

Assignment 2: Operational Semantics

CS - 456 – Programming Languages

September 11, 2014

General Information

- Due Date: 09/25/2014 by midnight.
- Submission: A *theory* and a *programming* assignment will be created on BlackBoard. You have to submit both independently. Multiple submissions are possible before the deadline. Only the last one will be considered.
- Theory assignment: *individual*. Programming assignment: in *pairs* or individually.
- Use Piazza for Q&A.

Programming Assignment: 40%

This assignment can be resolved in groups of at most two.

- Solve exercise 26 of Chapter 2 of the book (pg. 58).
- New versions of `all.h`, `ast.c`, `definition-code.c` and `parse.c` to handle local variables are provided under the directory `impcore-with-locals` of the code. Run `diff -r bare/impcore bare/impcore-with-locals` to see what's different.
- You will have to modify the initial basis of `impcode.c` to deal with the new syntax.
- You will have to modify `eval.c` to give proper semantics to local variables.
- Local variables have precedence over formal parameters. Therefore if we define the following function:

```
(define func (n m) (n) (+ n m))
```

and we call `(func 3 2)` we would obtain 2 as result, because `n` in the expression `(+ n m)` refers to the local `n` which is initially 0.

- Feel free to modify any of the files as long as you preserve all the functionality of `ImpCore`.

- Document your solution in a Readme file.

Theory Assignment: 60%

Individual assignment. Exercise 3 requires much more time than exercises 1 and 2.

1. (Operational Semantics) Solve exercise 10 and 11 of Chapter 2 of the book (pg. 55). Provide the derivation tree. For exercise 11 you will need to provide 2 proof trees and argue that the resulting v_1 and v_2 are necessarily equal.
2. (Language Design) Solve exercise 13 of Chapter 2.
3. (Properties of Languages) Solve exercise 20 of Chapter 2 (pg. 56). The general proofs are by induction on the derivation trees of the semantics. A sample case of the proof is provided as additional material.

General Evaluation Criteria

The programming exercise will be subject to the same evaluation criteria as Assignment 1. Test your own implementation. Test cases are not provided. You can provide your test cases in your submission.

Submission

- Programming Assignment:
 1. Submit a single (zip or tar) file containing a directory with the following naming structure {lastname1}-{lastname2}-programming-hw2.
 2. All students must submit the programming assignment individually (people working in pairs will submit the same file) to simplify grading in blackboard.
- Theory assignment
 1. Submit each exercise *in order* and start each exercise in a different page.
 2. Number your pages and add the total number of pages submitted in the first page.
 3. Submit a single pdf file containing all your answers. Your file should be named {lastname}-theory-hw2.pdf.
 4. The preferred method for producing the pdf file is latex (`pdflatex`). Use the package `mathpartir.sty`, in particular the `\inferrule{}{}{}` command to write derivation trees. If needed a template can be provided in Piazza.
 5. If you cannot use latex, you can convert a handwritten document into pdf and submit it. Handwritten documents have to be easy to read and might be subject to a higher standard.