CSc/Math 573 homework 5 8 April 2008
DUE: Tue 22 April 2008

READING

See the course web page.

PROBLEMS

1. Rice’s Theorem for Function Indices

Consider this version of Rice’s Theorem stated in terms of function indices rather than set indices.

Theorem: (Rice). Let $P(\phi)$ be a property of functions†.

$\theta P = \{ i \mid P(\phi_i) \text{ is true} \}$ is decidable iff $\theta P = \varnothing \ \vee \ \theta P = \mathbb{N}$.

Prove this version of Rice’s theorem using the Recursion Theorem. *HINT:* Write A for θP. Suppose P is non-trivial, so that $A \neq \varnothing$ and $A \neq \mathbb{N}$. If A were recursive, then $A \leq_m \overline{A}$ via some t.c.f. f (Why?). Apply the Recursion Theorem to f to obtain a contradiction.

2. No Total Destroyer Function

The recursion theorem states:

$$(\forall t.c.f.\ f)\ (\exists x)\ (\forall y)\ \phi_x(y) = \phi_{f(x)}(y) \tag{RT}$$

(a) Write down the logical negation of equation (RT), i.e.,

$$(\exists t.c.f.\ f)\ \cdots \tag{$\tilde{}$RT}$$

Since (RT) is a theorem, the assertion (˜RT) must be *false*.

(b) Apply this observation to solve Homework 3.11, p. 57 of the Text.

3. Bounded Quantification

The characteristic function c_P of a k-ary predicate $P(\overline{y})$ is a k-ary function such that $c_P(\overline{y}) = 1$ whenever $P(\overline{y})$ is true, and $c_P(\overline{y}) = 0$ otherwise. A predicate P is said to be a *LOOP predicate* iff c_P is computable by a *LOOP* program.

If Q is a $k+1$-ary predicate, define its *bounded existential quantification* as the $k+1$-ary predicate EQ given by

$$EQ(\overline{n},\ m) \quad \Leftrightarrow \quad (\exists i \leq m)\ [Q(\overline{n},\ i)]$$

Similarly, its *bounded universal quantification* is the predicate

$$AQ(\overline{n},\ m) \quad \Leftrightarrow \quad (\forall i \leq m)\ [Q(\overline{n},\ i)]$$

(a) Prove: If Q is a *LOOP* predicate, then so are its bounded existential and bounded universal quantifications.

(b) What if the word ‘‘bounded’’ is removed in the above so that, for example, we get the k-ary predicate $EQ(\overline{n}) \ \Leftrightarrow \ (\exists i)\ Q(\overline{n}, i)$? Are the resulting predicates still *LOOP* predicates?

---

†for example, ‘‘ϕ is bijective’’, ‘‘ϕ is total’’. The property must be independent of the particular TM that may implement ϕ.

4. Complexity Class Comparisons

What, if any, is the relationship between each of the following pairs of complexity classes? Possible answers are: one is properly contained in the other (e.g., $\subsetneq$), they are equal, or the classes are *incomparable* with respect to inclusion (neither one contains the other). Prove your answers.

(a) $\textbf{DSPACE}(n^2)$ and $\textbf{DTIME}(n)$

(b) $\textbf{DTIME}(2^n)$ and $\textbf{DTIME}(3^n)$.

(c) $\textbf{NSPACE}(2^n)$ and $\textbf{NSPACE}(5^n)$

(d) $\textbf{DSPACE}(n)$ and $\textbf{DTIME}(\lceil \log n \rceil^n)$

5. Parameters Matter

A *clique* in an undirected graph G is a subgraph wherein every two nodes are connected by an edge. A *k-clique* is a clique with k nodes. Let $<G>$ denote the encoding of a graph as a string in some suitable manner. It is known that the following problem is **NP**-complete:

$$CLIQUE = \{ (<G>, k) \mid G \text{ is an undirected graph with a } k\text{-clique } \}.$$

A *triangle* in an undirected graph that is a 3-clique. Show that the problem $TRIANGLE \in \textbf{P}$, where

$$TRIANGLE = \{ <G> \mid G \text{ is an undirected graph with a 3-clique } \}.$$

(Moral: fixing the value of a problem parameter may change the complexity of the resulting ‘‘special case’’ of the problem.)

6. L_U Analogue

Define

$$U_{\textbf{NP}} = \{ <M, x, 1^t> \mid M \text{ is a } \textit{nondeterministic} \text{ TM that accepts input } x \text{ within } t \text{ steps } \}$$

Show that $U_{\textbf{NP}}$ is **NP**-complete.

$U_{\textbf{NP}}$ defined in this problem is the "**NP** complexity class analogue" of L_U (which you recall was complete in class **CE**.) *HINT:* Since $3SAT \in \textbf{NP}$, there is some fixed *nondeterministic* TM M_{3SAT} that decides $3SAT$ in (nondet.) time cn^k, for some fixed positive c and k. This also hints at the reduction to use ...