

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

Ludwig-Maximilians-Universität München
Institut für Informatik
Discussion on 15.10.2025
Homework due on 22.10.2025 at 16:00

Possible solution for Exercise Sheet 1 in Scientific and Technical English for Computer Scientists

The exercise sheets consist of in-class exercises and homework. The in-class exercises take place in 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 Chat-GPT 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 this would defeat the purpose of the exercises.

Homework 1-2 Paper Review Use Google Scholar² to search for some computer science papers. You can search for any subject that you have learned about in your courses, or anything else that interests you. For example, you could search for terms such as “object-oriented programming,” “databases,” and “finite automata.” Skim some of the papers you find interesting, then choose one, read it carefully, and write a brief essay about it. Your essay should answer these questions:

- a) In one or two sentences, what is the paper about?
- b) What do you like about it?
- c) What do you dislike about it?
- d) Which school of writing does it belong to?
- e) Did you find it hard or easy to read? Why?
- f) Was it written for experts or novices?
- g) What background knowledge does it assume? Are there any words you do not know and that are not immediately explained?

Upload the paper you selected along with your essay to Moodle. You can include the paper as an attachment to your submission.

¹<https://www.medien.ifi.lmu.de/lehre/Plagiate-IfI.pdf>

²<https://scholar.google.com/>

POSSIBLE SOLUTION:

“Pathless Scala: A Calculus for the Rest of Scala”^a presents a definition of the semantics of Scala that extends the existing Featherweight Java calculus. The paper clearly explains the author’s contribution with clever use of “negative space”—that is, it describes the state of the art and problems that currently cannot be solved. It gives a comprehensive overview of the notations used right at the start, making the examples easy to follow. It also uses language that clearly identifies the “characters” of certain actions, as within the phrase “we delegate casting [an expression to some type] to an auxiliary judgement.”

On the other hand, I would have liked to see more descriptive (type) variable names in some of their examples. Most of them are just capital letters, when they are not the dreaded *foo*.

The paper appears to belong to the plain English school: Abstract explanations are written in a concise way that leaves room for many examples and comments that help make it clear why certain choices were made.

The paper appears to have been written for an audience that (beyond being familiar with semantic theories and calculi for programming languages) is specifically familiar with the DOT calculus and the Scala ecosystem. It further assumes that the reader knows what *pathless* means in this context, and given that it is used in the paper’s title, it ought to have been explained. The paper also assumes familiarity with some Java concepts. For example, it describes Scala inheritance by explaining how it differs from Java inheritance.

^aGuillaume Martres, *SCALA 2021*, pp. 12–21, ACM, 2021.