Computational Complexity

Reading guide and Homework for **Week 05**

1 Reading Guide

You can read about:

- The TQBF problem and Savitch's Theorem. §4.2
- L, NL, reachability is NL-complete. §4.3.
- You should study the proof that NL = coNL, also in §4.3, though we probably didn't have enough time to cover it in class.
- We did not teach this, and probably won't have the time during the course, but there is a remarkable proof that connectivity in undirected graphs is in L. §21.4.
- We sketched the proof that Sokoban is PSPACE-complete. This is nowhere in the book. A reasonable proof would use the gadgets of:

  Joseph C. Culberson. 1997. *Sokoban is PSPACE-complete.*

  combined with the (much simpler) Turing-machine simulation of an earlier paper (which does not prove that Sokoban is PSPACE-hard):

  Dorit Dor and Uri Zwick. 1996. *Sokoban and other motion planning problems.*

  Possible project for the second part of the course: write down a complete proof of the PSPACE-hardness of Sokoban, with the above outline.

There were several details that we covered very quickly, and which **must** be looked at carefully, in order to properly understand what is going on. This kind of familiarity will be necessary to understand what's being said in class.

2 Exercise guide

Before doing the homework, you should train yourself with the following exercises from the book. Solutions will be provided for the exercises marked with an asterisk (see the webpage). You should still try to do them by yourself before looking at the solutions.

- ∗ 4.2.
- ∗ 4.3.
- • 4.7
- • 4.8
- • 4.9
- • How much space is required to execute a recursive procedure, i.e., a procedure which may call itself (possibly more than once)?
- ∗ 4.10

3 Homework

You should turn in solutions for the following exercises before the beginning of the next class. You can turn in solutions electronically (to `bruno.loff+homework@gmail.com`), or in paper, in which case we will scan them ourselves. If you given them in paper, please respect the following rule, which is meant to make scanning easy:

Solutions should be given in separate (not stapled) a4 sheets of paper, and your name and number should appear clearly on every page.

You should feel free to discuss the exercises with your colleagues, but when you write down the answer, you must do it alone, without any help.

Consider the following definition:

1. The problem of *reachability in implicitly given graphs* is as follows. We are given as input a sequence 1^n, a Turing machine M and a sequence 1^t. These three inputs define a graph $G(n, M, t)$ over the vertex set $\{0,1\}^n$, as follows: in $G(n, M, t)$, there is an edge from vertex $x \in \{0,1\}^n$ to vertex $y \in \{0,1\}^n$ if and only if $M(x, y)$ halts and outputs 1 within t steps.

   Our problem is then:

   $$\mathsf{ImplicitReach} = \{\langle 1^n, M, 1^t \rangle \mid \text{there is a path from } 0^n \text{ to } 1^n \text{ in } G(n, M, t)\}.$$

   Show that $\mathsf{ImplicitReach}$ is PSPACE-complete under polytime reductions.

2. A *two-player polynomial game* is a family of games defined by a number n and a polynomial-time Turing machine M, as follows. In the *length-n* version of the game, the game proceeds in $2n$ rounds. Player 1 chooses a string $x_1 \in \{0,1\}^n$, and shows it to Player 2. Then Player 2 chooses a string $x_2 \in \{0,1\}^n$, which can depend on x_1, and shows it to Player 1, *etc*, with Player 1 playing on odd rounds and Player 2 playing on even rounds, with each move allowed to depend on the previous moves. In the end, Player 1 wins if $M(x_1, \ldots, x_n)$ outputs 1, and Player 2 wins otherwise.

   Show that in any such game either Player 1 can always win, or Player 2 can always win.

   Show that the problem of deciding, when given 1^n and M, whether Player 1 or Player 2 can always win, is in PSPACE.

3. (Exercise 4.5 of the book) Show that $\mathsf{2SAT}$ is in NL. For this consider the following hints:

   Hint 1. Suppose your formula has a clause $x_1 \vee x_2$. Note that any satisfying assignment with $x_1 = 0$ must have $x_2 = 1$. The NL-complete problem is directed reachability. Can you create a graph where reachability means something.

   Hint 2. Create a graph with $2n$ nodes, one for each equality $x_i = 0/1$. Add the above implications (e.g. for clause $x_1 \vee x_2$, one would add the two directed edges $x_1 = 0 \to x_2 = 1$ and $x_2 = 0 \to x_1 = 1$). What does this graph tell you?