List of potential questions and answers for INFOF408 Calculability and Complexity (2019/2020)

December 19, 2019

Contents

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 by proving that every nondeterministic Turing machine has an equivalent deterministic Turing machine

4. Given a Turing machine M and a word w, give the computation of M on w. Does M accept 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. Define the notion of enumerator for a language. Prove that a language is Turing-recognizable if and only if it has an enumerator

7. Define the notion of multitape Turing machine. Prove that every multitape Turing machine has an equivalent single tape Turing machine

8. What is the Church-Turing thesis ? Give a list of arguments in favor of this thesis ? Why is it not a theorem

9. 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 $L_0 = \{w_i | w_i \notin L(M_i)$ is not Turing recognizable

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

5. Explain why proving $A_T M$ undecidable establishes that R $\neq$ RE

6. Prove that a language A $\in$ 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 $\text{HALT}_{\text{TM}} = \{(M, w) | M$ is a TM and M halts on $w\}$ is undecidable

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

3. Let $A_{\text{LBA}} = \{(B, w) | B$ is a linear bounded automate and B accepts $w\}$, explain why this language is decidable

4. State Rice's Theorem and prove it. Give two problems that can be proved undecidable by applying Rice's Theorem

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

4 Chapter 7: Time Complexity

1. Define the notion of time complexity 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 $O(t(n))$, 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. Give an example of a problem which is in NP and prove its membership NP using the notion of certificate

8. 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

9. Explain why NP $\subseteq$ 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 $B \in P$ then $A \in P$

12. Prove that 3SAT is polynomial time reducible to CLIQUE

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

14. Give the main constructions and arguments underlying the proof of the Cook-Levin Theorem that establishes that SAT is NP-Complete

15. Prove that 3SAT $\leq_P$ VERTEX-COVER

16. Define the HAMILTONIAN-PATH problem and prove it is NP-complete

17. Give a dynamic programming algorithm for the PARTITION problem and explain why it does not imply P = NP

18. Give a dynamic programming algorithm for the SUBSET-SUM problem and explain why it does not imply P = NP

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 $O(t(n))$ space but we believe cannot be solved in $O(t(n))$ time

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

3. State and prove Savitch's theorem

4. Define the classes PSPACE, NPSPACE, coPSPACE and coNPSPACE. Explain why all those classes are equal

5. Define the notion of PSPACE-Completness and give two examples of problems that are PSPACE-Complete

6. Explain why TQBF can be solved in polynomial space

7. Give the main arguments and constructions underlying the proof that TQBF is PSPACE-Complete. Explain why the proof of Cook-Levin theorem cannot be applied directly to show that TQBF is PSPACE-Complete

8. Prove that Generalized Geography is PSPACE-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

6 Chapter 9: Intractability

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

2. Knowing that the equivalence between extended regular expressions is ExpSpace-Complete, show by applying the space hierarchy theorem that this problem can not be solved in deterministic polynomial time, nor in nondeterministic polynomial time

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

4. What do we know about the relations between the complexity classes P, NP, PSPACE, EXPTIME, and EXPSPACE? Explain