THEORY OF COMPUTATION
Problem session - 8

Prof. Dan A. Simovici

UMB

Problem 1: Let s be the code of an instruction in $\mathcal{S}$. Prove that the label, the variable number, the instruction type can be determined using recursive functions of s. Also, if s is the code of a conditional jump statement, prove that the label to which instruction with code s is pointing can also be determined using a primitive recursive function.

Problem 2: Find the program $\mathcal{P}$ such that $\#(\mathcal{P}) = 1000$.

Problem 3: Prove or disprove: if $f(x_1, \ldots, x_n)$ is a total function such that $f(x_1, \ldots, x_n) \leqslant k$ for all $x_1, \ldots, x_n$ and some constant k, then f is computable.

Problem 4: Let $f : \mathbb{N} \longrightarrow \mathbb{N}$ be a bijection. Prove that if f is computable, then so is f^{-1}.

Problem 5: Let $\mathrm{HALT}^1(x)$ be defined as

$$\mathrm{HALT}^1(x) = HALT(\ell(x), r(x))$$

for $x \in \mathbb{N}$. Show that HALT^1 is not computable.

Problem 6: The state of a program $\mathcal{P}$ was defined as a set of list of equations of the form $V = m$. Recall that the standard list of variables is

$$Y, X_1, Z_1, X_2, Z_2, \ldots$$

Note that the input variables occupy even numbered positions on this list (X_1 is on the 2^{nd} place, X_2 is on the 4^{th} place, etc.). The state of a program $\mathcal{P}$ is encoded by the number

$$S = [a_1, a_2, \ldots, a_n],$$

where a_i is the value assumed by variable V_i in the list above. Prove that:

1. The initial state of $\mathcal{P}$ is encoded by the number $\prod_{i=1}^{n} (p_{2i})^{x_i}$.
2. For a prime number p_i we have $p_i | S$ if and only if the state of the program contains the equation $V_i = a_i$, where $a_i \neq 0$.

Problem 1: Let s be the code of an instruction in $\mathcal{S}$. Prove that the label, the variable number, the instruction type can be determined using recursive functions of s. Also, if s is the code of a conditional jump statement, prove that the label to which instruction with code s is pointing can also be determined using a primitive recursive function.

Solution for Problem 1: Let $s = \#(I) = \langle a, \langle b, c \rangle\rangle$. The following primitive recursive function do the job:

$$\begin{aligned} \text{label}(s) &: \quad a = \ell(s), \\ \text{var}(s) &: \quad c + 1 = r(r(s)) + 1, \\ \text{instr}(s) &: \quad b = \ell(r(s)), \\ \text{label}'(s) &: \quad b \dot{-} 2 = \ell(r(s)) \dot{-} 2, \end{aligned}$$

where the last equality holds if $b > 2$.

Problem 2: Find the program $\mathcal{P}$ such that $\#(\mathcal{P}) = 1000$.
Solution for Problem 2: Suppose that $\mathcal{P}$ consists of instructions $I_1, \ldots, I_k$. Then, $[\#(I_1), \cdots, \#(I_k)] = \#(\mathcal{P}) + 1 = 1001$. Note that 1001 can be factored as $1001 = 7 \cdot 11 \cdot 13$ and that

$$1001 = 2^0 \cdot 3^0 \cdot 5^0 \cdot 7^1 \cdot 11^1 \cdot 13^1.$$

Therefore, $\mathcal{P}$ consists of 6 instructions $I_1, \ldots, I_6$. The first three unlabeled instructions are $Y \leftarrow Y$.

