Verification of LOTOS Specifications using Term Rewriting Techniques

Carron Elizabeth Kirkwood

Submitted for the Degree of Doctor of Philosophy.

Research carried out in the Department of Computing Science,
University of Glasgow.

Abstract

Recently the use of formal methods in describing and analysing the behaviour of (computer) systems has become more common. This has resulted in the proliferation of a wide variety of different specification formalisms, together with analytical techniques and methodologies for specification development. The particular specification formalism adopted for this study is LOTOS, an ISO standard formal description technique. Although there are many works dealing with how to write LOTOS specifications and how to develop a LOTOS specification from the initial abstract requirements specification to concrete implementation, relatively few works are concerned with the problems of expressing and proving the correctness of LOTOS specifications, i.e. *verification*. The main objective of this thesis is to address this shortfall by investigating the meaning of verification as it relates to concurrent systems in general, and in particular to those systems described using LOTOS. Further goals are to automate the verification process using equational reasoning and term rewriting, and also to attempt to make the results of this work, both theoretical and practical, as accessible to LOTOS practitioners as possible.

After introducing the LOTOS language and related formalisms, the thesis continues with a survey of approaches to verification of concurrent systems with a view to identifying those approaches suitable for use in verification of properties of systems specified using LOTOS. Both general methodology and specific implementation techniques are considered. As a result of this survey, two useful approaches are identified. Both are based on the technique of expressing the correctness of a LOTOS specification by comparison with another, typically more abstract, specification. The second approach, covered later in the thesis, uses logic for the more abstract specification. The main part of the thesis is concerned with the first approach, in which both specifications are described in LOTOS, and the comparison is expressed by a behavioural equivalence or preorder relation. This approach is further explored by means of proofs based on the paradigm of equational reasoning, implemented by term rewriting.

Initially, only *Basic* LOTOS (i.e. the process algebra) is considered. A complete (i.e. confluent and terminating) rule set for weak bisimulation congruence over a subset of Basic LOTOS is developed using RRL (Rewrite Rule Laboratory). Although fully automatic, this proof technique is found to be insufficient for anything other than finite toy examples. In order to give more power, the rule set is supplemented by an incomplete set of rules expressing the expansion law. The incompleteness of the rule set necessitates the use of a *strategy* in applying the rules, as indiscriminate application of the rules may lead to non-termination of the rewriting. A case study illustrates the use of these rules, and also the effect of different interpretations of the verification requirement on the outcome of the proof.

This proof technique, as a result of the deficiencies of the tool on which it is based, has two major failings: an inability to handle recursion, and no opportunity for user control in the proof. Moving to a different tool, PAM (Process Algebra Manipulator), allows correction of these faults, but at the cost of automation. The new implementation acts merely as computerised pencil and paper, although tactics can be defined which allow some degree of automation. Equations may be applied in either direction, therefore completion is no longer as important. (Note that the tactic language could be used to describe a a complete set of rules which would give an automatic proof technique, therefore some effort towards completion is still desirable. However, since LOTOS weak bisimulation congruence is undecidable, there can never be a complete rule set for deciding equivalence of terms from the full LOTOS language.) The composition of the rule set is reconsidered, with a view to using alternative axiomatisations of weak bisimulation congruence: two main axiomatisations are described and their relative merits compared. The axiomatisation of other LOTOS relations is also considered. In particular, we consider the pitfalls of axiomatising the **cred** preorder relation.

In order to demonstrate the use of the PAM proof system developed, the case study, modified to use recursion, is re-examined. Four other examples taken from the literature, one substantial, the others fairly small, are also investigated to further demonstrate the applicability of the PAM proof system to a variety of examples.

The above approach considers Basic LOTOS only; to be more generally applicable the verification of properties of *full* LOTOS specifications (i.e. including abstract data types) must also be studied. Methods for proving the equivalence of full LOTOS specifications are examined, including a modification of the technique used successfully above. The application of this technique is illustrated via proofs of the equivalence of three variants of the well-known *stack* example. The proofs are carried out by hand as neither of the implementation tools used above are able to handle data types. The approaches of other authors to verification of full LOTOS specifications are also described and illustrated by examples in order to propose an approach to verification comprising several complementary techniques.

Finally, the verification of LOTOS specifications where the abstract requirements are expressed using temporal/modal logic is briefly considered. Specific reference is made to the existing linear temporal logic used in conjunction with LOTOS and also to the use of HML (Hennessy-Milner Logic) in conjunction with CCS. The possibility of using HML with Basic LOTOS is discussed at length, with examples drawn from earlier in the thesis. Also considered is the possibility of extending the logic for use with full LOTOS. Both of these proposals require further investigation.

Contents

Chapter 1

Introduction

1.1 Background

The last few years have seen an increasing interest in the use of *formal methods* in the design and analysis of computer systems. By formal methods, we mean the application of mathematical concepts to the modelling of a real world problem. Most commonly this means the use of a formal specification language, i.e. one which has a formal mathematical semantics, to describe the system. The use of formal methods allows us to build a (simplified) mathematical model of a real world phenomenon which can then be analysed using mathematical techniques. The results gained can then be used to deduce results about the properties of the real world system. Of course, the applicability of these results to the real world system is highly dependent on the accuracy of our mathematical model.

Use of formal methods can aid practitioners in two main ways. Firstly, by virtue of their mathematical basis, formal specifications allow clear, precise and unambiguous descriptions of a system; moreover, descriptions can be written without reference to implementation issues. Secondly, the existence of formal semantics for the specification language makes rigorous mathematical analysis, i.e. *verification*, of the specification possible. Such analysis can improve confidence in the correctness of the design and/or implementation. It may also lead to a better/deeper understanding of the system, especially in cases where the analysis detects some (non-trivial) error in the specification.

A variety of different specification formalisms are in current use in many different areas of application, e.g. VDM, Z, denotational semantics, petri nets, finite state automata, process algebras, ... Although the choice seems endless, it may be narrowed down by the requirements of the area of application. We are interested in the verification (and therefore the specification) of *concurrent systems*. For this application area, a popular means of specification is *process algebra*. Prominent examples of process algebras include ACP [BK84], CCS [Mil80, Mil89b], CIRCAL [Mil85], CSP [Hoa85], SCCS (Synchronous CCS) [Mil89b], and MEIJE [AB84]. They are prominent because of *popularity*, i.e. used in many applications and studies, both academic and industrial, or *theoretical significance*, e.g. it has been shown in [dS85] that SCCS and MEIJE are *universal process algebras*, in the sense that all other process algebras can be described in terms of SCCS or MEIJE.

It should be realised that universality is not necessarily a desirable feature of a language. While a universal process algebra is able to describe every sort of system, it may be that these descriptions are clumsy. More importantly, because it is so general, the theory for verification may be less rich than if we used a language which has a smaller scope and is perhaps specifically designed for our purpose. For this reason neither of the universal process algebras mentioned above are considered further here. The matter of why variety is important in process algebras is further discussed in [BBH+91].

Another consideration not mentioned above is *standardisation*. Although the formalisms above are of *academic* significance, they have not been widely adopted for industrial use. For internationally standardised specification formalisms we must look to ISO (the International Standards Organisation).

One of the tasks undertaken by ISO is the development of *open systems* which will provide a uniform framework for communication throughout the world. This means that the protocols controlling communication must behave in exactly the same way whenever and wherever they are executed, regardless of local variables such as implementation language, machine architecture and physical location.

The focal point of the standardisation process is the seven layer Reference Model [ISO74] which describes the interaction between systems in an abstract, implementation independent way. Obviously, given the aims of ISO, the first essential is to have a specification which cannot be interpreted in any way other than that intended by the specifier, implying the use of a formal specification language. With this in mind, ISO have introduced three internationally standard formal description techniques (fdts) to support the Reference Model: Estelle [ISO90], SDL [CCI88] and LOTOS [ISO88]. These are to be used to give formal specifications of the services and protocols which make up the Reference Model. [Vis90] gives a historical overview of the development of these languages and related standards, and [Tur93] gives an introduction to, and comparison of, the languages.

Support for the formal description techniques is provided in the form of European research and development projects. In the past these projects have resulted in development methodologies and analytical tools for the fdts. Initially the projects were joint ventures, encompassing all three fdts, for example, the SEDOS project [vEVD89]. More recently, as interest in the different languages has grown, the projects have been specific to particular fdts, e.g. LOTOSPHERE [vSPV92]. Following this lead, we also concentrate on just one of the fdts, LOTOS, for the following reasons.

