Technical Writing and Speaking in English
Exercise 2: Writing a scientific article

Danilo Carastan-Santos and Enikő Kevi

September 29, 2023

1 Instructions

The objective is to give you hands-on experience writing a scientific article and teach you to detail, explain, organize, and visually present scientific results. We will focus on a specific segment of the paper each week and spend the second half of the lecture writing. The alleviating factor of this exercise is that the topics are well-known and related to your M1 MOSIG courses. Each of you will write your own paper, but you will work in pairs to review and provide feedback to each other.

During each session you will:

1. Write the lecture's corresponding section of your paper. Example: if the lecture covers the methodology section, you write the methodology section of your paper.
2. Revise your writing according to all technical writing guidelines covered during the course. Example: The class covered the concept of sustaining energy in the text, you will revise your text according to the sustaining energy guidelines (plus previous guidelines).
3. Revise the text of your partner, following every guideline of the course.

2 List of Topics

You must choose **one** (and only one) topic from the list below. The topics only give a high-level overview, so it is your task to design the technical details. For example, depending on the topic, you may create an experimental protocol, write code, create some figures, and so on. [1] Out of all your collected material, you have to decide what are the important parts and concepts to present in your scientific article.

2.1 Compiler Design

Topic 1: Semantics, Inference Systems, Typing

Choose a compiler/interpreter related subject that focuses on the importance of formal languages and the necessary formal methods needed to parse programming languages. You can choose to write a

[1] Use this as an opportunity to deepen your knowledge of one of the first semester's courses.

paper about semantics, inference systems (for example, Hoare calculus), typing, or other subjects that fit this topic. Make sure to detail your chosen concept precisely, state corresponding theorems and proofs, and if applicable, include examples.

Topic 2: Optimization Techniques

You can choose to explore optimization techniques employed by compilers to improve the efficiency and performance of the generated code. Focus on specific optimization methods such as Loop Optimization, Register Allocation, or Common Sub-expression Elimination. Your research paper should delve into the theoretical foundations, algorithms, and practical implementations of the chosen optimization technique within the context of compiler design.

Key points to cover in your paper can include how the optimization technique identifies and optimizes code segments, the impact on runtime performance, and the trade-offs between code size and execution speed. You can also discuss the role of optimizing compilers in modern software development, emphasizing the importance of generating efficient code for various target architectures.

2.2 Software Development Methodologies

Topic 3: Agile - A Comparative Study

Investigate and compare different Agile methodologies such as Scrum, Kanban, Extreme Programming (XP), or Lean. Your research paper should explore the principles, practices, and strengths of the chosen Agile methodologies, along with case studies or examples illustrating their real-world applications.

Your paper can include the roles and responsibilities within each methodology, the project management and planning aspects, the benefits of flexibility and adaptability, and the challenges associated with implementing Agile in various development environments. Furthermore, discuss how Agile methodologies have evolved over time and their impact on software development practices.

2.3 Operating Systems

Topic 4: Memory management

Memory management is a crucial feature of operating systems, with numerous sophisticated techniques. You can choose from a wide range of topics, like address spaces, memory API, address translation, segmentation, paging, and so on. In your research paper, explore the difficulty of memory management, investigate different algorithms, and construct experiments.

Additionally, you can discuss historical aspects of memory management and bring examples of how larger memory capacity causes new problems. You can also dive into recent research on non-volatile random-access memory (NVRAM).

Topic 5: Concurrency Control Mechanisms

You can compare different concurrency control mechanisms used in operating systems, such as Mutexes, Semaphores, Monitors, or Read-Write Locks. Explore in your research paper the principles, implementation details, and practical applications of these concurrency control mechanisms.

Include the theoretical foundations of concurrency control, how each mechanism prevents data races and ensures synchronization among concurrent threads or processes, and the trade-offs in terms

of performance, scalability, and ease of use. You can also discuss case studies or examples illustrating the use of these mechanisms in real-world operating systems or software systems.

2.4 Mathematics for Computer Science

Topic 6: Combinatorial Problems and Proof Techniques

You can choose to research any combinatorial problem or related proof technique. Detail the particularity of these kind of problems, their structure, and importance in real-life. You can examine one problem from different perspectives, or choose a group of problems which share some connection. Besides, you can revisit well-known theorems, provide explanations and visualization, when applicable.

2.5 Visual Computing

Topic 7: Image Processing and Image Segmentation

You can choose to focus on various image processing and image segmentation techniques, such as filters (Gaussian, median, etc.), edge detection, sharpening, and so on. Your research paper can explore the underlying principles, algorithms, and applications of the chosen image processing method. Make sure to cover the mathematical and computational aspects of the technique, its ability to handle various types of images (e.g., low-resolution, bi-colored, medical images, satellite images), and its advantages and limitations. You can also discuss real-world applications, such as object recognition, medical image analysis, and computer vision.

2.6 Algorithmic Problem Solving

Topic 8: Comprehensive Analysis of Famous Problems

Dive deeper into interesting problems and algorithm techniques discussed during the course, or go further and study famous problems, algorithms, or algorithmic complexity proofs and proof techniques. You can also compare different algorithms for the same problem from run time, memory usage, quality of solution, and generality (can the algorithm be used to solve other problems) perspectives.

Find edge-case instances of the studied problem, and try to identify which property of the problem makes the solving 'difficult'. You can prove the performance bounds of the studied algorithm (for example, the best possible run time). Make sure that your research paper includes visualization for the studied problem, pseudo-code for the algorithm, and figures from the experiments.