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

   \[
   \begin{array}{ccc}
   M1 & \rightarrow & S \quad H \quad \rightarrow \quad S' \quad H'
   \\
   \end{array}
   \]

   \[
   \begin{array}{ccc}
   M1 \rightarrow M2 & \rightarrow & S \quad H \quad z \quad z, R \quad \rightarrow \quad S' \quad H'
   \\
   \end{array}
   \]

   z represents any symbol in \( M1 \) \( \bigcup \) \( M2 \)

2. Given Turing Machines M1 and M2
   Notation for
   - Run M1
   - If \( x \) is current symbol
     - then Run M2

   \[
   \begin{array}{ccc}
   M1 & \rightarrow & S \quad H \quad \rightarrow \quad S' \quad H'
   \\
   \end{array}
   \]

   \[
   \begin{array}{ccc}
   M1 \times M2 & \rightarrow & S \quad H \quad x \quad x, R \quad \rightarrow \quad S' \quad H'
   \\
   \end{array}
   \]

   z represents any symbol in \( x \) is an element of \( M1 \) \( \bigcup \) \( M2 \)
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 \( \Gamma = \{a, b, c, B\} \)

z is any symbol in \( \Gamma \),

x is a specific symbol from \( \Gamma \),

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. \( R_{\neq a} \) - move right until you see anything that is not an a
8. \( L_{\neq a} \) - move left until you see anything that is not an a
9. h - halt in a final state
10. \[ \{a, b\} \rightarrow \] 5

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: abbabbb

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 $f: D \rightarrow \mathbb{R}$ is a TM $M$, which given input $d \in D$, halts with answer $f(d) \in \mathbb{R}$. 
Example: \( f(x + y) = x + y, x \) and \( y \) unary numbers.

\[
\begin{align*}
\text{start with:} & \quad 111+1111 \\
\uparrow & \\
\text{end with:} & \quad 111111 \\
\uparrow & 
\end{align*}
\]

Example: Copy a String, \( f(w) = w0w, w \in \Sigma^*, \Sigma = \{a, b, c\} \)

Denoted by \( C \)

\[
\begin{align*}
\text{start with:} & \quad abac \\
\uparrow & \\
\text{end with:} & \quad abac0abac \\
\uparrow & 
\end{align*}
\]

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.

\[
\begin{align*}
\text{start with:} & \quad \text{aaBbabca} \\
\text{end with:} & \quad \text{aaBBbaca}
\end{align*}
\]

**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

\[
\begin{align*}
\text{Example:} & \quad \text{Shift the string that is to the right of tape head to the left,} \\
& \quad \text{denote by $S_L$ (shift left)}
\end{align*}
\]

\[
\begin{align*}
\text{start with:} & \quad \text{babcaBba} \\
\text{end with:} & \quad \text{bacaBBba}
\end{align*}
\]

(similar to $S_R$)
**Example:** Add unary numbers

This time use shift.

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

- **start with:** 1111*11
  - ↑

- **end with:** 11111111
  - ↑