The LOTOS language has two parts: process algebra, which can be used to specify the control aspect of a system, and abstract data type, which can be used to specify the data manipulated by the system. The process algebra part was developed from the formalisms CCS and CSP; this allows us to link our study to both academic and industrial concerns. The theory behind CCS and CSP is now well developed, and there is a wide literature on many aspects of analysis of CCS and CSP specifications. Due to the close relationship between these formalisms and LOTOS it may be possible to transfer results obtained for these formalisms to the LOTOS setting. Our main reason for choosing to study LOTOS rather than CCS or CSP is its status as an international standard. We cannot investigate all aspects of the definition and application of LOTOS in one thesis, therefore we concentrate on *verification* of properties of LOTOS specifications.

The fdts of ISO were all designed with specification in mind; therefore the expressivity of the language was given more importance than the simplicity of the semantics. While specification in itself is a valuable undertaking, providing clear descriptions and perhaps a better understanding of the system, ideally, from our point of view, more emphasis should be put on methods for checking the correctness of a specification. The more expressive a language is, the more complex the semantics, and the harder it becomes to verify correctness of specifications in that language. Verification really needs to be considered right from the first stages of specification, rather than tackled after specification is completed; this point is made in [HJOP89]. This conflict between the expressive power of the language and the simplicity of its semantics is particularly relevant to LOTOS, which is encumbered by a verbose syntax, making analyses longer and more tedious, and also obscuring the simplicity of the underlying system.

Several forms of analysis can be considered for LOTOS, including debugging, simulation, testing and verification. While projects such as LOTOSPHERE have been successful in providing a structured development methodology together with some tool support, other aspects of the development process have been largely ignored. In particular, although much effort has been expended on, for example, testing techniques, relatively little has been directed towards the problems of verification, particularly verification of properties of *full* LOTOS specifications. A desire to rectify this imbalance is the main aim of this thesis; more specific aims and objectives are detailed in the next section. We are interested in verification because it provides rigorous mathematical analysis of a system. While verification cannot provide 100% certainty that a system is correct, it can greatly increase our confidence in the general correctness of the system, and the correctness with respect to particular properties (assuming the properties are expressed accurately). Verification can give more confidence in the system than a testing method can.

Another aspect of verification is *automation*. Even for small systems, analysis can be tedious and error prone, therefore it is vital to have some sort of machine assistance. Although full automation of any analyses is preferable, in practice we may have to settle for partial automation.

A wide variety of methods for automation exist for different aspects of verification. We consider a general method of proof and automation, the paradigm of equational reasoning implemented by term rewriting, and consider how two forms of analysis may be carried out by equational reasoning and automated by term rewriting. Process algebras are all associated with sets of laws or axioms corresponding to different notions of equivalence, making equational reasoning a natural proof technique to use.

The next section details our aims and objectives in the study of verification of properties of LOTOS specifications.

1.2 Aims and Objectives

The aims of this thesis are:

- to investigate the verification of concurrent systems described using the formal description technique LOTOS,
- to apply this knowledge by developing proof methods for verifying the correctness of LOTOS specifications. These methods should make use of existing proof tools rather than necessitating the implementation of new software. The proof technique used will be equational reasoning, automated by term rewriting.
- to make the results (both theoretical and practical) of this investigation accessible to LOTOS practitioners.

These aims will be achieved by the following objectives:

- to build knowledge of LOTOS, the LOTOS related process algebras CCS and CSP, and their associated approaches to verification,
- to use this knowledge in defining what verification means for concurrent systems in general, and more specifically for both Basic LOTOS *and* full LOTOS,
- from the above study, to identify an approach to verification which can be automated using equational reasoning and term rewriting,
- to develop a proof system (based on existing tools) implementing that approach to verification,
- to demonstrate the usefulness of the proof system through examples.

We believe these aims and objectives have been largely achieved in the thesis. Although we do not claim to have completely investigated verification for LOTOS (we have only briefly covered some aspects of verification, such as the use of temporal/modal logics in specifying the requirements of the system[1]), we have thoroughly explored one aspect of verification, namely that of proving two specifications (both described using LOTOS) are related by a behavioural equivalence or preorder. This has been achieved through theoretical and practical investigations, the practical work being carried out using equational reasoning and term rewriting tools. The proof technique we develop in the thesis is illustrated through several examples, some small, others medium sized.

The following section gives a more detailed account of the work carried out.

1.3 Overview

The thesis is organised around the main topic of verification; particular questions addressed are: what is meant by verification, what kind of verification can be carried out on LOTOS specifications, how can the proofs of verification can be automated, and what do those results tell us about the system under examination?

Chapter 2 contains an informal discussion of what is meant by *verification*. The view taken here is that verification is the formal, mathematical expression and proof of the correctness of a concrete description of a system with respect to some set of (formal) requirements. Of course, the requirements also constitute a description of the system, at a more abstract level. As the two descriptions need not be expressed using the same formalism, two main cases are considered; the bulk of the thesis is concerned with the case in which both descriptions are expressed using LOTOS. An alternative case in which the concrete description is expressed in LOTOS and the requirements are expressed using a temporal or modal logic is considered in chapter 11.

What does verification mean in relation to systems specified using LOTOS? In order to consider this question a good working knowledge of the semantics of LOTOS and the ways in which LOTOS specifications can be compared is required. It is also helpful to have the same knowledge for CCS and CSP, since LOTOS was developed from these formalisms. Since both CCS and CSP have a rich literature, it may be possible to adapt proof techniques from either of these formalisms for use with LOTOS. The three formalisms, LOTOS, CCS and CSP, are presented in some detail in chapter 3, including a summary of their main differences (and similarities). At this point we consider only Basic LOTOS, which aids the comparison with CCS and CSP. Full LOTOS is considered in chapter 10.

The survey of the three process algebras is followed by the first detailed section on verification,

[1]The investigation of temporal logic in conjunction with LOTOS is the basis of a SERC funded project, "Temporal Aspects of Verification of LOTOS Specifications", which will run over the next two years.

chapter 4. The technique of comparing two descriptions of a system where both are written using the same specification language is one which is commonly used in the process algebra literature. Many proof techniques and tools based on this approach are currently in use, and a wide variety of equivalence relations and/or preorders have been developed to express ways of comparing specifications. A side issue considered briefly here is the selection of the most appropriate relation for a given problem.

The available proof techniques and tools are surveyed, compared and considered for use in conjunction with LOTOS. Although there are some fast algorithms for deciding equivalence of processes (based on graph partition algorithms), in general these do not give any intuition in the case where the two specifications are not equivalent (they only answer yes or no). Such algorithms also rely on a special internal representation of the process specifications for their calculations. In this work our preference is for a proof technique in which no special intermediate forms are required, and which may give some insight into the workings of the system under consideration, especially in the case in which the two specifications are not equivalent. Equational reasoning is such a proof technique and is adopted for use in the practical work. Equational reasoning is automated by term rewriting. The basic theory of term rewriting, including Knuth-Bendix completion, is presented in chapter 5. We also discuss the discovery of an inconsistency, which we found by rewriting techniques, in the laws of weak bisimulation congruence of [ISO88].

Chapters 6 to 9 detail the various components of the practical work. The initial aim of the practical work is to form a complete (i.e. confluent and terminating) set of rewrite rules (giving a decision procedure) for LOTOS weak bisimulation congruence using the tool RRL (Rewrite Rule Laboratory). This is described in chapter 6. A complete rule set for a *subset* of the language is developed. No complete set for the full language can exist because weak bisimulation congruence is known to be undecidable. Several small examples of proofs by rewriting demonstrate the use of the rewrite rule set and also the need for rules expressing the full power of the expansion law (which allows parallelism to be expressed in terms of sequencing and choice). A set of rules to achieve this is developed. As the new set of rules is not complete, a strategy in applying the rules must be adopted, otherwise the rewriting may not terminate.

To illustrate the use of this verification technique, a case study is introduced in chapter 7. The case study has two purposes: firstly, to obtain a successful proof, using the rewrite rules developed with RRL, of the requirement that the specification of the system is satisfied by the implementation, and secondly, to discover the effect of different interpretations of this requirement on the outcome of the proof. It is interesting to note that under the initial, intuitive interpretation of the verification requirement the implementation cannot be proved to satisfy the specification. In the end the specification has to be altered (in a modular way, using the constraint oriented specification style) in order to complete the proof. As a result of this study a number of deficiencies in the original verification technique are identified; the most important is the inability to handle recursive processes, but also significant is the inflexibility of the RRL system and lack of opportunity for user intervention.

Chapter 8 introduces our second approach to using term rewriting proof techniques for verification of Basic LOTOS specifications. A different tool, PAM (Process Algebra Manipulator), which *can* perform proofs on specifications incorporating recursive processes, is adopted. This new power is balanced by the fact that PAM cannot perform proofs automatically; the user must guide every step. However, a number of tactics describing patterns of rule application may be defined, allowing some limited form of automation.

