Report on a Visit to ICOT

John Lloyd

Department of Computer Science
University of Melbourne
Parkville, Victoria 3052
Australia

1. Introduction

I was invited to visit ICOT for 3 weeks from Monday March 3 to Friday March 21, 1986. The suggested topic of collaborative research was "Declarative Debugging and Formal Semantics of GHC". I was very pleased to be able to accept the invitation because my research interests are close to those of Dr. Furukawa and his colleagues in the first laboratory and I felt a stimulating collaboration was likely. Of course, I was also pleased to have the opportunity to have a close look at this exciting project and assess for myself the progress made in the first five years.

2. Presentation of ICOT Research

The first week was mostly taken up with informal presentations of the research and development themes in the five laboratories at ICOT. I found these presentations most interesting and felt by the end of the first week that I had a good overview of the status of the project. During this week, Mr. Takeuchi of the first laboratory gave an informal presentation of his work on debugging GHC and a procedural semantics for GHC. I also gave an informal 2 hour presentation of my recent work on declarative error diagnosis for PROLOG programs. It seemed at that time that the most promising problem for collaborative research was the debugging of GHC programs.

3. Collaborative Research

During the second week, I began a closer study of the problem of debugging GHC programs. Mr. Takeuchi already had written a debugger for GHC, but it seemed to me that the formal basis of this debugger was unsatisfactory. After much discussion with Mr. Takeuchi and the discovery of some very strange examples, it became clear that a lot more work was necessary to properly understand debugging in GHC. I then decided to spend the remainder of my visit to ICOT on this problem.

The main results of this collaboration can be summarized as follows. Debugging in GHC is rather different to declarative debugging in PROLOG. This is because the commitment mechanism of GHC makes it difficult, and probably impossible, to give purely declarative definitions of the concepts involved in GHC debugging. For example, the concept of an intended interpretation for a GHC program has a very strong procedural flavour. The commitment mechanism of GHC also causes many buggy programs to deadlock. The possibility of deadlock makes the job of debugging GHC much more complicated than debugging PROLOG. In particular, we showed that debugging GHC programs by only using the final computation tree has some limitations not present in the PROLOG case. We were able to clear up the case of termination with incorrect answer and the case of deadlock fairly satisfactorily by results which showed the extent to which the debugging algorithm was complete. The case of failure turned out to be much more complicated and, although we made some progress, further work will be necessary to find a satisfactory method for this case.

Another of my research interests is deductive database systems. I gave a formal lecture on this topic in the first week and an informal presentation on stratified databases in the last week. Stratified databases (or programs) are a newly introduced class of databases which are important because they allow both negation and recursion and yet have a model theory which extends the well known model theory for definite databases. The reason that the model theory works in stratified databases is that, when suitably restricted, the usual mapping associated with a database, which maps interpretations to interpretations, becomes monotonic. This is an at first surprising result which is certain to be very fruitful. I am convinced that much of the theory of definite databases will eventually be extended by this technique to stratified databases. Already this technique has allowed the extension of a significant result about simplification of integrity constraint checking to stratified databases. From discussions during my visit, it seems that there could be immediate applications to problems in analogical reasoning, program verification and other areas.

Another idea which I think ICOT may find useful is the extended syntax provided by MU-PROLOG. This extended syntax allows arbitrary formulas in the bodies of clauses and queries. This syntax is very easy to implement and its extra expressive power suggests lots of applications in deductive database systems and expert systems.

I also gave a lecture on declarative error diagnosis to the Foundations of AI study group in the second week. This research puts the ideas of declarative debugging (also known as algorithmic debugging) on a firmer theoretical foundation. The practical outcome of this research is a debugging system I wrote which can handle the extended syntax and also highly coroutining programs. I believe that such declarative debugging systems will be an indispensable component of future logic programming systems.

4. Impressions of ICOT

My overall impression of the project was very positive. A great deal of interesting work has been done in the first five years. I was particularly impressed by the research on GHC in the first laboratory. GHC has an attractive elegance and simplicity which suggests it is an important contribution.

There are three aspects of the project that I would like to comment on.

The first point is that some of the researchers at ICOT do not seem to have a sufficiently strong background in the basic theory of Logic Programming. (This basic theory is essentially what is contained in the first three chapters of my book). I believe that all researchers need to know these results even if they are not theoreticians. The basic theory provides the "culture" necessary to carry out logic programming research. This problem could be easily overcome by starting a study group on the basic theoretical results.

The second point is that I was surprised at the small amount of interest at ICOT in deductive database systems and the related area of "database" logic programming systems, such as MU-PROLOG. The facilities offered by MU-PROLOG are especially important for building deductive database systems and expert systems, which are two of the main application areas of the Fifth Generation Project. I suggest a study of languages like MU-PROLOG and NU-PROLOG will prove fruitful.

Thirdly, it seems to me that ICOT is trying to build two incompatible inference machines - one an AND parallel machine for running GHC and the other an OR parallel knowledge base machine. With only five years left in the project, there seems to be a danger of there not being enough time to unify these very different approaches. Of course, these two different approaches stem from a basic property of Logic Programming that there are two kinds of programming - one is "database" programming, which requires OR parallelism, and the other is "system" programming, which requires AND parallelism. Certainly, at this stage of the research, it is very difficult to know which of these kinds of parallelism is best suited to be the basis of the hardware or whether it is possible to build hardware which handles both kinds of parallelism well.

In spite of these uncertainties, it seems more promising to start building now a single inference machine and a single knowledge base machine, which contains hardware for knowledge management and indexing but no inference capabilities.

