Report of My Research Visit at ICOT

Charles X.F. Ling 凌 晓峰
Department of Computer Science
University of Western Ontario
London, Ontario, Canada N6A 5B7
1-519-661-3515 (FAX)
Email: ling@csd.uwo.ca

During February 17 to February 28 of 1992, I had a research visit at ICOT (The Institute for New Generation Computer Techlology) in Tokyo. I learned basics of KL1 and wrote a prototype learning program in KL1; I gave a talk on the research I am doing on inventing new predicates in inductive logic programming; I had quite a few discussions with researchers working at ICOT; I attended several talks and the demonstration of PIM (Parallel Inference Machine) and Multi-PSI machines: and I visited several other universities in Tokyo. The short visit of two weeks is very fruitful.

Activities at ICOT

I had quite a few discussions with people at ICOT (especially in the Fifth Research Lab) about their research. **Hiroki Ishizaka** showed me his work in theoretical machine learning. I was impressed by the fine research hs has been conducting. The work on minimal multiple generalization, a natural extension of Plotkin's LGG (least general generalization) is particularly intersting to me. In discussion with **Jun Arima** on analogical reasoning, I suggested that a more general representation may be used to represent (and simplify) the Analogy Prime Rule, thus to have a more general account of valid analogical reasoning. I hope to look into the new representation for analogy in the near future with Arima-san. **Tadashi Kawamura** showed me his work on folding/unfolding of logic program transformation. We noted that folding and unfolding correspond to generalization (such as inverse resolution) and resolution deduction respectively in some sense. However, the folding is a semantics preserving operator, not a generalization operator. Thus some conditions should be satisfied. The main purpose of folding is not inductive learning, but equivalent program transformation for more efficient theorem proving. The synthesis of new terms is particularly intersting to me. I will study how we may use folding method to synthesize new terms in inductive inference. **Katsumi Inoue**'s work on consequence-finding method for abduction is also very intersting. The abduction problem of finding missing hypotheses is transferred to the problem of finding consequence (theorems) in some specific form. I'd like to look into this method in inductive learning: find Horn clause theory in the specific form (such as consisting of one unit clause and one Horn clause) via consequence-finding. **Masayuki Fujita** and I duscussed the application of KL1 in model generation theorem prover. I noticed some limitation of model generation theorem prover (such as it can not prove theorems with (true) universal variables) and some possible limitation of KL1.

I gave a talk on inventing necessary theoretical terms in inductive logic programming on Feb 26. During the talk Dr. Furukawa and Dr. Bry asked many good questions and suggested some new ideas for future research in the area.

I attended the demonstration of PIM and Multi-PSI led by Taki-san and Dr. Iwata, in which several applications were shown. I attended talks given by Dr. Furukawa (about FGCS project), Prof. R. Kott from INRIA (general introduction of INRIA), Dr. Chikayama and Dr. Ueda (on GHC and PIMOS). I was invited to the welcome lunch attended by most chief leaders of ICOT, and the farewell party with researchers in Fifth Lab and Dr. Furukawa.

A Learning Program in KL1

In my algorithm of inventing new terms, a proper partition of the theory is crucial; yet no "short-cut" exists so far for finding the right partition. A heuristic algorithm was designed, which examines all simple revisions of the current partition, and chooses the one which produces a conjecture with the minimal size. Since many alternative revisions exist, inductive learning and invention of new terms can be carried out for each revision *parallelly* and *independently*. This is a very good application of KL1 and PIM because the potential speed-up would be very large and the communication overhead among parallel processors is zero. In fact it is an independent OR-parallel problem, and because KL1 only supports AND-parallelism, it is transferred to AND-parallel via iteration.

So I learned basics of GHC and KL1 and wrote a KL1 program within a week. One processor (PE 0) generates revisions of the current partition and another processor (PE 1) dispatches each revision from PE 0 to different processors concurrently. Another processor (PE 2) waits for the results (the size the theory) of each processor and compares them and chooses the minimal one when the results are done concurrently. The program uses simple idea of stream (data) parallelism: the stream of partitions, the learning based on each partition, and the comparisons can be implemented parallelly, driven by the stream of data made available. The program runs successfully in the Pseudo Multi-PSI machine. I added load balance and tried it on the real parallel machine (Multi-PSI with 64 processors). It seems that because the program forks too many jobs to processors, the "while table" overflows. Unfortunately, all the Multi-PSI machines were scheduled for other users when I was free in the last two days, so I didn't get chance to try it on Multi-PSI.

I had some wonderful discussoins with Wada-san about load balance and he helped me running my program on the Pseudo and real parallel machine. At 4:30 Friday afternoon of my last working day, the program works on REAL Multi-PSI machine with 64 processors.

I hope this work will be continued as a useful application of KL1 and PIM with ICOT.

My First Reactions of KL1 and PIM Project

Don't take it seriously.

KL1 is a realistic extension of GHC, which uses guard, the only mechanism, for synchronization and data communication. The basic idea is that the guard of the clause has to be satisfied before the body of the clause can be executed, but the satisfaction of guard cannot instantiate any variables in the goal; otherwise the clause is suspended (until these variables are instantiated by execution of some other clauses). Thus synchronization is insured. The communication among processes is achieved by shared variables among subgoals. GHC provides a very simple and elegant solution to synchronization of AND-parallelism.

However, GHC/KL1 do not backtrack: if several clauses can be executed, one of them is chosen (randomly) and all the others are discarded and never tried. Nor does GHC/KL1 provide a direct way of OR-parallelism. Thus, the programs written in KL1 must be "deterministic". If OR-parallelism exists, it may be transferred into AND-parallelism by explicit iteration via stream parallelism. If backtracking is essential, it is transferred to another KL1 program by skillful and not-so-easy-to-understand continuation-based method (by Ueda).

But backtracking is an important feature of logic programming. It frees users from specifying the explicit and deterministic behaviors of the programs, leaving control to search and backtrack. So I guess it may be difficult to use KL1 for the applications such as (heuristic) search and backward-chaining reasoning. Why not provide backtrack in (one version of) KL1?

Independent choices exist in many AI problems, and large speed-up can be obtained via OR-parallelism. Nevertheless, using parallel machine with powerful processors (such as the ones in PIM and Multi-PSI) for mere AND-parallelism seems to over-kill. AND-parallelism can contain a lot of synchronization and communication among processors (subgoals on different processors). Yet the computation of each subgoal in each processor is really simple: pick a clause, check the guard, unify, and generate new processes (and pass them to new processors). Assume we have as many processors as we want, and we distribute each process to a different processor. AND-parallelism would have a small speed-up (compared with the number of processors taken) in general. Communication overhead would be too large. So each processor should be assigned many processes (jobs) and the communication among processors should be minimized. In this case why not create multiple environment and do OR-parallel in each processor, utilizing the power of powerful processors fully?

There must be many very good reasons why backtrack and OR-parallelism are not included and implemented in KL1. On the other hand, if KL1 was designed to be a *high level* language towards application of AI, it should be at least as powerful as Prolog. Besides, I think the choice of the language is *critical* for a long term project. It might be a better idea to have several versions of KL family around, and to see which one is better, or which one suits for which applications.

Academic Activities Outside ICOT

On Feb 19, I spent a full day visiting Professor Fumio Mizoguchi at Science University of Tokyo. I gave a talk (same as the one given at ICOT) in the seminar, and visited several labs of Prof. Mizoguchi. His students demonstrated some of their learning programs that extract qualitative difference from scene inputs. On Feb 24, I visited Masa Numao at Tokyo Institute of Technology. I discussed many technical details of inventing new terms with him and his students. On Feb 28, I had a informal visit at University of Tokyo, visiting some labs on medical image processing.

Acknowledge

I deeply thank Dr. Furukawa who invited me to visit ICOT, Prof. Mizoguchi who wrote the invitation letter for me. Dr. Iwata who spent a great deal of efforts on my visa, and Miss Karakawa for making many arrangement for me and taking me to Science University of Tokyo. I benefit a lot from the discussion with researchers at ICOT. ICOT provides a very stimulating and cooperative environment for conducting research.

CURRICULUM VITAE

NAME: LING, XIAOFENG (Charles)

RANK: Assistant Professor, The University of Western Ontario

TENURED: 1989.9.1 - 1995.6.30

FULL-TIME: Yes

DEGREE	UNIVERSITY	DEPARTMENT	YEAR
Ph.D.	U of Pennsylvania, 1985.9-1989.8	Comp. & Inf Sci.	1989
M.S.	U of Pennsylvania	Comp. & Inf Sci.	1987
Sc.	Shanghai Jiao-Tong U, 1981.9-1985.6	Computer Science	1985

EMPLOYMENT HISTORY

DATE	RANK & POSITION	DEPARTMENT	INSTITUTION
1989-prsnt	Assistant Professor	Computer Science	U of Western Ontario
1986-1989	Research Fellow	Computer Science	U of Pennsylvania
1988-1989	Software developer		DAISI Electronics

HONOURS

IJCAI-91 Travel Grant
Univ of Penn, Research Fellowship, 1986-1989
Shanghai Jiao-Tong U., Top Student Honour and Fellowship, 1982-1985

DEPARTMENTAL SCHOLARLY AND PROFESSIONAL ACTIVITIES:

Colloquium Chairman
Member of the PhD Committee, former member of The Graduate Committee,

GRADUATE SUPERVISIONS:

	SUCCESSFULLY COMPLETED	IN PROGRESS
A. DOCTORAL THESIS		
B. MASTER'S THESIS	1	1
C. POST-DOC/RESEARCH FELLOWS		1
D. PH.D. COMMITTEES SERVED ON		1
E. UNDERGRADUATE THESIS (490Y)		4

PROGRAM COMMITTEE: CSCSI Machine Learning Workshop

PAPER REVIEW FOR CONFERENCE/WORKSHOP/JOURNAL: about 20