More undecidable problems

Deepak D’Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

24 November 2011
Problem (a)

Is it decidable whether a given Turing machine has at least 481 states? Assume that the TM is given using the encoding below:

\[ 0^n10^m10^k10^s10^t10^r10^u10^v10^p10^a10^q10^b1010^p'10^a'10^q'10^b'100\cdots10^p''10^a''10^q''10^b''10. \]

Yes, it is. We can give a TM \( N \) which given \( \text{enc}(M) \) Counts the number of states in \( M \) upto 481. Accepts if it reaches 481, rejects otherwise.
More problems about Turing Machines

More decidable/undecidable problems

Problem (a)

Is it decidable whether a given Turing machine has at least 481 states? Assume that the TM is given using the encoding below:

\[ \begin{align*}
0^n & 1^m 10^k 10^s 10^t 10^r 10^u 10^v 1 0^p 10^q 10^b 10 1 0^p' 10^q' 10^b' 100 \ldots 1 0^p'' 10^q'' 10^b'' 10.
\end{align*} \]

Yes, it is.

We can give a TM \( N \) which given \( \text{enc}(M) \)

- Counts the number of states in \( M \) upto 481.
- Accepts if it reaches 481, rejects otherwise.
Problem (b)

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

00010000100101001000100010000 1 010010100100 1 0100100100100 1 010101010.
Problem (b)

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

Yes, it is.

We can give a TM $N$ which given $\text{enc}(M)$

- Uses 4 tapes: On the 4th tape it writes 481 0’s.
- Uses the first 3 tapes to simulate $M$ on input $\epsilon$, 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.
Problem (c)

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

00010000100101001000100010 01000101000100 1 01001001001001 010101010.
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 than 481 steps on each input $x$ of length upto 481. If so accept, else reject.

1 2 3 ...... 481 482

$\vdash a a b a b a a a b b$ ......
Problem (d)

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

0001000100101001000100010000 1 01000101000100 1 0100100100100 1 010101010.
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 than 481 steps on each input $x$ of length upto 481. If so accept, else reject.

\[
\begin{array}{cccccccc}
1 & 2 & 3 & \cdots & 481 & 482 \\
\hline
a & a & b & a & b & a & a & a & b & b & \cdots
\end{array}
\]
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 $\epsilon$?

0001000010010100100010000 1 01000101000100 1 0100100100100 1 010101010.

Yes, it is. Simulate $M$ on $\epsilon$ for up to $481 \cdot 482 \cdot k$ steps. If $M$ visits the 482nd cell, accept, else reject.
More problems about Turing Machines

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 $\varepsilon$?

Yes, it is.
Simulate $M$ on $\varepsilon$ for upto $m^{481} \cdot 482 \cdot k$ steps. If $M$ visits the 482nd cell, accept, else reject.

00010000100100100100010001000100 01000101000100 1 0100100100100 1 010101010.

1 2 3 \ldots 481 482

┌─ a a b a b a a a b b ┐

\ldots

"
Problem (f)

Is it decidable whether a given Turing machine accepts the null-string $\epsilon$?
Problem (f)
Is it decidable whether a given Turing machine accepts the null-string $\epsilon$?

No.
If it were decidable, say by a TM $N$, then we could use $N$ to decide HP as follows: Define a new machine $N'$ which given input $M \# \# x$, outputs the description of a machine $M'$ which:

- erases its input
- writes $x$ on its input tape
- Behaves like $M$ on $x$
- Accepts if $M$ halts on $x$.

$N'$ then calls $N$ with input $M'$. $N$ accepts $M'$ iff $M'$ accepts $\epsilon$ iff $M$ halts on $x$. 
Turing machine $M'$ for Problem (f)

$$L(M') = \begin{cases} A^* & \text{if } M \text{ halts on } x \\ \emptyset & \text{if } M \text{ does not halt on } x. \end{cases}$$
More problems about Turing Machines

More decidable/undecidable problems

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

Is it decidable whether a given Turing machine accepts all strings? That is, is $L(M) = A^*$?
More decidable/undecidable problems

Problem (i)

Is it decidable whether a given Turing machine accepts a finite set?
More problems about Turing Machines

More decidable/undecidable problems

Problem (j)

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

Given $M$ and $x$, build a new machine $M'$ that behaves as follows:

1. Saves its input $y$ on tape 2.
2. Writes $x$ on tape 1.
3. Runs as $M$ on $x$.
4. If $M$ gets into a halting state, then $M'$ takes back control, runs as $M_R$ on $y$, (Here $M_R$ is any TM that accepts a non-regular language $R$, say $R = \{a^n b^n | n \geq 0\}$).

$M'$ accepts iff $M_R$ accepts.
Problem (j)  

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

Given $M$ and $x$, build a new machine $M'$ that behaves as follows:

1. Saves its input $y$ on tape 2.
2. writes $x$ on tape 1.
3. runs as $M$ on $x$.
4. if $M$ gets into a halting state, then
   - $M'$ takes back control,
   - Runs as $M_R$ on $y$,
   - (Here $M_R$ is any TM that accepts a non-regular language $R$, say $R = \{a^n b^n \mid n \geq 0\}$).
   - $M'$ accepts iff $M_R$ accepts.
Turing machine $M'$ for Problem (j)

\[ L(M') = \begin{cases} 
R & \text{if } M \text{ halts on } x \\
\emptyset & \text{if } M \text{ does not halt on } x.
\end{cases} \]
Problem (k)

Is it decidable whether a given Turing machine accepts a CFL?
More problems about Turing Machines

More decidable/undecidable problems

Problem (I)
Is it decidable whether a given Turing machine accepts a recursive set?