Type checking Dependent (Record) Types and Subtyping

Gustavo Betarte

Instituto de Computación
Universidad de la República
Montevideo, Uruguay
email:gustun@fing.edu.uy

The subject of this talk is the specification and implementation of a proof checker for an extension of Martin-Löf's theory of logical types [ML87] with dependent record types and subtyping.

The original formulation of Martin-Löf's theory of types, from now on referred to as the logical framework, has been presented in [NPS89, CNSS94, Tas97]. The system of types that this calculus embodies are the type Set (the type of inductively defined sets), dependent function types and for each set A, the type of the elements of A.

The extension of the logical framework with dependent record types and subtyping is presented in [Tas97, BT99]. Dependent record types are just sequences of *fields* in which *labels* are declared as of certain types. These types, in turn, may not only depend on objects but also on labels. How this dependency is obtained is formally introduced in the rules for record types formation. Record objects, as in programming languages, are sequences of assignments of objects of appropriate types to labels. Each of these objects can be accessed by selecting the corresponding label of the record object. The mechanism of subtyping or type inclusion introduced is, in the first place, the one naturally induced by record types. However, once record inclusion is formally stipulated it is also required that rules of subtyping have to be given for the rest of the type formers.

In [Bet98] we have investigated an alternative formulation of the extended theory. In that formulation we make use of parameters, in the sense of [Coq91, Pol94], to stand for generic objects of the various types. The introduction of the notion of parameter allows to give a solution to the problems posed by the manipulation of "free names" in the presence of dependent types. We also present in [Bet98] the procedures for the mechanical verification of the forms of judgement of that variant of the extension.

In this talk we shall be mainly concerned with the design and specification of those procedures. A detailed and complete formulation of the constructions we shall present can be found in [Bet98, Bet00].

References

[Bet98] Betarte, G. (1998). *Dependent record types and algebraic structures in type theory.* Ph.D. thesis, PMG, Dept. of Computing Science, University of Göteborg and Chalmers University of Technology.

[BT99] Betarte, G., & Tasistro, A. (1999). *Extension of Martin-Löf's Type Theory with Record Types and Subtyping.* In Sambin, G., & Smith, J.M. (eds) *Twenty-five years of constructive type theory.* Oxford University Press. Pages 21–39.

[Bet00] Betarte, G. (2000). *Type Checking Dependent (Record) Types and Subtyping.* To be published in *Journal of Functional Programming.* Cambridge University Press.

[Coq91] Coquand, Th. (1991). An algorithm for testing conversion in type theory. *Pages 71–92 of: L Martin-Löfogical Frameworks, Huet G., Plotkin G. (eds.).* Cambridge University Press.

[CNSS94] Coquand, Th., Nordström, B., Smith, J.M., & von Sydow, B. (1994). Type theory and programming. *EATCS 52.*

[ML87] Martin-Löf, P. (1987). *Philosophical Implications of Type Theory.* Lectures given at the Facoltá de Lettere e Filosofia, Universitá degli Studi di Firenze, Florence, March 15th. - May 15th. Privately circulated notes.

[ML92] Martin-Löf, P. (1992). *Substitution calculus.* Talks given in Göteborg.

[NPS89] Nordström, B., Petersson, K., & Smith, J. M. (1989). *Programming in Martin-Löf's Type Theory.An Introduction.* Oxford University Press.

[Pol94] Pollack, R. (1994). *The Theory of LEGO: a proof checker for the Extended Calculus of Constructions.* Ph.D. thesis, University of Edinburgh.

[Tas97] Tasistro, A. (1997). *Substitution, record types and subtyping in type theory, with applications to the theory of programming.* Ph.D. thesis, PMG, Dept. of Computing Science, University of Göteborg and Chalmers University of Technology.