Solution cont'd

For the next three, the code is $\langle a, \langle b, c\rangle\rangle = 1$, so $2^a(2\langle b, c\rangle + 1) - 1 = 1$, that is $2^a(2\langle b, c\rangle + 1) = 2$, which means that $a = 1$ (so the label is A_1) and $\langle b, c\rangle = 0$, or $2^b(2c + 1) = 1$. In turn, this implies $b = 1$ and $c = 0$. The variable involved is still Y because $c = \#(V) - 1$ and the statement is $Y \leftarrow Y + 1$. The program is

$$
\begin{array}{ll}
 & Y \leftarrow Y \\
 & Y \leftarrow Y \\
 & Y \leftarrow Y \\
[A] & Y \leftarrow Y + 1 \\
[A] & Y \leftarrow Y + 1 \\
[A] & Y \leftarrow Y + 1
\end{array}
$$

Problem 3: Prove or disprove: if $f(x_1, \ldots, x_n)$ is a total function such that $f(x_1, \ldots, x_n) \leqslant k$ for all $x_1, \ldots, x_n$ and some constant k, then f is computable.

Solution for Problem 3: Note that $\mathrm{HALT}(x, y)$ is a total function and $\mathrm{HALT}(x, y) \leqslant 1$ for all x, y. Since HALT is not computable, if follows that the statement must be disproved.

Problem 4: Let $f : \mathbb{N} \longrightarrow \mathbb{N}$ be a bijection. Prove that if f is computable, then so is f^{-1}.

Solution for Problem 4: Let $\mathcal{P}$ be the program

$$\begin{array}{ll} [A] & Z_2 \leftarrow f(Z_1) \\ & Z_1 \leftarrow Z_1 + 1 \\ & \text{IF } Z_2 \neq X \text{ GOTO } A \\ & Y \leftarrow Z_1 - 1 \end{array}$$

Since f is a bijection, there exists exactly one x such that $f^{-1}(y) = x$ or $f(x) = y$. Therefore, the program $\mathcal{P}$ halts on any input and computes f^{-1}.

Problem 5: Let $\text{HALT}^1(z)$ be defined as

$$\text{HALT}^1(z) = HALT(\ell(z), r(z))$$

for $x \in \mathbb{N}$. Show that HALT^1 is not computable.

Solution for Problem 5: Let x, y be two arbitrary numbers in $\mathbb{N}$ and let $z = \langle x, y \rangle$. We have $\ell(z) = x$ and $r(z) = y$, hence $\text{HALT}(x, y) = \text{HALT}^1(z)$. Since $\text{HALT}(x, y)$ is not computable, if follows that HALT^1 is not computable.

Problem 6: The state of a program $\mathcal{P}$ was defined as a set of list of equations of the form $V = m$. Recall that the standard list of variables is

$$Y, X_1, Z_1, X_2, Z_2, \ldots$$

Note that the input variables occupy even numbered positions on this list (X_1 is on the 2^{nd} place, X_2 is on the 4^{th} place, etc.). The state of a program $\mathcal{P}$ is encoded by the number

$$S = [a_1, a_2, \ldots, a_n],$$

where a_i is the value assumed by variable V_i in the list above. Prove that:

1. The initial state of $\mathcal{P}$ is encoded by the number $\prod_{i=1}^{n} (p_{2i})^{x_i}$.
2. For a prime number p_i we have $p_i | S$ if and only if the state of the program contains the equation $V_i = a_i$, where $a_i \neq 0$.

Solution for Problem 6:

The initial state of the program is defined by

$$Y = 0, X_1 = x_1, Z_1 = 0, X_2 = x_2, Z_2 = 0, \ldots$$

Therefore, this state is encoded by

$$\begin{aligned} &2^0 \cdot 3^{x_1} \cdot 5^0 \cdot 7^{x_2} \cdot 11^0 \cdot \ldots \\ &= p_2^{x_1} p_4^{x_2} \cdots = \prod_{i=1}^{n} (p_{2i})^{x_i}. \end{aligned}$$

Solution for Problem 6 cont'd

If the state of the program $\mathcal{P}$ is encoded by

$$S = [a_1, a_2, \ldots, a_n],$$

we have $S = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}$. Thus, a prime p is a divisor of S if and only if $p = p_i$ for some i, $1 \leqslant i \leqslant n$, and $a_i > 0$. This shows that the state contains the equation $V_i = a_i$, where $a_i \neq 0$.