Turing Machines

Recall $(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

configurations, $f_m^k \xrightarrow{\text{one step}} f_m^{k+1}$

$M$ accepts $w$ iff $q_0 w \xrightarrow{f_m^k} q_{acc} w \in \Sigma^*$

$M$ rejects $w$ iff $q_0 w \xrightarrow{f_m^k} q_{rej} w \in \Sigma^*$

$M$ is a decider iff $\forall w \in \Sigma^*$

$M$ accepts $w$ or $M$ rejects $w$

Example: $\Sigma^2^n : n \geq 0$\), $\Sigma = \{0, 1\}$ input alphabet

Plan:
1. Check if one $0$, if yes accept
2. If more than one $0$, cross off every second $0$ (if odd reject)
3. Repeat above with remaining $0$'s

eg

$0\overline{0}0\overline{0}0\overline{0}0w$ reject

$\overline{0}\overline{0}0\overline{0}0\overline{0}0w$ need to reach start of the string to get back
to mark start of string we could use a new tape symbol but to keep TM simple we use a blank \( \mathbb{L} \) representing a 0 and the start.

**TM:**

![Diagram of TM states and transitions]

Notation: \( \rho \rightarrow a \rightarrow \delta \rightarrow \sigma \) mean \( \rho \rightarrow a \rightarrow \delta \rightarrow \sigma \)

Generalization of TM:

**k-tape TM**

\[ \text{State} \rightarrow \begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c|c} \hline w_1 & w_2 & \cdots & w_n \hline \hline \end{array} \]
Theorem: Every $k$-tape TM $M$ is equivalent to some 1-tape TM $M'$.

Proof: Basic idea:
- represent all $k$-tapes' contents on a single tape.

Support $M = (Q, \Sigma, \Gamma, \delta, \ldots)$

We create $M' = (Q', \Sigma, \Gamma', \delta', \ldots)$

Let $\# \in \Gamma'$ be a new symbol.

We represent all $k$-tapes' contents in $M'$ by

\[
\begin{array}{c}
\text{\# tape 1} \\
\text{\# tape 2} \\
\vdots \\
\text{\# tape k} \\
\end{array}
\]

Since each tape or infinite we only represent the portion that is used. The struc
We will represent the input tape as

# w_1 w_2 ... w_n #

but we also need to store head position for each
We put a * over each char if it
also has the head on or on it.

P^* = P * P^*

So... first convert w_1...w_n input to
above story.

To figure out what move to make, need to
store scanned symbol in state.

- Do h to R sweep recording the symbol
  under the dot

- To execute the moves for this step of M
  - sweep R to L and execute all the
    left moves
      (rewrite the symbol and dot to
       symbol just to the left
       if that symbol was a #
       put that dot back on the
       first symbol of that tape)
  - sweep L to R and execute all the
    right moves.
most right move are
simple having to... not more
to the right, except when
that is # (reached end
of word, form of tape)
and need to insert a
blank symbol and
shift the rest of the
tape to the right
during the sweep.

(to shift by c characters
keep track of queue of
most recent char)

(in state) reading at one end and
writing from the other.

Return to the left end.

Note: If original machine ran in T steps
new machine may take
O(kT^2) steps:

Original step costs = # cell/m

\[ \sum \leq kT + kT + 1 \]

Total O(kT^2)

This simulation is step by step
and the machine accepts


Alternative simulation: Multitracked

New set of symbols $\Gamma' = \Gamma \cup (\overline{\Gamma} \cup \overline{\Gamma})$

<table>
<thead>
<tr>
<th>$\overline{\Gamma}$</th>
<th>$\overline{\Gamma}$</th>
<th>$\overline{\Gamma}$</th>
<th>$\overline{\Gamma}$</th>
<th>$\overline{\Gamma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

each track represented as a "track"

Behaviour is same as before:
- Sweep L to R to collect scanned symbols
- Sweep R to L, then L moves
- Sweep L to R, then R moves
- (maybe make multi-tracked multi-dotted blanks to replace dot)

return to Lead

Time: $O(T^2)$ no sticky required

Fact: Every multiway 2-tape to 1-tape best possible simulation is $O(T^2)$

Next time: Nondeterministic TMs.