18-1/

$w \in L(t)$ iff $[q_0]w \Rightarrow^* u[q_a]v$

// side-bar: $\delta : (Q - \{q_a, q_r\}) \times \Gamma \to Q \times \Gamma \times \{L, R\}$

$\Rightarrow^*$ is the reflexive, transitive closure of $\Rightarrow$

$$\frac{\delta(q_i, a) = (q_j, b, L)}{uc[q_i]av \Rightarrow u[q_j]cbv} \qquad \frac{\delta(q_i, a) = (q_j, b, R)}{u[q_i]av \Rightarrow ub[q_j]v}$$

$u[q_i]v \Rightarrow \sqcup u[q_i]v \sqcup$

$q_i, q_j \in Q \quad a, b, c \in \Gamma$
$u, v \in \Gamma^*$

DFA on w → accept / reject

PDA → accept / diverge / reject

TM → accept / diverge (spinning/ unique new config every transition) / loop (a config steps to itself)

loop: $[q_0]w \Rightarrow^* u[q_i]v \Rightarrow^+ u[q_i]v$

diverge: $\forall u, q_i, v.\ [q_0]w \Rightarrow^* u[q_i]v \to$
$u[q_i]v \Rightarrow^+ x[q_j]y$ s.t.
$\neg(x = u \wedge q_i = q_j \wedge v = y)$

Two categories of TMs

decider is a TM that does not diverge on any input

recognizer is a TM that diverges on some input

$A \in \Sigma_0$ iff $\exists t \in$ deciders. $L(t) = A$

↗ language

$\Sigma_0 \subset \Sigma_1$

$A \in \Sigma_1$ iff $\exists t \in$ recognizers. $L(t) = A$

8-2

Is $\Sigma_1 \subset \Sigma_0$? Every language could have a decider.

Is ALL = Σ_1? All problems can be solved by TMs.

Enumerator is "like" a CFG "machine" : string -> bool

CFG/REX = "printer" : -> set(string)

$(Q, \Sigma, \Gamma, q_0, \delta, q_P)$

$w \in L(e)$ iff $[q_0] \Rightarrow^* u\,[q_P]\,w$

q_P is NOT a final state (i.e. you transition FROM it)

"diverging" "decide" for enumerators?

↳ print any order, maybe multiple times ↳ printing "in order" and once

---

Transducers := machines with output

f is a "computable function" → Turing transducer

$f(x) = y$ iff $[q_0]x \Rightarrow^* u\,[q_{halt}]\,y$

"decide" "diverging"

- total function — partial function

Coq is for total programming