Exercises about unrestricted grammars and universal Turing machines

1. Give an unrestricted (or context-sensitive) grammar, which produces language $L = \{a^i b^{2i} a^i | i > 0\}$. Give the derivation for string $aabbbbaa$ in the grammar!

2. Chomsky thought that natural language can be described by context-sensitive grammars, in which all productions can be represented in form:

   $S \to \epsilon$ or $\alpha A \beta \to \alpha \omega \beta$,

   in which A is non-terminal symbol and $\omega \neq \epsilon$. Give some example in in natural language, which needs such "context" α_β! (You can consider examples in your own native language, but explain the feature also in English.) Do you believe that all structures of natural language can be described with context-sensitive grammars?

3. The Universal Turing machine has invited all Turing machines of the universum into Universal Truring Symposium, which is hold in Hilbert's Hotel. The room reservations have been done with the codes c_M of the Turing machines. Is there enough rooms for all Turing machines, if there are no other guests in the hotel? How would you alocate the rooms?

4. What are the codes of the following Turing machines?

   (a) $\delta(q_0, 0) = (q_0, 0, R)$
   $\delta(q_0, 1) = (q_0, 1, R)$
   $\delta(q_0, <) = (q_{yes}, <, L)$

   (b) $\delta(q_0, 0) = (q_1, 0, R)$
   $\delta(q_0, 1) = (q_{no}, 1, R)$
   $\delta(q_0, <) = (q_{no}, <, L)$
   $\delta(q_1, 0) = (q_{yes}, 0, R)$
   $\delta(q_1, 1) = (q_0, 1, R)$
   $\delta(q_1, <) = (q_{yes}, <, L)$

5. Simulate the behaviour of the universal Turing machine, when it is given as its input $c_M w$ the code of the machine in b-part of the previous task and a string 0101!

6. Construct a finite automaton, which recognizes the legal codes of Turing machines. What is the corresponding regular expression?

7. Construct a standard Turing machine, which recognizes the legal codes of Turing machines.

8. Let the languages A and B be recursive enumerable. (i.e. there exists Turing machines M_A and M_B, which recognize the given languages and halt in "Yes"-case, but not necessarily in "No"-case.) Prove that languages $A \cup B$ and $A \cap B$ are also recursive enumerable i.e. you can construct for them Turing machines, which halt at least in "Yes"-case for all input strings!

9. Let A be a recursive enumerable language. Can you construct a Turing machine for its complement language $\overline{A}$ from the machine M_A by changing the accepting and rejecting final states?

10. Prove that language $\{c_M | M \text{ halts on input } \epsilon\}$ is recursive enumerable but not recursive! I.e. invent for it a Turing machine, which halts in "Yes"-case, but not necessarily in "No"-case!