Concurrent Logic Programming Languages and Partial Evaluation Report on a Visit to ICOT

Giorgio Levi

Dipartimento di Informatica
Università di Pisa
Corso Italia, 40, I56100 Pisa, Italy

January 29, 1988

1 Introduction

I spent the last three weeks of January 1988, as a visiting researcher at ICOT. I met with several researchers of ICOT, in particular those of the First Laboratory, and discussed their and our projects on language design and program transformation.

My research interests are, in fact, mainly in the areas of *semantics of logic languages* and *partial evaluation*. I believe that the semantics (and, in particular, the declarative semantics in the case of logic languages) should be used as a basic tool in language design and program optimization. In fact, the semantics can be of great help in the design of clean, simple, yet powerful languages and is necessary for proving the correctness of program transformation rules.

My second strong belief is that in the above mentioned research areas there should always be a strong relation between theory and practice. Therefore the First Laboratory was exactly my place. In that Laboratory, in fact, a relatively small group of very good researchers is doing excellent work both on the theory and the practice of concurrent logic programming, with an additional (and valuable) strong bias coming from other areas at ICOT, such as architectures and artificial intelligence applications.

On my first day at ICOT, I was welcomed by Doctor Iwata from the Research Planning Department, who introduced me to Doctor Fuchi, Director

of the Research Center, and presented me an overview of ICOT, through an excellent video tape. I then got in touch with Doctor Murakami, who, as main host, took in charge all the details of my stay at ICOT. On my first day I met Doctor Furukawa, Deputy Director of the Research Center, and I was introduced to Doctor Hasegawa, the chief of the First Laboratory. I had later the opportunity to meet the chiefs of other Laboratories.

Before entering into details, I want to say that the visit was really successful (at least, from my viewpoint), that I had many interactions with the ICOT researchers, that I succeeded in learning a lot and even in doing some original technical work, with the help of a friendly and stimulating environment.

I would also like to emphasize the quality of the wellcome I got from the ICOT people. I was invited to a welcome dinner by Doctor Furukawa and the colleagues of the First Laboratory, to a wellcome lunch and to a farewell party. My colleagues Fujita, Murakami, Okumura, Tanaka and Ueda showed me the delights of Japanes food, by taking me every day for lunch (and sometimes for dinner too) at a different restaurant. I would like especially to thank Doctor Iwata and his wife, who invited me at their home for an excellent japanese dinner and a tea ceremony, on Saturday the twentythird.

2 Presentation of ICOT Researches

My first technical meeting was with the researchers of the First Laboratory, involved in projects somewhat related to GHC, namely the colleagues Fujita, Murakami, Okumura, Tanaka and Ueda. They gave me an overall introduction to their current research activities, on GHC unfolding and verication, on partial evaluation, on layered-stream programming techniques and on extensions of GHC with metacalls, reflection and object-oriented features.

I was already familiar with most of their results, because of previous contacts and/or talks given at Logic Programming Conferences and Workshops. I was anywhere impressed by the quality of their results, and by the nature of their working style. In fact, each researcher has a large part of a project to take responsability for, and, at the same time, many people work on projects which are closely enough related, so as to make discussions and exchange fruitful and efficient. This seems to be a general positive feature of ICOT, since I saw many interactions to take place with other researchers

GHC. I also attended some lectures given at ICOT by other foreign visitors, namely two lectures by Prof. Ken Mc Aloon on Constraint Logic Programming and a nice application of the Stratification property, one lecture by Dr. Yvon Autret on Parallel Architectures and one lecture by Dr. Philippe Devienne on a very interesting method to study termination and complexity in rewriting systems.

3 My presentations at ICOT

I gave an informal presentation of the research activities of my logic programming group in Pisa to the colleagues of the GHC group and two formal talks.

