

Other Undecidable Problems

Strategy for more problems:

1. Assume for purpose of contradiction that problem P decid.
2. Show that if P were decidable, another problem Q (that we have shown undecidable) would be decidable. CONTR.

Th. $E_{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is empty} \}$ is undecidable.

Proof idea: We show if E_{TM} were decidable, then A_{TM} would be.

So suppose E_{TM} decided by R . We want decider S for A_{TM} .

Could we use R directly for S ? R accepts $\langle M \rangle$ iff $L(M) = \emptyset$

So if $L(M) = \emptyset$, then S should reject w .

But if $L(M) \neq \emptyset$, then we're stuck!????

1

Other Undecidable Problems

Trick: Instead of running R on $\langle M \rangle$, we run it on a different machine $\langle S^{M,w} \rangle$ (defined for specific M and w).

$S^{M,w}$ takes input x , and :

1. if $x \neq w$, it rejects.
2. if $x = w$, run M on input w , and accept if M does.

$S^{M,w}$ accepts at most one input, w ; and it accepts w iff M accepts w .

so $L(S^{M,w}) = \emptyset$ iff M rejects w

so $L(S^{M,w}) \neq \emptyset$ iff M accepts w

This allows us to use problem E_{TM} !

2

E_{TM} is Undecidable

Proof: Suppose E_{TM} were decidable by R . We show that A_{TM} would be decidable, a contradiction.

We first define $S^{M,w}$ for given M and w as:

On input x

1. if $x \neq w$, reject.
2. if $x = w$, run M on w and accept if M does.

We now define decider S for A_{TM} as follows:

On input $\langle M, w \rangle$

1. use M to construct $S^{M,w}$
2. Run R on $\langle S^{M,w} \rangle$
3. If R accepts, S rejects; if R rejects, S accepts.

3

Other Undecidable Problems

We can use other problems than A_{TM} to get a contradiction!

Th. $EQ_{TM} = \{\langle M1, M2 \rangle \mid M_i \text{ is a TM, } L(M1) = L(M2)\}$ is undecidable.

Proof: We show if EQ_{TM} is decidable, E_{TM} is, Contr.

Proof idea: EQ_{TM} tests if 2 languages =, E_{TM} if language = ϕ

So E_{TM} is a special case of EQ_{TM}

Proof: Suppose R decides EQ_{TM} . Construct S to decide E_{TM} as follows.

S: on input $\langle M \rangle$

1. Run R on $\langle M, Rej \rangle$, where Rej is TM with $L(Rej) = \phi$
2. If R accepts, S accepts; if R rejects, S rejects.

CONTR.!

4

Mapping Reducibility

Intuition: to solve a problem, reduce it to another

Ex. to solve problem of getting A in course, reduce to getting A on quizzes, on homework, and on final.

We've already used this idea to prove more problems undecidable

Ex : If HALT_{TM} is decidable, then A_{TM} is decidable.

But A_{TM} already shown undecidable, CONTR.

Mapping reducibility makes reduction of problems precise

$A \leq_m B$: intuitively, if B has a solution, can use to solve for A.

We will use computable mapping m of A to B

5

Computable Functions

TM as computer of output, not recognizer

Output is what is on tape when halt

**Def.: $f : \Sigma^* \rightarrow \Sigma^*$ is a *computable* function if there is some TM M which computes f:
on input w, M halts with just f(w) on its tape.**

Note: M must always halt!

Ex. HW 3 problem 6: TM that, with i and j on tape, computes $i*j$ (if on just 1 tape, and erased i and j, then this TM would compute * function)

Other computable functions: factorial($\langle n \rangle$)

gcd($\langle m, n \rangle$) prime($\langle n \rangle$) = $\langle \text{nth prime no} \rangle$

Transformers of TM:

$f(\langle M \rangle) = \langle M' \rangle$ where $L(M) = L(M')$, and M' never moves its tape off the left end

6

Mapping Reducibility

Def. Language A is *mapping reducible* to language B , written $A \leq_m B$, if there is a computable function

$f: \Sigma^* \rightarrow \Sigma^*$, where for every w in Σ^* ,
 $w \in A \iff f(w) \in B$

f is called a *reduction*

Can test membership in A by membership in B

Ex. $E_{TM} \leq_m EQ_{TM}$.

