Chapter 5 – Reducibility

CSC 320

Reductions

Basic idea: we don't have to solve problems from scratch. Use existing problems to solve new problems.

Can be useful in practice (e.g. SAT solvers) but is also a way to show that new problems are e.g., undecidable

Say, e.g., we have a language L and we want to know whether it is decidable.

Suppose we can show that *if* we *could* decide L, *then* we *could* decide A_{TM}

We conclude that we *can't* decide L

How do we show *if* . . . *then*? Reductions!

$HALT_{TM}$

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

Theorem: $HALT_{TM}$ is undecidable.

IDEA: Use machine for $HALT_{TM}$ to solve A_{TM} .
Proof: Assume there is a TM R which decides $HALT_{TM}$. Construct the TM S to decide A_{TM} as follows:

1. S calls R on input $\langle M, w\rangle$
2. If R rejects, reject.
3. If R accepts, simulate M on w until it halts (since R accepted we know this will happen).
4. If M accepts, accept. Else if M rejects, reject.

We have shown that if there is such an R then A_{TM} is decidable, but that is false, so our assumption is false and $HALT_{TM}$ is undecidable.

Test for Emptiness

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

Theorem: E_{TM} is undecidable.

Proof: For any M, w we can let M_1 be the TM which takes as input string x:

1. If $x \neq w$, M_1 rejects.
2. If $x = w$, M_1 runs M on input w and accepts if M does.

Now we construct TM S to decide A_{TM}. Let R be a hypothetical TM which decides E_{TM}:

S has input $\langle M, w\rangle$

1. Use $\langle M, w\rangle$ to construct M_1 as described above.
2. Run R on $\langle M_1\rangle$.
3. If R accepts, reject; if R rejects, accept.

If R decided the emptiness of $L(M_1)$, then S decides A_{TM}. Therefore R can't exist and E_{TM} is undecidable.

Equivalence of TM's

$EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1, M_2 \text{ are TM's and } L(M_1) = L(M_2)\}$.

Theorem: EQ_{TM} is undecidable.

IDEA: Can use such a TM to test if a language is empty.

Proof: Let R be the (hypothetical) TM which decides EQ_{TM}. Construct S to decide E_{TM} as follows: S is given $\langle M\rangle$ as input.

1. Run R on $\langle M, M'\rangle$,where M' is the TM which rejects all strings.

2. If R accepts, S accepts. If R rejects, S rejects.

If $L(M)$ is empty, then R accepts and S accepts. If $L(M)$ is not empty then R rejects and S rejects. So S decides E_{TM} . But E_{TM} is undecidable so R cannot exist and EQ_{TM} is undecidable.

Mapping Reducibility

We don't want to do an *ad hoc* argument every time we get a new problem. We will formalize what we have been doing using *mapping reducibilities*.

A function $f : \Sigma^* \to \Sigma^*$ is a *computable function* if some TM on every input w halts with just $f(w)$ on its tape.

Examples: $f(w) = w^R$, modifying a description of a TM $\langle M \rangle$ in a simple way.

Formal Definition

A language A is *mapping reducible* to a language B (written $A \leq_m B$ (if there is a computable function $f : \Sigma^* \rightarrow \Sigma^*$ where for every w

$$w \in A \text{ iff } f(w) \in B$$

Observation: $A \leq_m B$ iff $\overline{A} \leq_m \overline{B}$.

Proving Problems are Decidable

Theorem: If $A \leq_m B$ and B is decidable then A is decidable.

Proof: Let M be a TM which decides B. Construct a TM N which decides A using M as a subroutine:

1. Compute $f(w)$

2. Run M on $f(w)$

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

Proving Turing-recognizability

Theorem: If $A \leq_m B$ and B is Turing-recognizable then A is Turing-recognizable.

Corollary: If $A \leq_m B$ and A is not Turing-recognizable then B is not Turing-recognizable.

Recall that $\overline{A_{TM}}$ is not Turing recognizable. We use this to show that EQ_{TM} is neither Turing recognizable nor co-Turing recognizable.

Unrecognizability of EQ_{TM}

Theorem EQ_{TM} is not Turing recognizable

Proof We show $A_{TM} \leq_m \overline{EQ_{TM}}$, which implies $\overline{A_{TM}} \leq_m EQ_{TM}$.

Define f as follows: $f(\langle M, w\rangle) = \langle M_1, M_2\rangle$ where M_1, M_2 are machines such that

1. M_1 rejects all inputs;
2. For any input x, M_2 runs M on w and accepts if M accepts.

If $\langle M, w\rangle \in A_{TM}$ then M accepts w. But then $L(M_1) = \emptyset$ rejects everything and $L(M_2) = \Sigma^*$, so $\langle M_1, M_2\rangle \in \overline{EQ_{TM}}$.

If $\langle M, w\rangle \notin A_{TM}$ then $L(M_1) = L(M_2) = \emptyset$ reject and $\langle M_1, M_2\rangle \notin \overline{EQ_{TM}}$. So

f is a mapping reduction from A_{TM} to $\overline{EQ_{TM}}$.

Unrecognizability of $\overline{EQ_{TM}}$

Theorem: EQ_{TM} is not co-Turing recognizable

Proof: To show $\overline{EQ_{TM}}$ is not Turing recognizable, we show a reduction from A_{TM} to EQ_{TM}. Define g as follows: $g(\langle M, w\rangle) = \langle M_1, M_2\rangle$ where M_1, M_2 are machines such that

1. M_1 accepts all inputs;
2. For any input x, M_2 runs M on w and accepts if M accepts.

Similar argument as before except M accepts w iff both M_1 and M_2 accept.

Examples

Does M halt on the empty tape?

Is there any string which M halts on?

Given a TM and a state q, does the TM ever enter state q?