Exercise Sheet n°4

Exercise 1:

In this exercise we want to show that the problem of deciding when a given DFA has a finite or infinite language is Turing decidable.

We only consider – without loss of generality – the DFAs $(Q, \Sigma, \delta, q_0, F)$ where $Q = \{0, 1, \ldots, k-1\} = k$, $\Sigma = \{0, 1, \ldots, n-1\} = n$, and $q_0 = 0$ for some natural numbers $k > 0$ and $n > 0$.

1. Represent any DFA as a finite string on the finite alphabet

$$A = \{\langle, \rangle, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (,), \{, \}, , \}.$$

Formulate the problem of deciding when a given DFA has a finite or infinite language as the membership problem for a language $\mathrm{INF}_{\mathrm{DFA}}$ on A.

2. In a similar fashion, we consider the language $\mathrm{E}_{\mathrm{DFA}}$ consisting of the DFAs whose language is empty. Show that the language $\mathrm{E}_{\mathrm{DFA}}$ is Turing decidable.

3. Show that $\mathrm{INF}_{\mathrm{DFA}}$ is Turing decidable.

   Hint: The pumping lemma for languages recognised by DFA *and the previous point can be useful.*

Exercise 2:

The symbol ω denotes the set of all natural numbers. For a non empty set A, let $x \in A^{<\omega} \cup A^{\omega}$ be a finite or infinite sequence of elements of A. The domain of x is given by $\mathrm{Dom}(x) = n$ if $x \in A^n$ and $\mathrm{Dom}(x) = \omega$ if $x \in A^{\omega}$. For $k \in \mathrm{Dom}(x)$, $x_{\restriction k}$ is the restriction of x on k.

A *tree* on a set A is subset $T \subseteq A^{<\omega}$ closed by prefixes, i.e. such that for all $t \in T$ and all $k < \mathrm{Dom}(t)$, $t_{\restriction k} \in T$. An *infinite branch* of a tree T on a set A is a sequence $x \in A^{\omega}$ such that for all $n \in \omega$, $x_{\restriction n} \in T$.

1. Let T be a tree on a finite set. Show that T has no infinite branch if and only if T is finite.

Consider the following

Definition. *A language L is called* nondeterministic Turing decidable *if there exists a nondeterministic Turing machine N such that*

(i) for all input w, no computation of N on w loops for ever;

(ii) for all input w, $w \in L$ if and only if there is at least a computation of N on w which is accepting.

2. Show that a language is nondeterministic Turing decidable if and only if it is Turing decidable.

   Hint: Modify the proof of the equivalence between deterministic and nondeterministic Turing machine. You can use the following definition:

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \Delta, q_0, q_{acc}, q_{rej})$ be a non deterministic Turing machine and view Δ as a relation, in symbols $\Delta \subseteq (Q \times \Gamma) \times (Q \times \Gamma \times \{L, R\})$. For any input word $w \in \Sigma^*$ we define the *computation tree* of $\mathcal{M}$ on w as the tree $T_{\mathcal{M}}(w)$ on Δ given by

$$(s_0, \ldots, s_{n-1}) \in T_{\mathcal{M}}(w) \Leftrightarrow \begin{cases} s_i \in \Delta \text{ for } 0 \leq i \leq n-1 \text{ and} \\ \text{there exists a sequence of configurations} \\ (c_0, \ldots, c_n) \text{ such that} \\ \quad - \ c_0 = q_0 w, \text{ and} \\ \quad - \ c_i \text{ yields } c_{i+1} \text{ through } s_i \text{ for all } i < n. \end{cases}$$

Exercise 3:

1. Let Σ be a finite alphabet not containing $\sqcup$. Give a recursive coding $c : \Sigma^{<\omega} \to \{0,1\}^{<\omega}$.

2. Let $\Gamma = \Sigma \cup \{\sqcup\}$ and consider the recursive coding c you gave in 1. Show that for any Turing machine M with tape alphabet Γ, and input alphabet Σ, there exists a Turing machine M_c with input alphabet $\{0,1\}$ and tape alphabet $\{0,1,\sqcup\}$ such that for all $w \in \Sigma^{<\omega}$

$$M \text{ accepts } w \quad \text{iff} \quad M_c \text{ accepts } c(w).$$