CLASSIFICATION OF COMMUNICATION AND COOPERATION MECHANISMS FOR LOGICAL AND SYMBOLIC COMPUTATION SYSTEMS

Abstract. The combination of logical and symbolic computation systems has recently emerged from prototype extensions of stand-alone systems to the study of environments allowing interaction among several systems. Communication and cooperation mechanisms of systems performing any kind of mathematical service enable one to study and solve new classes of problems and to perform efficient computation by distributed specialized packages.

The classification of communication and cooperation methods for logical and symbolic computation systems given in this paper provides and surveys different methodologies for combining mathematical services and their characteristics, capabilities, requirements, and differences. The methods are illustrated by recent well-known examples.

We separate the classification into communication and cooperation methods. The former includes all aspects of the physical connection, the flow of mathematical information, the communication language(s) and its encoding, encryption, and knowledge sharing. The latter concerns the semantic aspects of architectures for cooperative problem solving.

1. Introduction

The design of general techniques to combine and integrate several systems has been initiated in many areas. For instance, the integration of theorem proving and symbolic mathematical computing has recently emerged from prototype extensions of single systems to the study of environments with interaction among distributed systems. However, there are no common languages, protocols, or standards for such interfaces.

Communication and cooperation mechanisms for logical and symbolic
computation systems enable to study and solve new classes of problems and to perform efficient computation through cooperating specialized packages. On the one hand, computer algebra systems (CAS) offer an extensive collection of efficient mathematical algorithms which could improve the efficiency of theorem proving systems (TPS). On the other hand, they ignore AI methods (e.g. theorem proving, planning of proofs and computations, machine learning) and their capabilities, e.g. verification of properties of mathematical objects using a TPS.

Basic architectures for performing communication among TPS and CAS are introduced in (Homann & Calmet, 1994). The classification given here is a result of generalizations and extensions of communication and cooperation mechanisms for software systems performing any kind of mathematical computation. We call such systems *mathematical services* (MS) which cover CAS and other symbolic computation packages, TPS, proof checkers and verification tools, numerical computation systems, visualization and type-setting applications, and format converters. This classification is illustrated by well-known recent examples of communication and cooperation mechanisms for both logical and symbolic computation systems. It provides and surveys different methodologies for combining such systems and their characteristics, capabilities, requirements, differences, and may guide the developments and selection of methods in this ongoing research. However, it must be pointed out that some of the presented architectures and communication methods are not specific to mathematical information and could be applied to combine other systems as well.

We separate the mechanisms into *communication* and *cooperation* methods. The former include all aspects of the physical connection, the flow of mathematical information, the communication language(s) and its encoding, encryption, and knowledge sharing. Communicating MS send and retrieve mathematical information and messages. The aspects of the “semantics” of these interactions are specified according to the level of cooperation among the distributed systems. Depending on their behaviour they can be classified into: master/slave, subpackage, black box, trust, extensible and exchangeable, consistency and closure.

As of today, there is no systematic investigation of the different possible methodologies to integrate heterogeneous mathematical systems. The goal of this paper is to fill this gap.

This paper is organized as follows. Section 2 gives an overview about architectures combining logical and symbolic computation systems. The advantages are illustrated by some recent well-known examples. The classification of such architectures based upon the features of the involved communication and cooperation methods is given in section 3 and section 4 respectively.