An important decision in setting up PAM is how to express the LOTOS laws, and indeed whether all laws are necessary for most examples. The relative merits of different solutions to this question are considered. This leads on to implementation of equivalences other than weak bisimulation congruence. Also discussed is the problem of axiomatising a preorder relation.

In chapter 9 the case study example is repeated (this time with recursive processes). In order to further show the utility of the system, a number of other examples are also presented. These are a simple radiation machine, the reader/writer problem, a nondeterministic candy machine and the scheduler. Of these, the most significant, and largest, is the radiation machine study. Proofs are presented of the safety (or not) of several variants of the machine; the most interesting are the proofs of safety. These could not be completed using PAM, as reasoning external to our proof system had to be employed. All of the examples are taken from the papers of other authors.

Until this point only one half of the LOTOS language has been considered; namely the process algebra part, Basic LOTOS. However, full LOTOS also incorporates an abstract data type language, ACT ONE. In chapter 10 the proof technique used so far is reviewed and the question of how the inclusion of data types might alter the verification process is considered. The modified proof technique is illustrated by means of an example. Three descriptions of the stack, each with varying emphasis on the process algebra, are compared using weak bisimulation congruence; the proofs are carried out by hand. Hand proofs are normally tedious and error prone: these are no exception. This "feature" is only exaggerated by including data types. The technique seems of limited value without automation (which is not possible due to the limitations of current tools) so approaches by other authors to the problem of verification of full LOTOS are also surveyed. The two main approaches we consider both work on the principle of removing the abstract data types from the specification to obtain a Basic LOTOS specification, and evaluating correctness using the better understood Basic LOTOS proof techniques. The first approach provides a method for encoding the data values of a full LOTOS specification in a Basic LOTOS one, and may be varied to preserve some, all, or none of the data type information. The other approach is really a method for deriving a process algebra specification from an abstract data type specification, preserving all data type information in the derivation. We illustrate both approaches by our own examples: one using the Stack example and a hand proof, the other using the radiation machine study of section 9.2 and the PAM implementation to automate the Basic LOTOS proof. In the absence of one really useful and generally applicable proof technique for verification of properties of full LOTOS specifications, it seems that a composite approach may be the best solution.

In the initial discussion of verification in chapter 2, two main approaches to verification were identified. Although the main approach of proving equivalence between LOTOS specifications was demonstrated in chapter 9 to be fairly successful for Basic LOTOS specifications, chapter 10 showed that it is not as suitable for full LOTOS specifications. The second approach to verification mentioned in chapter 2 considers the situation in which one description (usually the more abstract specification) is written using some form of logic. This allows the desirable properties of the system to be described in a more abstract, less constructive manner. The current state of verification with respect to this approach is surveyed in chapter 11. Although a linear temporal logic has been developed for use in conjunction with LOTOS, it is not satisfactory as the equivalence induced by the logic is the rather weak trace equivalence, meaning that deadlock properties are not preserved. We conjecture that a variant of HML (the logic commonly used with CCS) might be adapted for use with Basic LOTOS; we present the logic, outline the proof technique and give some re-specification, in logic, of earlier examples. A natural progression is to consider what sort of logic would be required for use with full LOTOS; we discuss this topic, illustrating the discussion by examples, but the possibility is not pursued. This work will be the subject of a future investigation, as mentioned earlier.

Finally, chapter 12 concludes our study with a discussion of what has been achieved, how far our work has gone towards meeting the original objectives, open problems and further work.

Four appendices are attached: appendix A consists of a survey of existing tools for verification of specifications written using process algebras and tools for LOTOS (proof tools and otherwise), appendix B presents the LOTOS syntax and semantics, and appendices C and D give the input files used in RRL and PAM respectively.

Acknowledgements

My interest in the topic of verification of LOTOS specifications began when I was employed on the SERC project “Verification Techniques for LOTOS” [VTL93]. I want to thank my supervisor Muffy Thomas for getting me involved in all of this, providing the stimulus for much the work described herein and for patiently proof-reading countless drafts of bits of this thesis.

During the course of the project I had many illuminating conversations with colleagues involved in the project from Royal Holloway and Bedford New College and Rutherford Appleton Laboratory. Amongst my colleagues, I particularly want to thank Phil Watson, who shared an office with me for three years, during which time he helped guide me through the hard maths! I also want to thank visitors to our project meetings: Jeremy Dick, who provided the case study example described in chapter 7, and friendly rivals in the field, Paula Inverardi and Monica Nesi.

The Department of Computing Science has been a great environment in which to work. I'd like to thank all my colleagues there, in particular, the members of the Formal Methods group, who probably never want to hear me talk about the case study example ever again. The University of Glasgow also played a part by funding me as a University Scholar from October 1992 to September 1993, allowing me to write this thesis.

The tools used in this work came from the following people: Deepak Kapur (RRL) and Huimin Lin (PAM). Thank you particularly to Huimin, who was always ready to answer my questions about PAM.

For moral support, I want to thank all my family; the Shanklands and the Kirkwoods. Last but by no means least, thanks to my best friend Derick, who supplied endless cups of tea, never stopped believing in me, and generally put up with my ranting and raving throughout the process.

Chapter 12

Conclusions

In this thesis we have introduced the topic of verification of properties of concurrent systems, in particular those described using LOTOS, in a manner suitable for those with no prior knowledge of the subject. We followed this with a thorough, practically-based investigation of verification of properties of LOTOS specifications expressed using comparison of two LOTOS specifications by a behavioural relation and equational reasoning.

We developed a partially automated proof technique based on equational reasoning, and used this, together with hand proofs where necessary, to study verification via particular examples. This allowed us to develop a greater understanding of the verification process and also demonstrated the utility of the proof system developed.

The main outcome of our work on equational reasoning and verification of properties of LOTOS specifications is that equational reasoning is highly suitable for carrying out *equivalence* proofs, but that the method begins to break down when partial specifications are considered; we are forced to write clumsy specifications, and were unable to (soundly) automate the proof process. This implies that a different proof paradigm should be adopted when considering *ordering* of specifications.

We investigated one method of dealing with partial specifications: the use of temporal or modal logic for specifications. We do not abandon LOTOS; a LOTOS expression may be used as the model in which we evaluate the validity of the logical specification. We made a preliminary study of the advantages and disadvantages of this approach, illustrating the use of logic for specification by examples drawn from the earlier part of the thesis. We showed how some of the examples for which the equational approach had been unsatisfactory are better treated using logic.

12.1 Detailed List of Achievements

The achievements of the thesis may be considered in four main groups. We began by introducing and surveying the field. This survey gives the necessary background for the main investigation of verification of properties of LOTOS specifications; the investigation had both theoretical and practical elements. During the practical work we made some contributions to the use of term rewriting for automation of process algebra proofs. We concluded by studying the use of logic with LOTOS.

Following this grouping, we list these achievements in more detail.

- We began by providing an introduction to verification of concurrent systems, process algebra, LOTOS, equational reasoning and logic which may be used as a springboard for other researchers entering the field. This makes the thesis self-contained by providing the background necessary for the main investigation of verification of properties of LOTOS specifications.
  - We surveyed the topic of verification of properties of LOTOS specifications. The introductory work comes in chapter 2, where we discuss possible interpretations of the term "verification", and chapter 4, where one particular approach to verification is described.
  - In chapter 3 we presented aspects of the three process algebras CCS, CSP and Basic LOTOS, including equivalence relations and proof techniques. The work is not new, but the presentation of the three together in a comparative manner is.
  - We presented those aspects of equational reasoning relevant to our work with LOTOS, namely proof by rewriting and Knuth-Bendix completion, in chapter 5.
  - In chapter 11 we presented the logics HML and modal mu-calculus.
  - As part of our survey of verification we also surveyed currently available proof tools which might be used with LOTOS. This is mentioned in chapter 4; the survey is given in more detail in appendix A.
  - The syntax and semantics of LOTOS is presented in appendix B. We found the presentation of the same information in the standard [ISO88] rather complex and poorly organised. Our intention was to provide a clearer presentation for ourselves (and we believe this has been achieved); others may also find our presentation easier to follow.
- The bulk of our work was related to the verification of properties of LOTOS specifications where the verification requirement is expressed by a behavioural equivalence and the proof carried out using equational reasoning.
  - We have surveyed and discussed the topic of verification of properties of LOTOS specifications, including two areas which have been largely ignored in the literature, namely

verification of properties of full LOTOS specifications and also the use of logic in specifying the requirements of a system; see chapters 2, 4, 10 and 11.

