University of Sussex
Informatics Spring 2020

Limits of Computation
Feedback to Exercises 1
Dr Bernhard Reus

Effective Procedures, Problems, Sets and Logic
(covers Lectures 1 – 2)

1. What do we mean by *effective procedure* in this module? What was an "effective procedure" for Alan Turing?

   Answer: An *effective procedure* is one that

   - is defined in terms of a finite number of exact instructions (each instruction can be in turn represented by a finite number of symbols).
   - must always produce the desired result in a finite number of steps.
   - can be carried out by a human with just paper and pencil (so it must be clear for us what the instructions mean!)
   - does not require any insight or ingenuity on the part of the human who carries it out, which means it is simple enough for a machine to do it too (and it does not need any "oracle").

   A function or procedure is "effectively calculable" meant for Turing that "its values can be found by some purely mechanical process". This he said, should be understood literally, leading to the first formalisation of computation in terms of what we now call "Turing machines".

2. For each problems below state whether it is a function problem (an optimisation problem) or a decision problem or whether it does not qualify as problem in our sense. Explain.

   (a) Given a finite array of integers what is the sum of all its values?
   Answer: Function Problem

(b) Given a an image of pixel size 1080 x 1920 in grayscale png format, does it depict a cat?
Answer: Not a problem, it's not definitely clear what "depicting a cat" is supposed to mean.

(c) Can an integer number n be divided by 7?
Answer: Decision Problem. Some might argue it is not clear enough what "can an integer be divided by 7" means, as it is not clearly stated which kind of division. In this case one could argue it is not a problem as we can't give a clear answer.

(d) Given the current published UK railway timetable, produce a connection with minimal changes from Hove to Birmingham University on any given day and for any departure time.
Answer: Optimisation (so Function) Problem

(e) Does a given Java program run on a Java SE10 virtual machine with input string "Turing" return the string "Icon"?
Answer: Decision Problem.

(f) What is the maximum value of a given function f on the real numbers in a given interval $[a, b]$?
Answer: Optimisation (so Function) Problem. Note that here a function "on real numbers" means a function from real numbers to the real numbers. One could argue that if it is not known what the codomain (result type) of f was, it would not be clear if we can say what a maximum value on this codomain is.

(g) What is the meaning of life?
Answer: Not a problem (neither uniform, nor do we know a definite answer).

3. When do we call a decision problem P-decidable? What is P in this case?

   Answer: If there is an effective procedure *of type P* that is a solution to this decision problem, i.e. it computes yes or no for any instance/input of the problem. P is here a language or any other description of notion of computation, in which the procedure is expressed/programmed.

   The following two questions recall notation about sets and logic that we will use in this module.

4. For the pairs of sets A and B in (a)-(e) below, explain whether $A \subset B$, whether $B \subset A$, whether $A = B$, and whether $A \neq B$.

   (a) $A = \mathbb{N} \times \mathbb{N}$ and $B = \mathbb{N}^2$
   Answer: $A = B$.

   (b) $A = \{ 1, 3, 5 \}$ and $B = \{ 1, 3, 5, 6 \}$
   Answer: $A \subset B$ so $A \neq B$.

   (c) $A = \{ 1, 3, 3, 3 \}$ and $B = \{ 1, 3 \}$
   Answer: $A = B$. This was difficult for some of you. Note that in sets we don't distinguish multiple occurrences of an element. So either an element (e.g. 3) is in the set or it is not.

   (d) $A = \{ x \in \mathbb{N} \mid x = x + 1 \}$ and $B = \emptyset$
   Answer: $A = B$. This was difficult for some of you. Note that there is no natural number x such that $x = x + 1$. So this is the empty set. Note that $\emptyset$ means empty set, i.e. $\{\}$.

   (e) $A = \{ x \in \mathbb{N} \mid \mathrm{even}(x) \ \wedge \ x < 11 \}$ and $B = \{ 0, 2, 4, 6, 8, 10 \}$
   Answer: $A = B$. Note that in computer science 0 is usually considered to be a natural number unless stated otherwise.

The following question is to recall logical connectives.

5. Which of the propositional logic formulae over propositions P and Q below are always true (in classical logic)? In cases where the formula is not always true state for what values of P and Q e.g. it does not hold.

   (a) $P \vee \neg P$ ($\vee$ denotes *disjunction*, $\neg$ denotes *negation*)
   Answer: always true. Note that in logic this law is also called "law of excluded middle" or "tertium non datur". This holds only for *classical logic*. That's the one we're using in general.

   (b) $P \Rightarrow Q$ ($\Rightarrow$ denotes *implication*)
   Answer: not always true, e.g. P true and Q false. Some students had problems with implication: true does not imply false, but false implies true as from an inconsistent assumption one can derive anything ("ex falso quod libet"). Classically we can also define $P \Rightarrow Q$ as $\neg P \vee Q$.

(c) $(P \wedge Q) \vee \neg P$ ($\wedge$ denotes *conjunction*)
Answer: not always true, e.g. P true and Q false

(d) $P \vee (P \Rightarrow Q)$
Answer: always true

(e) $P \Rightarrow (P \Rightarrow Q)$
Answer: not always true, e.g. P true and Q false

(f) $P \Rightarrow (Q \Rightarrow P)$
Answer: always true

(g) $(P \wedge (P \Rightarrow Q)) \Rightarrow Q$
Answer: always true ("Modus Ponens")