Homework 4

Problem 1. Give a model for the sentence

$$\begin{aligned}\phi_{\mathrm{lt}} = {} & \forall x\,[R_1(x,x)] \\ & \wedge \forall x,y\,[R_1(x,y) \leftrightarrow R_1(y,x)] \\ & \wedge \forall x,y,z\,[(R_1(x,y) \wedge R_1(y,z)) \rightarrow R_1(x,z)] \\ & \wedge \forall x,y\,[R_1(x,y) \rightarrow \neg R_2(x,y)] \\ & \wedge \forall x,y\,[\neg R_1(x,y) \rightarrow (R_2(x,y) \oplus R_2(y,x))] \\ & \wedge \forall x,y,z\,[(R_2(x,y) \wedge R_2(y,z)) \rightarrow R_2(x,z)] \\ & \wedge \forall x \exists y [R_2(x,y)].\end{aligned}$$

Solution. Taking the universe as $\mathbb{Z}$. Relation $R_1(x,y)$ as $x = y$, relation $R_2(x,y)$ as $x < y$.

Problem 2. Prove that the Halting problem with empty input

$$\mathrm{HALT}_\varepsilon = \{\langle M \rangle \mid M \text{ halts on empty input.}\}$$

is undecidable.

Solution. Assume there is a Turing machine H that decides $\mathrm{HALT}_\varepsilon$, we construct a Turing machine D as follows:
On input w,

1. Obtain its own code $\langle D \rangle$ by recursion theorem.
2. Run $H(\langle D \rangle)$. If H accepts, it enters a loop forever state; if H rejects, it halts.

It can be seen that if D halts on ε, it will loop forever, and if it loops on ε, it will halt, a contradiction.

The statement can also be proved by reducing the problem to HALT.

Problem 3. Show that any infinite subset of $\mathrm{MIN}_{\mathrm{TM}}$ is not Turing-recognizable where $\mathrm{MIN}_{\mathrm{TM}}$ is a language defined in the class.

Solution. The solution is similar to the proof in class, just note that for any infinite subset we can always find some D such that $|\langle D\rangle| > |\langle C\rangle|$.

Problem 4.

(a) Prove a special case of the S-m-n theorem, the Currying technique for Turing machines. That is, show that there is a computable function $S_1^1 : \Sigma^* \times \Sigma^* \to \Sigma^*$ mapping the description of Turing machine T and input x to the description of a Turing machine S such that (1) S on input y computes the same output as T on input $\langle x, y\rangle$ if T halts; and (2) S loops on input y if T loops on input $\langle x, y\rangle$.

(b) Prove Kleene's recursion theorem by item (a) and Roger's fixed-point theorem.

Solution. (a) First construct the described turing machine S: it first print x on the tape, then simulate T on input $\langle x, y\rangle$. We can construct a turing machine R that prints the description of S.

(b) For any computable function $T\colon \Sigma^* \times \Sigma^* \to \Sigma^*$, we define the function $s_T(x) = S_1^1(\langle T\rangle, x)$. It corresponds to a turing machine $S_x(y)$ that works as $T(x, y)$ forall y. We apply the roger fix point theorem to s_T, which says there is a turing machine R such that $\forall y, R(y) = s_T(\langle R\rangle) = T(\langle R\rangle, y)$. This is the Kleene's recursion theorem.