THEORY OF COMPUTATION
Halting Problem for TM and TM Variants - 21

Prof. Dan A. Simovici

UMB

1 The Halting Problem for Turing Machines

2 Nondeterministic Turing Machines

3 Variations on the Turing Machine Theme

4 Turing Machine Variants

A halting problem for a fixed TM $\mathcal{M}$ is the problem of finding an algorithm that will determine whether $\mathcal{M}$ will eventually halt when started with a given configuration.

Theorem

There is a TM $\mathcal{M}$ *with alphabet* $\{1\}$ *that has an unsolvable halting problem.*

Proof.

Let U be some r.e. set that is not recursive. For example, the set K introduced earlier would do. Let $\mathcal{K}$ be the corresponding TM. Thus, $\mathcal{K}$ accepts a string of 1s if and only if its length belongs to U. Hence, $x \in U$ if and only if $\mathcal{K}$ halts when started with the configuration

$$\underset{\underset{q_1}{\uparrow}}{B}\,1^x$$

Thus, of there were an algorithm for solving the halting problem for $\mathcal{K}$, it could be used to decide the membership of x in U. Since U is not recursive, such an algorithm cannot exist. □

Another unsolvable problem that concerns TMs.

Theorem

There exists a TM with alphabet $\{1\}$ and a state q_m such that there is no algorithm that can determine whether $\mathcal{M}$ will ever arrive to the state q_m when it begins in a given configuration.

Proof.

Let $\mathcal{K}$ be a TM with alphabet $\{1\}$ and set of states $\{q_1, \ldots, q_k\}$ that has an unsolvable halting problem.
Define the TM $\hat{\mathcal{K}}$ by adding to the quadruples of $\mathcal{K}$ the quadruples of the form

$$q_i \ B \ B \ q_{k+1}$$

for $1 \leqslant i \leqslant k$ for which no quadruple of $\mathcal{K}$ begins with $q_i B$. In addition, add

$$q_i \ 1 \ 1 \ q_{i+1}$$

when no quadruple of $\mathcal{K}$ begins with $q_i 1$. Thus, $\mathcal{K}$ eventually halts beginning with a given configuration if and only if $\hat{\mathcal{K}}$ eventually halts in the state q_{k+1}. □

Definition

A nondeterministic TM is an arbitrary finite set of quadruples.

Previously considered TMs are referred to as deterministic. In other words, the restriction that no two distinct quadruples may begin with the same pair of symbols $q_i s_j$ is dropped for non-deterministic TMs.

Definition

A configuration

$$\begin{array}{c} \cdots s_j \cdots \\ \uparrow \\ q_i \end{array}$$

is called terminal with respect to a nondeterministic Turing machine, and the machine is said to *halt*, if $\mathcal{M}$ contains no quadruple beginning with $q_i s_j$.

If c, c' are two configurations of a quadruple TM $\mathcal{M}$ c we write

$$c \vdash c'$$

to indicate that the transition from the configuration c to the configuration c' is permitted by one of the quadruples of $\mathcal{M}$.

Example

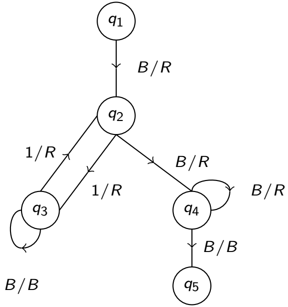
Consider the nondeterministic TM defined by the following quadruples:

$$\begin{gathered} q_1 \; B \; R \; q_2 \\ q_2 \; 1 \; R \; q_3 \\ q_2 \; B \; B \; q_4 \\ q_3 \; 1 \; R \; q_2 \\ q_3 \; B \; B \; q_3 \\ q_4 \; B \; R \; q_4 \\ q_4 \; B \; B \; q_5 \end{gathered}$$

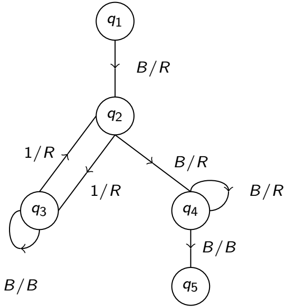
This TM is not deterministic because of the presence of the tuples

$$q_4 \; B \; R \; q_4 \text{ and } q_4 \; B \; B \; q_5$$

The state diagram is:



In this machine we have the computation:



$\underset{\substack{\uparrow\\ q_1}}{B}1111$

$\vdash B\underset{\substack{\uparrow\\ q_2}}{1}111$

$\vdash B1\underset{\substack{\uparrow\\ q_3}}{1}11$

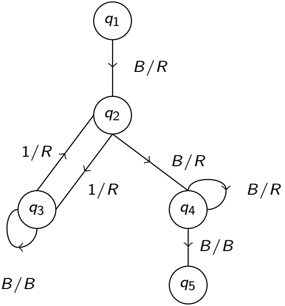
$\vdash B11\underset{\substack{\uparrow\\ q_2}}{1}1$

$\vdash B111\underset{\substack{\uparrow\\ q_3}}{1}$

$\vdash B1111\underset{\substack{\uparrow\\ q_2}}{B}B$

$\vdash B1111\underset{\substack{\uparrow\\ q_4}}{B}$

At this point the computation becomes nondeterministic.