- As part of the study of choices the user is faced with in the verification process we identified several possible criteria, given in section 4.2.2, which might help differentiate between the various equivalences/preorders.
- We have made a thorough investigation of one aspect of the verification topic, namely the method of comparing two LOTOS specifications in terms of a behavioural equivalence relation. The theoretical part of the study was carried out first for a portion of Basic LOTOS in chapter 4, extended to the complete language of Basic LOTOS, including recursion, in chapter 8, and finally extended to full LOTOS in chapter 10.
- The above approach to verification of properties of Basic LOTOS specifications has been implemented in a term rewriting framework. The initial implementation described in chapter 6 deals only with a subset of the language, but the system has evolved to include all features of Basic LOTOS. The system finally obtained is described in chapter 8. The ease with which it was possible to adapt and develop the system is a consequence of choosing the equational reasoning paradigm. This development also required changing the underlying equational reasoning tool. Note that the relabelling operator of LOTOS is slightly simplified in our implementation.
- The utility of the above proof system has been demonstrated via a number of examples, presented in chapter 9. We deliberately chose examples which had been presented by other authors using different proof systems as a means of avoiding unintentional bias towards examples suited to our proof system.
- When investigating verification of properties of full LOTOS specifications we considered the approaches of other authors to the problem in addition to considering how the above equational reasoning approach could be modified. In particular we studied the transformations from full LOTOS to Basic LOTOS detailed in [Bol92]. We investigated the properties of these transformations with respect to their use in verification of full LOTOS specifications, concluding that the results of verification carried out on the transformed specifications can only be extrapolated to the original full LOTOS specification for one of the transformations. The other transformation preserves very weak properties only. This was not considered in the original presentation of [Bol92]; our contribution is detailed in section 10.3.1.
- An important part of verification is specification; if the possible approaches to verification are borne in mind when specifying a system, the verification may be easier. We also contributed to research on specification in LOTOS.

The LOTOS language was developed for use in specification of communications and this is the area in which it is usually applied. We successfully used LOTOS for non-communications examples in chapter 7 and in chapter 9, showing that the language is applicable outside the originally conceived area of application.

We reviewed the language LOTOS in section 7.6. Of special relevance are the observations that some features of the LOTOS language make verification more difficult, in particular the disable operator, discussed in section 9.2.5, and also the hide operator, discussed in section 7.4.1.

- During the practical investigation of verification we used equational reasoning and term rewriting for automation; this resulted in the following contributions.
  - Early experiments in using a rewriting tool for proofs of equivalence were centered around an attempt to find a confluent and terminating set of rewrite rules for the LOTOS weak bisimulation congruence relation for a subset of the language; this is described in chapter 6. This experiment was successful in that such a rule set was developed, but unsuccessful in that the rule set did not have sufficient power for any but the simplest proofs.

    A complete rule set corresponding to the equivalence of the semantics is impossible to obtain because weak bisimulation is undecidable, therefore we also discussed the relative merits of different choices of rules and how they might affect the verification process, see section 8.3. In particular, we discussed the effects an incomplete set of rules might have on the verification process, and how that might necessitate the introduction of a strategy in applying the rules, see sections 6.4.2 and 8.3.
  - The above completion work had two side effects. The first was the generation of several diverging sequences of rules (useful for work detailed in [Wat92]), see section 6.4.3. The second was to show that the laws of weak bisimulation congruence given in [ISO88] are not sound (although this is easily corrected), see section 5.5.2.
  - The complete set of rules developed above was not powerful enough for any but the simplest examples, a fact easily ascertained by experiment. Our initial solution was to develop a set of rewrite rules which reduce a term according to the expansion law for parallelism; see section 6.4.1. (Note that the final implementation does not use these rules because this facility is built into PAM).
  - Although obtaining a set of rewrite rules for an equivalence relation is just a matter of orienting the axioms or laws, the process is not so simple for a preorder relation. In section 8.4 we presented two possible rule sets for the **cred** preorder, together with analysis of the effects of using these rewrite rules in proofs.

– Throughout the study various equational reasoning tools were used. The principle of equational reasoning is a simple and familiar one, which makes proof in this paradigm straightforward. Equational reasoning tools, on the other hand, are hard to use on the whole. This is due more to the status of these tools as research tools rather than a pieces of software engineered for industry; see particularly our remarks about RRL in section 7.6. However, we note that PAM, reviewed in appendix A, is also an equational reasoning tool and yet is easy to use. Perhaps its simple graphical interface shows the way of the future for such tools.

- As a result of the shortcomings of the proof system developed above, we identified a need for an alternative approach to specification and verification. We studied the use of logic in specifying the requirements of a system, with a LOTOS specification being used as the model in which those requirements are evaluated.

  – In chapter 11 we presented HML and the modal mu-calculus and proposed that, although defined for use with CCS, they could also be used in conjunction with LOTOS since both are based on the model of labelled transition systems. We outlined a suitable proof technique and gave several examples of the use of the modal mu-calculus in expressing properties of Basic LOTOS specifications. In particular we considered examples from earlier in the thesis for which the equational approach had been unsuitable.

  – The use of the modal mu-calculus for Basic LOTOS seems straightforward, but the modification of the logic for use with full LOTOS may be more difficult. We discussed possible extensions to the modal mu-calculus and proof system required for use with full LOTOS, illustrating those requirements by means of selected examples; see section 11.6.

Our original aims have been only partially met in that we have only thoroughly researched one particular approach to verification of LOTOS specifications. Specifically, we have not fully considered verification of LOTOS specifications with respect to logical requirements specifications (but see further work below). We believe that with respect to verification of equivalence/ordering of two LOTOS specifications we have achieved our original goals, including the goal to present the work as simply and clearly as possible, making our work easily understood by a newcomer to the subject, although this is of course a rather subjective evaluation.

12.2 Further Work

There are two kinds of work discussed here: work which follows on naturally from the work of the thesis, and work which is indirectly related to the main body of the thesis work.

12.2.1 Work Directly Related to the Thesis

This category contains four main topics: development of the PAM proof system, further case studies, further investigation of verification for full LOTOS, in particular methods of automating proofs, and use of modal or temporal logic for full LOTOS.

Developing the PAM System There are several ways in which the proof system could be further developed. The most obvious one is to add axiomatisations for other relations. This could also mean finding a better axiomatisation for the **cred** preorder, although really we cannot properly express preorders in the equational framework.

We remarked in chapter 10 that a new version of PAM which can handle parameterised processes is under development. The addition of parameters to processes, both gate parameters and data type parameters, is important if our system is to be used for real LOTOS verification, therefore we anticipate modifying our approach to utilise these new features.

A further development, which also depends on development of PAM, is implementation of the LOTOS relabelling operator as described in [ISO88], rather than the current simplified version.

More Examples It is clear that we have only attempted fairly small examples in the studies of chapters 7 and 9; although we note that our system was easy to use, and, for most of these examples, proofs were completed quickly. An important question is: can our method be scaled up to deal with larger examples? The easiest way to answer this question is to attempt verification proofs involving larger, more complex specifications.

Development of Proof Techniques for full LOTOS Verification Although we were able to carry out some verification of properties of full LOTOS specifications, these results were not satisfactory; the (hand) proofs were complex and tedious. While some of the difficulty lies in the lack of automated tools, there has also been very little research on verification techniques for full LOTOS. In particular, the method of constructing a bisimulation in the stack proof of section 10.2.4 is very tedious; it would be useful to find a better way, which could be easily automated, to prove two full LOTOS specifications equivalent.

Developing a Modal/Temporal Logic for full LOTOS While the use of the modal mu-calculus for Basic LOTOS is perfectly valid, since both are based on labelled transition systems, it seems much harder to generalise the modal mu-calculus for use with full LOTOS. As seen in chapter 11, the problems lie more in the development of a proof system; we can already formulate properties which use data. This work will be continued in the SERC-funded project "Temporal Aspects of Verification of LOTOS Specifications".

12.2.2 Work Indirectly Related to the Thesis

The remainder of these topics for further work are ones which were encountered during our investigations but which are somewhat tangential to the main body of the work.

Criteria for Choosing a Relation One of the most difficult parts of verification lies in interpreting the verification requirements; a particular aspect of this is choosing which of the many equivalence and preorder relations defined for process algebras is most appropriate for a given example. In section 4.2.2 we postulated a number of possible criteria which might be used in making this choice; these remain to be more thoroughly investigated.

Specification Styles To what extent does the style of specification affect the verification? It is clear that some specifications exclude certain methods; see, for example, the third stack and equivalence proofs of section 10.2.6. A possible direction for future work lies in determining whether such problems can be classified, identified in advance, and avoided.

