Distributed Modal Theorem Proving with KE

Jeremy Pitt\(^1\) and Jim Cunningham\(^2\)

\(^1\) Intelligent Communication Systems, Department of Electrical & Electronic Engineering, Imperial College of Science, Technology, and Medicine, Exhibition Road, London, SW7 2BT, UK. Email: j.pitt@ic.ac.uk URL: http://www.ics.ee.ic.ac.uk/~jvp/

\(^2\) Department of Computing, Imperial College of Science, Technology, and Medicine, Huxley Building, 180, Queen's Gate, London, SW7 2BZ, UK. Email: rjc@doc.ic.ac.uk URL: http://medlar.doc.ic.ac.uk/~rjc/

Abstract. This paper describes an approach to distributed modal theorem proving by bringing together and exploiting two software packages. The first is the implementation of a theorem prover for normal modal logics based on KE and a generalization of Fitting's prefixed tableaux. The second is a library for implementing brokered inter-process communication over the internet. We describe three demonstrators which combine these implementations and illustrate potential applications of the new technology, enabling theorem provers connected to the internet to cooperate, compete, or be used by third parties.

1 Introduction

This paper describes an approach to distributed modal theorem proving. This is based on two software implementations. The first is \(\Box\)KE, an extension of the first-order theorem prover leanKE \([23]\) to include modal rules for theorem proving in normal modal logics. The second is the CSF (Cooperation Services Framework) \([21]\), a library for providing and invoking services between autonomous software processes executing in a distributed open computing environment.

\(\Box\)KE is a sound and complete theorem prover for first-order classical logic, which is based on the calculus KE \([11]\) and free variable rules for eliminating quantifiers. It has been implemented in ICL/ECRC's ECL/PS Prolog \([12]\), and in \([23]\) its performance is evaluated on some standard problems and compared to the tableau theorem prover leanTAP \([3]\). As part of the CEC Medlar II project (Esprit 6471) we have investigated extending \(\Box\)KE to handle normal modal logics.

The CSF was first investigated in the CEC GOAL project (Esprit 6283), and was designed to provide distributed services to autonomous software processes running in a federated information system, based on normalized, brokered, interactions between them. It has been further developed in the Medlar II project, where it was proposed as a means to integrate and combine specialised inference engines \([22]\), and to exploit a common recent practice of making automated reasoning systems available through the WWW, e.g. SCAN \([19]\) and the Logics Workbench \([18]\).

The Logics Workbench is an integrated, interactive system which provides general reasoning facilities in a wide variety of propositional logics. Indeed, considerable research effort has been devoted to developing general proof systems for families of non-classical logics: see, for example, \([10]\) on substructural logics, \([14]\) on modal logics, and \([15]\) on "any" logic. We have investigated a generalization of Fitting's prefixed tableaux of \([14]\) for the family of normal modal logics, and have implemented the system \(\Box\)KE, an extension of
the propositional core of leanKE to include rules for the modal operators □ and ◊. For each normal modal logic in □KE, the form of the modal rules remains the same but the side-conditions on each rule (and on closure) can vary.

In this paper, we bring together the themes of distributed computing and modal theorem proving. We describe a series of demonstrative applications in which it is possible to launch separate □KE processes, each configured to handle a certain modal logic or logics. Using the CSF, these processes can then be coupled in such a way that they can cooperate or compete, can be sited centrally or distributedly, and can be accessed by other users or applications via the internet.

In section 2 we motivate and summarize our approach to prefixed KE for normal modal logics, give a specification of □KE, describe its implementation, and present some preliminary results. In section 3 we focus on the important features of the CSF for inter-process communication and cooperation. We then draw the two themes of the paper together in section 4, where we implement three demonstrators which illustrate the potential of □KE and the CSF for distributed modal theorem proving. Some final comments and concluding remarks are made in section 5.

2 Modal KE

2.1 Fitting's Prefixed Tableaux

In automated reasoning with modal logics, the idea of using a prefix, or label, to denote either a world and/or a path to a world, can be found in the works of for example [24, 6, 15], amongst others. The research reported here builds primarily on that of [14], but seeks to redress certain inefficiencies.

In Fitting's prefixed tableaux for modal logics [14], the standard tableau machinery is extended with names for possible worlds, which are prefixes, and which are used in such a way that the accessibility relation between possible worlds is reflected in the syntactic features of those names. If σ is a prefix and X is a formula, then by the prefixed formula σ : X we mean that X is true in the world named by σ. A prefix in [14] is simply a non-empty finite sequence of positive integers. This choice of representation allows syntactic recognition of whether one world is accessible from another or not, given the accessibility relation between worlds defined in a frame for a particular logic. For example, in K, the world named by the prefix (1,2,1,4) is accessible from (1,2,1). (In fact, in any normal modal logic, given any prefix σ and any integer n, the prefix formed by appending n to σ, σn, is always accessible from σ.)

Other conditions on accessibility between prefixes are given below, which is taken from [14], extended with condition 5:

1. the general condition if σi is accessible from σ for every i;
2. the reverse (or symmetric) condition if σ is accessible from σi for every i;
3. the identity (or reflexive) condition if σ is accessible from itself;
4. the transitive condition if τ is accessible from σ whenever σ is a proper initial segment of τ;
5. the euclidean condition if σiτ is accessible from σj for every i except j;
6. the universal condition if every prefix is accessible from any prefix.

The modal rules used to extend the usual tableau rules, for the logics that Fitting considered, can be stated as follows: