

CS525 Winter 2012 \ Chapter #6 Preparation

Ariel Stolerman

6.4)

Let $A'_{TM} = \{\langle M, w \rangle \mid M \text{ is an Oracle TM and } M^{A_{TM}} \text{ accepts } w\}$. Following is a proof that A'_{TM} is undecidable relative to A_{TM} :

Assume by contradiction that A'_{TM} is decidable relative to A_{TM} , then there exists a TM R with oracle access to A_{TM} , denoted $R^{A_{TM}}$ that decides A'_{TM} . We will define the following TM S :

“For input w :

- If w is not a proper encoding of a TM with oracle access to a decider of A_{TM} , *reject*.
- Otherwise, denote that machine $T^{A_{TM}}$.
- Simulate R on $\langle T^{A_{TM}}, T^{A_{TM}} \rangle$. If it accepts, *reject*. Otherwise, *accept*.”

This is basically applying the diagonalization argument in the same manner as done for A_{TM} . Note that S is a TM with oracle access to A_{TM} , as it uses R which is a decider for A'_{TM} with oracle access to A_{TM} , therefore $S \in A'_{TM}$. S is defined to return the opposite of any TM with oracle access to A_{TM} when simulated on itself. Since S is a candidate for A'_{TM} itself, when given the code of itself $\langle S \rangle$, it is defined to do the opposite of what it does – a contradiction. Therefore R cannot exist, and so A'_{TM} is undecidable relative to A_{TM} .

6.13)

Following is a proof that for each m , the theory $Th(F_m)$, where $F_m = (Z_m, +, \times)$ is a model over the group $Z_m = \{0, 1, 2, \dots, m - 1\}$ and the relations $+, \times$ computed modulo m , is decidable:

Denote a simple addition or multiplication modulo m , consisting of vectors of size 3 where the first and second rows are the arguments and the third row is the result. Without loss of generality, we will look at $+$. For any given m , there are m^2 true additions, i.e. combinations that denote a correct addition modulo m . We can check for a given string over $\Sigma_3 =$

$\left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}$ that is represents a true addition modulo m as follows:

- Make sure all rows represent numbers in $\{0, 1, \dots, m - 1\}$.
- If yes, check whether the third row fits the result expected for the first two rows (out of the m^2 options).
- If yes, it is a true addition modulo m , otherwise it is not.

Similar method can be applied on multiplication modulo m .

Next, for any given formula of the form $\varphi = Q_1 x_1 \dots Q_l x_l [\psi(x_1, \dots, x_l)]$, where Q_i are quantifiers, x_i are variables and ψ is a quantifier-free formula with the $+$ and \times modulo m relations (and the standard operators, e.g. \neg), we check all possible assignments of all x_i for the corresponding quantifier Q_i as follows:

Denote $I_i(x_1, \dots, x_l) = \psi(x_1, \dots, x_l)$. For any $i > 0$:

- If $Q_i = \exists$: $I_{i-1}(x_1, \dots, x_{i-1}) = \bigvee_{k=0}^{m-1} I_i(x_1, \dots, x_{i-1}, k)$

- If $Q_i = \forall: I_{i-1} = \bigwedge_{k=0}^{m-1} I_i(x_1, \dots, x_{i-1}, k)$

Eventually, we get $I_0 \equiv \varphi$, and it can be calculated whether it is a true statement in finite time for any given m . Therefore $Th(F_m)$ is decidable for any m .

6.14)

Following is a proof that for any two languages A, B a language J exists such that $A \leq_T J$ and $B \leq_T J$:

Let $J = \{\bar{0}a \mid a \in A\} \cup \{\bar{1}b \mid b \in B\}$, where $\bar{0}, \bar{1}$ are symbols that do not appear in any $w \in A \cup B$. We can then define a TM M that is decidable relative to J as follows:

“For input w :

- Check with the oracle of J whether $\bar{0}w \in J$.
- If it accepts, *accept*. Otherwise, *reject*.”

Clearly $w \in A \Leftrightarrow \bar{0}w \in J$. In a similar manner we can construct a TM that decides B by mapping w to $\bar{1}w$ and querying the oracle of J . Therefore both $A \leq_T J$ and $B \leq_T J$, as required.

6.15)

Following is a proof that for any language A , a language B exists, where $A \leq_T B$ and $B \not\leq_T A$:

For any language A we can define the following language B as follows:

$$B = \{\langle M, w \rangle \mid M \text{ is a TM with oracle access to } A \text{ and } M^A \text{ accepts } w\}$$

$A \leq_T B$:

Define a TM N with oracle access to A as follows:

“On input w :

- Query the oracle of A if $w \in A$.
- If yes, *accept*. Otherwise, *reject*.”

Now we show a construction of a TM R that given an oracle access to B decides A :

“On input w :

- Construct the description of N as defined above.
- Query the oracle of B if $\langle N, w \rangle \in B$.
- If yes, *accept*. Otherwise, *reject*.”

Clearly N decides A by simply query A 's oracle, that is N is a TM with oracle access to A . An oracle for B helps to decide whether N^A accepts w , which happens iff $w \in A$. Therefore R is a decider for A with an oracle access to B , so $A \leq_T B$.

$B \not\leq_T A$:

Assume by contradiction that $B \leq_T A$, then there exists a TM S with an oracle access to A such that S^A decides B . We can construct the following TM C :

“On input w

- If w is not a proper encoding of a TM, *reject*. If yes, denote $M := w$.

- Simulate S on $\langle M, M \rangle$, which uses an oracle access to A .
- If S accepts, *reject*. Otherwise, *accept*."

C is defined as a TM which uses S , which is a TM with oracle access to A , therefore C is a TM with oracle access to A .

Consider an enumeration of all TMs with oracle access to A , applied on their own description:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle \dots$
M_1^A	Accept	Reject	Accept
M_2^A	Reject	Accept	Accept
M_3^A	Accept	Reject	Reject
	...		
C	Reject	Reject	Accept

Since C is a TM with oracle access to A , then there exists some integer j such that C is the TM in that index in the enumeration above. But then if $\langle C, C \rangle \in B$ then C^A accepts $\langle C \rangle$, but then C^A rejects $\langle C \rangle$ by the definition of C , in contradiction. Therefore S cannot exist, and so $B \not\leq_T A$.

6.16)

Following is a proof that there exist two languages A, B that are Turing-incomparable, i.e. $A \not\leq_T B$ and $B \not\leq_T A$:

6.18)

Following is a proof that there exist languages that are not recognizable by an oracle Turing machine with oracle for A_{TM} :
 The proof follows corollary 4.18 (page 178). The set of legal encodings of Turing machines with oracle access to A_{TM} is countable (the same argument as used for proving the set of all standard Turing machines is countable). However the set of all languages is uncountable: any language A can be mapped to an infinite binary vector (sequence of 0's and 1's) corresponding to its characteristic vector χ_A , as described in the book. Therefore the set of all languages cannot be put into a correspondence with the set of all Turing machines with oracle access to A_{TM} , therefore some languages exist that are not recognizable by any Turing machine with oracle access to A_{TM} .