Prof. Dr. Jasmin Blanchette
Dr. Martin Desharnais-Schäfer
Dr. Michael Kirsten
Elisabeth Lempa

Ludwig-Maximilians-Universität München
Institut für Informatik
Discussion on 19.11.2025
Homework due on 26.11.2025 at 16:00

Possible solution for Exercise Sheet 6 in

Scientific and Technical English for Computer Scientists

The exercise sheets consist of in-class exercises and homework. The in-class exercises take place during the second half of the lecture time slots. The homework, which is optional and ungraded, can be submitted via the "Homework" section in Moodle. The homework is subject to peer review.

Unless indicated otherwise, generative artificial intelligence assistants such as ChatGPT may be used, as long as you acknowledge how you use them as specified by the Institute's policy on plagiarism.[1] However, you may not use such tools to generate peer reviews for you. In addition, we strongly recommend that you do not use them to generate entire solutions, since that would defeat the purpose of the exercises.

Homework 6-3 ***Punctuation Galore*** Write a text explaining the concept of deadlocks in computer science using as many different punctuation marks as possible. Use them correctly. Your text should be at most 250 words long.

POSSIBLE SOLUTION:

What is a "deadlock"? A deadlock is a—generally undesirable—condition that can occur in concurrent/distributed computing, in which no member of some set of processes can proceed, because every process is waiting for another one to take an action (such as: send a message or release a lock on a resource). For a deadlock to occur, four conditions must be met [28]: mutual exclusion (multiple unsharable resources), resource holding (a process is holding at least one resource and requesting a second one, which is in turn held by another process), no external preemption (only the process currently holding a resource can release it; there is no scheduler that globally assigns resource locks), and circular wait (there exists a set of waiting processes $\{P_1, \ldots, P_n\}$ so that P_1 is waiting for a resource held by P_2, P_2 is waiting for a resource held by P_3, and so on, until P_n, which is waiting for a resource held by… P_1!).

[1] `https://www.medien.ifi.lmu.de/lehre/Plagiate-IfI.pdf`