An Experiment using Term Rewriting Techniques for Concurrency

Carron Kirkwood*
Department of Computing Science
University of Glasgow

August 13, 1997

Extended Abstract

Introduction

The purpose of this extended abstract is to report on a preliminary investigation into the use of term rewriting theorem proving to support verification of formal description techniques for concurrent systems. The RRL [1, 2] system, particularly Knuth-Bendix completion and equational proofs, is used to investigate verification of specifications written in the formal description language LOTOS. This work was completed as part of the SERC/IED ERIL project, which is investigating the use of Equational Reasoning for LOTOS. A more complete presentation of the results given here may be found in [3]. That paper also reports on complementary work, also undertaken as part of the ERIL project, using the LP theorem prover [4] for equational and inductive proofs about CSP [5].

The examples we present are all small ones, but are already at the level where hand verification is tedious and error prone. They execute immediately in RRL, and give hope that the same techniques can be used on much larger examples.

Experiments using LOTOS and RRL

LOTOS (Language of Temporal Ordering Specification) is based on the concept of specifying a system in terms of observable behaviour, i.e. events. Constraints may be expressed on the order of events, but there is no explicit representation of time. The communication and change of information within a system is expressed by the structure of the events.

LOTOS is made up of two distinct parts. The process part describes the behaviour of processes and is based on CCS, with some modifications drawn from other concurrency formalisms, such as CSP. The abstract data type part describes the observable behaviour of the data being exchanged by the processes. The application of term rewriting systems to abstract data types is a well researched area;

*Funded by SERC grant gr/f 35371/ 4/1/1477, Verification Techniques for LOTOS.

what is not so clear is how suitable such systems are for theorem proving in the process algebra of LOTOS.

The aim of our experiments was to generate a *complete* set of rewrite rules from the axioms of weak bisimulation congruence presented in appendix B.2.2 of the LOTOS standard.

For a set of rewrite rules to be *complete* two properties are required: the equational system must be *confluent* and *terminating*. These properties ensure that every term has a unique normal form and give us a decision procedure for equality of terms.

A set of rules with these properties can be used in a variety of ways. We have a decision procedure for the word problem specified by the original equational system. We have a check for consistency and completeness of algebraic data types. We can carry out proofs by inductionless induction, and proofs by refutation in First Order Predicate Calculus.

The Rewrite Rule Laboratory (RRL) was used to attempt to generate a complete set of rules. RRL can perform completion in the presence of a-c operators and has a limited form of order-sortedness.

The set of axioms used was built up incrementally, adding new axioms one by one, and rejecting those which cause the completion algorithm to fail or diverge. At present only a small subset of the axioms in appendix B.2.2 have been successfully run through the completion procedure. However, even this small set of rewrite rules can be useful.

Equational Proofs

With the complete set of rules obtained, simple equational proofs may now be carried out which were not possible before completion. Examples of these are:

$$(A \gg \mathbf{stop}) \gg B \equiv A \gg \mathbf{stop}$$
$$((A \gg \mathbf{exit}) \gg \mathbf{exit}) \gg B \equiv (A \gg \mathbf{exit}) \gg B$$

For a more complex example, see figure 1.

This example shows that a two-way buffer is the same as two one-way buffers in parallel. Unfortunately, this example is a little forced as recursion is not part of our restricted LOTOS. This means that these buffers only handle one data element and then die.

The second statement in this example is far easier to read than the first, demonstrating that simply-defined processes in parallel are frequently easier to understand than large process written without parallelism.

The rules already generated are not enough to prove the buffer example; laws relating some of the higher level constructs (such as parallelism) to the basic constructs of action sequencing and choice are also required. These are known as the expansion laws. Any LOTOS process can be written using just action sequencing and choice. The expansion law required here is the following:

$$\begin{aligned}
\mathrm{B} \,|[A]|\, \mathrm{C} = \;& [\,]\, \{b_i;\, (B_i \,|[A]|\, \mathrm{C}) \mid \mathrm{name}\ (b_i \notin A, i \in I\} \\
& [\,][\,]\, \{c_j;\, (\mathrm{B} \,|[A]|\, C_j) \mid \mathrm{name}\ (c_j \notin A, j \in J\} \\
& [\,][\,]\, \{\mathrm{a};\, (B_i \,|[A]|\, C_j) \mid \mathrm{a} = b_i = c_j,\ \mathrm{name}\ (\mathrm{a}) \in A, i \in I, j \in J\}
\end{aligned}$$

```
(x1;   (x2;   (y1; y2; exit
                 [] (y2; y1; exit))
          [] (y1; x2; y2; exit))
[] (x2;   (x1;   (y1; y2; exit
                 [] y2; y1; exit)
          [] (y2; x1; y1; exit)))
```

is weak bisimulation congruent to

(x1; y1; exit) ||| (x2; y2; exit)

Figure 1: Buffer example

This law cannot be entered straight into RRL because generalised choice is not a part of the restricted LOTOS used, so two useful instantiations of this expansion law are derived:

$$a; A \mid\mid\mid b; B == (a; (A \mid\mid\mid b; B))\,[]\,(b; (B \mid\mid\mid a; A)))$$
$$a; A \mid\mid\mid \mathbf{exit} == a; (A \mid\mid\mid \mathbf{exit})$$

When these are added to the complete set of rules, the equivalence in figure 1 may be proved as an equational theorem.

