

CSE105 (spring 2008): Homework 4 (updated)

Instructor: Daniele Micciancio

Due on Tuesday, June 3, 2008, noon.

Solutions to the homework should be typed up, printed out, and turned in at the beginning of class. A homework submission page for the JFLAP files will appear on the course webpage shortly. The files must be named *exactly* as shown in the problem description.

1 Turing Machines

Consider the language $L = \{wvz\#v : u, v, z \in \{0, 1\}^*\}$. In other words, L is the set of all pairs $w\#v$ of binary words separated by special symbol $\#$ such that w contains v as a substring. Show that the language L is decidable by giving a TM that terminates on all inputs and accepts precisely the strings in L .

You should design your TM using JFLAP, and submit your solution as a JFLAP file `hw4-1.jff` on the course webpage. (Just as a hint about the size of a reasonable solution to this problem, I could easily design a TM with 15 states, and a smaller number of states is probably sufficient.)

2 Undecidability (updated)

The second problem given in the first version of this homework assignment (now listed as problem 4) turned out to be much harder than intended. You should skip that problem, and solve problem 5.13 from the textbook (Sipser, 2nd edition) instead. For your convenience, the text of the problem is reported below. This problem is in the same spirit as the old problem 2, but it is much easier to solve.

Problem (Sipser 5.13): A **useless state** in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a Turing machine has any useless states. Formulate this problem as a language and show that it is undecidable.

3 Map reductions

Consider the languages

$$EQ_{TM} = \{\langle M_1, M_2 \rangle : L(M_1) = L(M_2)\} \quad SUBSET_{TM} = \{\langle M_1, M_2 \rangle : L(M_1) \subseteq L(M_2)\}.$$

Give a map reduction from EQ_{TM} to $SUBSET_{TM}$, and prove your reduction correct. (Map reductions have not been covered yet. They will be covered in class on May 27.)

4 Undecidability (Optional)

This is the old version of problem 2, which turned out to be much harder than intended. You may still attempt to solve the original problem (reported in the next page) if you like, and if you do, you can submit your solution with the rest of the homework. No formal/partial credit will be assigned to problem 4, so submit your solution only if you think you solved it correctly.

Original problem. Consider the following problem: given a TM M , determine if the number of states in M is minimal (i.e., M uses the smallest possible number of states), or there exists an equivalent Turing machine M' with fewer states than M . (Two Turing machines are equivalent if they accept the same language, i.e., $L(M) = L(M')$.)¹ Formally, you are asked to prove that the language

$$L_2 = \{\langle M \rangle : M \text{ is a TM and for any } M' \text{ with strictly fewer states than } M, L(M) \neq L(M')\}.$$

Remark (this is not a hint to help you solve the problem, just a side remark): the above problem is related to typical program optimization problems. Say you wrote a program M , and you are trying to optimize its size (e.g., to fit into a small embedded computing device with very little memory). You want to find the smallest possible program M' that is equivalent in functionality to M . The above problem shows that even figuring out that you have found the best solution (let alone actually finding it) is a very hard problem. In practice, compilers can only make a best effort attempt at producing small code, with no guarantees that the result produced by the compiler is indeed optimal.

¹Other definitions of equivalent TMs are possible. E.g., one may want that not only $L(M) = L(M')$, but also that M and M' terminate on exactly the same set of strings. For simplicity, in this problem, we disregard termination issues, and only consider the condition $L(M) = L(M')$ in the definition of equivalence.