In the first presentation I gave an overview of our recent achievements on the following research areas:

a) Design and semantic characterization of a Logic+Functional language, featuring infinite data structures and partial functions (LEAF project funded by the European Community under ESPRIT).

b) Declarative semantics of logic languages (and their extensions, including concurrent logic languages). This was also the topic of my first talk.

c) Design and implementation of a Knowledge Base Management System, based on Prolog, Data Bases, metaprogramming and partial evaluation of metaprograms (project Epsilon, funded by the European Community under ESPRIT). This was the subject of my second talk.

d) Theoretical studies on negation and partial evaluation (definition of a class of logic programs, for which negation as finite failure is correct and complete and for which there exists a correct strategy for recursion detection in partial evaluation).

e) Definition of an algebra of logic programs, with a set of algebraic operators acting on logic program theories (including intensional negation) (LML project). The algebra provides a compositional semantics to logic programs and allows to model updates on programs in a purely declarative way. The theory is currently being used to model the multiple worlds feature of Epsilon and its inheritance mechanisms.

My first formal talk was on a new semantics for pure logic programs and on the declarative semantics of committed-choice languages. The new semantics defines models containing universally quantified formulas and allows a complete declarative characterization of computed answer substi-

tutions, thus realizing the real equivalence between the operational and the declarative semantics. This new semantics is the basis of our formalizations of the semantics of concurrent logic languages. On this topic, I first presented some recent results on a fixpoint characterization of the set of finite failures, caused by the commit operator. The last part of the talk was related to the semantics of synchronization. I presented essentially the contents of the paper we presented at the last Logic Programming Conference. However, I made some efforts to apply the general construction to the case of Flat GHC. This effort and the comments I had after the talk originated the definition of a different class of models for Flat GHC, which is described in the enclosed report and that I will describe in a later Section.

The second talk was an overview of the Epsilon system, with a particular emphasis on the partial evaluation algorithm, which is used as a systematic (and automatic) tool to compile new inference engines, defined as Prolog metaprograms. I discussed the peculiarities of our algorithm (it handles full Prolog and is incremental, using the multiple world feature) and reported on some performance analyses. I finally presented some recent results on a methodology to define metainterpreters, which allows to combine their functionalities automatically.

4 Discussions

I had several interesting and stimulating discussions, particularly with the researchers of the GHC group, with Dr. Furukawa and Dr. Hasegawa, on different topics, including the semantics of perpetual processes and recursion detection in partial evaluation. I will here report on our discussions on four relevant issues.

KL-2 design

I discussed with Dr. Tanaka the design of the new language, which should extend GHC with metaprogramming capabilities, reflection and object-orientation. I suggested to take into consideration some recent results on higher-order logic programming, which seem to offer a logical basis which allows to define most of the necessary constructs. Our algebra of logic programs, defined in the LML project, could also be useful.

Partial evaluation in Prolog

I had several discussions with Mr. Fujita. I presented our recent "multiple partial evaluation algorithm", which allows to define forward and backward binding propagation, even when unfolding is not possible.

Mr. Fujita explained me its new partial evaluation algorithm with constraints. The idea seems really innovative and relevant, as shown by the improvements obtained in some examples.

After my first exposition to the CAL system, we discussed the applicability of the CAL constraint solver as a theorem prover to simplify the sets of delayed primitive calls in the bodies of the clauses generated by partial evaluation. Such a combination would really result in something very similar to Dr. Futamura's "Generalized Partial Computation". The technique could also be combined with the current constraint handling in Mr. Fujita's algorithm. I was informed by Mr. Fujita that he is currently studying this problem and this seems to me a very good decision.

Unfolding and partial evaluation in GHC

I had some discussions with Mr. Fujita on the nature of a partial evaluation system for GHC. We agreed that two parallel goals should be pursued, the first one aiming at a powerful interactive program transformation system, the second one aiming at an automatic compilation system, to be used in relation to GHC metaprograms.

The discussion on the use of constraints in the partial evaluator of GHC, at Mr. Fujita's talk, was also very interesting.

