Institut for Matematik og Datalogi
Syddansk Universitet

March 5, 2020
JFB

DM553/MM850 – Complexity and Computability 2020 – Lecture 9

Lecture, March 4

We continued on chapter 5, starting with section 5.3 (we skipped the last part of section 5.1, having to do with reductions via computation histories). In section 5.2, we covered the reduction from A_{TM} to MDCP, but still need to do the reduction from MPCP to PCP.

Lecture, March 17

We will finish chapter 5 (the reduction from MPCP to PCP). Then, we will begin on NP-Completeness. starting with Definitions 7.7, 7.9, 7.12 and section 7.3 in Sipser's book, introducing definitions and showing that 3-SAT and CLIQUE are in NP. Note that some of this is also in chapter 34 in the CLRS book.

Lecture, March 20

We will first show that 3-SAT is NP-Complete. To do this, we assume that CNF-SAT is NP-Complete. The proof that 3-SAT is NP-Complete combines the proofs in both the Sipser and the CLRS books. Then, we will cover the proof that SATISFIABILITY (actually CNF-SAT) is NP-Complete, from section 7.4 in Sipser's textbook.

Problems to be discussed on March 24

From CLRS do:

1. The following argument is incorrect. Find the most important error.

   Consider the following algorithm:

   Input: $n \in \mathbb{N}$
   for $i = 2$ to $n-1$ **do**
   check if i divides n
   if it does **then** output i
   endfor
   output -1 if no output yet

Checking if i divides n can be done in time $O(\log^3 n)$ via binary search for an integer k such that $n = i \cdot k$ (the multiplication can clearly be done in time $O(\log^2 n)$).

Thus, the total running time is $O(n \cdot \log^3 n)$ in the worst case. Since $O(n \cdot \log^3 n) \subset O(n^2)$, and n^2 is a polynomial, this algorithm runs in polynomial time. Thus, we have an efficient algorithm for factoring, so we can break RSA, a famous cryptosystem which is believed to be secure.

2. Suppose you have a Boolean formula form, with exactly three literals per clause. Show how to add some constant number of clauses (also with exactly three literals per clause) to F to create a formula F' which is guaranteed to be false.

3. In the CLRS textbook, do the following:

   - 34.2-5, 34.2-10.
   - 34.3-7.
   - 34.4-4, 34.4-5, 34.4-6, 34.4-7. (Probably for next time.)