Presentations — winter semester 2024/25

- Presentations are assigned to students during the semester. Every student will receive an e-mail with the information about the number of the presentation that was assigned to her/him. The web page of the course will contain the date when the presentations were assigned. A student who have not obtained an assignment of the presentation till this date should contact the tutor as soon as possible (for example by e-mail).

- You should bring prepared written form of the presentation and show your understanding of the topic at specified terms at the end of the semester (terms will be published on the web page of the course). The most important is how you will **present** the given topic where you must demostrate your understanding in such a way that you would be able to explain it to your colleagues and to the tutor during **15 minutes**. The tutor will check how you are prepared for that.

- A main source for each presentation is a particular artircle mentioned in the statement of the topic of the presentation. This article should basically contain all relevant information necessary for preparing presentation for the given topic.

  All articles for topics of the presentations are available in electronic form in MS Teams in team `Theoretical Computer Science 2024/25` in channel `General` on panel `Files` in folder

  `Documents / General / Presentations / Sources`.

  Of course, you can also use any information from other publicly available sources.

  You must prepare the given topic yourself, together with the written form of the preparation for the topic. All used sources should be referenced.

- Do not send the written preparation to the tutor. Bring it in a paper form to the presentation. You can use it during the presentation. You will submit it to the tutor after the presentation. It can be prepared on a computer or written by hand, with pictures, etc. It must contain enough information to assess whether it is a reasonable preparation prepared by you (and not for example just copied parts of texts from some other sources).

  This preparation should not have a form of slides but rather a coherent complete text.

  There must also be your name and login stated at the beginning of this preparation.

Topic 1 — *NP-completeness of Minesweeper*

Minesweeper is a well-known game where we have a given area consisting of square cells where each of them can contain a mine. At the beginning, all squares are covered. They are uncovered by clicking on them and uncovered squares display information about the number of mines on neighbouring cells (as a number in the interval from 1 to 8). The goal is to avoid clicking on a mine, to mark the positions containing mines, and to uncover all squares that do not contain mines.

Consider the following problem:

Input: A position in Minesweeper where some squares are uncovered, some of them are covered, and some of them are marked as mines.

Question: Is the given position consistent in the sense that there exists a possibility how mines can be placed in the covered squares in a way correspoding to the displayed information?

Show that this problem is NP-complete.

The main source for this topic is article

R. Kaye: Minesweeper is NP-complete, *The Mathematical Intelligencer*, Vol. 22, No. 2, pp. 9–15, 2000.
DOI: `https://doi.org/10.1007/BF03025367`

Topic 2 — *PSPACE-hardness of game GO*

Game GO is a well-known board game played by two players where one of them plays with black stones and the other with white stones. The players alternately put these stones on a board. The standard size of this board is a grid of size 19×19 but we can consider a more general variant with grid of size $n \times n$ where n is an arbitrary natural number.

Consider the following problem:

Input: Position in game GO in a board of arbitrary size $n \times n$ where some black and white stones are already put on the board, and information, which player has the next turn.

Question: Has the player, which has the next turn, a winning strategy in the given position?

Show that this problem is PSPACE-hard.

The main source for this topic is article

D. Lichtenstein, M. Sipser: GO Is Polynomial-Space Hard, *Journal of the Association for Computing Machinery*, Vol. 27, No. 2, pp. 393–401, 1980.
DOI: `https://doi.org/10.1145/322186.322201`