $enc(M)$

- Uses 4 tapes: On the 4th tape it writes 481 0's.
- Uses the first 3 tapes to simulate M on input ϵ, like the universal TM U.
- Blanks out a 0 from 4th tape for each 1-step simulation done by U.
- Rejects if M halts before all 0's are blanked out on 4th tape, accepts otherwise.

More decidable/undecidable problems

Problem (c)

Is it decidable whether a given Turing machine takes more than 481 steps on *some* input without halting?

More decidable/undecidable problems

Problem (c)

Is it decidable whether a given Turing machine takes more than 481 steps on *some* input without halting?

Yes, it is.

Check if M runs for more that 481 steps on each input x of length upto 481. If so accept, else reject. Note: In 481 steps, it can read at most the first 481 symbols of the input.

	1	2	3		·······		481	482			
$\vdash$	a	a	b	a	b	a	a	a	$\flat$	$\flat$	······

More decidable/undecidable problems

Problem (d)

Is it decidable whether a given Turing machine takes more than 481 steps on *all* inputs without halting?

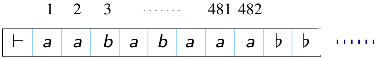
More decidable/undecidable problems

Problem (d)

Is it decidable whether a given Turing machine takes more than 481 steps on *all* inputs without halting?

Yes, it is.

Check if M runs for more that 481 steps on each input x of length upto 481. If so accept, else reject. Note: If M takes more than 481 steps on all inputs of length at most 481, then it will take more than 481 steps on all inputs.



More decidable/undecidable problems

Problem (e)

Is it decidable whether a given Turing machine moves its head more than 481 cells away from the left-end marker, on input ϵ?

More decidable/undecidable problems

Problem (e)

Is it decidable whether a given Turing machine moves its head more than 481 cells away from the left-end marker, on input ϵ?

Yes, it is.

Simulate M on ϵ for upto $m^{481} \cdot 482 \cdot k$ steps.

If M has k states and m tape symbols, and never moves more than 481 tape cells away from the left endmarker, then there are only $m^{481} \cdot 482 \cdot k$ configurations it could possibly ever be in, one for each choice of tape contents that fit within 481 tape cells, head position and state. If it runs for any longer, it must be in a loop, because it must have repeated a configuration.

If M visits the 483rd cell, declare it to be in a loop, else accept.

	1	2	3	·······			481	482			
$\vdash$	a	a	b	a	b	a	a	a	$\flat$	$\flat$	······

More decidable/undecidable problems

Problem (f)

Is it decidable whether a given Turing machine accepts the null-string ϵ?

More decidable/undecidable problems

Problem (f)

Is it decidable whether a given Turing machine accepts the null-string ϵ?

No.
If it were decidable, then, we could use the machine that decides it to decide HP as follows: Define a new machine M' which given input y, outputs the description of a machine M' which:

1. Erases its input y
2. Writes x on its input tape
3. Behaves like M on x
4. Accepts if M halts on x.

- Note that M' does the same thing on all inputs y.
- If M halts on x, then M' accepts its input y and if M does not halt on x, then M' does not halt on y and therefore does not accept y.
- Thus,

$$L(M') = \begin{cases} \Sigma^* & \text{if } M \text{ halts on } x \\ \emptyset & \text{if } M \text{ does not halt on } x \end{cases}$$

- Now, if we could decide whether a given machine accepts ϵ, we could apply this decision procedure to M' just constructed and this would tell whether M halts on x.

Reducing halting problem

- Given M and x, construct M', then ask whether M' accepts ϵ.
- The answer to the latter question is yes iff M halts on x.
- Since we know that the halting problem is undecidable, it must also be undecidable whether a given machine accepts ϵ.

Reductions

- Reduction is one of the techniques used to show that a certain problem is undecidable.
- We just proved that the ability to decide problem (f) above gives us the ability to decide halting problem.
- Since we know that the halting problem is undecidable, problem (f) should be undecidable too.
- We say that halting problem is reduced to problem (f).

More decidable/undecidable problems

Problem (g)

Is it decidable whether a given Turing machine accepts any string at all? That is, is $L(M) \neq \emptyset$?

- The reduction technique used to show problem (f) is undecidable works here too.

More decidable/undecidable problems

Problem (h)

Is it decidable whether a given Turing machine accepts all strings? That is, is $L(M) = \Sigma^*$?

- Again, the same reduction technique works here.

More decidable/undecidable problems

Problem (i)

Is it decidable whether a given Turing machine accepts a finite set?

- Note that the construction of M' constructed in the earlier slides is such that

$$\begin{array}{rcccl} M \text{ halts on } x & \Rightarrow & L(M') = \Sigma^* & \Rightarrow & L(M') \text{ is infinite} \\ M \text{does not halt on } x & \Rightarrow & L(M') = \emptyset & \Rightarrow & L(M') \text{ is finite} \end{array}$$

More decidable/undecidable problems

Problem (j)

Is it decidable whether a given Turing machine accepts a regular set?

- Pick your favourite r.e. but non-recursive set A, halting problem will do.
- Given M and x, build a new machine M'' that does the following on input y:
  1. saves y on a separate track of its tape.
  2. writes x on a different track
  3. runs M on input x
  4. if M halts on x, then M'' runs a machine accepting A on its original input y, and accepts if that machine accepts.

More decidable/undecidable problems

- Either M does not halt on x, in which case the simulation in the third step in the previous slide never halts and M'' never accepts any string; or M does halt on x, in which case M'' accepts its input y iff $y \in A$.
- Thus,

$$L(M'') = \begin{cases} A & \text{if } M \text{ halts on } x \\ \emptyset & \text{if } M \text{ does not halt on } x \end{cases}$$

- Since A is not regular and $\emptyset$ is regular, if one could decide whether a given TM accepts a regular set, then one could apply this decision procedure to M'' and this would tell whether M halts on x.

More decidable/undecidable problems

Problem (k)

Is it decidable whether a given Turing machine accepts a CFL?

- Same reduction used for problem (j) will work, since A is not a CFL.

More decidable/undecidable problems

Problem (l)

Is it decidable whether a given Turing machine accepts a recursive set?

- Same reduction used for problem (j) will work, since A is not a recursive set.