We may have:

$B1111\underset{\overset{\uparrow}{q_4}}{B}$

$\vdash$

$B1111\underset{\overset{\uparrow}{q_5}}{B}111$

OR

$B1111\underset{\overset{\uparrow}{q_4}}{B}$

$\vdash$

$B1111B\underset{\overset{\uparrow}{q_4}}{B}$

$\vdash$

$B1111BB\underset{\overset{\uparrow}{q_4}}{B}$

$\vdash$

...

Definition

Let $A = \{s_1, \ldots, s_n\}$ be an alphabet and let $u \in A^*$. The nondeterministic TM accepts the word u if there exists a sequence of configurations $\gamma_1, \ldots, \gamma_m$ such that:

1. $$\gamma_1 \vdash \gamma_2 \vdash \cdots \vdash \gamma_m.$$
2. γ_1 is the configuration $$\underset{\overset{\uparrow}{q_1}}{s_0}\, u$$
3. γ_m is terminal with respect to $\mathcal{M}$.

The sequence $\gamma_1, \ldots, \gamma_m$ is called an accepting computation for u. The language accepted by $\mathcal{M}$ is the set of all $u \in A^*$ that are accepted by $\mathcal{M}$.

Note that:

- a nondeterministic TM accepts a word u if there exists an accepting computation which starts with the configuration $\underset{\substack{\uparrow \\ q_1}}{s_0\, u}$
- this does not preclude the existence on a non-accepting computation which starts with $\underset{\substack{\uparrow \\ q_1}}{s_0\, u}$.

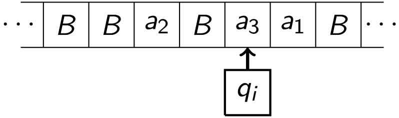
For acceptance it is only necessary that there is some sequence of configurations leading to a terminal configuration. In other words, a deterministic TM must “guess” an sequence of configuration leading to acceptance.

A previous result can now be reformulated for nondeterministic TMs:

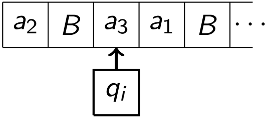
Theorem

For every r.e. language there is a nondeterministic TM $\mathcal{M}$ *that accepts* L*.*

The one-way TMs with quadruples:
We replace the tape that is infinite in both directions with a tape that is infinite in one direction only:



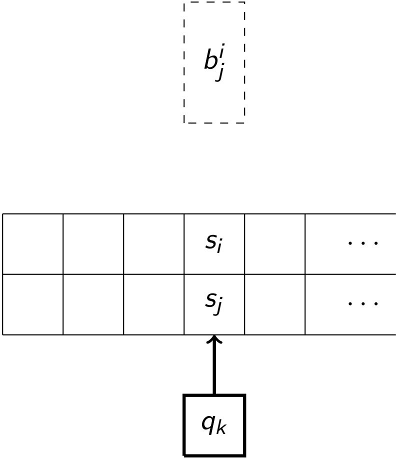
is replaced by



For one-way infinite tape machines (with quadruples) it is necessary to make a decision about the effect of a quadruple $q_i s_j L q_k$ when the head is at the left end of the tape. We assume an instruction to move left is interpreted as a halt if the head is at the leftmost square.

Clearly, anything a TM can do on an one-way infinite tape it can do on a bilaterally infinite tape.

Another idea is to consider TMs with one-way, two track tapes (an upper and a lower track).



b_j^i is equivalent
to having s_i on the upper
track and s_j on the lower
track

Let $\mathcal{M}$ be a TM with alphabet $A = \{s_1, \ldots, s_n\}$ and states $q_1, \ldots, q_k$ which computes a function $g(x)$ on A_0, where $A_0 \subseteq A$. The initial configuration for $x \in A_0^*$ is

$$\underset{\overset{\uparrow}{q_1}}{B}\, x$$

The goal is to construct a TM $\overline{\mathcal{M}}$ that computed g on an one-way infinite tape.

The initial configuration for $\overline{\mathcal{M}}$ is

$$\# \underset{\overset{\uparrow}{q_1}}{B} x$$

where $\#$ is a special symbol that will occupy the leftmost square of the tape for most of the computation.

The alphabet of $\overline{\mathcal{M}}$ is $A \cup \{\#\} \cup \{b^i_j \mid 1 \leqslant i, j \leqslant n\}$. The symbol b^i_j indicates that s_i is on the upper track and s_j is on the lower track.

