- Similarly, given an $i$,

  we can find $w_i$, the string at index $i$.

  [Exercise].

- What if we could number the turing machines also, so that a 'number' either represents a unique TM, or it does not represent any TM?

- The requirement is only this much. So, it is OK even if there are two different numbers for the same TM.
- This can be done, by encoding the 'moves' ($\delta$) of the turing machine using binary strings. [Details later.]

$$M \equiv move_1 \cdot move_2 \cdot move_3 \cdot \ldots\ldots move_n$$

- $move_1$ is of the form

  $\delta(q_1, a) = (q_2, b, D)$ where $D$ is $L$ or $R$.

  These five things uniquely represent one move:

  $q_1$ (state-old), $a$ (symbol-old),
  $q_2$ (new state), $b$ (new symbol), $D$ (direction).

- Encoding of $move_i$ must capture these.

- So let us assume we can encode a machine M as

$$\text{move}_1 \cdot \text{move}_2 \cdot \;\cdots\cdots\; \text{move}_n$$

[you may think of the details to fill in].

- Now, one machine may have different encodings (moves written in a different order, or states/symbols in different order).

- But one encoding fixes the machine.
- If we take the 'value' of this binary string as the number for the machine,
  - One number corresponds to exactly one machine, or no machine.

Now, consider an (infinite) table:

| | $M_1$ | $M_2$ | $\cdots\cdots$ | $M_j$ | $\cdots\cdots$ | $M_{1059}$ | $\cdots\cdots$ |
|---|---|---|---|---|---|---|---|
| $w_1$ | | | | | | | |
| $w_2$ | | | | | | | |
| $\vdots$ | | | | | | | |
| $w_i$ | | | | $\boxed{\phantom{x}}$ $Entry_{i,j}$ | | | |
| $\vdots$ | | | | | | | |

$$Entry_{i,j} = \begin{cases} 1 & \text{if } M_j \text{ accepts } w_i \\ 0 & \text{otherwise.} \end{cases}$$

Now we are ready to define a language, $L_d$, using the diagonal entries in this table.

Def$^n$: $L_d = \{ w_i : \text{Entry}_{i,i} = 0 \}$

Claim: $L_d$ is NOT recognizable!
(There is no machine $M_j$ that recognizes $L_d$).

Why?

- If there was a machine $M_j$ that recognized this language $L_d$, what would be the entry $j,j$ in this table?
- Or, can $M_j$ accept the string $w_j$?

- $Entry_{j,j} = 1$ means
  $M_j$ accepts $w_j$;
  But by defn of $L_d$, $w_j$ is not in $L_d$!
  ($w_j \in L_d$ only if $entry_{j,j} = 0$)
- $Entry_{j,j} = 0 \Rightarrow$ OK, $w_j \in L_d$ (by defn of $L_d$)
  but $M_j$ does not accept $w_j$!
- So, there is no $M_j$ that recognizes $L_d$.

In other words,
there is no machine $M_j$
that accepts strings in $L_d$
and does not accept strings not in $L_d$.

- That proves our claim:
$L_d$ is not turing recognizable.