Marie Curie project SImPL

1/4/2010 – 31/3/2012
Final Report

Vladimir Komendantsky

18th May 2012

The project SImPL[1] was dedicated to provably-correct extensions of regular expressions with notions of a pattern and a backreference, and to mechanisations of decision algorithms on regular expressions in the proof assistant Coq [1]. By a coincidence, `simpl` is the name of a computational tactic in Coq that is used for computing proofs following the method of computational (a.k.a. small-scale) reflection, one of the key methods employed in SImPL.

Regular expressions [12, 10] are a formalism ideally suited to specification and implementation with formal methods. They are essential for text processing and form the basis of most markup schema languages. Regular expressions are useful in the production of syntax highlighting systems, data validation, speech processing, optical character recognition, and in many other situations when we attempt to recognise patterns in data. Extended versions of regular expressions are used in search engines such as Google Code Search. In fact, there is a difference between what is understood by the term *regular expression* in programming and in theoretical computer science. Different software based on regular expressions has in each case its own "RegEx flavour": ECMAScript, Perl-style, GNU RegEx, Microsoft Word, POSIX Basic/Extended RegEx (with extensions), Vim, and many others. In this project, I worked with an algebraic definition of regular expression matching that rests upon the concept of *partial derivatives*. I appropriately extended algebraic matching of regular expressions to account for *backreferences*. The project has yielded several peer-refereed papers [7, 3, 6, 5, 9, 8].

The following objectives have been duly reached: *Objective 1.* Theoretical representation of extended, or, *practical*, regular expressions in constructive dependent type theory. *Objective 2.* A Coq library for regular languages and automata that includes features not present in available related libraries, such as backreferences and partial derivatives of regular expressions. *Objective 3.* A formally certified `grep`-like extended regular expression parser (in other words, a formally certified compiler of extended regular expressions into finite automata).

A much broader aim of SImPL was helping to provide robust and transparent data infrastructure for the future Internet (which is a part of the European Commission *ICT Challenge 1: Pervasive and Trustworthy Network and Service Infrastructures*). The primary object of research was data, in contrast with computation, in the sense of the duality emphasised in the seminal paper [2] with respect to [11]. Therefore the intended application of the results of the project is formal data certification. Application to proving computational correctness was not perceived as a specific goal. However,

[1] `http://archive.cs.st-andrews.ac.uk/SImPL/`

due to the foundational nature of regular expressions, for example, in relationship to concurrency, the results on decision methods for extended regular expressions can be employed in proving correctness of data-parallelism.

References

[1] Coq development team. The Coq proof assistant reference manual. `http://coq.inria.fr/refman/`.

[2] K. Fisher, Y. Mandelbaum, and D. Walker. The next 700 data description languages. In *POPL '06*, pages 2–15. ACM, 2006.

[3] V. Komendantsky. Application of monadic substitution to recursive type containment. In S. Escobar, editor, *Pre-proceedings of the 10th Workshop on Reduction Strategies in Rewriting and Programming (WRS 2011)*, Wrocław, Poland, 29 May 2011.

[4] V. Komendantsky. Formal proofs of the prebase theorem of Mirkin, 2011. Coq script available at `http://www.cs.st-andrews.ac.uk/~vk/doc/prebase.v`.

[5] V. Komendantsky. Packed views of pre-structured data. In *Workshop on Computer Theorem Proving Components for Educational Software, THedu 2011*, Wroclaw, Poland, 31 July 2011. Extended abstract.

[6] V. Komendantsky. Regular expression containment as a proof search problem. In S. Lengrand, editor, *Proceedings of the International Workshop on Proof-Search in Axiomatic Theories and Type Theories (PSATTT'11)*, Wrocław, Poland, 30 July 2011.

[7] V. Komendantsky. Subtyping by folding an inductive relation into a coinductive one. In R. Peña, editor, *Post-proceedings of the 12th International Symposium on Trends in Functional Programming (TFP 2011)*, LNCS, Madrid, Spain, 16-18 May 2011. Springer.

[8] V. Komendantsky. Matching problem for regular expressions with variables. In *Proceedings of the 13th International Symposium on Trends in Functional Programming (TFP 2012)*, 2012. To appear.

[9] V. Komendantsky. Reflexive toolbox for regular expression matching: Verification of functional programs in Coq+Ssreflect. In *The 6th ACM SIGPLAN Workshop Programming Languages meet Program Verification (PLPV'12)*, Philadelphia, USA, 24 January 2012. For contributed proofs, see [4].

[10] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Infor. and Comput.*, 110(2):366–390, 1994.

[11] P. J. Landin. The next 700 programming languages. *Commun. ACM*, 9(3):157–166, 1966.

[12] S. Yu. Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*, volume 1: Word, language, grammar, pages 41–110. Springer-Verlag, 1997.