UG3 Computability and Intractability (2011-2012)

Exercise Sheet 1

The deadline for this coursework is 4pm on Thursday 20 October; submit your solutions to the ITO. Please note that multiple submissions are not allowed. Your work will be marked and returned, but the mark will not contribute to the overall mark for the course. Note that the marks for questions are not always related to their length or difficulty.

In your answers you should aim for clarity, conciseness and correctness; look at your answers with an objective eye, e.g., imagine that somebody else gave them to you to check. You should therefore make an early start to give yourself time to consider your answers before submitting the final version. This coursework should take about 10 hours of work to complete.

1. For this question we use the notation of Note 1 and assume that both inputs and outputs are encoded as natural numbers.

   Recall that a partial function from $\mathbb{N}$ to $\mathbb{N}$ is a function that takes a natural number as input and is either undefined for that input or returns a natural number as result.

   Let f and g be partial functions from $\mathbb{N}$ to $\mathbb{N}$. We say that g *extends* f if for all natural numbers n such that $f(n)$ is defined we have that $g(n)$ is also defined and $g(n) = f(n)$. (In other words g might be defined for more inputs than f but whenever f is defined then so is g and it agrees with f.)

   Define the function f by

$$f(n) = \begin{cases} P_n(n) + 1, & \text{if } P_n \text{ returns a result on input } n, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

   (a) Do you think that f is computable (see p.5 of Note 1)? Give a brief justification of your answer (bear in mind that there are infinitely many P_n). *[3 marks]*

   (b) Let g be any computable function that extends f. Prove that g is not total, i.e., there is at least one natural number m such that $g(m)$ is not defined. *[4 marks]*

   (c) Use the preceding two parts to define a total function that is not computable and explain why this is the case. [*Note:* You *must* answer this question as asked, i.e., an alternative proof that does not rely on parts (a) and (b) is not acceptable.] *[4 marks]*

2. Design a Turing machine `copy` with input alphabet $\{ 0, 1, \$, @ \}$ having the following behaviour:

- When started with an input string $\$B@$ where B is any binary string (including the empty string) `copy` accepts the input and the final tape is $\$B@B$. It doesn't matter which symbol is scanned when the machine stops so you can make this as simple as possible; there are two natural choices.
- When started with any other input we do not care about the behaviour of the machine—so you need not even think about this case.

(a) Create a file containing your machine in the format used by the Turing machine simulator. Submit a printout of this and email a copy to your tutor. [*6 marks*]

(b) Run your machine with input \$@ and submit a printout of the trace (make all transitions printable). [*2 marks*]

(c) Run your machine with input \$101@ and submit a printout of the trace (again make all transitions printable). [*3 marks*]

3. Let Σ be a finite alphabet and L a recursive language over Σ, i.e., there is a Turing machine that recognizes L and always halts. Define the language L' (over Σ) by

$$L' = \{\, w \mid \text{ there is a string } u \in \Sigma^* \text{ s.t. } wu \in L \,\},$$

i.e., a string is in L' if and only if it is a prefix of a string in L.

(a) Prove that there is a Turing machine that recognizes L' (the machine might not always halt). For this part you should outline the proposed machine in fairly high level terms but take care of any important points. (You may assume that there is a Turing machine that lists the strings of Σ^* using some sensible convention.) [*7 marks*]

(b) Can the assumption that L is recursive be weakened to L is recursively enumerable? (Of course even if the answer is "yes" it would probably be necessary to change the construction used in the preceding part.) For this part you need only justify your answer in very high level terms. [*4 marks*]

Rahul Santhanam, Sunday 8 October