Nondeterminism In the Login case study example of chapter 7 we originally anticipated that the conjecture expressing correctness and hence the proof could be split into three parts due to the disjoint nature of the protocols. In attempting the subproofs we discovered that the use of the **hide** operator led to extra nondeterminism and the proof could not proceed. We could not prove anything about the correctness of the parts of the conjecture, and therefore nothing could be deduced about the correctness of the system as a whole. However, the technique of divide and conquer as a method of simplifying problems is both commonly used and valuable. Since the use of **hide** causes problems by introducing internal events, the problem may be that we have not as yet found the right method of splitting the conjecture up, or of expressing the correctness of the parts. As we have seen in chapter 11, one approach to this problem involves the use of logic. If we want to remain within process algebra a solution could lie in *relativized bisimulation* [LM92], where bisimulation is measured with respect to an environment which expresses "allowed" actions.

We conclude with a discussion of the possible impact of our work on academics in this field and also on the wider LOTOS community.

12.3 Prospects for this Work

The thesis as a whole may be useful to other researchers getting started in the area of verification of properties of concurrent systems; we provide an introduction to the main topics of this area, a thorough study of the applicability of equational reasoning techniques to such verification, and also preliminary investigations of the use of temporal or modal logic for use with LOTOS. It is important to point out that the area of verification has been largely ignored by the LOTOS community in favour of validation methods such as testing and simulation, therefore few, if any, large scale works on verification of properties of LOTOS specifications, such as our own, exist.

Given that LOTOS is an ISO standard and therefore used by industry, particularly the telecommunications industry, we must also consider the impact of our work on the wider LOTOS community, i.e. outside academia.

Through PAM, our system provides an environment in which to reason about Basic LOTOS which is easy to use and also to extend (only a knowledge of LOTOS is required; the form of the PAM files is straightforward and requires no special coding ability). However, the quality and robustness of tools demanded by industrial practitioners is much higher than we have yet attained; our tool is still in the early stages of development. We require to carry out further tests, particularly on larger examples. In addition, our proof system relies to a large extent on the skill of the user in guiding the proof process, which requires a significant investment in terms of time both in preliminary study and in the proof process.

As long as tool support for LOTOS continues to be concentrated in the areas of simulation, testing and translation, there is little future for verification in the wider LOTOS community. Although verification can give us greater confidence in the correctness of our systems, perhaps, relative to the amount of work required to gain that confidence, the gain is not as great as can be achieved through use of testing and simulation tools which require less effort on the part of the user to obtain results. Nevertheless, we hope that our work will help lead to a greater understanding of verification and the development of better tools and techniques for verification in the future.

Bibliography

[AB84] D. Austry and G. Boudol. Algèbre de processus et synchronisation. *Theoretical Computer Science*, 30:90–131, 1984.

[AB90] G.J. Akkerman and J.C.M. Baeten. Term Rewriting Analysis in Process Algebra. Technical Report P9006, University of Amsterdam, 1990.

[Abr87] S. Abramsky. Observation Equivalence as a Testing Equivalence. *Theoretical Computer Science*, 53:225–241, 1987.

[Aju89] I. Ajubi. Formal Description of the OSI Session Layer: Session Protocol. In P.H.J. van Eijk, C.A. Vissers, and M. Diaz, editors, *The Formal Description Technique LOTOS*, pages 153–210. Elsevier Science Publishers B.V. (North-Holland), 1989.

[Ald89] R. Alderen. COOPER: the compositional construction of a canonical tester. In S. Vuong, editor, *Formal Description Techniques, II*, pages 13–18. Elsevier Science Publishers B.V. (North-Holland), 1989.

[AW92] H.R. Andersen and G. Winskel. Compositional Checking of Satisfaction. In K.G. Larsen and A. Skou, editors, *Proceedings of CAV 91*, LNCS 575, pages 24–36, 1992.

[BA91] G. Bruns and S. Anderson. The Formalization and Analysis of a Communications Protocol. Technical Report ECS-LFCS-91-137, LFCS, University of Edinburgh, 1991.

[Bae90] J.C.M. Baeten, editor. *Applications of Process Algebra*. Number 17 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1990.

[Bai91] J. Baillie. A CCS case study: a safety-critical system. *Software Engineering Journal*, pages 159–167, July 1991.

[BBH+91] J.C. Baeten, J.A. Bergstra, C.A.R. Hoare, R. Milner, J. Parrow, and R. de Simone. The Variety of Process Algebra. Deliverable ESPRIT Basic Research Action 3006, CONCUR (R. Milner and F. Moller, eds.), University of Edinburgh, 1991.

[BHR84] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A Theory of Communicating Sequential Processes. *Journal of the Association for Computing Machinery*, 31(3):560–599, 1984.

[BIN92] M. Boreale, P. Inverardi, and M. Nesi. Complete sets of axioms for finite basic LOTOS behavioural equivalences. *Information Processing Letters*, 43:155–160, 1992.

[BK84] J.A. Bergstra and J.W. Klop. Process Algebra for Synchronous Communication. *Information and Control*, 60(1/3):109–137, 1984.

[BK91] E. Brinksma and P. Kars. From Data Structure to Process Structure. Technical Report Memorandum INF-91-38/TIOS-91-11, University of Twente, 1991.

[Bol92] T. Bolognesi, editor. Catalogue of LOTOS Correctness Preserving Transformations. Technical Report Lo/WP1/T1.2/N0045, The LOTOSPHERE Esprit Project, 1992. Task 1.2 deliverable. LOTOSPHERE information disseminated by J. Lagemaat, email `lagemaat@cs.utwente.nl`.

[Boo89] R. Booth. An Evaluation of the LCF Theorem Prover using LOTOS. In S. Vuong, editor, *Formal Description Techniques, II*, pages 83–100. Elsevier Science Publishers B.V. (North-Holland), 1989.

[BR83] S.D. Brookes and W.C. Rounds. Behavioural Equivalence Relations induced by Programming Logics. In *Proceedings of ICALP 83*, LNCS 154, pages 97–108. Springer-Verlag, 1983.

[Bra92] J. Bradfield. A proof assistant for symbolic model checking. Technical Report ECS-LFCS-92-199, University of Edinburgh, 1992.

[Bri88a] E. Brinksma. A Theory for the Derivation of Tests. In S. Aggarwal and K. Sabnani, editors, *Protocol Specification, Testing, and Verification, VIII*, pages 63–74. Elsevier Science Publishers B.V. (North-Holland), 1988.

[Bri88b] E. Brinksma. *On the Design of Extended LOTOS*. PhD thesis, University of Twente, 1988.

[Bri89] E. Brinksma. Constraint-oriented specification in a constructive formal description technique. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness*, LNCS 430, pages 130–152. Springer-Verlag, 1989. REX School/Workshop, Mook, The Netherlands, May/June 1989.

[Bri92] E. Brinksma. From Data Structure to Process Structure. In K.G. Larsen and A. Skou, editors, *Proceedings of CAV 91*, LNCS 575, pages 244–254, 1992.

[BS87] T. Bolognesi and S.A. Smolka. Fundamental Results for the Verification of Observational Equivalence: a Survey. In H. Rudin and C.H. West, editors, *Protocol Specification, Testing, and Verification, VII*, pages 165–179. Elsevier Science Publishers B.V. (North-Holland), 1987.

[BS90] J. Bradfield and C. Stirling. Verifying Temporal Properties of Processes. In *CONCUR 90*, LNCS 458, pages 115–125, 1990.

[BS94] B. Berthomieu and T. Le Sergent. Programming with Behaviors in an ML framework — The Syntax and Semantics of LCS. In *Proceedings of ESOP*, 1994. To appear in LNCS.

[BSS87] E. Brinksma, G. Scollo, and C. Steenbergen. LOTOS Specifications, their Implementations and their Tests. In B. Sarikaya and G.V. Bochmann, editors, *Protocol Specification, Testing, and Verification, VI*, pages 349–360. Elsevier Science Publishers B.V. (North-Holland), 1987.

[CCI88] CCITT. *Specification and Description Language (SDL) Recommendations Z.100*, 1988.

[CES86] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic Verification of Finite State Concurrent Systems using Temporal Logic Specifications. In *ACM TOPLAS*, volume 8, 1986.

[CH90] R. Cleaveland and M. Hennessy. Testing Equivalence as a Bisimulation Equivalence. In J. Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, LNCS 407, pages 11–23, 1990.

[CIN91] A. Camilleri, P. Inverardi, and M. Nesi. Combining Interaction and Automation in Process Algebra Verification. In S. Abramsky and T.S.E. Maibaum, editors, *Proceedings of TAPSOFT 91*, volume II, pages 283–296. Springer-Verlag, 1991.

[Cle89] R. Cleaveland. Tableau-Based Model Checking in the Propositional Mu-Calculus. Technical Report 2/89, University of Sussex, 1989.

