Parallel Functional Skeletons in Concurrent Clean[1]

Zoltán Horváth, Viktória Zsók
Department of General Computer Science, University Eötvös Loránd, Budapest
hz@ludens.elte.hu, zsv@valerie.inf.elte.hu

Pascal Serrarens, Rinus Plasmeijer
Faculty of Mathematics and Computer Science, University Nijmegen
pascalrs@cs.kun.nl, rinus@cs.kun.nl

The behaviour of the concurrent and parallel programs can be specified in a functional style. In this paper we introduce a set of polymorphic higher order functions to formulate our abstract parallel programming theorems [2] in a pure functional style.

Functional programs are inherently parallel, but the evaluation of an expression in parallel is not always worthwhile. Evaluation strategies [4] are used to define the concrete program's dynamic behaviour, the evaluation degree and the control of the parallelism, separating dynamic evaluation issues and static requirements. A skeleton is a parameterized algorithmic scheme. Imperative skeletons are the counterpart of the abstract program of a programming theorem. Functional skeletons correspond to the generalized specification part of a programming theorem.

We investigate the parallel functional computation of associative operators, many-argument elementwise processable functions, pipelined functions composition and simple reactive systems in the parallel functional language Concurrent Clean [3]. Concurrent Clean provides tools for controlling parallel evaluation by annotations and language primitives for explicit thread creation and message passing. We build both transformational and reactive skeletons [1]. We develop the concrete parallel functional skeletons expressed in Concurrent Clean by reusing the intermediate results of the same refinement process which was applied for synthetizing the abstract parallel programs.

References

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

[2] Fóthi Á.- Horváth Z.- Kozsik T.: Parallel Elementwise Processing – A Novel Version. in Varga L., ed., *Proceedings of the Fourth Symposium on Programming Languages and Software Tools*, Visegrád, Hungary, June 9-10, 1995 (1995), pp. 180-194. and in *Annales Uni. Sci. Budapest de R. Eötvös Nom. Sectio Computatorica*, Tom. XVII. (1998), pp. 105-124.

[3] Serrarens, P.R.: Explicit Message Passing for Concurrent Clean. In K. Hammond et al.: *Proc. of Implementation of Functional Languages, IFL'98*, pp. 298-308, Sept. 1998, London, to appear in LNCS.

[4] Trinder, P.W., Hammond, K., Loidl, H.W., Peyton Jones, S.J.: Algorithm + Strategy = Parallelism. *Journal of Functional Programming* 8 (1): pp. 23-60, 1998.

[1]Supported by the Hungarian State Eötvös Fellowship, by the Hungarian National Science Research Grant (OTKA), Grant Nr. T032178 and by FKFP 0206/1997.