# A Macro language for Turing Machines

You need to learn this simple language. I will use it and I expect you to use it on HW and tests (for exams I'll give you a handout with the details).

(1) Define some basic machines

- Symbol writing machines

  For each $x \in \Gamma$, define $M_x$, written as just $x$, to be a machine that writes $x$. Read-write head ends up in original position.

- Head moving machines

  R: for each $x \in \Gamma$, $\delta(s, x) = (h, x, \rightarrow)$
  L: for each $x \in \Gamma$, $\delta(s, x) = (h, x, \leftarrow)$

- Machines that simply halt:

  $h$, which simply halts (don't care whether it accepts).
  $n$, which halts and rejects.
  $y$, which halts and accepts.

# Checking Inputs and Combining Machines

Machines to:

- Check the tape and branch based on what character we see, and
- Combine the basic machines to form larger ones.

To do this, we need two forms:

- $M_1\ M_2$

- $M_1 \xrightarrow{\text{<condition>}} M_2$

# Turing Machines Macros Cont'd

Example:

$$\begin{tikzcd} >M_1 \arrow[r, "a"] \arrow[d, "b"'] & M_2 \\ M_3 & \end{tikzcd}$$

- Start in the start state of $M_1$.
- Compute until $M_1$ reaches one of its halt states, which are not halt states in the combined machine.
- Examine the tape and take the appropriate transition.
- Start in the start state of the next machine, etc.
- Halt if any component reaches a halt state and has no place to go.
- If any component fails to halt, then the entire machine may fail to halt.

# More macros

$$\begin{tikzcd} M_1 \arrow[r, shift left, "a"] \arrow[r, shift right, "b"'] & M_2 \end{tikzcd}$$ becomes $M_1 \xrightarrow{a,\ b} M_2$

$M_1 \xrightarrow{\text{all elems of } \Gamma} M_2$ becomes $M_1 \longrightarrow M_2$ or $M_1 M_2$

**Variables**

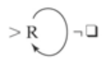
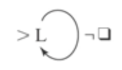
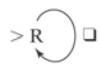
$M_1 \xrightarrow[\text{except a}]{\text{all elems of } \Gamma} M_2$ becomes $M_1 \xrightarrow{x \leftarrow \neg a} M_2$
and $x$ takes on the value of the current square

$M_1 \xrightarrow{a,\ b} M_2$ becomes $M_1 \xrightarrow{x \leftarrow a,\ b} M_2$
and $x$ takes on the value of the current square

$M_1 \xrightarrow{x = y} M_2$
if $x = y$ then take the transition

e.g., $> \xrightarrow{x \leftarrow \neg Æ} Rx$ if the current square is not blank, go right and copy it.

## Blank/Non-blank Search Machines

| | | |
|---|---|---|
|  | Find the first blank square to the right of the current square. | $R_{Æ}$ |
|  | Find the first blank square to the left of the current square. | $L_{Æ}$ |
|  | Find the first nonblank square to the right of the current square. | $R_{\neg Æ}$ |
|  | Find the first nonblank square to the left of the current square | $L_{\neg Æ}$ |

## More Search Machines

| | |
|---|---|
| $L_a$ | Find the first occurrence of a to the left of the current square. |
| $R_{a,b}$ | Find the first occurrence of a or b to the right of the current square. |
| $\begin{array}{l} L_{a,b} \xrightarrow{a} M_1 \\ \downarrow b \\ M_2 \end{array}$ | Find the first occurrence of a or b to the left of the current square, then go to $M_1$ if the detected character is a; go to $M_2$ if the detected character is b. |
| $L_{x \leftarrow a,b}$ | Find the first occurrence of a or b to the left of the current square and set $x$ to the value found. |
| $L_{x \leftarrow a,b}Rx$ | Find the first occurrence of a or b to the left of the current square, set $x$ to the value found, move one square to the right, and write $x$ (a or b). |

## An Example

Input: $\underline{Æ}\, w \quad w \in \{1\}^*$

Output: $\underline{Æ}\, w^3$

Example: $\underline{Æ}$111Æ Æ Æ Æ Æ Æ Æ Æ Æ Æ Æ Æ Æ Æ

$M =$