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

Ludwig-Maximilians-Universität München
Institut für Informatik
Discussion on 30.10.2026
Homework due on 06.11.2026 at 10:15

Possible solution for Exercise Sheet 3 in

Scientific and Technical English

WiSe 2026/27

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.

In-class exercise 3-1 ***Transatlantic Medley*** The following text inconsistently mixes the rules of American and British English. Identify as many constructs as possible that are specific to either language variant.

> "Alex, we have a small problem", said the woman. Her irritated tone did not match the way she casually leant against the filing cabinet. "Is that a check mark?"
>
> Alex tried to analyse the splotches on the page. "Looks like an exclamation point to me."
>
> "Bloody hell. A pipe broke in the archive room, and now the writing on these is entirely unrecognisable."
>
> 'Actually, that might be more of a midsize problem then, Dr Arbour. If it's unfeasible to discern the grades, we're obligated to repeat the exam within the autumn term.' He was sceptical they would be able to organise that in time, but that was a moot point just now.
>
> Dr. Arbour frowned, and dropped the papers she was holding into the rubbish bin. "Have the committee ruled on that yet?"

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

"Not yet. They tabled it until after the curricula are dealt with. But you know how the maths department's vice president feels about issues of this caliber. I doubt they'll rule in our favour."

"We could always randomize the marks," suggested Dr Arbour.

"If we do that, I'll need to go update my CV." A stunt like that was sure to get his teaching licence revoked.

POSSIBLE SOLUTION:

- American English (AmE) uses double quotes (" ") and puts commas and periods within the closing quote where British English (BrE) uses simple quotes (' ') and puts commas and periods inside or outside, based on the sentence's logic.
- AmE favors *leaned*, BrE *leant* (but both are correct either way).
- AmE *check mark*, BrE *tick*.
- AmE *file cabinet*, BrE *filing cabinet*—BrE favors gerunds in such constructions.
- AmE *analyze*, BrE *analyse*.
- AmE *exclamation point*, BrE *exclamation mark*.
- *Bloody hell* is a very British way to curse.
- AmE *grades*, BrE *marks*.
- AmE *unrecognizable*, BrE *unrecognisable*.
- AmE *midsize*, BrE *medium size*.
- AmE *Dr.*, BrE *Dr*—no period if the last letter of the abbreviation is also the last letter of the word.
- Red herring: AmE *arbor*, BrE *arbour*, but as it is somebody's name, the rule does not apply.
- Some sources claim that BrE should say *infeasible* instead of *unfeasible*.
- AmE *obligated* (unless it is in specific social contexts), BrE *obliged* (except in a legal text).
- AmE *fall*, BrE *autumn*.
- AmE *skeptical*, BrE *sceptical*.

- AmE *organize*, BrE *organise*.
- AmE *moot* means "not worth debating," whereas BrE *moot* means "remains up for debate."
- AmE *trash can*, BrE *rubbish bin*.
- AmE *has the committee ruled*, BrE *have the committee ruled* (if the emphasis is on the individual members).
- AmE *to table an issue* means "to postpone or suspend consideration," BrE *to table an issue* means "to begin consideration."
- Red herring: *Dealt* is always *dealt*.
- Red herring: *Vice president* is always *vice*, but *vice* as in *vice grip* is *vise* in AmE, *vice* in BrE.
- AmE *math*, BrE *maths*.
- AmE *caliber*, BrE *calibre*.
- AmE *favor*, BrE *favour*.
- AmE *randomize*, BrE *randomise*.
- AmE *résumé*, BrE *CV*.
- AmE *license* (both noun and verb), BrE *licence* (as a noun).

In-class exercise 3-2 ***Said in So Many Words*** The following text explains the concept of tail recursion.

> Within the discipline of functional programming, in order for a function to be described as tail recursive, it must fulfill the following criterion: Whenever a function is called recursively, whatever result this recursive call yields must already constitute the final result of the function itself, without any additional computations occurring that further modify this recursive result.
>
> Compared with the naive approach, such an implementation can be said to reduce the memory space a function requires for its execution: Because of the fact that the values of intermediate recursive calls do not need to be processed further, it is not necessary for the compiler to store every return call location up until the point where the entire recursion terminates.
>
> It should, however, also be noted that this constraint on compiler optimization is not always equally applicable to all programming languages. For

instance, if we consider the language Haskell as an example, we find that because evaluations are performed lazily, these considerations become redundant in any and all cases where the evaluation of the recursive call can be delayed up until such a point where its result is actually needed in the computation.

a) Rewrite the above text to make it as concise as you can.

POSSIBLE SOLUTION:

A function is *tail recursive* if the result of the recursive call is also the result of the function call. This can improve space complexity: As recursive call results are not processed further, the compiler can discard intermediate recursive calls' return locations. However, in, e.g., Haskell, this optimization is unnecessary when evaluation of the recursive call is delayed due to laziness.

b) Compare your version with the original version in terms of word count.

POSSIBLE SOLUTION:

- Original version: 191 words.
- Rewrite: 62 words.

The rewritten version is only about 32% of the length of the original.

c) Which version do you find better—e.g., more easily comprehensible, more engaging, or more pleasing to read?

POSSIBLE SOLUTION:

While the new version comes across as more focused and to the point, which improves its understandability, some of the changes in favor of conciseness were made at the expense of clarity, leaving us with hard-to-parse phrasings such as "discard intermediate recursive calls' return locations." Additionally, some of the context that was omitted ("in functional programming" and "compared with the naive approach") might have been helpful.

d) Are there further improvements you could make to your rewritten version?

POSSIBLE SOLUTION:

In functional programming, a function is said to be *tail recursive* if the result of every recursive call inside the function is also the final result of the

function call itself. Tail recursion can be used to improve space complexity: Because the results of recursive calls are not processed further within the function itself, the compiler can discard the return locations of the intermediate recursive calls. However, in a language such as Haskell, laziness can delay the evaluation of recursive calls, achieving an equivalent optimization without the need for tail recursion.