CSL705/CS355N: Theory of Computation

Tutorial sheet: Coding and Gödelization

1. We may code finite sets of natural numbers by the function fs which for any finite set $T = \{a_1, \ldots, a_m\}$, is defined as $fs(T) = \prod_{a_i \in T} p_{a_i+1}$ where p_n is the n-the prime (with $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, etc.).
   (a) What is $fs(\emptyset)$?
   (b) Prove that fs is primitve recursive.
   (c) Prove that the predicate $isfs : \mathbb{N} \to \{0, 1\}$ which determines whether a given number n encodes a finite set of natural numbers, is also primitive recursive.
2. Prove that the "*head*" and "*tail*" functions on lists are both primitive recursive.
3. Prove that every integer is primitive recursive.
4. Prove that every rational number (regarded as an infinite sequence of decimal digits) is primitive recursive.
5. What distinguishes a modern machine in a fundamental fashion from the URM machine is the fact that unlike the case of the URM, it is not possible to determine from the program text, the amount of memory a program may consume (compare this for instance with the $\rho(P)$ which we have freely used in many programs). This is mainly due to the fact that the architecture of the modern machine allows for *indexing* registers and *indirect* addressing mecahnisms. Suppose we add the following indirect addressing mechanisms to create a new and more "powerful" machine called "URM++".

opcode	instruction	semantics	Verbal description
4	$\mathsf{CI}(m, n)$	$R_n :=!R_{!R_m+1}$	Copy *indirectly* into R_n from (register referred to by) R_m
5	$\mathsf{AI}(m, n)$	$R_{!R_n+1} :=!R_m$	Assign *indirectly* the contents of register (referred to by) R_n the contents of R_m
6	$\mathsf{AII}(m, n)$	$R_{!R_n+1} :=!R_{!R_m+1}$	Assign *indirectly* the contents of register (referred to by) R_m to the register (referred to by) R_n

   Notice the +1 in each of the subscripts to the registers. Since our registers have only positive indices, whereas register contents could be 0, we map the content k in a register to denote the register R_{k+1}.
   (a) Write a (non-terminating) URM++ program which starting from the empty memory map gradually fills up all the registers with non-zero values.
   (b) Prove that the function ρ is a partial recursive function.
6. (a) Give a Gödel numbering of URM++ programs.
   (b) Show that there are primitive recursive functions which implement the indirect addressing instructions.
7. (a) Show that for any URM configuration, the contents of any register may be obtained by a primtive recursive function.
   (b) Show that the function ρ, which for any URM program yields the maximum index of register used, is primitive recursive.
8. Prove that $\mapsto$ is a primitive recursive function.