Formal Verification of Coherence for a Shared Memory Multiprocessor Model

Manuel Barrio-Solórzano[1], M. Encarnación Beato[2], Carlos E. Cuesta[1], and Pablo de la Fuente[1]

[1] Departamento de Informática. Universidad de Valladolid, Spain
[2] Escuela de Informática. Universidad Pontificia de Salamanca, Spain

Abstract. The specification and verification of shared-memory multiprocessor cache coherence protocols is a paradigmatic example of parallel technologies where formal methods can be applied. In this paper we present the specification and verification of a cache protocol and a set of formalisms which are based on *'process theory'*. System correctness is not established by simple techniques such as testing and simulation, but 'ensured' in terms of the underlying formalism. In order to manipulate the specification and verify the properties we have used an automated tool —namely the 'Edinburgh Concurrency Workbench' (CWB).

1 Introduction

Formal methods are mathematically based techniques for specifying and verifying complex hardware and software systems [3]. This paper emphasizes their application to parallel processing and distributed computing systems, where the main source of complexity is due to the co-existence of multiple, simultaneously active, and interacting agents. The specification and verification of shared-memory multiprocessor cache coherence protocols is a paradigmatic example of parallel technologies where formal methods can be applied. This kind of systems are composed of a set of elements which need to be coordinated by means of a reliable communication protocol.

In this paper we have chosen a cache coherence protocol as working example which will be developed through several stages of specification and verification. Cache coherence protocols range from simple *"snooping"* ones to complex *"directory-based"* frameworks [12]. In order to make a first approximation to the subject we will stick to one that belongs to the second group: the CC-NUMA protocol. Although its description is relatively simple it allows the definition of non-trivial properties. The verification of these properties illustrates the expressiveness and potentiality of the formalisms and how they could be applied to more complex examples.

In order to deal with a formal specification and verification of the cache coherence protocol, it is essential to use a mathematically-based technique or formal method. There are several formalisms that could be used to tackle this problem. Any protocol can be successfully described in terms of *processes* (con-

V. Malyshkin (Ed.): PaCT 2001, LNCS 2127, pp. 17–26, 2001.

current agents which interact in accordance with a predefined pattern of communication [4]) and consequently modelled by using a process-based formalism. Specifically, we have used a process algebra —CCS [9]— for the specification, and an associated temporal logic —μ-calculus [7,13]— for the coherence verification. Furthermore, for these formal methods an automated tool —'Edinburgh Concurrency Workbench[1]', CWB [1]— is available. Basically, it allows the definition, manipulation and verification of processes and temporal properties.

This paper has been structured in four sections following this first introduction. Section 2 deals with process-oriented specification and verification of protocols; a justification of the chosen formalisms can be found here. Section 3 describes the CC-NUMA cache coherence protocol. First of all, its specification in terms of communicating processes is presented; secondly, coherence restrictions are defined in terms of temporal properties which are then automatically verified with regard to the previous specification. A brief summary and conclusions, as well as future lines of research, are discussed in Section 4.

2 Specification and Verification of Protocols in CCS

Cache coherence protocols (in general, any communication protocol) can be described in terms of *'objects'* [10] which operate concurrently and interact in accordance with a predefined pattern of communication. This idea of communicating objects fits with the concept of *process* which allows the definition of an observable behaviour by means of all possible communications.

There are different process theories, most of them with a notion of behaviour pattern of objects [5] or machine for performing actions [4]. Moreover, some of them are based on a well-founded underlying formalism and so are suitable to be used to formally specify, manipulate and verify cache coherence protocols. From all existing formal process theories we have chosen the process algebra CCS ('Calculus of Communicating Systems' [9] and the μ-calculus [14,13] as a complementary temporal logic which allows the definition of properties to be verified in relation to the specified processes (other formal methods can lead to valid results as well). The main features are:

1. The CCS is a process theory which allows a formal manipulation of concurrent communicating processes.
2. There are higher-order extensions where dynamic structures communicating not only interaction channels but whole processes can be modelled.
3. The CCS is complemented by a temporal logic —the μ-calculus— which allows the definition of temporal properties. The process of verifying whether a certain process satisfies a property can be automated.
4. An automated tool —the CWB [1]— is available.

[1] See http://www.dcs.ed.ac.uk/home/cwb/index.html (29/10/1999)