Lecture #14: Oracle Computations
Exercises and Review

Questions for Review

$\mathcal{S}$-Programs with Oracles

1. What is an ***$\mathcal{S}$-program with an oracle***? How is it different from the $\mathcal{S}$-programs that have already been defined and studied in this course?
2. Suppose that $G : \mathbb{N} \to \mathbb{N}$ is some partial (or total) function. Describe what happens when an ***oracle statement***

$$V \leftarrow O(V)$$

is executed during a ***G-computation*** of a program $\mathcal{P}$ with an oracle.

3. When a regular $\mathcal{S}$-program $\mathcal{P}$ is executed on inputs $x_1, x_2, \ldots, x_n \in \mathbb{N}$, the computation either halts (with an output value returned) or it runs indefinitely (and on output value is defined).

   Describe a *third* thing that can happen when an ***$\mathcal{S}$-program $\mathcal{P}$ with an oracle*** is executed on inputs $x_1, x_2, \ldots, x_n \in \mathbb{N}$. What is the output considered to be in *this* case?
4. Suppose $G : \mathbb{N} \to \mathbb{N}$ is a partial (or total) function, $\mathcal{P}$ is an $\mathcal{S}$-program with an oracle, and m is a positive integer. Define the function

$$\Phi^{(m)}_{\mathcal{P},G} : \mathbb{N}^m \to \mathbb{N}$$

as precisely as you can.

G-Computable Functions

Now suppose that $G : \mathbb{N} \to \mathbb{N}$ is a ***total function***.

5. What does it mean for a partial function to be ***G-computable*** (called ***partially G-computable*** or ***G-partial recursive* in older references**)?

6. What does it mean for a *total* function to be ***G-computable***, or ***G-recursive***?
7. How is the set of computable partial or total functions related to the set of ***G-computable functions***?
8. How is the set of computable total functions related to the set of ***G-computable functions***?
9. How is the set of computable partial or total functions related to the set of ***I-computable functions***, where $I : \mathbb{N} \to \mathbb{N}$ is the identify function?
10. Is G a ***G-computable function***? Why (or why not)?
11. Briefly explain why the set of all G-computable functions is a ***PRC class***.
12. What is a ***relativized theorem***? What is a ***relativized proof***?
13. Suppose that $f : \mathbb{N} \to \mathbb{N}$ is a partial (or total) function, $G, H : \mathbb{N} \to \mathbb{N}$ are ***total*** functions, that f is partially G-computable, and G is H-computable. What else can be concluded about f? Why?
14. Describe the relationship between the set of ***computable*** total functions and the set of ***G-computable*** functions, when $G : \mathbb{N} \to \mathbb{N}$ is a ***computable*** total function.

f-Computable Functions

Let $f : \mathbb{N}^n \to \mathbb{N}$ be a total function such that $n \geq 2$.

15. Say what it means for a function g to be ***(partially) f-computable***.
16. Is f an f-computable function? Why (or why not)?

Relativization of Universality

17. Describe a new (and very minor) restriction on $\mathcal{S}$-programs with oracles that is needed to ensure that the ***length*** of a program is computable from its encoding.
18. State the ***Relativized Universality Theorem*** and say (briefly) how this is proved.
19. State the ***Relativized Step Counter Theorem*** and say (briefly) how *this* is proved.
20. Describe the function $\{u\} : \mathbb{N} \to \mathbb{N}$ where $u \in \mathbb{N}$.
21. State the ***Finiteness Theorem*** and say (briefly) how this is proved.
22. State the ***Strengthened Parameter Theorem.*** Why is this considered to be a "strengthened" version of the Parameter Theorem?
23. State the ***Relativized and Strengthened Parameter Theorem.*** Why does the proof relativize?

Problems To Be Solved

1. Recall that if $S \subseteq \mathbb{N}$ then the ***characteristic function*** of S is the total predicate

$$p_S : \mathbb{N} \to \{0, 1\}$$

   such that, for all $n \in \mathbb{N}$,

$$p_S(n) = \begin{cases} 1 & \text{if } n \in S, \\ 0 & \text{if } n \notin S. \end{cases}$$

   Recall, as well, that the ***complement*** $\overline{S}$ of a set $S \subseteq \mathbb{N}$ is the set

$$\overline{S} = \{n \in \mathbb{N} \mid n \notin S\}.$$

   Prove that $p_{\overline{S}}$ is p_S-computable for every set $S \subseteq \mathbb{N}$.

2. Use this to prove that there exist sets $A, B \subseteq \mathbb{N}$ such that

   (a) p_B is p_A-computable, but

   (b) p_B is ***not*** many-one reducible to p_A.