TRANSLATING CONCURRENT RSL INTO PVS

A. Dasso, A. Funes, G. Montejano, D. Riesco, R. Uzal
{arisdas, afunes, gmonte, driesco, ruzal}@unsl.edu.ar

SEG
Universidad Nacional de San Luis
Ejército de los Andes 950
D5700HHW San Luis
Argentina
http://sel.unsl.edu.ar/
Tel.: +54 (0) 2652 42 4027 ext. 251
Fax: +54 (0) 2652 43 0224

ABSTRACT

Translating RSL into PVS has been a way of obtaining a proof tool for RSL. However there are several styles or constructs of RSL that are harder to translate than others. One of them is concurrent RSL since PVS has no similar construct. We propose to study the problems involved in translating Concurrent RSL into PVS.

INTRODUCTION

"PVS is a Prototype Verification System for the development and analysis of formal specifications. The PVS system consists of a specification language, a parser, a type checker, a prover, specification libraries and various browsing tools." [OSR99]

"RAISE is an acronym for 'Rigorous Approach to Industrial Software Engineering'. It was the name of a CEC funded ESPRIT project and now gives its name to a formal specification language, the RAISE Specification Language (RSL [RLG92]), an associated method and a set of tools … the tools have been commercially available for sometime". [RMG95].

As we said above RAISE is supported by several tools some of which are commercially available but not in the public domain. On the other hand, there are a number of tools that can be freely accessed and that can help in the development process when using RSL. These tools are developed and maintained at UNU/IIST, see [GEO01], but the set of these tools does not include a prover.

PVS has a potent prover that is freely available, so translating RSL into PVS has the benefit of producing a tool that can be freely available and allows a specification written in RSL to be proved in PVS using the PVS prover.

Since we are dealing with formal languages the translation should also be correct which implies dealing with the logics and other aspects of both languages.

Although RSL and PVS share some similarities in their logics –both are based in typed first order logic– there are differences in their logics as well as in the language. This difference is most marked by the fact that PVS has no corresponding construct for concurrency as has RSL.

PAST, CURRENT AND FUTURE WORK

Some of us have already worked in the translation of most of RSL into PVS, see [DaGe02]. In doing so we have dealt with the different semantics frameworks of both languages, but we have not included the concurrent constructs of RSL.

Our present work is centred in solving the theoretical problems that arise in dealing with concurrency, while reserving for future work the implementation of those constructs that can be safely translated, where safely means that the translation is sound (for a complete discussion on this subject see [DaGe02] and specially [GeDa03]).

REFERENCES

[OSR99] S. Owre, N. Shankar, J. M. Rushby, D. W. Stringer-Calvert, "PVS Language Reference", SRI International, Computer Science Laboratory, 1999.

[RMG95] The RAISE Method Group, "The RAISE Development Method", Prentice Hall International (UK), 1995.

[GEO01] Chris George, "RAISE Tools User Guide", Technical Report 227, UNU/IIST, P.O. Box 3058, Macau, February 2001, url = http://www.iist.unu.edu

[DaGe02] A. Dasso, C George, "", Technical Report 256, UNU/IIST, P.O. Box 3058, Macau, May 2002, url = http://www.iist.unu.edu

[RLG92] The RAISE Language Group, "The RAISE Specification Language", Prentice Hall International (UK), 1992.

[GeDa03] C. George, A. Dasso, "Using PVS as a proof tool for RSL", to be presented at the 12th International FME Symposium. Pisa, Italy, September 8-14, 2003.