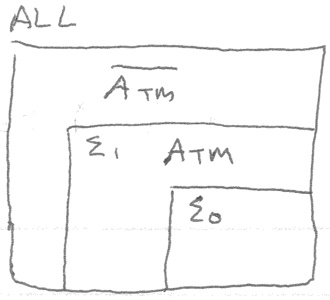
25-1



1. $|ALL| \neq |\Sigma_1|$
2. $A_{TM} \notin \Sigma_0$ (undecidable)
3. $A \in \Sigma_0$ iff $A \in \Sigma_1 \wedge \bar{A} \in \Sigma_1$

   $\Rightarrow$

   $\overline{A_{TM}} \notin \Sigma_1$

"A" is Undecidable $\equiv$ $A \notin \Sigma_0$ (no idea how long it takes)

"A" is Uncomputable $\equiv$ $A \notin \Sigma_1$ (no computer can do it)

no better strategy than testing

Computable function (transducer)

$f : \Sigma^* \to \Sigma^*$ (TM with w as the input reaches HALT/ACC with $f(w)$ on tape)

$w \to f(w)$

"011 + 100" $\to$ "111"

Mapping - Reducible

A language X is mapping-reducible to a language Y

$(X \leq_m Y)$

There exists a computable function, f

$\forall w \in \Sigma^*, \quad w \in X$ iff $f(w) \in Y$

- $\Sigma = \{0,1\}$ $\quad X = \{0^n \mid n \text{ is prime}\}$ $\quad Y = \{1^n \mid n \text{ is a prime}\}$
- $\Sigma = \{0,1\}$ $\quad X = \{01^n \mid n \text{ is even}\}$ $\quad Y = \{0^n \mid n \text{ is } \cancel{\text{odd}}\}$

  $f(1\Sigma^*) = 1 \qquad f(01^n \mid n \text{ is odd}) = 1$

  $f(01^n \mid n \text{ is even}) = 0$ or 0^{n+1}

Σ_0 and Σ_1

$A \leq_m B$ and $B \in \Sigma_0$, then $A \in \Sigma_0$

$\quad f \qquad\qquad M \qquad\qquad$ fun $w \Rightarrow M(f(w))$

$A \leq_m B$ and $A \notin \Sigma_0$, then $B \notin \Sigma_0$

25-2 (1) $A_{TM} \notin \Sigma_0$

(2) $\forall \cancel{X}, Y.\ X \leq_m Y$ and $X \notin \Sigma_0$, then $Y \notin \Sigma_0$

$X = A_{TM}$

$\Downarrow$

$Y \notin \Sigma_0$ if $A_{TM} \leq_m Y$

$\Downarrow$ $\forall w.$

$\exists f : \Sigma^* \to \Sigma^*.\ w \in A_{TM}$ iff $f(w) \in Y$

$A_{TM} = \{ \langle M, w \rangle \mid M$ is a TM and M accepts $w \}$

$Y = HALT_{TM} = \{ \langle M, w \rangle \mid M$ is a TM and M halts on $w \}$

$f(\langle M, w \rangle)$ return #true iff M accepts w

given answer to $\forall m', w'.\ HALT(\langle m', w' \rangle)$

1. Run $HALT_{TM}(\langle M, w \rangle)$
2. If accepts, simulate M on w and return what it does
3. If not, we reject

A_{TM}

$E_{TM} = \{ \langle M \rangle \mid M$ is a TM and $L(M) = \emptyset \}$

$A_{TM}(\langle M, w \rangle)$ = construct M' s.t. $L(M') = \emptyset$ iff M accepts w

run $E_{TM}(M')$

$M'(x)$ = simulate M on w
if accepts, reject x
o.w. accept x

$EQ_{TM} = \{ \langle A, B \rangle \mid A$ and B are TMs and $L(A) = L(B) \}$

$E_{TM}(\langle M \rangle) = EQ_{TM}(\langle M,$ "On input w, reject w" $\rangle)$

f

25-3

$REG_{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) \in REG \}$

$A_{TM} \le_m REG_{TM}$

$A_{TM}(\langle M, w \rangle) = REG_{TM}(\langle M' \rangle)$

f

$L(M') \in REG$ iff M accepts w

$M'(\text{input } x) =$ if (M accepts w) ie w/ simulation

be regular = accept x ($\Sigma^* \in REG$

else

not be regular = accept x if $x = 0^n 1^n$

$L(M') = \Sigma^*$ if M accepts w

$\{0^n 1^n\}$ o.w

$CFL_{TM} \notin \Sigma_0$

$\{ w \# w \}$

Rice's Theorem = "Any non-trivial property of Turing machines is undecidable"

Our computers are DFAs

but scalable

so LBA

LBA is a linear-bounded automaton

It is a TM with a finite tape, $|tape| = k \times |w|$

$|tape| = |w| + 1$ $\quad \Sigma = \{0, 1\}$

$\Sigma \quad \Gamma \quad |\Gamma| = 1$ million

$= |w|$

A TM could	An LBA can
accept	accept
reject	reject
loop (return to previous config)	loop
diverge (always see new config)	~~diverge~~

25-4) To diverge, there must an infinite number of configs

Suppose $M \in LBA$ w/ q states and g symbols in Γ
and $|w| = n$

$$g^n \times q \times n$$

total number of configs

A_{LBA} is decidable:

On input $<M, w>$ ($M \in LBA$)

run M on w for $g^n q n$ steps

if acc $\Rightarrow$ acc, rej $\Rightarrow$ rej, o.w reject

In real world, q = proceessor state

n = memory (in bits)

head = p.c. (part of q)

$100mB = 2^{20} \times 100 \times 8 = n$ $\quad q = 64 \times 64$

$2^{23} \times 100$ $\quad = 2^6 \times 2^6$

2^{30} $\quad = 2^{12}$

$2^{2^{30}} \times 2^{12} = 2^{2^{30}+12}$

A_{DFA}, A_{CFG}, E_{DFA}, E_{CFG}, $\in$ LBAs $\qquad \forall$ CFF $\in$ LBA

$E_{LBA} = \{<M> \mid M$ is an LBA and $L(M) = \emptyset\}$

$A_{TM} \leq_m E_{LBA}$ $\qquad$ in: $<M, w>$ $\longrightarrow$ out: $<LBA'>$

$L(LBA') = \emptyset$ iff M accepts w

LBA' "verifies" histories of M

history = config # config # ... $\qquad$ config = tape [q] tape

verified history = initial config is $[q_0]\,w$

last config is $u\,[q_a]\,v$

if config i is $u\,[q_i]\,v$ and config $i+1$ is $u'\,[q_j]\,v'$ then

$u\,[q_i]\,v \Rightarrow u'\,[q_j]\,v'$