A Grammatical Approach for Parallel Control of Declarative Programming Languages

A. M. Thanos, PhD thesis review

National Technical University of Athens
Department of Electrical and Computer Engineering
9 Iroon Polytechniou str. 15773 Zografou Campus, Hellas
Tel: +30 1 7721530, Fax: +30 1 7722496
thanos@cslab.ece.ntua.gr

The thesis presents how different control operations on parallel implementations of declarative programming languages (Logic Programming (LP), Functional Logic Programming (FLP), Concurrent Constraint Logic Programming (CCLP)) can be modelled. A program analysis method based on Attribute Grammar dependency graphs is used. The PAGE framework, which facilitates the development of parallel implementation of declarative languages, is presented. As in compiler technology, AGs are used as a specification language for the description of the programming paradigms under consideration. Each of the programming paradigms is outlined from a transformation table or a combination of them. These tables consist of transformation actions that have to be applied, under some conditions. The transformations are described in the form of AG semantic rules. With this analysis a large part of the control of such languages is discerned, named control rationale, which can be specified in a programmable way. The remaining control part forms a non-programmable layer, named control principles, which follows the restrictions of the underlying hardware architecture of each implementation. While the principles component of the control is a set of mechanisms depending on the underlying implementation of the system (computational model), control rationale component is a set of specifications described via proper semantic rules and reflects the special operational semantics of the programming paradigm under consideration. Each of the paradigms follows the operational semantics induced by the dependency graphs corresponding to an AG. Operationally complete execution is guaranteed by following the evaluation order prescribed by the dependency graphs. The methodology could be proved to be a very valuable tool for supporting the automation of the implementation of more general classes of parallel programming paradigms. The system has been implemented and tested in a wide range of architectures, exhibiting encouraging results.