The Projection of Systolic Programs

C. Lengauer and J. W. Sanders

Programming Research Group, Oxford University Computing Laboratory, 8–11 Keble Road, Oxford OX1 3QD, UK

Keywords: Communicating sequential processes; occam; Systolic array

Abstract. A scheme is presented which transforms systolic programs with a two-dimensional structure to one dimension. The elementary steps of the transformation are justified by theorems in the theory of communicating sequential processes and the scheme is demonstrated with an example in occam: matrix composition/decomposition.

1. Introduction

We combine two types of formal refinement to transform a two-dimensional systolic program to one dimension.

1.1. Systolic Design

Systolic arrays are particularly regular distributed processor networks capable of processing large amounts of data quickly by accepting streams of input and producing streams of output [KuL80]. Typical applications are to image or signal processing; ours is an algorithm which subsumes matrix composition and decomposition.

2 On leave from the Department of Computer Sciences, The University of Texas at Austin, Taylor Hall 2.124, Austin, Texas 78712-1188, USA.
3 Correspondence and offprint requests to: C. Lengauer, Department of Computer Science, University of Edinburgh, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, Scotland, UK.
Systolic arrays are usually realised in hardware. We are interested in realising them in software, because then they can run on one of the families of distributed computers (now plentiful) capable of emulating systolic arrays. We are led to express such software in a distributed programming language that provides constructs for process definition and communication. The production of that software is relatively straightforward if the program's process and channel structure, which matches the processor and communication structure of the systolic array, also matches the distributed computer. That is not always the case. The distributed computer may not offer the processor layout and interconnections that the systolic program prescribes.

We consider one specific such case: the processor layout of the machine has fewer dimensions than the process layout of the systolic program. In this case, a projection, i.e., a transformation of the process layout of the systolic program, is required. We consider the transformation of two-dimensional systolic programs into one-dimensional systolic programs. There are programming environments that permit the specification of a mapping from software processes to hardware processors (e.g., for a transputer network [INM84]), which makes explicit program projections unnecessary. We require this mapping to be the identity in order to avoid inefficiencies caused by the software simulation of channel communication.

1.2. Formal Methods

The method we use to justify the projection from two dimensions to one appears to be novel. It can be thought of as a variant of a hybrid of refinement techniques used in "formal methods". There, criteria for the refinement of sequential systems involve a relation between the states of the two systems [Hoa86, Nip86]; criteria for the refinement of concurrent systems enable one system to be replaced by another in any environment [Hoa85, Jac87]. We employ a technique of state relabelling which enables one system to replace another in any of a restricted class of environments. We hope this feature will be useful in other contexts. The refinement, as usual, makes a program more specific for the machine at hand: by postulating a one-dimensional systolic architecture, it leads from the ideal two-dimensional design to a one-dimensional implementation.

2. The Problem

We are given three square matrices: A, B and C. Our goal is to establish that C is the matrix product of A and B:

\[ (\forall i, j : 0 \leq i, j < n : c_{i,j} = \sum_{k:0 \leq k < n} a_{i,k} \cdot b_{k,j}) \]

That goal may be achieved in different ways, depending on which of the matrices are to be determined. We consider two possibilities. Because we wish to derive a systolic solution we shall assume that the matrices are distinct program objects, i.e., they do not share elements.

2.1. Matrix Composition

A and B are input and C is output. A and B uniquely determine C.