I then had several discussions on GHC unfolding with Dr. Furukawa, Dr. Murakami and Dr. Ueda. The main problem here is, of course, the correctness of the transformations with respect to the language operational semantics. This problem seems to have been solved by the set of rules proposed in the paper by Furukawa, Okumura and Murakami. My feeling, however, is that we need a set of complete unfolding rules, in order to build partial evaluation systems similar to those that have been defined for Prolog. Completeness means here that it is possible to remove by unfolding all the procedure calls (apart from the usual recursive calls). I tried to suggest a solution to this problem and the results are contained in the enclosed report.

Declarative semantics of FGHC

The discussion with Dr. Murakami and Dr. Ueda has been mainly related to my new declarative characterization of a variation of Flat GHC. The discussions I had with them were of great help for my own work and allowed us to understand what is the essence of the GHC semantics that my construction does not succeed in modeling, i.e. the fact that in some programs which have a cyclic producer-consumer relation in the body, we need the ability to model the data structures computed by partial computations.

In the middle of my stay, I had the opportunity to visit ETL at Tsukuba,

within the First Laboratory and even with researchers of the other Laboratories.

I will report on the discussions with the researchers of this group later. I want just to say that GHC seems really to be a major achievement and that I asked to have a copy of its compiler, so as to allow us to make some experiments in Pisa on the practice of concurrent logic programming.

Dr. Aiba gave me a quite interesting presentation of the CAL system. Their constraint solver for non-linear equations is really impressive. I was not fully familiar with the Buchberger algorithm, which seems really powerful. Also, the idea of extending the constraint solver with quantifier elimination and partial order relations seems to be very promising. It is a very nice example of use of term rewriting techniques in the logic programming environment. Most of the existing CLP systems use in fact constraint solvers, based on algorithms (for example, the simplex algorithm), which are effective and powerful, yet have no much of the flavour of logic programming. This is not the case of TRS techniques which are, on the contrary, quite close to the logic programming paradigm. An open problem seems to be the problem of handling negation. Even if full logical negation cannot be handled by the Buchberger algorithm, I feel that something similar to negation as finite failure could be defined and handled. We have discussed the possible interactions between CAL and GHC and the applicability of CAL to make partial evaluation more effective. I will report on the last point later. This seems anyway a rather important issue, that makes CAL very interesting even for my own partial evaluation project. I intend to investigate this problem on my return to Pisa, if I will have a chance of using CAL on our Unix machines.

Miss Susaki made me a very useful presentation on the research activities of the PIM group in the Fourth Laboratory. She gave me an overview of the Multi-PSI and PIM-1 architectures, of the operating system and of the various language layers. We discussed about the adequacy of FGHC as a language for writing operating systems. I wonder whether some of the extensions defined in Sho-en could have been defined by FGHC metainterpreters and then be compiled down to FGHC, as the Concurrent Prolog group did in the Logix system. I am not an expert in parallel architectures, even if I am involved in a project, where our Logic+Functional language LEAF is being implemented, starting from a Flat committed-choice subset. I was then very interested in the compilation of KL1 and in the definition of the user language A'UM.

I attended a talk my Mr. Fujita on its partial evaluation system for

where I discussed the possibility for one of our young researchers to spend some time at ETL, under a recent Exchange Agreement between Italy and Japan. There, I met Dr. Taisuke Sato and I had a quite interesting discussion on his research on program transformations and our research on semantics and LML.

5 My work at ICOT

I don't want to spend too many words on the results of my own research activity in these three weeks, since it is fully described in the enclosed report. I just want to mention that my work was strogly influenced by my being at ICOT, by results of ICOT researches and by my discussions with ICOT people.

The starting point was my realizing, after my talk, that the models obtained for GHC from our general construction, were too complex and hard to understand. So I had the idea of defining models as sets of Flat GHC unit clauses. The technical part, where I define the various declarative semantics, is based on a notion of guarded unification, that was found by studying and understanding the last transformation in the already mentioned GHC unfolding paper. My semantics (and the corresponding set of unfolding rules) still does not model GHC deadlocks. The already mentioned discussions with Dr. Murakami and Dr. Ueda made me understand the real nature of the problem and I hope to obtain more results in the future.

The conclusion is that my paper is really an ICOT paper and I would be honoured if it could appear as an ICOT Technical Report.

6 Conclusion

My visit was really fruitful, and not only from the scientific viewpoint. It gave me a better understanding of the Japanese culture and way of life. As already noted, I was impressed by the ICOT way of working, which is quite different from the one we have in our universities and industrial research centers. In the universities, we are used to a strongly individual work, usually only theoretical, with communications taking place essentially through the leader. Our industrial research centers have a reasonable organization, yet they do not succeed in obtaining innovative scientific results. The ICOT organization is exactly what I tried to do in my country, mainly with the

transformation of myself into a complex parallel communication switching device. An environment like ICOT would really make my life easier. I was, of course, impressed by the wish that ICOT people have to transform their even foundational research into industrial products. I think that this attitude, together with the organization and the high quality of the personnel, is one of the main reasons of the success of the Institute.

Concerning my stay, I feel that such exchanges are really useful and I hope that more will take place in the future in both directions. One thing that could really make me happy, would be the possibility of defining some Research agreement between ICOT and the University of Pisa, which could then serve also as a basis for researcher exchange. The organization of

Prof. Giorgio Levi

Expertise: Concurrent Logic Programming Languages

Recent publifications:

1. G. Levi and C. Palamidessi,
   "The declarative Semantics of Logical Read-Only Variables,"
   in Symp. on Logic Programming, 1985.
2. G. Levi,
   "Object Level reflection Inference Rules by Partial Evaluation,"
   (Extended Abstract),
   in P. Maes and D. Nardi, (eds.):
   Workshop on Meta-Level Architectures and Reflection, Sardinia, October 1986, (to appear)
3. G. Levi and C. Palamidessi,
   "An Approach to The Declarative semantics of Synchronization in Logic Languages,"
   inFourth International Conference on Logic Programming, 1987.
4. G. Levi, C. Palamindessi, P.G. Bosco, E. Giovannetti, and C. Moiso
   "A Complete Semantic Characterization of K-leaf: A Logic Language with Partial Functions,"
   in Symp. on Logic Programming, 1987.

5. G. Levi and G. Sardu,
Partial Evaluation of Meta-Programs in a Multiple World's Logic Languages,"
in A.P. Ershov, D. Bjørner and N.D. Jones (eds.):
Proc. of Workshop on Partial Evaluation and Mixed Computation, Gl. Avernæs, Denmark, October 1987,
North-Holland, 1988. (to appear)

Prof. Giorgio Levi

Expertise: Concurrent Logic Programming Languages

Recent publifications:

1. G. Levi and C. Palamidessi,
"The declarative Semantics of Logical Read-Only Variables,"
in Symp. on Logic Programming, 1985.
2. G. Levi,
"Object Level reflection Inference Rules by Partial Evaluation,"
(Extended Abstract),
in P. Maes and D. Nardi, (eds.):
Workshop on Meta-Level Architectures and Reflection, Sardinia, October 1986, (to appear)
3. G. Levi and C. Palamidessi,
"An Approach to The Declarative semantics of Synchronization in Logic Languages,"
inFourth International Conference on Logic Programming, 1987.
4. G. Levi, C. Palamindessi, P.G. Bosco, E. Giovannetti, and C. Moiso
"A Complete Semantic Characterization of K-leaf: A Logic Language with Partial Functions,"
in Symp. on Logic Programming, 1987.
5. G. Levi and G. Sardu,
Partial Evaluation of Meta-Programs in a Multiple World's Logic Languages,"
in A.P. Ershov, D. Bjørner and N.D. Jones (eds.):
Proc. of Workshop on Partial Evaluation and Mixed Computation, Gl. Avernæs, Denmark, October 1987,
North-Holland, 1988. (to appear)