[Cle91] R. Cleaveland. On Automatically Explaining Bisimulation Inequivalence. In E.M. Clarke and R.P. Kurshan, editors, *Proceedings of CAV 90*, LNCS 531, pages 364–372. Springer-Verlag, 1991.

[CN91] P. Curran and K. Norrie. Specification of an ISO Protocol in LOTOS. Technical report, University of London, 1991.

[CN92] P. Curran and K. J. Norrie. An approach to verifying concurrent systems — a medical information bus (MIB) case study. In *Proceedings of the 5th annual IEEE symposium on computer-based medical systems*, 1992.

[Com91] *The Computer Journal*, 34(1), 1991. Special Issue on Term Rewriting.

[CPS89] R. Cleveland, J. Parrow, and B. Steffen. The Concurrency Workbench. In J. Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, LNCS 407, pages 24–37. Springer-Verlag, 1989.

[CR90] S.J. Colwill and G.H.B. Rafsanjani. Towards Machine-Assisted Formal Validation of LOTOS Specifications. Technical report, British Telecom, 1990.

[De 87] R. De Nicola. Extensional Equivalences for Transition Systems. *Acta Informatica*, 24:211–237, 1987.

[Der82] N. Dershowitz. Orderings for Term Rewriting Systems. *Theoretical Computer Science*, 17:279–301, 1982.

[DFGR92] R. De Nicola, A. Fantechi, S. Gnesi, and G. Ristori. An action based framework for verifying logical and behavioural properties of concurrent systems. In K.G. Larsen and A. Skou, editors, *Proceedings of CAV 91*, LNCS 575, pages 37–47, 1992.

[DH84] R. De Nicola and M.C.B. Hennessy. Testing Equivalences for Processes. *Theoretical Computer Science*, 34:83–133, 1984.

[Dic90] A.J.J. Dick. A Case Study for the ERIL Project. Private communication, 1990.

[Dic91] A.J.J. Dick. An Introduction to Knuth-Bendix Completion. *The Computer Journal*, 34(1):–, 1991. Special Issue on Term Rewriting.

[DIN89] R. De Nicola, P. Inverardi, and M. Nesi. Using the Axiomatic Presentation of Behavioural Equivalences for Manipulating CCS Expressions. In J. Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, LNCS 407, pages 54–67, 1989.

[DIN91] R. De Nicola, P. Inverardi, and M. Nesi. Equational Reasoning about LOTOS Specifications: A Rewriting Approach. In *Proceedings of 6th International Workshop on Software Specification and Design*, pages 148–155. IEEE Press, 1991.

[DMdS90] G. Doumenc, E. Madelaine, and R. de Simone. Proving process calculi translations in ECRINS: The PureLOTOS $\rightarrow$ MEIJE Examples. Technical Report RR 1192, INRIA, 1990.

[DP91] D. Duce and F. Paterno. A Formal Specification of a Graphics Systen in the Framework of the Computer Graphics Reference Model. Technical Report RAL-91-065, Rutherford Appleton Laboratory, September 1991.

[dS85] R. de Simone. Higher-level synchronizing devices in Meije-SCCS. *Theoretical Computer Science*, 37:245–267, 1985.

[EBB+86] H. Ehrig, J. Buntrok, P. Boehm, F. Nurnberg K-P. Hasler, C. Rieckhoff, and J. de Meer. Towards an Algebraic Semantics of the ISO-Specification Language LOTOS. Technical Report SEDOS/C2/N58, ESPRIT SEDOS Project, 1986.

[EFJ90] P. Ernberg, L. Fredlund, and B. Jonsson. Specification and Validation of a Simple Overtaking Protocol using LOTOS. Technical Report T9006, Swedish Institute of Computer Science, 1990.

[EFP91] P. Ernberg, L. Fredlund, and J. Parrow. An Extended Bibliography of Case Studies. Deliverable ESPRIT Basic Research Action 3006, CONCUR (R. Milner and F. Moller, eds.), University of Edinburgh, 1991.

[EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1: Equations and Initial Semantics.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1985.

[Ern91] P. Ernberg. CCS as a method of specification and verification: Analysis of a case study. Technical Report T91:05, Swedish Institute of Computer Science, 1991.

[FGL89] A. Fantechi, S. Gnesi, and C. Laneve. An Expressive Temporal Logic for LOTOS. In S. Vuong, editor, *Formal Description Techniques, II*, pages 261–276. Elsevier Science Publishers B.V. (North-Holland), 1989.

[FGR90] A. Fantechi, S. Gnesi, and G. Ristori. Compositional Logic Semantics and LOTOS. In L. Logrippo, R. L. Probert, and H. Ural, editors, *Protocol Specification, Testing, and Verification, X*, pages 365–378. Elsevier Science Publishers B.V. (North-Holland), 1990.

[Fid93] C. Fidge. Comparison of CCS, CSP and LOTOS. Notes from a seminar given at FORTE 93, but not published as part of the proceedings, 1993.

[Fle87] M. Fletcher. The Boyer-Moore Theorem Prover and LOTOS. Research & Technology Memorandum RT62/014/87, British Telecom, Ipswich, 1987.

[FLS90] M. Faci, L. Logrippo, and B. Stépien. Formal Specifications of Telephone Systems in LOTOS. In E. Brinksma, G. Scollo, and C. A. Vissers, editors, *Protocol Specification, Testing, and Verification, IX*, pages 25–36. Elsevier Science Publishers B.V. (North-Holland), 1990.

[FO91] L. Fredlund and F. Orava. Modelling Dynamic Communication Structures in LOTOS. In *Formal Description Techniques, IV*, 1991.

[GH91] J.F. Groote and H. Hüttel. Undecidable Equivalences for Basic Process Algebra. Technical Report ECS-LFCS-91-169, LFCS, University of Edinburgh, 1991.

[GL91] S. Gallouzi and L. Logrippo. A Hoare-style Proof System for LOTOS. In J. Quemada, J. Mañas, and E. Vásquez, editors, *Formal Description Techniques, III*, pages 49–62. Elsevier Science Publishers B.V. (North-Holland), 1991.

[GLO91] S. Gallouzi, L. Logrippo, and A. Obaid. An Expressive Trace Theory for LOTOS. In B. Jonsson, J. Parrow, and B. Pehrson, editors, *Protocol Specification, Testing, and Verification, XI*, pages 159–175. Elsevier Science Publishers B.V. (North-Holland), 1991.

[GLZ89] J.C. Godskesen, K.G. Larsen, and M. Zeeberg. TAV (Tools for Automatic Verification): Users Manual. Technical report, Aalborg University, 1989.

[GM92] J.F. Groote and F. Moller. Verification of Parallel Systems via Decomposition. In W.R. Cleaveland, editor, *CONCUR'92*, LNCS 630, pages 62–76. Springer-Verlag, 1992. Third International Conference on Concurrency Theory.

[Gor88] M.J.C. Gordon. HOL: A Proof Generating System for Higher-Order Logic. In G. Birtwistle and P.A. Subrahmanyam, editors, *VLSI Specification, Verification and Synthesis*, pages 73–128. Kluwer Academic Publishers, 1988.

[Got87] R. Gotzhein. Specifying Abstract Data Types with LOTOS. In B. Sarikaya and G.V. Bochmann, editors, *Protocol Specification, Testing, and Verification, VI*, pages 15–26. Elsevier Science Publishers B.V. (North-Holland), 1987.

[Gro87] R. Groenwald. Verification of a sliding window protocol by means of process algebra. Technical Report P8701, University of Amsterdam, 1987.

[Hen88] M. Hennessy. *Algebraic Theory of Processes*. MIT Press, 1988.

[HJOP89] H. Hansson, B. Jonsson, F. Orava, and B. Pehrson. Specification for Verification. In S. Vuong, editor, *Formal Description Techniques, II*, pages 227–244. Elsevier Science Publishers B.V. (North-Holland), 1989.

[HKK91] M. Hermann, C. Kirchner, and H. Kirchner. Implementations of Term Rewriting Systems. *The Computer Journal*, 34(1):20–33, 1991. Special Issue on Term Rewriting.

[HM85] M. Hennessy and R. Milner. Algebraic Laws for Nondeterminism and Concurrency. *Journal of the Association for Computing Machinery*, 32(1):137–161, 1985.

[HO82] G. Huet and D.C. Oppen. Equations and Rewrite Rules - A Survey. In R. Book, editor, *Formal Languages: Perspectives and Open Problems*. Academic Press, 1982.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall International, 1985.

[Hüt91] H. Hüttel. Silence is Golden: Branching Bisimilarity is Decidable for Context-Free Processes. Technical Report ECS-LFCS-91-173, LFCS, University of Edinburgh, 1991.

