

Assignment for Semantics and Domain Theory (IMC011)

Herman Geuvers

Goal

You should write a short report (pdf, maximum 10 pp) about a formal language or a feature of a formal language that has an operational semantics (OS) and a denotational semantics (DS). You should formally describe this operational semantics and a denotational semantics and relate them, like we have done for PCF and While in the course. The primary goal is to give the proper definitions, give some enlightening example(s) and *state* the relations between operational and denotational semantics via lemmas that you claim to hold. Please speculate about how you would prove these lemmas if you had the time.

NB. The language or feature you discuss doesn't have to be new: there is a lot of research in this field, studying the OS and DS of programming languages and programming language features. Explaining some existing results in your own words, with your own examples, or slightly adapting existing results is fine. But of course you should refer to the literature you have read, also webpages, blogposts, etc.

Grading

Your assignment will be evaluated on the following points

1. Interesting feature, level of complexity.
2. Correctness of the definitions.
3. Motivating and enlightening example(s).
4. Correctness of the relations between the definitions: have they been really thought through well?
5. Quality of the report: explanation of problem and definitions, good mixture of formal definitions and informal explanations.
6. Independency (“zelfstandigheid”): how much help/suggestions were received?
7. Plus features: Remarkably smart definitions; additional proofs and properties.

Suggestions

Extending PCF or While with additional programming features or varying on the known semantics of PCF or While is a good idea. Don't make it too complex: study one feature at a time or in isolation; a lot of the real complexity in semantics of programming languages arises through feature interaction.

In case you have questions: please consult the teacher.

Concrete suggestions:

- Extend While with blocks and procedures. The OS and DS are defined in [1]. You can take this and explain it in your own words, discussing the choices between *static* and *dynamic* scope, and do some of the exercises relating them that are in [1].
- Extend While with a continuation semantics towards dealing with exceptions. The OS for exceptions (raise/handle) is given in [1]; the continuation based DS is also defined in [1]. You can take this and give an OS and DS for While with handle/raise, or think of other exception mechanisms like break/continue from Java.
- In the second set of exercises for this course (see the webpage of the course), the “for-loop” has been mentioned: how could one give semantics to it and how does this depend on the *static* and *dynamic* scope of the variables? You can make the OS and DS of the for-loop precise, discussing the various options and problems.
- Extend While with expressions whose evaluation has a side-effect, like x^{++} or $++x$ in Java. ($y := x^{++}$ assigns the value of x to y and then increments the value of x .)
- Extend PCF with monadic types, to capture side-effects (e.g. I/O) or exceptions (e.g. letting `pred 0` raise an exception).
- Extend PCF with other types, e.g. a list data type, and give the OS and the DS for that, discussing the possible choices, e.g. call-by-name vs. call-by-value.

[1] Hanne Riis Nielson and Flemming Nielson: Semantics with applications, Wiley 1999. See the website of the course for the freely available pdf file.