Abstract
Gamma is a minimal language based on conditional multiset rewriting. The virtues of this paradigm in terms of systematic program construction and design of programs for highly parallel machines have been demonstrated in previous papers. We introduce here sequential and parallel operators for combining Gamma programs and we study their properties. The main focus of the paper is to give conditions under which sequential composition can be transformed into parallel composition and vice versa. Such transformations are especially valuable for adapting Gamma programs for execution on a particular target architecture.

1. Introduction

We have seen in the last decade the emergence of a new generation of parallel machines featuring a huge number of processors, the most significant representative being the Connection Machine [8] with its 64k processing elements. In order to make these architectures upgradable and cost effective, individual components are kept as simple as possible and they are connected in a loosely coupled way. These unique features impose the need for the design of new programming techniques for exploiting these machines efficiently. It is very unlikely that a programmer can mentally manage the details of the decomposition of a program into a great number of individual tasks. The programmer should rather be provided with a high-level programming language in which parallelism is left implicit and an associated compiler which is able to detect this implicit parallelism and to map it effectively onto the parallel architecture. So these massively parallel
machines raise interesting challenges in the fields of parallel language design and compiler construction.

The Gamma formalism was proposed a few years ago to allow the description of programs without artificial sequentiality (by artificial we mean sequentiality that is not implied by the logic of the program). A simple example illustrates this point. The problem is to find the maximum element of a non-empty set. In Gamma such a program can be written as follows:

\[
\text{maxset: } x, y \rightarrow y \iff x \leq y
\]

\(x \leq y\) specifies a property to be satisfied by the selected elements \(x\) and \(y\); these elements are replaced in the set by the value \(y\). Nothing is said in this definition about the order of evaluation of the comparisons; if several disjoint pairs of elements satisfy \(x \leq y\), the comparisons and replacements can even be performed in parallel. An intuitive way of describing the meaning of a Gamma program is the metaphor of the chemical reaction: the set can be seen as a chemical solution; \(R(x,y) = x \leq y\) is called the reaction condition and \(A(x,y) = \{y\}\) is the action. The computation terminates when a stable state is reached, that is to say when no elements of the set satisfy the reaction condition.

We illustrate the programming style of the language with two examples. The following program returns the set of prime numbers smaller than \(n\).

\[
\text{pn}(n) = \text{prime}\_\text{numbers} \{2, ..., n\}
\]

\[
\text{prime}\_\text{numbers: } x, y \rightarrow y \iff \text{multiple}(x,y)
\]

where \(\text{multiple}(x,y)\) holds if \(x\) is a multiple of \(y\). The program proceeds by removing from the set \(\{2, ..., n\}\) elements that are multiples of another element in the set.

Next we consider the sorting problem: the goal is to organize the elements of an array in increasing order. We use a multiset of pairs (index, value) and the program exchanges ill-ordered values until all values are well-ordered.

\[
\text{sort: } (i,v), (j,w) \rightarrow (i,w), (j,v) \iff (i > j) \text{ and } (v < w)
\]

The interested reader may find in [3] a series of examples (string processing problems, graph problems, geometric problems, ...) illustrating the Gamma style of programming. The possibility of getting rid of artificial sequentiality in Gamma has two important consequences:

- It makes Gamma suitable as an intermediate language in the program derivation process allowing the programmer to design a very abstract version of his program in the first