$f: \Sigma^* \rightarrow \Sigma^*$

$\langle M \rangle \mapsto f(\langle M \rangle) = \langle M, \text{Rej} \rangle$ (where $L(\text{Rej}) = \emptyset$)

f appends to $\langle M \rangle$ the repr. $\langle \text{Rej} \rangle$; $f(w) = w \text{ ow.}$

$\langle M \rangle \in E_{TM} \iff f(\langle M \rangle) \in EQ_{TM}$

f is computable:

7

Uses of Mapping Reducibility

Th: If $A \leq_m B$ and B is decidable, then A is decid.

Proof: Let D be the decider for B , and f the reduc.
 $A \rightarrow B$. We define decider D' for A as follows:

D' : on input w

1. compute $f(w)$
2. Run D on input $f(w)$; if D accepts, D' accepts.
 if D rejects, D' rejects.

Since $w \in A \iff f(w) \in B$ by def. map. reduc.,
 and D' accepts $w \iff D$ accepts $f(w)$,
 we conclude D' accepts A .

Since D is a decider, D' is a decider.

Therefore, A is decidable.

8

Uses of Mapping Reducibility

Cor: If $A \leq_m B$ and A is undecidable, then B is undecidable.

Proof: Suppose B were decidable. By the prev.

Th., since $A \leq_m B$, then A would be decidable.

Contradiction.

Ex. If $A_{TM} \leq_m \text{HALT}_{TM}$ and A_{TM} undecidable

$\longrightarrow \text{HALT}_{TM}$ undecidable

To show $A_{TM} \leq_m \text{HALT}_{TM}$, need computable

$f : \langle M, w \rangle \longrightarrow \langle M', w' \rangle$

$\langle M, w \rangle$ in A_{TM} iff $\langle M', w' \rangle$ in HALT_{TM}

9

Uses of Mapping Reducibility

TM F computes f as follows:

F : on input $\langle M, w \rangle$

1. Construct M' :

M' : on input x

1. Run M on x .

2. If M accepts, M' accepts.

If M rejects, loop infinitely.

2. Output $\langle M', w \rangle$

Then $\langle M, w \rangle$ in $A_{TM} \iff M \text{ accepts } w \iff$

$M' \text{ halts on } w \iff$

$\langle M', w \rangle$ in HALT_{TM}

10

Uses of Mapping Reducibility

Th. E_{TM} is undecidable (*revisited*)

A_{TM} reduced to E_{TM}

Qu: is it a mapping reducibility?

$\langle M, w \rangle \longrightarrow \langle M' \rangle$

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

$\langle M, w \rangle \in A_{TM} \iff \langle M' \rangle \in E_{TM}$

so we showed $A_{TM} \leq_m \overline{E_{TM}}$

not $A_{TM} \leq_m E_{TM}$ (*can't be done*)

We could still prove the th., because

E_{TM} decidable $\iff \overline{E_{TM}}$ decidable

11

Turing-Recognizability & Mapping Reducibility

Th. If $A \leq_m B$ and B Turing-recognizable, then
 A is Turing-recognizable.

Proof: Suppose B is recognized by TM R , and
 $f: A \rightarrow B$ is a reduction. We define recognizer TM
 M for A as follows:

M : on input w

1. compute $f(w)$
2. run R on input $f(w)$; if R accepts, M acc.
 if R rejects, M rejects.

Since $w \in A \iff f(w) \in B$, by def. map. reduc.,
 and M accepts $w \iff R$ accepts $f(w)$
 we conclude M accepts A . M is certainly a
 recognizer. Therefore, A is Turing-recognizable.

12