CSL705: Theory of Computation

Tutorial: Turing Machines and Turing Computability

1. Prove that for every NPDA, there exists a Turing machine which accepts the same language.
2. Design a deterministic turing machine to multiply two non-negative integers.
3. Any relation R between various sets is said to be Turing-computable if there exists a Turing machine which implements the *characteristic function* of R. If $R \subseteq A_1 \times A_2 \times \cdots \times A_n$ then the *characteristic function* of R is the function χ_R such that for any $\vec{a} \in A_1 \times A_2 \times \cdots \times A_n$, $\chi_R(\vec{a}) = 1$ whenever $\vec{a} \in R$ and $\chi_R(\vec{a}) = 0$ otherwise.
4. Design a deterministic turing machine which implements the "greater than" relation on the naturals.
5. Prove that the addition and comparison of rational numbers are Turing computable
6. Let T be a (deterministic) Turing machine which accepts a language $L \subseteq \Sigma^*$. Design a Turing machine T' which decides membership in L.
7. Let T_f be a Turing machine which implements a partial function $f : D \rightharpoonup E$ and let T_g be a Turing machine which implements another function $g : C \rightharpoonup D$. Prove that there exists a Turing machine $T_{f \circ g}$ which implements $f \circ g$.
8. Prove that every Turing-computable function may be transformed into a language. Further show that there exists a Turing machine that accepts the language.