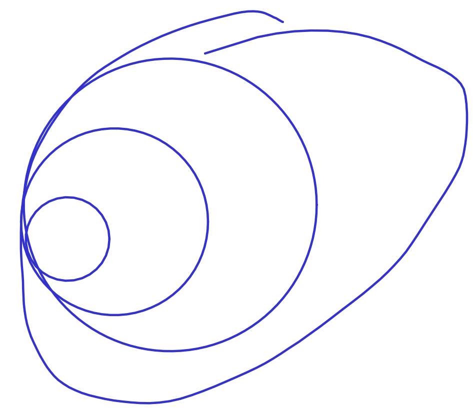
CSE322 Theory of Comput. (L23)



$L1 \leq_T L2$ $\qquad$ $L1 \leq_m L2$

Turing-reduction

A different kind of reduction

Many-one reduction
(aka. mapping reduction)

Thm: HALT is undecidable.

Proof: Assume HALT is decided by D. Construct decider B for A-TM using D.

B(<M,w>):

// Construct M', w' such that B(<M,w>) accepts iff D(<M',w'>) accepts.

// Does this work? M' = M, w' = w ???

D is unable to distinguish between halt-accept and halt-reject.

$\boxed{\text{Call D(<M',w'>). If D accepts, B accepts. If D rejects, B rejects.}}$

Lemma:-

<M,w> ∈ ATM iff <M',w> ∈ HALT

def DATM(<M,w>):

w' = w

M' = M'(x):

a. Run M on x.

b. When M accepts x, M' accepts x.

c. When M rejects x, M' loops and never halts.

def Red(<M,w>):

w'=w

Construct M' as given below

output <M',w'>

* If DHALT() accepts, DATM accepts

If DHALT() rejects, DATM rejects.

DHALT(<M',w>)

many-one Reduction

L1 $\le_m$ L2 if ... there is a TM / algorithm / reducer R s.t.

* R (instance of L1) -> instance of L2
* x is in L1 iff R(x) is in L2 Correctness claim

A-TM $\le_m$ HALT A-TM m-reduced to HALT

L1 $\le_T$ L2 if ... there is a TM / algorithm / reducer R s.t. R decides (or recognizes) L1 by making calls to a TM for deciding (or recognizing) L2.

Properties of many-one reduction

For any language L,

Is $L \leq_m L$?

R(x): return x

$x \in L$ iff $R(x) = x \in L$.

$L1 \leq_m^{R12} L2$ $L2 \leq_m^{R23} L3$

Suppose language L1 reduces to language L2 and L2 reduces to language L3.

Show: $L1 \leq_m L3$

def R13 (x) : // x should be an instance of type L1

y = R12(X) → $x \in L1$ iff $y \in L2$

$x \in L1$ iff $R1(x) \in L3$.

z = R23(Y) → $Y \in L2$ iff $z \in L3$

output z

If $L1 \leq_m^{R12} L2$, show L1-complement $\leq_m$ L2-complement $\overline{\text{A-TM}} \leq \overline{\text{HALT}}$

$x \in L1$ iff $R12(x) \in L2$

$x \in \overline{L1}$ iff $R12(x) \in \overline{L2}$

<= : At least "as hard as"

$x \in L1$ iff $R12(x) \in L2$

Suppose $L1 \leq_m^{R12} L2$

L2 is atleast as hard as L1

(L1 is not harder than L2)

1. If L2 is decidable ... ? by D2

then L1 is decidable

def D1(x):

y = R12(X)

Run D2(y) and do whatever D2 does

D1 always halts ✓

if $x \in L1$, D1 accepts x

if D1 accepts x, $x \in L1$

2. If L1 is undecidable ... ?

then L2 is undecidable.

3. If L2 is recognizable ... ?

then L1 is recognizable

4. If L1 is not recognizable ... ?

then L2 is not recognizable.

