1 Simulating $k$-tape TM by 1-tape TM

For a $k$-tape TM $M_k$, we construct a 1-tape TM $M_1$ deciding the same language. For symbol $\sigma$ in $M_k$’s alphabet, we use symbol $↓\sigma$ to represent symbol $\sigma$ with a tape head on it. Suppose at position $i$ of tape, the $M_k$ has symbols $x_1, \ldots, x_k$, then we write $x_1, \ldots, x_k$ sequentially on the tape of $M_1$.

<table>
<thead>
<tr>
<th>$k$-tape TM $M_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tapes:</td>
</tr>
<tr>
<td>$x_1 \ldots x_n #$</td>
</tr>
<tr>
<td>$$#</td>
</tr>
<tr>
<td>$$#</td>
</tr>
<tr>
<td>$\ldots$</td>
</tr>
<tr>
<td>$$#</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1-tape TM $M_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tapes:</td>
</tr>
<tr>
<td>$x_1 \ldots x_n #$</td>
</tr>
</tbody>
</table>

Initialize $M_1$:
1. Replace $\$ with $\$$\ldots\$
2. Go left one
3. Replace $x_i$ with $x_i \# \# \ldots \#$
4. Go left until we see $\#$
5. Replace $x_i$ with $x_i \ldots \#$
6. Replace $\#$ with $\# \ldots \#$
Time: $O(n)$.

To simulate one step of $M_k$:
1. Find the head locations of each tape head locations of each tape in $M_k$.
2. Start at beginning.
3. Go through tape. When we see a symbol whose $i$-th coordinate is a symbol $\sigma$, remember that $\sigma$ is under $i$-th tape.
4. Look up the actions that we need to perform: Move $i$-th tape head left or right/Write $\sigma'$ under $i$-th tape head / Halt / Accept / Reject.

The size of current tape is bounded by $\max(n, T(n))$. So to simulate one step of $M_k$, $M_1$ takes $O(n + T(n))$ steps. The overall running time is $O(T(n)(n + T(n))) = O(n^2)$. 

Problems computable by a $k$-tape TM in polynomial time can be computed by a 1-tape TM in polynomial time. Therefore “polynomial time” is independent of the number of tapes, so it is a robust model of efficiently computable problems.

## 2 Time bounded Church-Turing thesis

### 2.1 Communication complexity of string equality

Let language $Pal = \{x \in \{0, 1\}^* \mid x = x^R\}$. $Pal$ is the set of palindromes.

**Theorem 2.1.** If there exists a 1-tape TM that decides $Pal$, then $T_M(n) \in \Omega(n^2)$.

The proof idea is to consider the *communication complexity* of deciding if two strings are equal. Suppose Alice knows binary string $x$, and Bob knows binary string $y$. They want to communicate to each other to know whether $x = y$. If $x$ and $y$ has length $n$, then they can decide $x = y$ by sending $n$ bits.

We claim that, to decide whether $x = y$, the number of bits sent by Alice and Bob is at least $n$. Let $C(x, y) = b_1, \ldots, b_t$ be the bits needed to communicate.

**Lemma 2.2.** If $C(x, x) = C(y, y)$, then $C(x, y) = C(y, x) = C(x, x) = C(y, y)$.

The proof is by induction in the string length. Note that this lemma also applies to deciding whether $x = y^R$.

In general, if $x, y$ are selected from a domain $S$, then the bits needed to communicate is $t \geq \log |S|$.

### 2.2 Simulation time lower bound

Let language $L = \{x0^{n/2}x^R\}$, where $x \in \{0, 1\}^{n/4}$. (Strings in $L$ can be considered as the “worst case” of strings in $Pal$. If $L$ requires time $\Omega(n^2)$, so does $Pal$. Sorry for the
mistake of mentioning reducing from Pal to L in previous version of this notes, which is the wrong direction of proving the lower bound of Pal from the lower bound of L.)

We claim that a 2-tape TM can decide L in O(n) (by simply comparing each bit) but any 1-tape TM deciding L requires time $\Omega(n^2)$.

For string $x0^{n/2}y$, deciding whether $x = y^R$ can be considered as Alice and Bob holding $x$ and $y$ respectively, while they communicate using the tape head of TM $M$.

Define $CN_i$ as the number of times the tape head crossing the $i$-th 0 in the middle. By definition, it is obvious that

$$\sum_{i=1}^{n/2} CN_i \leq T_M(n)$$

Because there are $n/2$ positions of 0’s, we get

$$\min_i CN_i \leq \frac{T_M(n)}{n/2} = \frac{2T_M(n)}{n}$$

So the communication bits used to decide $x = y$ is $O(T_M(n)/n)$.

Let set $S_i = \{x \mid \arg \min CN_i(x) = i\}$. The number of different communication sequences is $2^{n/4}$. Again because there are $n/2$ positions of 0’s, there exists a position $i$ s.t. $|S_i| \geq 2^{n/4}$. To distinguish these $x$’s, the communication bits needed to go through the $i$-th 0 is $\log |S_i|$.

Thus,

$$c \cdot \frac{T_M(n)}{n} \geq \log |S_i| \geq \log \frac{2^{n/4}}{n/2},$$

we have $T_M(n) = \Omega(n^2)$. 