List of the potential questions for "Calculability and Complexity"

JF Raskin

December 21, 2010

In addition to all the questions that have been formulated on slides marked "Exercises", here is the list of potential questions for the exam of "Calculability and complexity".

1 Chapter 3: The Church-Turing Thesis

1. Explain how decision problems can be formalized by the notion of language of finite words. Develop an example of a language that formalizes a problem.
2. Define the notion of configuration of a Turing machine. Define the notion of computation of a deterministic Turing machine M on a word w.
3. Are nondeterministic Turing machines more powerful than deterministic Turing machines as deciders ?Justify your answer.
4. Given a Turing machine M and a word w, give the computation of M on w. Does M accepts w ?
5. Define when a Turing machine recognizes a language L and when it decides a language L. What is the fundamental difference between those two notions ? Why do we use deciders to formalize the intuitive notion of algorithm ?
6. What is the Church-Turing thesis ? Give a list of arguments in favor of this thesis ? Why is it not a theorem ?
7. Why don't we use the notions of finite automata or context-free grammars to formalize algorithms ?

2 Chapter 4: Decidability

1. Why is the set of Turing machines countable ?

2. Prove that the set of subsets of an infinite countable set is not countable.

3. Prove that the language $L_0 = \{w_i \mid w_i \notin L(M_i)\}$ is not Turing recognizable.

4. Prove that the language $A_{\mathsf{TM}} = \{(M, w) \mid M \text{ is a TM and } M \text{ accepts } w\}$ is undecidable.

5. Explain why proving A_{TM} undecidable establishes that $\mathsf{R} \neq \mathsf{RE}$.

6. Prove that a language $A \in \mathsf{R}$ iff A is Turing-recognizable and co-Turing-recognizable.

3 Chapter 5: Reducibility

1. Explain the technique of "the reduction" with the proof that $\mathsf{HALT}_{\mathsf{TM}} = \{(M, w) \mid M \text{ is a TM and } M \text{ halts on } w\}$ is undecidable.

2. Explain the technique of "the reduction" with the proof that $\mathsf{REGULAR}_{\mathsf{TM}} = \{(M) \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$ is undecidable.

3. Let $\mathsf{A}_{\mathsf{LBA}} = \{(B, w) \mid B \text{ is a linear bounded automata and } B \text{ accepts } w\}$, explain why this language is decidable.

4. Let $\mathsf{ALL}_{\mathsf{CFG}} = \{(G) \mid G \text{ is a CFG and } L(G) = \Sigma^* \}$ and assume that it is undecidable, prove that $\mathsf{EQ}_{\mathsf{CFG}} = \{(G_1, G_2) \mid G_1 \text{ and } G_2 \text{ are CFGs and } L(G_1) = L(G_2)\}$ is also undecidable.

5. With the help of the slides, explain the proof that the problem POST is undecidable.

6. Define the notion of "reduction function" $f : \Sigma^* \to \Sigma^*$ from a problem A to a problem B and prove the following theorem: "If $A \leq_m B$ and B is decidable then A is decidable" where $A \leq_m B$ reads "A is reducible to B".

4 Chapter 7: Time Complexity

1. Define the notion of time complexity $\mathsf{TIME}(f(n))$. Explain why we can say that this is a worst-case measure of complexity ? Why do we use big **O** notation when we reason on the time complexity of a Turing machine ?

2. Define the big **O** notation and the small **o** notation. Give examples and explain the relation that exists between those two notions.

3. Why are deterministic polynomial time Turing machines suitable to study the class P ?

4. Define the running time of a Turing machine that is a decider. If M is a nondeterministic machine that decides the language L in $\mathbf{O}(t(n))$, how can we bound the running time of a deterministic Turing machine that decides L ?

5. Define the class P and explain why it is an important complexity class.

6. Given two examples of problems and their natural encodings. Suggest other encodings that are not reasonable and explain why.

7. We have given two definitions of the class NP. One uses "certificates" and one uses nondeterministic Turing machines. Recall the two definitions and prove that they are equivalent.

8. Give an example of a problem which is in NP and prove the membership using the notion of certificate.

9. Explain why $\mathsf{NP} \subseteq \mathsf{ExpTime}$.

10. Define the notion of NP complete problem and explain why this notion is important.

11. Define the notion of "polynomial time mapping reducible". Show that if A is polynomial time reducible to B and $B \in \mathsf{P}$ then $A \in \mathsf{P}$.

12. Illustrate on an example that $\mathsf{3SAT}$ is polynomial time reducible to CLIQUE.

13. Prove the following two statements: (i) if $B \in \mathsf{NP} - \mathsf{Complete}$ and $B \in \mathsf{P}$, then $\mathsf{P} = \mathsf{NP}$, (ii) if $B \in \mathsf{NP} - \mathsf{Complete}$ and $B \leq_P C$, and $C \in \mathsf{NP}$ then C is $\mathsf{NP} - \mathsf{Complete}$.

14. With the help of the slides, summarize in one page the main arguments that show that SAT is $\mathsf{NP} - \mathsf{Complete}$.

15. Illustrate with an example that $\mathsf{3SAT} \leq_P \mathsf{VERTEX} - \mathsf{COVER}$, and explain the main ideas of the reduction.

5 Chapter 8: Space Complexity

1. Define the notion of space complexity. Illustrate the difference between time complexity and space complexity by showing an example of problem that can be solved in $\mathbf{O}(t(n))$ space but we believe cannot be solved in $\mathbf{O}(t(n))$ time.

2. If a language $L \in \mathsf{SPACE}(\mathbf{O}(t(n))$ with $t(n) \geq n$, what can we say about the time complexity of L ?

3. With the help of the slides, summarize in one page the main arguments underlying Savitch's theorem.

4. Define the classes PSPACE, $\mathsf{NPSPACE}$, $\mathsf{coPSPACE}$ and $\mathsf{coNPSPACE}$. Explain why all those classes are equal.

5. Define the notion of $\mathsf{PSPACE} - \mathsf{Completness}$ and give two examples of problems that are $\mathsf{PSPACE} - \mathsf{Complete}$.

6. Explain why TQBF can be solved in polynomial space.

7. Explain why the proof of Cook-Levin theorem cannot be applied directly to show that TQBF is $\mathsf{PSPACE} - \mathsf{Complete}$.

8. With the help of the slides, summarize in one page the proof that Generalized Geography is $\mathsf{PSPACE} - \mathsf{Complete}$.

9. Define the class L and NL. In those definitions we use Turing machines with a read only input tape and read/write working tape, explain why.

10. Explain why PATH is in NL, and why we believe that it is not in L.

11. Explain how we can bound the time complexity of a Turing machine which decides a language in L.

12. Define the notion of log-space transducer and explain why it is required when proving the theorem: "if $A \leq_L B$ and $B \in \mathsf{L}$ then $A \in \mathsf{L}$".

6 Chapter 9: Intractability

1. Define the notion of space constructible function $f : N \to N$. Give an example of such a function and explain why it is so.

2. Knowing that the equivalence between extended regular expressions is $\mathsf{ExpSpace} - \mathsf{Hard}$, show by applying the space hierarchy theorem that this problem can not be solved in deterministic polynomial time, nor in non-deterministic polynomial time.

3. Define the notion of Turing machine with oracle. Define the set of languages that can be solved in $\mathsf{P}^{\mathsf{SAT}}$, define a language that belongs to this set and which is not believed to be in NP (explain why).