Programming Language Design
Assignment 1 2022

Torben Mogensen and Hans Hüttel

February 7, 2022

This assignment is *individual*, so you are not allowed to discuss it with other students. All questions should be addressed to teachers and TAs. If you use material from the Internet or books, cite the sources. Plagiarism *will* be reported.

Assignment 1 counts 15% of the grade for the course, but you are required to get at least 33% of the possible score for every assignment, so you can not count on passing by the later assignments only. You are expected to use around 9 hours in total on this assignment, including the resubmission (if any). Note that this estimate assumes that you have solved the exercises suggested in slides/videos and participated in the plenary and TA sessions. If not, you should expect to use considerably more time for this assignment.

The deadline for the assignment is **Friday February 18 at 16:00 (4:00 PM)**. The individual exercises below are given percentages that provide a rough idea how much they count (and how much time you are expected to use on them). These percentages do *not* translate directly to a grade, but will give a rough idea of how much each exercise counts.

In normal circumstances, feedback will be given by your TA no later than February 25. Note that, even if the TAs find no errors in your submission, this is no guarantee that it is perfect, so we strongly recommend that you take time to improve your submission for resubmission regardless of the feedback you get. You can do so until **March 7 at 16:00 (4:00 PM)**. Note that resubmission is made as a separate mandatory assignment on Absalon. If you resubmit, the resubmission is used for grading, otherwise your first submission will be used for grading.

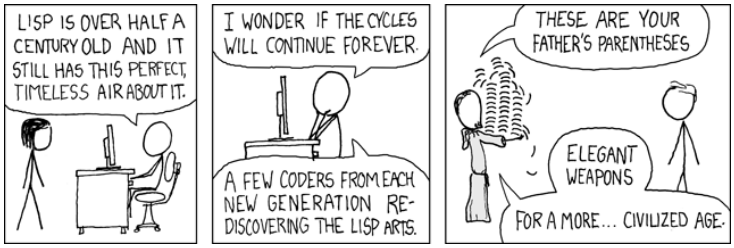
The assignments consists of several exercises, some from the notes and some specified in the text below. You should hand in a single PDF file with your answers and a zip-file with your code. Hand-in is through Absalon. Your submission must be written in English.

A1.1) (10%) **Lambda calculus**

a. Show one step in reducing the following term. Show the reduced term before and after substitution.

$$((\lambda x\,.\,(x\ x))\ (\lambda x\,.\,(x\ x)))$$

A1.2) (40%) **PLD LISP**



If you have not already done so, install PLD LISP as explained in the session description for the History sessions. Then do the following

a. Write in PLD LISP a function `sum` that given a list of numbers return their sum. For example, `(sum '(1 2 3 4))` should return `10`. You can assume that the argument to `sum` is a simple list containing only numbers. You can use functions from `listfunctions.le` in your solution (remember to load them first).

b. Write a similar function in another language. Compare the different syntaxes for size, readability and effort to program of the two versions. Explain your reasoning.

A1.3) (15%) **Bootstrapping**

You have the following components at your disposal:

- A machine running ARM machine code
- An interpreter for F# written in ARM machine code
- An interpreter for LISP written in F#

a. Show Bratman diagrams for these components.

b. Show with diagrams how you, given these components, can execute a program written in LISP.

A1.4) (35%) **Comments**

The following problem concerns design issues that arise when the syntax of a programming language allows programmers to put comments in their code.

a. Here are a few lines of C++ that somebody wrote.

```
/* Initialize the variable a
   a = 0;
/* Set the minimum size */
   min = 100;
```

What is the problem with this piece of code?

b. In some programming languages, comments can be nested – that is, inside a comment there can be other comments.

Provide a list of advantages and disadvantages to allowing nested comments. The advantages/disadvantages can be related to readability, to writability, to syntax checking or to something else.

For each advantage/disadvantage argue why this is an advantage/disadvantage and give a small example of code (just as in was done in the subproblem above) that illustrates this particular advantage/disadvantage.

c. Some programming languages allow for compiler directives in comments. A pragma comment is a compiler directive that tells the compiler to leave a comment in the object file that is generated. This comment can then be read by later compiler phases and the linker and be used to modify the behaviour of these phases. What are the advantages and disadvantages to having such compiler directives in comments?

Provide a list of advantages and disadvantages and in each case give a small example of code that illustrates this particular advantage/disadvantage.