Program Semantics, Verification, and Construction - Part I
MAP-i
2008-2009

This assignment is to be accomplished by groups of two students. Each group must:

1. Choose **a topic** from the list bellow. Please send an email to `nam@ncc.up.pt` with the name/email of the group members and the topics ordered by preference (higher, first). Topics will be allocated by order of email arrival.
2. Write an essay about the allocated topic. Due date: one week before the presentation session.
3. A 20 minutes presentation in a date to be announced (before December 20).

Topics

λ-calculus, deduction systems, type systems, Curry-Howard isomorphism and semantics

1. **Combinatory logic:** definition; relation with the λ-calculus; relation with Hilbert-style proofs. See [PU96], chapter 5.
2. **Expressibility:** the simply typed λ-calculus (vs. the pure system), system T. See [PU96], chapters 3, 11.
3. **Classical propositional logic:** Curry-Howard correspondence for classical propositional logic and control expressions. See [PU96], chapter 8 and [Gri90].
4. **Sequent calculi:** Curry-Howard correspondence for intuitionistic propositional logic presented in sequent calculus. Discuss if we have a one-to-one map. See [PU96] chapter 7 and [BG02].
5. **Type systems I** Consider an extension the simple type system for the λ-calculus, for a term language consisting of the λ-calculus, integers, arithmetic operators $(+, -, \times)$, *if then else*, pairs, projections (*first* e *second*), local variable definitions *let* $x = t_1$ *in* t_2, where the scope of x is t_2 and a *rec* operator $y.(\lambda x.t)$, where the function $y = \lambda x.t$ is defined recursively (i.e. y occurs free in t). Present a type system, an operational and a denotational semantics using *call by value*, show their equivalence and type preservation (i.e subject reduction) for the operational semantics.

   See [Win93] chapter 11 (chapter 9 or [Fer04] for an introduction).

6. **Type systems II** The same type system and questions of topic 5 but using *call by name*. See [Win93] chapter 11 (chapter 9 or [Fer04] for an introduction).

References

[BG02] Henk Barendregt and Sylvia Ghilezan. Lambda terms for natural dedution, sequent calculus and cut elimination. *J. of Functional Programming*, 2002.

[Fer04] Maribel Fernandez. *Programming Languages and Operational Semantics.* King's College Publications, 2004.

[Gri90] Timothy G. Griffin. A formulae-as-types notion of control. In *In Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages*, pages 47–58. ACM Press, 1990.

[PU96] Morten B. Sorensen Pawel Urzyczyn. Lecture on the curry-howard isomorphism. Technical report, University of Copenhagen, 1996. http://zls.mimuw.edu.pl/ urzy/ftp.html.

[Win93] Glynn Winskel. *The Formal Semantics of Programming Languages.* MIT Press, 1993.