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 $M$ is the problem of finding an algorithm that will determine whether $M$ will eventually halt when started with a given configuration.

**Theorem**

*There is a TM $M$ 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

\[ B 1^x \]

\[ \uparrow \]

$q_1$

Thus, if 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 \( M \) 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 $1 \leq i \leq k$ 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 $q_{k+1}$. \qed
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 $q_is_j$ is dropped for non-deterministic TMs.
A configuration

\[ \ldots s_j \ldots \]

\[ \uparrow q_i \]

is called **terminal** with respect to a nondeterministic Turing machine, and the machine is said to **halt**, if \( M \) contains no quadruple beginning with \( q_is_j \).
If $c$, $c'$ are two configurations of a quadruple TM $M$ 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 $M$. 
Example

Consider the nondeterministic TM defined by the following quadruples:

\[ q_1 \ B \ R \ q_2 \]
\[ q_2 \ 1 \ R \ q_3 \]
\[ q_2 \ B \ B \ q_4 \]
\[ q_3 \ 1 \ R \ q_2 \]
\[ q_3 \ B \ B \ q_3 \]
\[ q_4 \ B \ R \ q_4 \]
\[ q_4 \ B \ B \ q_5 \]

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:

```
q1
\downarrow B/R
q2
\downarrow 1/R \rightarrow 1/R
q3
\overset{1/R}{\rightarrow} B/R
q4
\overset{B/R}{\rightarrow} B/B
q5
```

Nondeterministic Turing Machines
In this machine we have the computation:
At this point the computation becomes nondeterministic.

We may have:

- $B1111B \uparrow q_4$
- $B1111111 \uparrow q_5$

OR

- $B1111B \uparrow q_4$
- $B1111B \uparrow q_4$
- $B1111BBB \uparrow q_4$

...
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. $\gamma_1$ is the configuration
   
   \[
   \begin{array}{c}
   s_0 u \\
   q_1
   \end{array}
   \]

2. $\gamma_m$ is terminal with respect to $M$, and

3. $\gamma_1 \vdash \gamma_2 \vdash \cdots \vdash \gamma_m$.

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

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

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 $M$ 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:

\[
\begin{array}{cccccccc}
& & B & B & a_2 & B & a_3 & a_1 & B & \cdots \\
\cdots & & & & & & & & & \\
\end{array}
\]

\[
\begin{array}{cccccccc}
B & B & a_2 & B & a_3 & a_1 & B & \cdots \\
\cdots & & & & & & & & \\
\end{array}
\]

is replaced by

\[
\begin{array}{cccccccc}
a_2 & B & a_3 & a_1 & B & \cdots \\
\cdots & & & & & & & \\
\end{array}
\]

\[
\begin{array}{cccccccc}
& & & & & & & & \\
q_i & & & & & & & \\
\end{array}
\]
For one-way infinite tape machines (with quadruples) it is necessary to make a decision about the effect of a quadruple $q_i; s; j Lq_k$ 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).

\[ b_j^i \]

\[ s_i \quad \cdots \]
\[ s_j \quad \cdots \]

\[ q_k \]

\( b_j^i \) 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

$$
\begin{array}{c}
B \\
q_1
\end{array}
\uparrow
x
$$

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

$$\# B x \uparrow q_1$$

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

The alphabet of $M$ is $A \cup \{\#\} \cup \{b^i_j \mid 1 \leq i, j \leq n\}$. 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 $\overline{M}$ are $q_1, q_2, q_3, q_4, q_5$ and

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

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

BEGINNING, MIDDLE, END.
BEGINNING serves to copy the input on the upper track putting blanks on the corresponding lower track and consists of the following quadruples:

\[
\begin{align*}
q_1 & \rightarrow B \ R \ q_2 \\
q_2 & \rightarrow s_i \ R \ q_2 \quad \text{for } 1 \leq i \leq n \\
q_2 & \rightarrow B \ L \ q_3 \\
q_3 & \rightarrow s_i \ b_i^j \ q_3 \quad \text{for } 0 \leq i \leq n \\
q_3 & \rightarrow b_i^j \ L \ q_3 \quad \text{for } 0 \leq i \leq n \\
q_3 & \rightarrow \# \ R \ q_1
\end{align*}
\]
Starting with the configuration

\[
\# B s_2 s_1 s_3
\]

\[q_1\]

BEGINNING will halt in the configuration

\[
\# b_0^0 b_0^2 b_0^1 b_0^3 B
\]

\[q_1\]

Note that \(b_0^0\) is different from \(s_0 = B\).
MIDDLE will consist of quadruples corresponding to those of \( M \) as well as additional quadruples.

<table>
<thead>
<tr>
<th>Quadr. in ( M )</th>
<th>Quadr. in ( \tilde{M} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>((a)) ( q_i s_j s_k q_\ell )</td>
<td>( \tilde{q}<em>i b^j_m b^k_m \tilde{q}</em>\ell 0 \leq m \leq n )</td>
</tr>
<tr>
<td>((b)) ( q_i s_j R q_\ell )</td>
<td>( \tilde{q}<em>i b^j_m R \tilde{q}</em>\ell 0 \leq m \leq n )</td>
</tr>
<tr>
<td>((c)) ( q_i s_j L q_\ell )</td>
<td>( \tilde{q}<em>i b^j_m L \tilde{q}</em>\ell 0 \leq m \leq n )</td>
</tr>
<tr>
<td>((d))</td>
<td></td>
</tr>
<tr>
<td>((e))</td>
<td></td>
</tr>
</tbody>
</table>

\( \tilde{q}_i \) 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 vice versa.
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 $0 \leq m \leq n$ and $0 \leq i, j \leq n$) include in END the quadruples

$$\bar{q}_i b_m^i b_m^j q_4 \text{ and } \bar{q}_i b_m^j b_m^i q_4.$$ 

Also, include

$$q_4 b_i^j L q_4 \text{ and } q_4 \# B q_5$$
For each initial configuration for which $M$ halts, the effect of BEGINNING, MIDDLE and this part of END is to ultimately produce a configuration of the form:

$$B b_1^{i_1} b_2^{i_2} \cdots b_k^{i_k}$$

\[ q_5 \]
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.