Institut for Matematik og Datalogi
Syddansk Universitet

Exam Assignment 2
Complexity and Computability — 2018

This is the second of three sets of problems (assignments) which together with the oral exam in June (June 27, 28, or 29) constitute the exam in DM553. This second set of problems must be solved individually, not in groups.

The assignment is due at 10:15 on Tuesday, May 1. You may write this either in Danish or English. Write your full name clearly on the first page of your assignment (on the top, if it's not a cover page). Turn it in as a PDF file via Blackboard through your DM553 course. The assignment hand-in is in the menu for the course and is called "SDU Assignment". Keep the receipt it gives you proving that you turned your assignment in on time. Blackboard will not allow you to turn in an assignment late.

Cheating on this assignment is viewed as cheating on an exam. Do not talk with anyone other than David Hammer or Joan Boyar about the assignment, and do not show your solutions to anyone. If you have questions about the assignment, come to Joan Boyar or David Hammer.

Assignment 2

Do the following problems. Write clear, complete answers, but not longer than necessary.

Note that some of the parts from the later problems (particularly problem 7) will be easier to do after the lecture on April 10 or 12 and discussion section on April 18 or 19, but most can be started on now.

1. (a) For proving the undecidability of the Post Correspondence Problem, the notation $*u*$ is defined on page 233 of Sipser's textbook. Consider the alphabet $\{a, b\}$. What is $*abbabaa*$ over the alphabet $\{a, b, *\}$?

(b) Design a (possibly multitape) deterministic Turing machine which decides
$$\{wcw^Rc *w* \mid w \in \{a,b\}^+\}$$
over the alphabet $\{a, b, c, *\}$. Recall that w^R is w reversed and $*w*$ is as defined on page 233 of Sipser's book. Give a formal description, either with the complete transition function or with a state diagram. **Also explain in words how your Turing machine works.**

2. Show that A is decidable if and only if $A <_m \{a, b, ab, ba\}$.

3. Prove that if the languages L_1 and L_2 are decidable, then $L_1 \cup \overline{L_2}$ is also decidable.

4. Consider $L = \{\langle P, R\rangle \mid P$ is a pushdown automaton, R is a regular expression, and $L(P) \cup \overline{L(R)} = \Sigma^*\}$. Prove that L is undecidable.

5. Consider $L = \{\langle M\rangle \mid M$ is a TM, and $\exists w \in \Sigma^*$, such that M does not halt on input $w\}$. Prove that L is undecidable.

6. Use Rice's Theorem to show that
$$\{\langle M\rangle \mid M \text{ is a TM and } |L(M)| \text{ is divisible by } 3\}$$
is undecidable.

7. Suppose G is a graph representing an adventure game, "Monster Control". Each vertex in G represents a "room", and each edge represents a way of going from one room to another (i.e., a tunnel). You have t monster friends, who all like you, but hate each other. In fact they will kill each other if they can look each other in the eye, which they can do if they are both are in the same room or in adjacent rooms connected by a tunnel (unless at least one is blindfolded). Originally, all of your monsters are placed in separate cages in the "Entrance Hall" (one of the rooms). These are the only cages that are strong enough to keep the monster in it inside and the monsters not in it outside. You can blindfold a monster at any time, if you first burn a flare in front of it to temporarily blind it, but it in exactly 5 minutes it will tear the blindfold off. You can make it through one tunnel and one room during that time frame of 5 minutes. If you burn another flame 4 minutes and 55

seconds after the previous, the monster will see that flame long enough to become blind again, so you can blindfold it again immediately and walk through another tunnel and room. You are given a total of s flares and s blindfolds you can use to lead monsters past other monsters or rooms near other monsters. A monster will stay in a room if you put it there, unless it can look another monster in the eye. The goal is to place all of the t monsters in different rooms so that none of them will eat each other. You have a map of G in advance. The question is, "Is it possible to win this game by placing the t monsters in t different rooms in such a way that none of them eat each other, using no more than s flares and s blindfolds?"

(a) Show that it is NP-Complete to determine if an instance of Monster Control is winnable. (An instance consists of G, s and t.) This is the recognition version of the problem (a decision problem).

(b) In this problem, consider a simplified version of the game, call Magical Monster Control. Now, you can magically place the monsters in the different rooms without taking them there. Thus, no flares or blindfolds are necessary, so you can assume that $s = 0$. (Note that you still cannot put monsters into two rooms connected by a tunnel.) Suppose that you have an efficient (polynomial time) algorithm A for determining if an instance of Magical Monster Control is winnable (a black-box polynomial time algorithm for the recognition version). Give a polynomial time algorithm (using A) to determine a strategy for winning the instance when it is winnable. (Thus, you should solve the optimization version, using a black-box, polynomial time algorithm for the recognition version of the problem.)

(c) Suppose, in Magical Monster Control, each room has at most two magic tunnels going off from it. Is the recognition version of the problem still NP-Complete? Prove your answer. (You may assume that $P \neq NP$.)