HOMEWORK N. 3, MATHEMATICAL LOGIC FOR COMPUTER SCIENCE 2020/2021

ASSIGNMENT: DO 2 EXERCISES, ONE FROM EACH GROUP.
DEADLINE: APRIL 15, 2021.

1. Computable, c.e. and not computable Problems

N.B. In the following we use $(\varphi_i)_{i\in\mathbf{N}}$ to refer to a fixed programming system (model of computation). You can reason in terms of the one you prefer, be it Turing Machines $(M_i)_{i\in\mathbf{N}}$, the class $\mathcal{C}$, Σ_1-definable functions, or in terms of pseudo-code.

Exercise 1.1. Write down the sentence F_M associated as in the proof of Trakhtenbrot and Church's Theorems to the Turing Machine described below, where the alphabet is $\{0,1\}$, $\#$ is the blank symbol, q_0 the intiial state, q_r the reject state. The machine M has the following transition function:

$$(q_0, 0) \mapsto (q_r, \#, +1)$$

$$(q_0, 1) \mapsto (q_0, \#, +1)$$

$$(q_0, \#) \mapsto (q_0, \#, +1)$$

Describe the canonical model of the sentence F_M.

Exercise 1.2. Show by informal arguments the following points.

(1) If $L_1 \cap L_2$ is not computable and L_2 is computable then L_1 is not computable.
(2) If $L_1 \cup L_2$ is not computable and L_2 is computable then L_1 is not computable.
(3) If $L_1 \setminus L_2$ is not computable and L_2 is computable then L_1 is not computable.
(4) If $L_2 \setminus L_1$ is not computable and L_2 is computable then L_1 is not computable.
(5) If L_1 and L_2 are computably enumerable then $L_1 \cup L_2$ and $L_1 \cap L_2$ are computably enumerable.

Exercise 1.3. Assume that the set $\{i : \varphi_i \text{ is a total function }\}$ is not computable. Show (by an informal argument) that the set $\{(i,j) : \forall n(\varphi_i(n)\uparrow \text{ iff } \varphi_j(n)\uparrow\}$ is not computable. (The notation $\uparrow$ indicates that the function is not defined on the given input).

Exercise 1.4. Let C be a set of c.e. sets. If C is not closed under supersets, then $\{i : dom(\varphi_i) \in C\}$ is not c.e. (Hint: you can assume that the set $\{i : \varphi_i(i)\uparrow\}$ is not c.e.).

Exercise 1.5. Argue informally whether the following sets are computable, c.e., or not computable.

(1) $\{(i,j) :$ given j as input, M_i never moves left$\}$.
(2) $\{(i,j) :$ during the computation on j, M_i moves left three times in a row$\}$.

(Hint: you can assume that the Halting Set $\{(i,j) : M_i(j) \text{ accepts}\}$ is not computable).

Exercise 1.6. Show that if we define the minimalization operator by dropping the requirement that the function is defined on all values smaller than the first value on which the output is 0 then we can define some non-computable function.

More precisely, consider the class $\mathcal{C}^*$ defined as the smallest calss of functions containing the initial functions and closed under composition and under the following operator of minimalization: if $\psi(\vec{x}, y)$ is in $\mathcal{C}^*$ then the function $\varphi(\vec{x})$ = the minimum y such that $\psi(\vec{x}, y) = 0$, if any; is also in $\mathcal{C}^*$. Show that the class $\mathcal{C}^*$ contains a non-computable function.

2. NP PROBLEMS

Exercise 2.1. Show, by writing a formula, that the following properties are definable in Existential Second-Order Logic over the class of finite graphs:

- Even: the graph has an even number of elements.
- Hamiltonian: the graph contains a cycle which visits each vertex exactly once.
- Bipartite: the graph is bipartite.
- Perfect Matching: the graph has a perfect matching.

Exercise 2.2. Express the transitive closure query in Existential Second-Order Logic: write a formula $T(x, y)$ that holds of two elements of a finite structure if and only if they are in the transitive closure of a given binary relation R.

Express the Unreachability query in Existential Second-Order Logic: the Unreachability query, relative to a relation symbol R singles out the pairs (a, b) such that there is not R-path from a to b.

Does the negation of your sentence express Reachability?

Exercise 2.3. Let's call a formula a formula in Universal Second Order Logic if it has the form $\forall R_1 \dots \forall R_n F$, where F is a first-order formula. Show that the property of being a connected graph is expressible by a Universal Second Order formula using only quantification on relations $R_1, \dots, R_n$ of arity 1.

(Hint: start by defining the negation of connectivity).

Exercise 2.4. Let S be a binary relation symbol. Assume that S is interpreted as the standard linear order on the set $\{0, 1, \dots, n-1\}$. Define two formulas $F(x)$ and $L(x)$ expressing, respectively, that x is the first and that x is the last element of the ordering. Consider the set of pairs $(i, j) \in \{0, 1 \dots, n-1\} \times \{0, 1, \dots, n-1\}$. To use these pairs to count up to n^2 we identify each pair with its position in the lexicographic ordering induced by S on the set of pairs. Define a formula $S^2(x_1, x_2, y_1, y_2)$ that expresses that the number represented by (x_1, x_2) is the immediate predecessor of the number coded by (y_1, y_2).

Discuss how to generalize your answer to the lexicographic ordering on k tuples, for $k \geq 2$.

Exercise 2.5. Show that there is no formula $F(x, y, z)$ in first-order logic in the language of orders such that for all n, if $a, b, c < n$ then $a + b = c$ if and only if $(\{0, 1, \dots, n-1\}, <) \models F[a, b, c]$.

(Hint: Use some inexpressibility result we proved).