3C04 - Exercises 1

Sean Holden
Department of Computer Science
University College London

Exercises marked with a star are a little more difficult than others. Exercises marked with two stars are more challenging still.

The final exercise is particularly important.

1. One of the first decision problems presented in the course was as follows. An instance is a polynomial $p(x_1, x_2, \ldots, x_n)$ in the variables $x_1, x_2, \ldots, x_n$ (where n forms part of the instance) with integer coefficients. For example, we might have, $n = 2$ and,

   $$p(x_1, x_2) = c_0 + c_1 x_1 + c_2 x_1^2 x_2 + c_3 x_1 x_2^3$$

   where c_0, c_1, c_2, c_3 are integers. The question is: does the equation,

   $$p(x_1, x_2, \ldots, x_n) = 0$$

   have a solution which is all integers? Design an encoding scheme for this decision problem[1].

2. Design a Turing machine to perform unary division by 2. (It is assumed that we're only interested in integer division. So, for example, 7/2 = 3.)

3. Design a Turing machine to perform unary subtraction. (It is assumed that we're only interested in positive integers and we therefore give a result of zero if the true result would be negative. So, for example, 8 - 7 = 1, although 8 - 11 = 0.)

4. Let T be the finite alphabet $T = \{a, b, c\}$. Any finite string $x \in T^\star$ can be divided into blocks, each block containing contiguous copies of the same symbol, in the obvious way. For example,

   $$x = \underbrace{aaaaaaaaa}_{\text{block}}\underbrace{bbbbbbbbb}_{\text{block}}\underbrace{aaaaaaaa}_{\text{block}}\underbrace{cccccccccc}_{\text{block}}\underbrace{bbbbbbbbb}_{\text{block}}$$

   Any single symbol s appearing between two symbols t and u where $s \neq t$ and $s \neq u$ also counts as a block.

   For any $s \in T$ and any $x \in T^\star$, let $\texttt{blocks}(s, x)$ denote the number of blocks of the symbol s in the string x. For example, if we have,

   $$x = aaaabbcacbbbbbabbbabbbbbbccca$$

[1] Equations of this kind are called *diophantine equations*, after the Greek mathematician *Diophantus of Alexandria* of the third century BC.

then,

$$\begin{aligned}
\texttt{blocks}(a, x) &= 5 \\
\texttt{blocks}(b, x) &= 4 \\
\texttt{blocks}(c, x) &= 3
\end{aligned}$$

Design a TM, that decides the language,

$$L = \{x \in T^\star : \texttt{blocks}(a, x) = \texttt{blocks}(b, x) = \texttt{blocks}(c, x)\}$$

Be careful to state any assumptions that you make. Your solution should consist of a formal description of the TM, including a diagram describing f, and a short description in English of how your TM works.

5. Comment on the following argument. Let A be a finite alphabet, let a be any symbol in A, and let s be any infinite string composed of symbols from A. It is possible to design a Turing machines that tells us whether or not any such string s contains any such symbol a. The machine merely searches along s. It outputs a 1 if it eventually finds a and outputs a 0 otherwise.

6. $\star$Provide a discussion, of not more than two sides of A4 in length, of the following question. In particular, provide a specific answer to the question, and explain why you have reached your conclusion.

We are interested in a particular, informally stated decision problem, and we have formalised it by introducing an alphabet Σ and an encoding scheme, such that we obtain a corresponding language L. The encoding scheme possesses all the required properties. Also, we have demonstrated that L is not recursive*, and so we conclude that the problem is not solvable. Is it possible, by using a* different *encoding scheme (and possibly a different alphabet) to make the problem solvable?*