[IN90] P. Inverardi and M. Nesi. A Rewriting Strategy to Verify Observational Congruence. *Information Processing Letters*, 35:191–199, 1990.

[ISO74] International Organisation for Standardisation. *The Reference Model for Open Systems Interconnection*, 1974.

[ISO88] International Organisation for Standardisation. *Information Processing Systems — Open Systems Interconnection — LOTOS — A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour*, 1988.

[ISO90] International Organisation for Standardisation. *Information Processing Systems — Open Systems Interconnection — Estelle — Formal Description Technique Based on an Extended State Transition Model*, 1990.

[IYK90] H. Ichikawa, K. Yamanaka, and J. Kato. Incremental Specification in LOTOS. In L. Logrippo, R. L. Probert, and H. Ural, editors, *Protocol Specification, Testing, and Verification, X*, pages 183–196. Elsevier Science Publishers B.V. (North-Holland), 1990.

[KB70] D.E. Knuth and P.B. Bendix. Simple Word Problems in Universal Algebras. *Computational Problems in Abstract Algebra*, pages 263–297, 1970.

[Kir91] C. Kirkwood. An Experiment using Term Rewriting Techniques for Concurrency. In S.L. Peyton-Jones, G. Hutton, and C.K. Holst, editors, *Functional Programming, Glasgow 1990*, pages 196–200. Springer-Verlag, 1991. Extended abstract.

[Kir92] C. Kirkwood. A Case Study for the ERIL Project. Technical Report 1992/R4, University of Glasgow, 1992.

[Kir93] C. Kirkwood. Automating (Specification $\equiv$ Implementation) using Equational Reasoning and LOTOS. In M.-C. Gaudel and J.-P. Jouannaud, editors, *TAPSOFT '93: Theory and Practice of Software Development*, LNCS 668, pages 544–558, 1993.

[KN90] C. Kirkwood and K. Norrie. Some Experiments using Term Rewriting Techniques for Concurrency. Technical Report CSD-TR-623, Royal Holloway and Bedford New College, 1990.

[KN91] C. Kirkwood and K. Norrie. Some Experiments using Term Rewriting Techniques for Concurrency. In J. Quemada, J. Mañas, and E. Vásquez, editors, *Formal Description Techniques, III*, pages 527–530. Elsevier Science Publishers B.V. (North-Holland), 1991. Extended Abstract.

[Koz83] D. Kozen. Results on the Propositional μ-Calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[KS83] P.C. Kanellakis and S.A. Smolka. CCS expressions, finite state processes, and three problems of equivalence. In *Proceedings of 2nd ACM Symposium on Principles of Distributed Computing*, pages 228–240, 1983.

[KS90] P.C. Kanellakis and S.A. Smolka. CCS expressions, finite state processes, and three problems of equivalence. *Information and Computation*, 86:43–68, 1990.

[KZ87] D. Kapur and H. Zhang. *RRL : Rewrite Rule Laboratory User's Manual*, 1987. Revised May 1989. Available by anonymous ftp from `herky.cs.uiowa.edu`.

[Lam77] L. Lamport. Proving the Correctness of Multiprocess Programs. *IEEE Transactions on Software Engineering*, SE-3(2):125–143, 1977.

[Lan89] St. Lange. Towards a Set of Inference Rules for Solving Divergence in Knuth-Bendix Completion. In K.P. Jantke, editor, *Proceedings of Analogical and Inductive Inference 89*, LNCS 397, pages 304–316. Springer-Verlag, 1989.

[Lan90] R. Langerak. A testing theory for LOTOS using deadlock detection. In E. Brinksma, G. Scollo, and C. A. Vissers, editors, *Protocol Specification, Testing, and Verification, IX*, pages 87–98. Elsevier Science Publishers B.V. (North-Holland), 1990.

[Lan92] R. Langerak. *Transformations and Semantics for LOTOS*. PhD thesis, University of Twente, 1992.

[Lar86] K.G. Larsen. *Context-Dependent Bisimulation between Processes*. PhD thesis, University of Edinburgh, 1986.

[Lar90a] K.G. Larsen. Modal Specifications. In J. Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, LNCS 407, pages 232–246, 1990.

[Lar90b] K.G. Larsen. Proof Systems for Satisfiability in Hennessy-Milner Logic with Recursion. *Theoretical Computer Science*, 72:256–288, 1990.

[Led87] G.J. Leduc. The Intertwining of Data Types and Processes in LOTOS. In H. Rudin and C.H. West, editors, *Protocol Specification, Testing, and Verification, VII*, pages 123–136. Elsevier Science Publishers B.V. (North-Holland), 1987.

[Lin91] H. Lin. PAM User Manual (Version 0.6). Technical Report 9/91, University of Sussex, 1991.

[Lin92] H. Lin. PAM : A Process Algebra Manipulator. In K.G. Larsen and A. Skou, editors, *Proceedings of CAV 91*, LNCS 575, pages 136–146, 1992.

[LITE] M. Caneve and E. Salvatori, editors. LITE User Manual. Technical Report Lo/WP2/N0034/V08, The LOTOSPHERE Esprit Project, 1992. LOTOSPHERE information disseminated by J. Lagemaat, email `lagemaat@cs.utwente.nl`.

[LM92] K.G. Larsen and R. Milner. A Compositional Protocol Verification Using Relativized Bisimulation. *Information and Computation*, 99:80–108, 1992.

[LT91] K.G. Larsen and B. Thomsen. Partial specifications and compositional verification. *Theoretical Computer Science*, 88:15–32, 1991.

[LX90] K.G. Larsen and L. Xinxin. Compositionality Through an Operational Semantics of Contexts. In M.S. Paterson, editor, *Automata, Languages and Programming (ICALP 90)*, LNCS 443, pages 526–539, 1990.

[Mad92] E. Madelaine. Verification Tools from the CONCUR Project. *EATCS Bulletin*, 47, 1992.

[MFV89] C. Miguel, A. Fernández, and L. Vidaller. LOTOS Extended with Probabilistic Behaviours. *Formal Aspects of Computing*, 5(3):253–281, 1989.

[Mil80] R. Milner. *A Calculus of Communicating Systems*. LNCS 92. Springer-Verlag, 1980.

[Mil85] G. Milne. Circal and the representation of communication, concurrency and time. *ACM Transactions on Programming Languages and Systems*, 7:270–298, 1985.

[Mil89a] R. Milner. A Complete Axiomatisation for Observation Congruence of Finite-state Behaviours. *Information and Control*, 81(2):227–247, 1989.

[Mil89b] R. Milner. *Communication and Concurrency*. Prentice-Hall International, 1989.

[MM92] W. Mao and G.J. Milne. An Automated Proof Technique for Finite-State Machine Equivalence. In K.G. Larsen and A. Skou, editors, *Proceedings of CAV 91*, LNCS 575, pages 233–243, 1992.

[Mol90] F. Moller. The importance of the left merge operator in process algebras. In M.S. Paterson, editor, *Automata, Languages and Programming (ICALP 90)*, LNCS 443, pages 752–764, 1990.

[Mol91] F. Moller. The Edinburgh Concurrency Workbench (Version 6.0). Technical Report LFCS-TN-34, LFCS, University of Edinburgh, 1991.

[MP89] Z. Manna and A. Pnueli. The Anchored Version of the Temporal Framework. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, LNCS 354, pages 201–284. Springer-Verlag, 1989. REX School/Workshop, Noordwijkerhout, The Netherlands, May/June 1988.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Volume 1: Specification*. Springer-Verlag, 1992.

[MPW92] R. Milner, J. Parrow, and D. Walker. A Calculus of Mobile Processes, Parts I and II. *Information and Computation*, 100(1):1–40 and 41–77, 1992.

[MT90] F. Moller and C. Tofts. A Temporal Calculus of Communicating Systems. In *Proceedings of CONCUR 90*, LNCS 458, pages 401–415, 1990.

[MV89] E. Madelaine and D. Vergamini. Auto: A verification tool for distributed systems using reduction of finite automata networks. In S. Vuong, editor, *Formal Description Techniques, II*, pages 61–66. Elsevier Science Publishers B.V. (North-Holland), 1989.

[MV91a] E. Madelaine and D. Vergamini. Finiteness conditions and structural construction of automata for all process algebras. In E.M. Clarke and R.P. Kurshan, editors, *Proceedings of CAV 90*, LNCS 531, pages 353–363, 1991.

[MV91b] E. Madelaine and D. Vergamini. Specification and Verification of a Sliding Window Protocol in LOTOS. In K.R. Parker and G.A. Rose, editors, *Formal Description Techniques, IV*, volume C-2 of *IFIP Transactions*. Elsevier Science Publishers B.V. (North-Holland), 1991.

