$$\{w\#w \mid w \in \{0,1\}^*\}$$

A DTM $M = (Q, \{0,1\}, \Gamma, \delta, q_s, q_a, q_r)$ does this as follows:

$$\begin{aligned}
\delta(q_s, \sqcup) &= (q_r, \sqcup, R) \\
\delta(q_s, \#) &= (e_0, \#, R) \\
\delta(e_0, \sqcup) &= (q_a, \sqcup, R) \\
\delta(e_0, 0) &= (q_r, 0, R) \\
\delta(e_0, 1) &= (q_r, 1, R) \\
\delta(q_s, 0) &= (r_0, x, R) \\
\delta(r_0, 0) &= (r_0, 0, R) \\
\delta(r_0, 1) &= (r_0, 1, R) \\
\delta(r_0, \sqcup) &= (q_r, \sqcup, R) \\
\delta(r_0, \#) &= (t_0, \#, R) \\
\delta(t_0, x) &= (t_0, x, R) \\
\delta(t_0, 1) &= (q_r, 1, R) \\
\delta(t_0, 0) &= (m_1, x, L) \\
\delta(q_s, 1) &= (r_1, x, R) \\
\delta(r_1, 0) &= (r_1, 0, R) \\
\delta(r_1, 1) &= (r_1, 1, R) \\
\delta(r_1, \sqcup) &= (q_r, \sqcup, R) \\
\delta(r_1, \#) &= (t_1, \#, R) \\
\delta(t_1, x) &= (t_1, x, R) \\
\delta(t_1, 0) &= (q_r, 0, R) \\
\delta(t_1, 1) &= (m_1, x, L)
\end{aligned}$$

$\{w\#w \mid w \in \{0,1\}^*\}$ (continued)

$$
\begin{aligned}
\delta(m_1, x) &= (m_1, x, L) \\
\delta(m_1, \#) &= (m_0, \#, L) \\
\delta(m_0, 0) &= (m_0, 0, L) \\
\delta(m_0, 1) &= (m_0, 1, L) \\
\delta(m_0, x) &= (q_n, x, R) \\
\delta(q_n, 0) &= (r_0, x, R) \\
\delta(q_n, 1) &= (r_1, x, R) \\
\delta(q_n, \#) &= (e_1, \#, R) \\
\delta(e_1, x) &= (e_1, x, R) \\
\delta(e_1, \sqcup) &= (q_a, \sqcup, R) \\
\delta(e_1, 0) &= (q_r, 0, R) \\
\delta(e_1, 1) &= (q_r, 1, R)
\end{aligned}
$$

Shift input right and add # at the left end

A DTM $M = (Q, \{0,1\}, \Gamma, \delta, q_s, q_a, q_r)$ does this as follows:

$$\begin{aligned}
\delta(q_s, 0) &= (q_0, \#, R)\\
\delta(q_s, 1) &= (q_1, \#, R)\\
\delta(q_0, 0) &= (q_0, 0, R)\\
\delta(q_0, 1) &= (q_1, 0, R)\\
\delta(q_0, \sqcup) &= (q_2, 0, L)\\
\delta(q_1, 1) &= (q_1, 1, R)\\
\delta(q_1, 0) &= (q_0, 1, R)\\
\delta(q_1, \sqcup) &= (q_2, 1, L)
\end{aligned}$$

In state q_2, move to the left end.

$$\{ww \mid w \in \{0,1\}^*\}$$

Repeat until all symbols are marked:

> Mark the first unmarked symbol from the left end by $\vee$.
>
> Mark the first unmarked symbol from the right end by $\wedge$.

Repeat until all symbols are unmarked:

> Move to the left end.
>
> Find the first symbol that is marked with $\vee$.
>
> Unmark the symbol and store in the state.
>
> Find the first symbol that is marked with $\wedge$.
>
> > If none found halt and reject.
>
> Unmark the symbol.
>
> > If not same as the stored one, halt and reject.

Halt and accept.

$$\{0^{2^n} \mid n \geq 0\}$$

Repeat the following steps:

> If number of unmarked 0's is odd and more than one halt and reject.
>
> Move left to right marking every other unmarked 0.
>
> If single unmarked 0 halt and accept.

Primality testing: Example from "Automata and Computability" by Dexter Kozen

Input: $\#aaa\cdots aa\#$
Problem: Decide if p is prime.
If $p\ =\ 0$ or $p\ = 1$ halt and reject.

Erase the first a and replace the last a by \$.

Repeat until the first non-blank symbol is \$:

(Invariant: the index of the first non-blank symbol is prime.)

Mark the first non-blank symbol with $\wedge$.

Mark all symbols until the first non-blank symbol with $'$.

Repeat until the right end of the tape:

(erase all symbols in positions that are multiples of the current prime number.)

Erase the next non-blank symbol with the mark $\wedge$.

Shift the $'$ marks and the $\wedge$ mark to the next block.

If \$ is marked with $\wedge$ then halt and reject.

Move to the left end of the tape.

Halt and accept.

TM as a Function Computer

M is said to compute f iff: starting with w on the tape

the TM ends up with $f(w)$ on the tape in the accept state.

Partial function.

Incrementing a binary number

$w = w_1 w_2 \cdots w_n$: input binary number

assume that the input tape contains $\# w_1 w_2 \cdots w_n$

Move to the right-most end;

Repeat the following steps:

Current symbol is 0: change it to 1; move to the left end; halt and accept;

Current symbol is 1: change it to 0; move left by 1 position;

Current symbol is #:

change it to 1;

shift right by one position the non-blank part of tape;

move to the left end;

write a #;

halt and accept;

Generate the lexicographically next binary string

$w = w_1 w_2 \cdots w_n$: a string over the alphabet $\{0, 1\}$

assume that the input tape contains $\# w_1 w_2 \cdots w_n$

Move to the right-most end;

Repeat the following steps:

Current symbol is 0: change it to 1; move to the left end; halt and accept;

Current symbol is 1: change it to 0; move left by 1 position;

Current symbol is #:

change it to 0;

shift right the non-blank part of the tape by one position;

move to the left end;

write a #;

halt and accept;

Summary

- TMs can recognize languages; can decide as well.
- TMs can compute functions: partial functions.
- Standard computer instructions can be simulated using TM instructions.

Can write TM programs at high level.

Examples in chapter 3.