Axioms causing Infinite Sequences of Rewrite Rules

As mentioned earlier, some axioms of appendix B.2.2 caused the completion procedure to diverge, generating an infinite sequence of rewrite rules. It would be useful if these rules could be generalised by producing a finite set of rules that subsume the infinite sequence of rules and are terminating and confluent. Further details of various techniques for generalising infinite sequences of rules may be found in [6, 7, 8].

Some attempts were made to generalise the particular sequences obtained. For example, the law distributing **hide** across $\gg$,

$$\textbf{hide}\ \mathrm{A}\ \textbf{in}\ (B_1 \gg B_2) \equiv (\textbf{hide}\ A\ \textbf{in}\ B_1) \gg (\textbf{hide}\ A\ \textbf{in}\ B_2)$$

produces two distinct infinite sequences of rewrite rules. The generalisations for these are as follows.

$B_1 \gg B_2 \rightarrow B_1$
where B_1 has **stop** *as the rightmost argument*
and B_1 is built from applications of **hide** *and $\gg$.*

and

$(B_1 \gg B_2) \gg B_3 \rightarrow B_1 \gg B_3$
where B_2 is built from applications of **hide**, *$\gg$ and* **exit**,
and B_2 is a subterm of B_1.

The sequences generated by this rules can be eliminated by adding two new rules.

hide A **in stop** $\equiv$ **stop** *since* $L(\textbf{stop}) = \emptyset$
hide A **in exit** $\equiv$ **exit** *since* $L(\textbf{exit}) = \emptyset$

However, most of the infinite sequences generated by our experiments were not so easy to eliminate. For example:

The rules in the infinite sequence generated by the addition of the axiom for associativity of choice (A [] B) [] C == A [] (B [] C) follow this basic pattern:

$A \,[]\, B \rightarrow B$

This rule is of course undesirable as it makes choice totally unfair (it will always choose the second argument). Also it is not an exact generalisation (i.e. not every instantiation is a rule) and so adding it would imply unsound equivalences. We know that A is a subterm of B (by inspection of the rules generated), but there is a more complex relationship between these two terms, which deserves further investigation.

Other Results

It is expected that the weak bisimulation congruence laws in appendix B of the LOTOS standard will be sound. It was suspected that this was not true and our experiments revealed an inconsistency.

When deciding which laws to use we entered:

$A \gg \textbf{stop} \equiv A \;|||\; \textbf{stop}$

When taken with other rules in the rule set, this rule causes the following contradiction:

$\mathbf{i}; \textbf{stop} \equiv \textbf{stop}$

Under weak bisimulation congruence this is false as the two equivalent processes are required to behave in the same way in all contexts. To test this condition it is necessary only to test the way in which the processes behave when part of a choice statement.

Conclusions and Further Work

We conducted an experiment in the application of term rewriting to the concurrency formalism, LOTOS. While the results we obtained were hopeful, we appreciate that there is still much work to be carried out in this area.

This preliminary investigation has been successful in that not only have we have shown that term rewriting is useful in reasoning about concurrency, we have also uncovered several areas for further development.

Acknowledgements

We should like to thank Deepak Kapur for letting us have RRL. We would especially like to thank Ursula Martin, Muffy Thomas and Phil Watson, our colleagues.

References

[1] D. Kapur and H. Zhang, RRL : Rewrite Rule Laboratory User's Manual, 1987, Revised May 1989.

[2] H. Zhang, D. Kapur, M.S. Krishnamoorthy: *A Mechanizable Induction Principle for Equational Specifications*, Proc. 9th Intl. Conference on Automated Deduction, LNCS 310 (1988).

[3] C. Kirkwood and K. Norrie, Some Experiments using Term Rewriting Techniques for Concurrency, Technical Report — Royal Holloway and Bedford New College, 1990.

[4] S.J. Garland, J.V. Guttag: *An Overview of LP, The Larch Prover*, Proc. Rewriting Techniques and Applications, 3rd Intl. Conference, LNCS 355 (1989), 137-151.

[5] C. A. R. Hoare, Communicating Sequential Processes, Prentice-Hall, 1985.

(September 1988)

[6] M. Thomas and K.P. Jantke, Inductive Inference for Solving Divergence in Knuth-Bendix Completion, Proc. Analogical and Inductive Inference '89, GDR, LNCS 367, Springer-Verlag, 1989.

[7] St. Lange, Towards a Set of Inference Rules for Solving Divergence in Knuth-Bendix Completion, Proc. Analogical and Inductive Inference '89, GDR, LNCS 367, Springer-Verlag, 1989.

[8] M. Thomas and P. Watson, Solving Divergence in Knuth-Bendix Completion by Enriching Signatures, in preparation, 1990.