CS 3719 (Theory of Computation and Algorithms) – Lecture 20

Antonina Kolokolova*

February 24, 2011

0.1 Examples of reductions

Recall that $A \leq_m B$ iff there exists a computable function f such that $\forall x \in \Sigma_A^* \; x \in A$ iff $f(x) \in B$. The notation suggests that "A is at most as hard to solve as B". Often we use the reduction to prove hardness for problems of comparable complexity, but sometimes it is not the case: A can be a lot simpler than B.

Example 1. $\{uu | u \in \{0,1\}^*\} \leq_m A_{TM}$
We need to define $f(x) =< M, w >$. Take a Turing machine, say, which accepts an empty string and rejects everything else. So the description of M is simple: $Q = \{q_0, q_{accept}, q_{reject}\}$, $\Sigma = \{0,1\}$, $\Gamma = \{0,1,\sqcup\}$ $\delta = \{(q_0, 0) \rightarrow (q_{reject}, 0, R), (q_0, 1) \rightarrow (q_{reject}, 0, R), (q_0, \sqcup) \rightarrow (q_{accept}, 0, R)$. Now, define $f(x) =< M, \epsilon >$ if $x = uu$ for some $u \in \{0,1\}^*$ and $f(x) =< M, 0 >$ otherwise. This is a computable function, and it has the desired property that $x \in \{uu | u \in \{0,1\}^*\} \iff f(x) \in A_{TM}$.

The most useful use of reduction is, though, to prove that certain problems are hard by reducing known hard problems to them.

Example 2. Let $HaltB = \{\langle M \rangle |$ TM M halts on blank input $\}$. We will show that $A_{TM} \leq_m$ HaltB
Let $x \in \Sigma^*$, and assume that $x = \langle M, w \rangle$ where M is a Turing Machine.
(If x is not of this form, then we can let $f(x)$ be anything not in HaltB. In general we will assume that the input is "well-formed" since this will always be easy to test for.)

We will let $f(x) = \langle M' \rangle$ where M' works as follows on a blank tape (we don't care what M' does on a non-blank tape).

*The material in this set of notes came from many sources, in particular "Introduction to Theory of Computation" by Sipser and course notes of U. of Toronto CS 364 and SFU CS 710.

M' :on input x'
write w on the tape
simulate M running on input w;
if and when M halts and accepts, M' halts and accepts;
if and when M halts and rejects, M' goes into an infinite loop.

Clearly f is computable. It is also easy to see that $x \in A_{TM} \Leftrightarrow f(x) \in$ HaltB, since M accepts $w \Leftrightarrow M'$ halts on blank tape.

Corollary 19. *$\overline{HaltB}$ is not semi-decidable.*

Proof. We know HaltB is semi-decidable but not decidable, so $\overline{\text{HaltB}}$ is not semi-decidable. □

Example 3. Let NE be the language consisting of Turing Machines that accept a nonempty language. That is, NE $= \{\langle M\rangle \mid M$ is a Turing Machine and $\mathcal{L}(M) \neq \emptyset\}$.

Lemma 20. *NE is semi-decidable, but not decidable. (Hence, $\overline{NE}$ is not semi-decidable.)*

Proof. We will design a Turing Machine M_{NE} such that NE $= \mathcal{L}(M_{NE})$ M_{NE} behaves as follows:

M_{NE}: on input x
if x is not of the form $\langle M\rangle$, reject
for $i = 1$ to ∞
run M on all inputs of length $\leq i$ for i steps;
if and when it is discovered that M accepts some input, M_{NE} halts and accepts.

That is, M_{NE} does the following (let $\Sigma = \{0, 1\}$): run M on all inputs of length ≤ 1 for 1 steps; if M accepted ϵ or 0 or 1 in one transition then accept; otherwise
run M on all inputs of length ≤ 2 for 2 steps; if M accepted one of $\epsilon, 0, 1, 00, 01, 10, 11$ then accept; otherwise
run M on all inputs of length ≤ 3 for 3 steps; etc.

Clearly NE $= \mathcal{L}(M_{NE})$, so NE is semi-decidable.

To show that NE is not decidable we will prove that HaltB $\leq_m$ NE.
Let x be an input for HaltB, and assume that x is well-formed, that is, $x = \langle M\rangle$.
Define $f(x) = \langle M'\rangle$ where M' works as follows.

M' on input x
erase x and run M on the blank tape;
if and when M halts, M' halts and accepts.

Clearly $\langle M \rangle \in \text{HaltB} \Leftrightarrow \langle M' \rangle \in \text{NE}$, so $\text{HaltB} \leq_m \text{NE}$, so NE is not decidable.

This type of reduction, doing the same on all inputs can help with a surprising number of problems. I like to call them "All-or-nothing" reductions. The resulting language is either Σ^* (and thus includes every subset one might be interested in) or $\emptyset$. □

Example 4. Let T be the language of "total" machines, that is, of machines that halt on every input. $\text{T} = \{\langle M \rangle \mid M \text{ is a Turing Machine that halts on every input}\}$.

Lemma 21. *Neither T nor $\overline{T}$ is semi-decidable.*

Proof. We first show that $\text{HaltB} \leq_m \text{T}$, implying that $\overline{\text{HaltB}} \leq_m \overline{\text{T}}$, implying that $\overline{\text{T}}$ is not semi-decidable.

Let $f(\langle M \rangle) = \langle M' \rangle$ (and if input to f is not a proper encoding of a Turing machine, it is an M' that just loops on every input). We will do an "all-or-nothing" reduction again:

M' : on input x
erase input and run M on the blank tape.
if M accepts, accept. If M rejects, reject.

We have M halts on the blank tape $\Leftrightarrow$ M' halts on every input, so we are done.

We next show that $\text{HaltB} \leq_m \overline{\text{T}}$, implying that $\overline{\text{HaltB}} \leq_m \text{T}$, implying that T is not semi-decidable. This reduction is a bit tricky.
Assume that the input for HaltB, and assume is well-formed: $\langle M \rangle$.
Let $f(\langle M \rangle) = \langle M' \rangle$ where M' is as follows.

M' :On input x
Simulate M on the blank tape for $|x|$ steps;
if M halts within $|x|$ steps, then M' goes into an infinite loop;
if M doesn't halt within $|x|$ steps, then M' halts (and, say, accepts).

Clearly M halts on the blank tape $\Leftrightarrow$ M' is not a total machine. □

Example 5. Let $\text{All} = \{\langle M \rangle \mid M \text{ is a Turing Machine that accepts every input}\}$.

We will show in one shot that *All* is neither semi-decidable nor co-semi-decidable by reducing T to it: $T \leq All$. Now, $f(\langle M \rangle) = \langle M' \rangle$ where M' is the same as M except every occurrence of q_{reject} in M is changed to q_{accept}.