Combining Turing Machines

We will define notation that will make it easier to look at more complicated Turing machines

1. Given Turing Machines M1 and M2
   Notation for
   • Run M1
   • Run M2

2. Given Turing Machines M1 and M2
   Notation for
   • Run M1
   • If x is current symbol
     – then Run M2
3. Given Turing Machines M1, M2, and M3

Notation for

- Run M1
- If x is current symbol
  - then Run M2
  - else Run M3

More Notation for Simplifying Turing Machines

Suppose Γ={a,b,c,B}

- z is any symbol in Γ
- x is a specific symbol from Γ

1. s - start
2. R - move right
3. L - move left

4. x - write x (and don’t move)

5. R_a - move right until you see an a

6. L_a - move left until you see an a

7. R_{¬a} - move right until you see anything that is not an a

8. L_{¬a} - move left until you see anything that is not an a

9. h - halt in a final state

10. \[ a, b \mapsto w \]
    
    If the current symbol is a or b, let w represent the current symbol.
Example

Assume input string \( w \in \Sigma^+, \Sigma = \{a, b\} \).

If \(|w|\) is odd, then write a \( b \) at the end of the string. The tape head should finish pointing at the leftmost symbol of \( w \).

input: bab, output: babb

input: ba, output: ba

What is the running time?
Example

Assume input string $w \in \Sigma^+, \Sigma = \{a, b\}, |w| > 0$

For each $a$ in the string, append a $b$ to the end of the string.

input: $abbabb$, output: $abbabbbb$

The tape head should finish pointing at the leftmost symbol of $w$.

Turing's Thesis  
Any computation that can be carried out by a mechanical means can be performed by a TM.

Definition: An algorithm for a function $f: \mathbb{D} \rightarrow \mathbb{R}$ is a TM $M$, which given input $d \in \mathbb{D}$, halts with answer $f(d) \in \mathbb{R}$.

Example: $f(x + y) = x + y$, $x$ and $y$ unary numbers.

\begin{align*}
\text{start with: } & 111 + 1111 \\
& \uparrow \\
\text{end with: } & 1111111 \\
& \uparrow
\end{align*}
**Example:** Copy a String, \( f(w) = w0w, \ w \in \Sigma^*, \ \Sigma = \{a, b, c\} \)

Denoted by \( C \)

<table>
<thead>
<tr>
<th>Start with:</th>
<th>abac</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\uparrow</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>End with:</th>
<th>abac0abac</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\uparrow</td>
</tr>
</tbody>
</table>

Algorithm:

- Write a 0 at end of string
- For each symbol in string
  - make a copy of the symbol
**Example:** Shift the string that is to the left of the tape head to the right, denoted by $S_R$ (shift right)

Below, “ba” is to the left of the tape head, so shift “ba” to the right.

```
start with: aaBbabca
↑
end with:   aaBBbaca
↑
```

Algorithm:

- remember symbol to the right and erase it
- for each symbol to the left do
  - shift the symbol one cell to the right
- replace first symbol erased
- move tape head to appropriate position
**Example:** Shift the string that is to the right of tape head to the left, denote by $S_L$ (shift left)

- start with: $\text{babcaBba}$
- end with: $\text{bacaBBba}$

(similar to $S_R$)
Example: Add unary numbers
This time use shift.

Example: Multiply two unary numbers, f(x*y)=x+y, x and y unary numbers. Assume x,y>0.

\[
\begin{align*}
\text{start with:} & \quad 1111 \cdot 11 \\
& \uparrow \\
\text{end with:} & \quad 11111111 \\
& \uparrow 
\end{align*}
\]