A Turing Machine has an infinite tape of characters and can read & write to tape, then stop as it pleases.

TM $t = \langle Q, \Sigma, \Gamma, q_0, q_a, q_r, \delta \rangle$

Q = a finite set (states)

Σ = an alphabet $\sqcup \notin \Sigma$ (input alphabet)

Γ = an alphabet $\{\sqcup\} \cup \Sigma \subseteq \Gamma$ (tape alphabet)

$q_0 \in Q$ (start state)

$q_a \in Q$ (accept state) $q_a \neq q_r$

$q_r \in Q$ (reject state)

δ (transition function) : $(Q - \{q_a, q_r\}) \times \Gamma$

$\rightarrow (Q \times \Gamma \times \{L, R\})$

TM t is in config $c_0 = ua[q_i]bv$ $v, u \in \Gamma^*$

then the next config $c_1 = \ldots$ $a, b \in \Gamma$

1) Consult $\delta(q_i, b) = (q_j, c, dir)$

2) If dir = L,

$c_1 = u[q_j]acv$

If dir = R,

$c_1 = uac[q_j]v$

A TM t steps from c_n to c_{n+1} ($c_n \Rightarrow c_{n+1}$) iff

$$\frac{\delta(q_i, b) = (q_j, c, L)}{ua[q_i]bv \Rightarrow u[q_j]acv}$$

$$\frac{\delta(q_i, b) = (q_j, c, R)}{ua[q_i]bv \Rightarrow uac[q_j]v}$$

18-2)

The transitive, reflexive closure of $\Rightarrow$
written $\Rightarrow^*$

we call runs

A TM t runs from c_n to c_m $(c_n \Rightarrow^* c_m)$

$$\frac{}{c_0 \Rightarrow^* c_0}\ \text{(reflexive)} \qquad \frac{c_i \Rightarrow c_j \qquad c_j \Rightarrow^* c_k}{c_i \Rightarrow^* c_k} \qquad \frac{\sqcup\, c_i \sqcup \Rightarrow^* c_k}{c_p \Rightarrow^* c_k}$$

(option 1: add blanks as needed

A string $w \in \Sigma^*$ is accepted by TM t
written as accepts (t, w)
iff

$$[q_0]\, w \Rightarrow^* u\, [q_a]\, v$$
$$u, v \in \Gamma^*$$

Option 2:

$$\text{accepts}(t, w) ::= \exists n \in \mathbb{N}, m \in \mathbb{N}.\ \sqcup^n [q_0]\, w \sqcup^m \Rightarrow^* u\, [q_a]\, v$$

```
interface Tape {
  char ~~heat~~ head();
  Tape left();
  Tape ~~mog~~ right();
}
```

```
class Something : Tape {
  char h; Tape u, v;
  head() -> h          right() -> v, tol(u, h)
  left() -> u, ton(h, v)
}
      tol(u', h') =
           new Something(u, h ++ u' :: h'
```

```
Class Nothing {
 tol ( something (      , w,
```

8-31

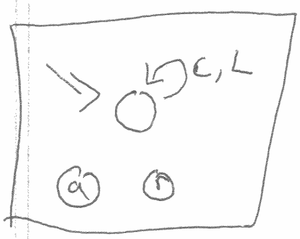
$L(TM\ t) = \{ w \mid w \in \Sigma^* \text{ s.t } \text{accepts}(t, w) \}$

$A = L$

$\text{rejects}(t, w) :=$

$[q_0]\, w \Rightarrow^* u\, [q_r]\, v$

$R(TM\ t) = \{ w \mid w \in \Sigma^* \text{ s.t } \text{rejects}(t, w) \}$



$\exists t \in TM$

$\overline{A(t)} \neq R(t)$ $\quad A(t) \cup R(t) \neq ALL$

however

$d \in DFA$ $\quad \forall d \in DFA$

$\overline{A(d)} = R(d)$ $\quad A(d) \cup R(d) = ALL$

Every TM t on string w could either

ACCEPT, REJECT, or LOOP

$w \in A(t)$ $\quad w \in R(t)$ $\quad w \in \overline{A(t) \cup R(t)}$

Every TM t is either

(1) A decider $\quad$ Either ACC or REJ

Σ_0 $\quad A(t) \cup R(t) = ALL$

$\overline{A(t) \cup R(t)} = \emptyset$

(2) ~~An~~ A recognizer

Σ_1 $\quad \{ACC, REJ, LOOP\}$

$A(t) \cup R(t) \neq ALL$

$w \in \overline{A(t) \cup R(t)}$

Σ_0 is a set of langs where $A \in \Sigma_0$ means $\exists t \in \text{decider}, L(t) = A$

Σ_1 ~~w~~ ... $A \in \Sigma_1$ means $\exists t \in \text{rec}, L(t) = A$

18-4

$\Sigma_0 = \Sigma_1$? $\Sigma_0 \subsetneq \Sigma_1$?

$\Sigma_1 \subsetneq \Sigma_0$? What about ALL?

A transducer f = TM except no q_a or q_r
but there is q_h

f transduces w into v
iff

$$[q_0]w \Rightarrow^* [q_h]v$$

An enumertor e = TM except no q_a or q_r
but there is q_p
but $\delta : Q \times \Gamma$
q_p is not the end

e prints w iff

$$[q_0] \Rightarrow^* [q_p]w$$

fun	accepts?	L
Meeley/Moore	DFA	REX
	PDA	CFG
transducer	TM	enemerator