Parallel Functional Reactive Skeletons in Concurrent Clean*

Zoltán Horváth, Viktória Zsók†
Pascal Serrarens, Rinus Plasmeijer ‡

Abstract

The skeletons are parameterised algorithmic schemes. We can use them to control parallel execution of the programs. Skeletons in functional programming languages are expressed as higher order functions and they allow to implement the well-known parallel programming paradigm in portable, efficient programs. The skeletons can be parameterised triply: by a function that computes the value of the result, by a strategy determining the dynamic behaviour and by types. Reactive systems are systems which have some interactions with their environment, but doesn't give a final result, usually they are non-deterministic. In this paper we would like to present a skeletal approach of the reactive systems in the parallel lazy functional language Concurrent Clean. Concurrent Clean provides process annotations for explicit thread creation and data-driven message passing system for the communication.

References

[1] P. Achten, R. Plasmeijer: Interactive Objects in Clean, In: C. Clack et al., eds., *Implementation of Functional Languages, 9th International Workshop, IFL'97*, St. Andrews, Scotland, UK, Sept. 1997, Selected Papers, LNCS, Vol. 1467, pp. 304–321, Springer-Verlag, 1998.

[2] K.M. Chandy, J. Misra: *Parallel program design: a foundation*, Addison-Wesley, 1989.

[3] L.A. Galán, C. Pareja, R. Peña: Functional Skeletons Generate Process Topologies in Eden, In: *Int. Symp. on Programming Languages, Implementations Logics and Programs, PLILP'96*, Aachen, Germany, LNCS, Vol. 1140, pp. 289–303, Springer-Verlag, 1996.

[4] Á. Fóthi, Z. Horváth, T. Kozsik: Parallel Elementwise Processing – A Novel Version. In: L. Varga, ed., *Proceedings of the Fourth Symposium on Programming Languages and Software Tools*, Visegrád, Hungary, June 9-10, 1995, pp. 180–194,

*OTKA ....

†Department of General Computer Science, H-1117 Budapest, Pázmány Péter sétány 1/D, Hungary, e-mail: hz@inf.elte.hu, zsv@inf.elte.hu

‡Faculty of Mathematics and Computer Science, University of Nijmegen, NL-6500 GL Nijmegen, PO. Box 9010, The Netherlands, e-mail: pascalrs@cs.kun.nl, rinus@cs.kun.nl

and in: *Annales Univ. Sci. Budapest de L. Eötvös Nom. Sectio Computatorica*, Vol. 17, pp. 105–124, 1998.

[5] Horváth Z.: *A relational programming model of parallel programs*, PhD thesis, in Hungarian. PhD program in Informatics, Department Group Informatics, University Eötvös Loránd Budapest, Hungary, 1996.

[6] M. Kesseler: Constructing Skeletons, In: *The Implementation of Functional Languages on Parallel Machines with Distributed Memory*, PhD Thesis, Catholic University Nijmegen, pp. 153–173, The Netherlands, 1996.

[7] P.R. Serrarens, R. Plasmeijer: Explicit Message Passing for Concurrent Clean, In: K. Hammond et al., eds., *Implementation of Functional Languages, 10th International Workshop, IFL'98*, London, UK, Sept. 1998, Selected Papers, LNCS, Vol. 1595, pp. 229–245, Springer-Verlag, 1999.

[8] R. Plasmeijer, M. van Eekelen, *Functional Programming and Parallel Graph Rewriting*, Addison-Wesley, 1993.

[10] P.W. Trinder, K. Hammond, H.W. Loidl, S.J. Peyton Jones: Algorithm + Strategy = Parallelism, *Journal of Functional Programming*, Vol. 8, No. 1, pp 23–60, 1998.

[11] C. Okasaki: *Edison User's Guide (Haskell version)*, http://www.cs.columbia.edu/ cdo/edison/, May, 1999.

[12] M. Cole: Algorithmic Skeletons, In: K. Hammond, G. Michaelson, eds., *Research Directions in Parallel Functional Programming*, pp. 289–303, Springer-Verlag, 1999.

[13] M. Cole: Algorithmic Skeletons: Structured Management of Parallel Computation, *Research Monographs in Parallel and Distributed Computing*, The MIT Press, Cambridge, Massachusetts, 1989.