Introduction to Theoretical Computer Science

Exercise Sheet: Week 5. Answers

(1) Let *Even* be the decision problem $(\mathbb{N}, \{n : n \text{ is even}\})$ and *Odd* be $(\mathbb{N}, \{n : n \text{ is odd}\})$. Give m-reductions between the two problems.

Use $n \mapsto n + 1$. You **cannot** *just flip the answer, because that can't be done with m-reductions.*

(2) We remarked that (computable) predicates are (computable) functions. A function $f : \mathbb{N} \to \mathbb{N}$ is also a predicate: its *characteristic predicate* $\chi_f(x, y)$ is true iff $y = f(x)$. Show that f is computable iff χ_f is computable.

Assume f is computable. Then to compute $\chi_f(x, y)$, just see if $f(x) = y$. Conversely, assume χ_f is computable. To compute $f(x)$, compute $\chi_f(x, y)$ for $y = 0, \ldots$ until it returns true, which it will because f is a function.

(3) Suppose that Q_1 and Q_2 are semidecidable queries over the same domain D. Show that $Q_1 \cup Q_2$ and $Q_1 \cap Q_2$ are semidecidable.

Trivial

(4) Suppose that L is a decidable language over some alphabet Σ. Show that the language L^* is decidable.

Just check all possible decompositions of the input string.

(5) Following on from question 2: Given a computable predicate $\psi(x, y)$, is it decidable whether ψ is the characteristic predicate of some function f?

No. We need to check that ψ is functional, in that each x maps to exactly one y. Because we have no bounds, it is only semi-decidable to check that ψ maps a given x to some *y (compute each $\psi(x, y)$ in turn until it says true), and doing it for all x is Π^0_2; and it's only co-semi-decidable that ψ maps a given x to* at most one *y (compute as before, returning false when you see the second 'true') (and so still Π^0_1 to do this for all x).*

To actually prove undecidability, we can reduce H to this problem, thus. Let M be a machine. Define the predicate ψ as follows:

$$\psi(x, y) = \begin{cases} 1 & \textit{if } M \textit{ halts after exactly } y \textit{ steps} \\ 0 & \textit{otherwise} \end{cases}$$

This is clearly a computable predicate, and is the characteristic predicate of the partial function

$$f(x) = \textit{execution time of } M \textit{ or } \perp$$

which is a function iff M has a finite execution time, i.e. halts.

More slickly and more powerfully, we can prove Π^0_2-hardness directly *by reduction from UH. Define* $\psi' : \mathbb{N}^2 \to \{0, 1\}$ *by*

$$\psi'(x, y) = 1 \quad \textit{iff } M \textit{ halts in exactly } y \textit{ steps on input } x$$

ψ' *is the graph of a total function iff* M *halts on every input.*