Show: E-TM $\le_m$ EQ-TM

def Red($\langle M \rangle$):

construct a TM N s.t. N rejects all strings $L(N) = \phi$

return $\langle M, N \rangle$

Claim:- $\langle M \rangle \in$ ETM iff $\langle M, N \rangle = \text{Red}(\langle M \rangle) \in$ EQTM.

Show: EQ-TM is NOT R.E.

Level1: Reduce $\overline{ATM}$ <= EQ-TM

Construct a computable function f() s.t.

(a) f(M,w) = (M1,M2) s.t.

(b) M doesn't accept w iff L(M1) = L(M2)

$L(M_1) = \emptyset$

$M_2 = M$

M accepts w iff $L(M_2) \neq \emptyset$

def Red(<M,w>):

Construct M_1 that rejects all strings

Construct $M_w(x)$:

if $x \neq w$: rejects

else : simulates M(w) and does whatever M does

return $\langle M_1, M_w \rangle$

If M accepts w, $L(M_w) = \{w\}$

If M doesn't accept w $L(M_w) = \emptyset$

Show: EQ-TM-complement is NOT R.E.

Level1: Reduce ATM-complement <= EQ-TM

M accepts w iff $L(M_2) = \Sigma^*$

$\Updownarrow$

ATM $\leq_m$ EQ-TM

$L(M_1) = \Sigma^*$

def Red(<M, w>):

Construct M_1 that accepts every string

Construct M_2 : $M_2(x)$: if $x \neq w$: accept

else do whatever $M(x)$ does.

return $\langle M_1, M_2 \rangle$

$$L(M_2) = \begin{cases} \Sigma^* & \text{if } w \in L(M) \\ \Sigma^* - \{w\} & \text{if } w \notin L(M) \end{cases}$$

Thm: EQ-TM is neither RE nor coRE !!!

Show that ATM cannot be many-one reduced to ETM.

Proof by contradiction: Assume $ATM \leq_m ETM$.

$$\Updownarrow$$

$\overline{ATM} \leq_m \overline{ETM}$

$\overline{ATM} \notin RE$, $\overline{ETM} \in RE$

Contradiction

$\overline{ETM} = \{\langle M \rangle \mid L(M) \neq \phi\}$

Non-trivial property of TM Lang.

P : any "non-trivial" property (language) of languages of TMs.

P = { ⟨M⟩ : M is a TM and L(M) has some property}

* There are some TMs in P and some TMs not in P.
* Whether ⟨M⟩ is in P or not depends ONLY on L(M).

LP1 = { ⟨M⟩ : L(M) is infinite }

LP2 = { ⟨M⟩ : M has more than 10 states }

LP3 = { ⟨M⟩ : M accepts 110101 }

LP4 = { ⟨M⟩ : L(M) = {} }

LP5 = { ⟨M⟩ : there is some w for which M(w) moves left at some point}

LP6 = { ⟨M⟩ : M uses at most 10 extra cells on empty input }

LP7 = { ⟨M⟩ : M has even number of states }

LP8 = { ⟨M⟩ : M('hello') has 'world' written on tape at some point of time}

LP9 = { ⟨M⟩ : any string accepted by M can be recognized by some TM with even number of states}

LP10 = {⟨M⟩ : M is a TM}

$\{\langle M\rangle : L(M) \subseteq \Sigma^*\}$

Rice's theorem

$\{\langle M\rangle$: M visits both q_{acc} & $q_{rej}\}$

LP = { <M> : L(M) has non-trivial property of language P} is undecidable !!!

Can you decide if ... ?

$\{\langle M\rangle : L(M)$ has even parity strings $\}$

L' : reg lang.

* TM accepts only (all or some) strings of length 12345 ?

$\{\langle M\rangle : L(M) = L'\}$

* TM accepts a finite set of languages ?
* TM accepts some regular language ?
* TM accepts some CFL ?

$\{\langle M\rangle : L(M)$ is regular$\}$

* TM accepts language L(N) where N is a given TM ?

[Rice] non-trivial Semantic properties of TMs are not decidable.