The states of $\overline{\mathcal{M}}$ are q_1, q_2, q_3, q_4, q_5 and

$$\{\overline{q_i}, \tilde{q}_i \mid 1 \leqslant i \leqslant K\},$$

as well as some other additional states.
There are three groups of quadruples of $\overline{\mathcal{M}}$:

$$\text{BEGINNING}, \text{MIDDLE}, \text{END}.$$

BEGINNING serves to copy the input on the upper track puting blanks on the corresponding lower track and consists of the following quadruples:

$$\begin{array}{ll}
q_1\ B\ R\ q_2 & \\
q_2\ s_i\ R\ q_2 & \text{for } 1 \leqslant i \leqslant n \\
q_2\ B\ L\ q_3 & \\
q_3\ s_i\ b_0^i\ q_3 & \text{for } 0 \leqslant i \leqslant n \\
q_3\ b_0^i\ L\ q_3 & \text{for } 0 \leqslant i \leqslant n \\
q_3\ \#\ R\ \overline{q}_1 &
\end{array}$$

Starting with the configuration

$$\#\,\underset{\overset{\uparrow}{q_1}}{B}\,s_2 s_1 s_3$$

BEGINNING will halt in the configuration

$$\#\,\underset{\overset{\uparrow}{\overline{q}_1}}{b_0^0}\,b_0^2 b_0^1 b_0^3 B$$

Note that b_0^0 is different from $s_0 = B$.

MIDDLE will consist of quadruples corresponding to those of $\mathcal{M}$ as well as additional quadruples.

	Quadr. in M	Quadr. in$\overline{\mathcal{M}}$
(a)	$q_i\ s_j\ s_k\ q_\ell$	$\overline{q}_i\ b^j_m\ b^k_m\ \overline{q}_\ell\quad 0 \leqslant m \leqslant n$
		$\tilde{q}_i\ b^m_j\ b^m_k\ \tilde{q}_\ell\quad 0 \leqslant m \leqslant n$
(b)	$q_i\ s_j\ R\ q_\ell$	$\overline{q}_i\ b^j_m\ R\ \overline{q}_\ell\quad 0 \leqslant m \leqslant n$
		$\tilde{q}\ b^m_j\ L\ \tilde{q}_\ell\quad 0 \leqslant m \leqslant n$
(c)	$q_i\ s_j\ L\ q_\ell$	$\overline{q}_i\ b^j_m\ L\ \overline{q}_\ell\quad 0 \leqslant m \leqslant n$
		$\tilde{q}_i\ b^m_j\ R\ \tilde{q}_\ell\quad 0 \leqslant m \leqslant n$
(d)		$\overline{q}_i\ B\ b^0_0\ \overline{q}_i\quad 1 \leqslant i \leqslant K$
		$\tilde{q}_i\ B\ b^0_0\ \tilde{q}_i\quad 1 \leqslant i \leqslant K$
(e)		$\overline{q}_i\ \#\ R\ \tilde{q}_i\quad 1 \leqslant i \leqslant K$
		$\tilde{q}_i\ \#\ R\ \overline{q}_i\quad 1 \leqslant i \leqslant K$

$\overline{q}_i$ and $\tilde{q}_i$ correspond to actions on the upper track and lower track, respectively.

Note that:

- in (b) and (c) the lower track left and right are reversed;
- quadruples in (d) replace single blanks B by double blanks b_0^0 as needed;
- quadruples (e) arrange for switchover from the upper to the lower track and viceversa.

The END part translates the output into a word on the original alphabet A, taking into account that the output is split between two tracks.

When $\mathcal{M}$ contains no quadruple beginning with $q_i s_j$ (for $0 \leqslant m \leqslant n$ and $0 \leqslant i, j \leqslant n$ include in END the quadruples

$$\overline{q}_i \ b^j_m \ b^j_m \ q_4 \text{ and } \tilde{q}_i \ b^m_j \ b^m_j \ q_4.$$

Also, include

$$q_4 \ b^i_j \ L \ q_4 \text{ and } q_4 \ \# \ B \ q_5$$

For each initial configuration for which $\mathcal{M}$ halts, the effect of BEGINNING, MIDDLE and this part of END is to ultimately produce a configuration of the form:

$$\underset{\substack{\uparrow \\ q_5}}{B}\, b_{j_1}^{i_1}\, b_{j_2}^{i_2} \cdots b_{j_k}^{i_k}$$

The remaining task of END is to convert this tape content into

$$s_{j_k} s_{j_{k-1}} \cdots s_{j_1} s_{i_1} s_{i_2} \cdots s_{i_k}$$

Instead of giving quadruples for doing this we could use the macros of the Post-Turing language, which can be translated readily into quadruples. This is useful because the Post-Turing language can shift block on the tape.