[Naj87] E. Najm. A Verification Oriented Specification in LOTOS of the Transport Protocol. In H. Rudin and C.H. West, editors, *Protocol Specification, Testing, and Verification, VII*, pages 181–203. Elsevier Science Publishers B.V. (North-Holland), 1987.

[Nes92] M. Nesi. A Formalization of the Process Algebra CCS in Higher Order Logic. Technical Report 278, University of Cambridge Computer Laboratory, 1992.

[New42] M.H.A. Newman. On Theories with a Combinatorial Definition of Equivalence. *Annals of Mathematics*, 43(2):223–243, 1942.

[OP91] F. Orava and J. Parrow. An Algebraic Verification of a Mobile Network. Technical Report R9102, Swedish Institute of Computer Science, 1991. To appear in FACS.

[Par81] D. Park. Concurrency and Automata on Infinite Sequences. In *Theoretical Computer Science, 5th GI Conference*, LNCS 104, pages 167–183, 1981.

[Par88] J. Parrow. Verifying a CSMA/CD-protocol with CCS. In S. Aggerwal and K. Sabnani, editors, *Protocol Specification, Testing, and Verification, VIII*, pages 373–384. Springer-Verlag, 1988.

[Ple87] U. Pletat. Algebraic Specifications of Abstract Data Types with CCS: An Operational Junction. In B. Sarikaya and G.V. Bochmann, editors, *Protocol Specification, Testing, and Verification, VI*, pages 361–372. Elsevier Science Publishers B.V. (North-Holland), 1987.

[PT87] R. Paige and R.E. Tarjan. Three Partition Refinement Algorithms. *SIAM Journal of Computing*, 16(6):973–989, 1987.

[QAF90] J. Quemada, A. Azcorra, and D. Frutos. A Timed Calculus for LOTOS. In E. Brinksma, G. Scollo, and C. A. Vissers, editors, *Protocol Specification, Testing, and Verification, IX*. Elsevier Science Publishers B.V. (North-Holland), 1990.

[QFA89] J. Quemada, D. Frutos, and A. Azcorra. TIC: A TImed Calculus. *Formal Aspects of Computing*, 5(3):224–252, 1989.

[Raf92] G.H.B. Rafsanjani. A Data Type Specification for the Process Part of Basic LOTOS — An Axiomatic Semantics. In C.M.I. Rattray and R.G. Clark, editors, *The Unified Computation Laboratory*, pages 321–332. Oxford University Press, 1992.

[RS91] S. Ramanathan and G. Sivakumar. Rewrite Systems for Protocol Specification and Verification. In J. Quemada, J. Mañas, and E. Vásquez, editors, *Formal Description Techniques, III*, pages 79–94. Elsevier Science Publishers B.V. (North-Holland), 1991.

[RvB91] N. Rico and G. v. Bochmann. Performance description and analysis for distributed systems using a variant of LOTOS. In B. Jonsson, J. Parrow, and B. Pehrson, editors, *Protocol Specification, Testing, and Verification, XI*, pages 199–213. Elsevier Science Publishers B.V. (North-Holland), 1991.

[Sco89] G. Scollo. Formal Description of the OSI Session Layer: Transport Service. In P.H.J. van Eijk, C.A. Vissers, and M. Diaz, editors, *The Formal Description Technique LOTOS*, pages 97–116. Elsevier Science Publishers B.V. (North-Holland), 1989.

[Sti87] C. Stirling. Modal Logics for Communicating Systems. *Theoretical Computer Science*, 49:311–347, 1987.

[Sti91] C. Stirling. An Introduction to Modal and Temporal Logics for CCS. In A. Yonezawa, editor, *Concurrency: Theory, Language, and Architecture*, LNCS 491, pages 2–20. Springer-Verlag, 1991. UK/Japan Workshop, Oxford, UK, September 1989.

[SW90] C. Stirling and D. Walker. CCS, liveness, and local model checking in the linear time mu-calculus. In J. Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, LNCS 407, pages 166–178, 1990.

[Tho93] M. Thomas. A Translator for ASN.1 into LOTOS. In M. Diaz and R. Groz, editors, *Formal Description Techniques, V*, pages 37–52. Elsevier Science Publishers B.V. (North-Holland), 1993.

[Tho94] M. Thomas. The Story of the Therac-25 in LOTOS. *High Integrity Systems Journal*, 1(1):3–15, 1994.

[TJ89] M. Thomas and K.P. Jantke. Inductive Inference for Solving Divergence in Knuth-Bendix Completion. In *Proceedings of Analogical and Inductive Inference 89*, LNCS 397. Springer-Verlag, 1989.

[Tur92] K. Turner. Constraint-Oriented Specification in LOTOS — The Compositional Specification of a File Handler. Lecture given at University of Glasgow, 1992.

[Tur93] K.J. Turner, editor. *Using Formal Description Techniques: An Introduction to Estelle, LOTOS and SDL*. John Wiley and Sons, 1993.

[TW93] M. Thomas and P. Watson. Solving Divergence in Knuth-Bendix Completion by Enriching Signatures. *Theoretical Computer Science*, 112:145–185, 1993.

[vE89] P. van Eijk. Tools for LOTOS Specification Style Transformation. In S. Vuong, editor, *Formal Description Techniques, II*, pages 43–52. Elsevier Science Publishers B.V. (North-Holland), 1989.

[vEKvS90] P. van Eijk, H. Kremer, and M. van Sinderen. On the use of specification styles for automated protocol implementation from LOTOS to C. In L. Logrippo, R. L. Probert, and H. Ural, editors, *Protocol Specification, Testing, and Verification, X*, pages 157–168. Elsevier Science Publishers B.V. (North-Holland), 1990.

[vEVD89] P.H.J. van Eijk, C.A. Vissers, and M. Diaz, editors. *The Formal Description Technique LOTOS*. Elsevier Science Publishers B.V. (North-Holland), 1989.

[vG86] R.J. van Glabbeek. Notes on the Methodology of CCS and CSP. Technical Report CS-R8624, Centrum voor Wiskunde en Informatica, Amsterdam, 1986.

[vG90] R.J. van Glabbeek. *Comparative Concurrency Semantics and Refinement of Actions.* PhD thesis, Centrum voor Wiskunde en Informatica, Amsterdam, 1990.

[Vis90] C. Vissers. FDTs for Open Distributed Systems, A Retrospective and a Prospective View. In L. Logrippo, R. L. Probert, and H. Ural, editors, *Protocol Specification, Testing, and Verification, X*, pages 341–362. Elsevier Science Publishers B.V. (North-Holland), 1990. Invited paper.

[vS89] M. van Sinderen. Formal Description of the OSI Session Layer: Session Service. In P.H.J. van Eijk, C.A. Vissers, and M. Diaz, editors, *The Formal Description Technique LOTOS*, pages 117–152. Elsevier Science Publishers B.V. (North-Holland), 1989.

[vSPV92] M. van Sinderen, L. Pires, and C.A. Vissers. Protocol Design and Implementation using Formal Methods. Technical Report Memoranda Informatica 92-19, TIOS 92-19, Universiteit Twente, 1992. To appear in The Computer Journal.

[VSvSB91] C.A. Vissers, G. Scollo, M. van Sinderen, and E. Brinksma. Specification styles in distributed systems design and verification. *Theoretical Computer Science*, 89:179–206, 1991.

[VTL93] Verification Techniques for LOTOS specifications. Final project report. Project information available from M. Thomas, email `muffy@dcs.gla.ac.uk`, 1993.

[Wal89] D.J. Walker. Automated Analysis of Mutual Exclusion Algorithms using CCS. *Formal Aspects of Computing*, 1(3):273–292, 1989.

[Wat92] P. Watson. The expressive power of recurrence terms. Technical Report FM-92-6, Department of Computing Science, University of Glasgow, 1992. Also submitted for publication.

[Wez90] C. Wezeman. The Co-op Method for Compositional Derivation of Conformance Testers. In E. Brinksma, G. Scollo, and C. A. Vissers, editors, *Protocol Specification, Testing, and Verification, IX*, pages 145–158. Elsevier Science Publishers B.V. (North-Holland), 1990.

[CONCUR] W.R. Cleaveland, editor. *CONCUR'92*, LNCS 630. Springer-Verlag, 1992. Proceedings of the Third International Conference on Concurrency Theory.

[FORTE] M. Diaz and R. Groz, editors. *Formal Description Techniques, V.* Elsevier Science Publishers B.V. (North-Holland), 1993.

[PSTV] B. Jonsson, J. Parrow, and B. Pehrson, editors. *Protocol Specification, Testing, and Verification, XI.* Elsevier Science Publishers B.V. (North-Holland), 1991.