The main function of the knowledge base machine would simply be the very fast retrieval of information which the inference machine requires. The most obvious form in which to store this information would be as database clauses (or, more generally, as first order formulas). Based on the research so far, a possible approach would be to build an AND parallel inference machine to run GHC. This approach then seems to require the compilation of "database" programs into GHC. This compilation process must be able to handle the extended syntax and coroutining capabilities of systems such as MU-PROLOG. The other potentially difficult problem is smoothly interfacing the AND parallel machine and the knowledge base machine. While this approach is unlikely to produce a machine which can run OR parallel programs as fast as a specialised OR parallel machine, it has the great attraction that a single machine would be produced which would presumably handle a large class of applications well. My reservation with this approach is that it is being driven too much by the demands of the machine and not enough by the demands of the application areas.

Finally, let me reiterate that my overwhelming impression of the project was positive. There is much exciting research being done by an enthusiastic and capable group of researchers. I fully expect the fifth generation computers to have a profound impact on the way we do computing.

5. Visits to other Institutions

I also had the opportunity to visit two other institutions in Japan. During the second week, I made a brief visit to the Fujitsu Research Laboratories in Kawasaki. Fujitsu has a large AI lab (with 300 researchers). I was given a presentation on their FACOM alfa machine and other interesting research themes in the laboratory.

At the end of the second week, I spent a weekend at the Research Institute for Fundamental Information Science at Kyushu University. On the Saturday, I gave two lectures, one on deductive database systems and one on declarative error diagnosis. Members of the Institute then presented their research. I found that we had many common research interests and the visit was a very stimulating one.

Acknowledgements

First let me thank Dr. Fuchi for inviting me to ICOT. I learned a great deal during the visit and my hope is that ICOT got something in return.

Particular thanks go to Mr. Kusama of ICOT who many arrangements before and during my visit.

My thanks also to Dr. Furukawa and his colleagues in the first laboratory for providing such a stimulating research environment. In particular, I had many fruitful discussions with Mr. Takeuchi.

Mr. Seki made many arrangements for me during my visit, especially the most important one of where to have lunch! He was always available when I needed assistance. Mr. Seki and Mr. Ohki accompanied me on a very successful expedition to Shinjuku to buy a camera. Also Mr. Ohki and Ms. Semba gave up a Sunday afternoon to show me around Tokyo, which I greatly appreciated.

For my visit to Fujitsu, I thank Director Sato and Dr. Sugimoto. The visit to the restaurant in the evening was truly memorable!

Thanks also go to Professor Arikawa for the invitation to visit Kyushu University and yet another memorable banquet. Dr. Haraguchi kindly escorted me to Kyushu and took care of all the arrangements. Particular thanks go to him and Dr. Takeya who took me sightseeing to Mt. Aso on the Sunday.

Finally, my thanks to all the people I met in Japan for their friendliness and generosity.

RESEARCH RESUME

Name John Wylie LLOYD

Current Appointment

Senior Lecturer in Computer Science at Melbourne University.

Research Interests

My research interests are in Logic Programming and its impact on Database Systems and Artificial Intelligence.

Logic Programming is currently attracting substantial interest, much of which is due to the wide publicity given to the Japanese 5th Generation Computer Project. However, there are very strong technical reasons why Logic Programming will play an increasingly important role in Computer Science. The most fundamental reason is due to the fact that logical inference can be regarded as the fundamental unit of computation. From this perspective, many diverse areas of Computer Science can be unified and simplified. For example, PROLOG is now commonly used as the core of database systems, as a systems programming language and as an Artificial Intelligence programming language.

The following is a list of the specific research topics in which I am interested.

(a) *Theoretical Aspects of Logic Programming*

There are many important unresolved theoretical issues in Logic Programming. My interests here are mainly concerned with negation (esp. negation as failure), completeness, control and concurrency.

(b) *Theoretical Aspects of Deductive Database Systems*

A firm theoretical foundation for deductive database systems is just emerging. My interests here concern soundness and completeness of query evaluation, logic as a query language, integrity constraints and various update issues.

(c) *Implementation of Deductive Database Systems*

PROLOG provides a satisfactory query evaluator for deductive database systems. However, PROLOG needs to be augmented with indexing schemes, query languages and so on, before it can properly be regarded as a deductive database system. My interests here concern clause indexing, query optimization, integrity constraint implementation and query language design and implementation.

(d) *Programming Environments*

One of the weaknesses of current PROLOG systems is their programming environments. This is a major reason why many programmers still prefer LISP to PROLOG. I have recently been investigating declarative error diagnosis, starting from the work of Shapiro and Ferrand. I have written a debugging system which finds errors in extended syntax programs by asking questions of the programmer. Typically, it asks if an atom is satisfiable (in the intended interpretation) and sometimes asks for values of variables. The programmer does not need to have any understanding whatsoever of the computational behaviour of the PROLOG system. I believe such debugging systems are crucial for the future development of PROLOG because they offer the potential for writing and debugging programs at a very high and purely declarative level.

Machine Intelligence Project

I am one of the principal investigators for the Machine Intelligence Project at Melbourne University. This project is being funded by the Department of Science. One major task of this project is the development of a deductive database system based on MU-PROLOG, whose performance will be competitive with state-of-the-art relational database systems. The other major task is the development of a compiler version of MU-PROLOG.

Book on Foundations of Logic Programming

I have written this book to provide the first account of the mathematical foundations of Logic Programming. The purpose of the book is to collect, in a unified and comprehensive manner, the basic results of Logic Programming previously available only in widely scattered research papers.

Associations with Journals and Conferences

Editorial Advisor for The Journal of Logic Programming

Member of Editorial Board for Information Technology: Research & Development

Member of Program Committee for Third International Logic Programming Conference to be held in London in 1986.