Computational Logic in Australia*

Australian research institutions have a long and distinguished tradition in logic related ventures. This commitment continues to be very strong today among a large and diverse research community. The work being performed extends across all areas of logic into the opening up of new fields of endeavour. It ranges from the classical to the non-classical; theoretical to the practical. The strength of this work is attested to by a recent issue of the journal Logique et Analyse [1] devoted to contemporary logical research in Asutralia.

In this article we highlight some of the institutions involved and projects currently being undertaken inlogic related research (especially computational logic). The article has been put together from contributions by researchers at the respective institutions. These contributions have been edited for clarity and uniformity. Any queries regarding content should be addressed to the relevant institution.

- **Automated Reasoning Project, Australian National University**

The Automated Reasoning Project (ARP) is a department in the Research School of Information Sciences and Engineering at the Australian National University in Canberra. It is a small group conducting research into pure and applied logic, particularly non-classical logics like relevant logics, modal logics, substructural logics and display logic. Its other main emphasis has been on automatic theorem proving, both for classical logic and for certain non-classical systems. More recent research involves problems from Artificial Intelligence like propositional planning, constructive negation for knowledge representation, a proof theory for description logics, and the use of paraconsistent logics in belief revision.

Enquiries from prospective students or visitors are welcomed. The ARP has a visitors program which enables scientists from Australian or overseas institutions to spend time in the Project. Funds are available to cover the expenses of selected visitors. They also run an annual Logic Summer School which is intended for beginning graduate or advanced undergraduate students typically from Computer Science, Mathematics or Philosophy backgrounds. Follow the links from their home page `http://arp.anu.edu.au` or send email to `Timothy.Surendonk@arp.anu.edu.au` for information about PhD scholarships, the Summer School, and other matters of relevance to prospective students or visitors.

- **Australian Artificial Intelligence Institute**

The design of systems that are required to perform high-level management and control tasks in complex dynamic environments is becoming of increasing commercial importance. Such systems include the management and control of air traffic systems, telecommunications networks, business processes, space vehicles, and medical services. Experience in applying conventional software techniques to develop such systems has shown that they are very difficult and very expensive to build, verify, and maintain. Agent-oriented systems, based on a radically different view of computational entities, offer prospects for a qualitative change in this position.

A number of different approaches have emerged as candidates for the study of agent-oriented systems One such architecture views the system as a rational agent having certain *mental attitudes* of Belief, Desire and Intention (BDI), representing, respectively, the information, motivational, and deliberative states of the agent. These mental attitudes determine the system's behaviour and are critical for achieving adequate or optimal performance when deliberation is subject to resource bounds.

The research and development work in the broad area of BDI agents being carried out at the Australian Artificial Intelligence Institute (AAII) can be divided into three broad categories:

- *Foundations*. This strand of research examines the logical foundations of the mental attitudes of beliefs, desires, goals, intentions, plans, commitment, obligations, etc., and the processes, such as deliberation, means-end reasoning, and reconsideration that operate on these mental attitudes. AAII has actively pursued this area by working on the goal of formalizing various mental attitudes and processes as building-blocks that can be used by others as a framework for extension and refinement.

*Edited by Maurice Pagnucco, Knowledge Systems Group, Department of Artificial Intelligence, School of Computer Science and Engineering, University of New South Wales, NSW 2050 Australia. Email: morri@cse.unsw.edu.au.

- *Systems*. This involves building tools that helps one to design, develop, and deploy BDI agents. The Procedural Reasoning System (PRS) and its successor dMARS are some of the exemplars in this category. AAII has pioneered the development of dMARS, a C++ based software development environment for building and testing BDI agents. The system development effort has been supported by the foundational research and research into areas that bridge the gap between these two categories, such as abstract architectures for building BDI agents and agent-oriented languages that capture the essential features of dMARS. The system has also been continuously refined based on a number of agent-based applications that have been built using dMARS as the underlying system.
- *Applications*. This strand of development involves close interaction with end-users in specific domains, such as air-traffic management, business process management, air-combat modelling, etc. to develop and deploy real-world solutions to problems that can be solved better by using BDI agents. AAII has worked with a number of commercial clients to develop application systems targetted for specific problems using dMARS. This has provided important feedback for the further development of dMARS and raised interesting questions for foundational research.

The success of AAII, both in terms of its fundamental contributions to the foundational research and its commercial success in deploying systems for clients, is due to the healthy interaction between all of the above areas of research and development. For further information on AAII's specific achievements in these areas visit `http://www.aaii.oz.au/`. For some recent publications, refer to [9, 10, 24, 25, 28].

• Griffith University

Computational logic related research at the School of Computing and Information Technology spans a wide spectrum of areas including knowledge representation, non-monotonic reasoning, belief revision, deductive databases, logic-based approaches to rational agency, constraint logic programming and parallel/distributed approaches to logical inference.

Representation and reasoning with incomplete and changing information: Grigoris Antoniou, David Billington, Aditya Ghose, Michael Maher, Abdul Sattar and Koji Tanaka (also of the University of Queensland) are engaged in projects that investigate theoretical frameworks for non-monotonic reasoning and belief revision, including efficient implementations, connections with constraint-based reasoning and probabilistic frameworks for knowledge representation, as well as practical applications in industry and commerce. Specific projects include:

- a study of the relationship between defeasible logic and established non-monotonic reasoning frameworks for logic programs (Grigoris Antoniou, David Billington and Michael Maher).
- development a domain-independent toolkit for nonmonotonic reasoning and belief revision (as part of Grigoris Antoniou's collaboration with Mary-Anne Williams of the University of Newcastle on the CIN project).
- a study of theory refinement and abstraction in the context of default reasoning (Grigoris Antoniou in collaboration with Kym MacNish of the University of Western Australia).
- investigation of paraconsistent logics and their applications (Abdul Sattar and Koji Tanaka in collaboration with Graham Priest of the University of Queensland)
- a project to develop a first-order defeasible logic (David Billington).

External collaborators who contribute to these projects include Norman Foo, Abhaya Nayak and Maurice Pagnucco of the University of New South Wales, Randy Goebel of the University of Alberta, Edmonton, Canada and Scott Goodwin of the University of Regina, Canada.

Deductive object-oriented databases and languages. Rodney Topor and Peter Bernus have been working with Chris Higgins and Gill Dobbie (of Victoria University, Wellington, New Zealand) on the foundations of deductive object-oriented databases and languages. Chris Higgins has developed a detailed meta-logic theory in which objects are represented by theories. Rodney Topor and Gill Dobbie have developed an alternative standard-model semantics that naturally describes concepts such as method overriding and multiple inheritance.

Formal architectures for rational agents. Abdul Sattar and Aditya Ghose are investigating how rational agents (taking the notion of a *strong* agent) revise mental attitudes such as beliefs, desires and intentions in the context of a formal model of rational agency called the BDI (or Belief-Desire-Intention) architecture. This research involves collaboration with Wayne

Wobcke of the University of Sydney and Anand Rao of the Australian Artificial Intelligence Institute.

Parallel processing of logic programs on distributed systems. This project takes advantage of highly scalable parallelism in distributed systems to speed up logic-based artificial intelligence application systems. Abdul Sattar, Chengzheng Sun and Zhiyi Huang are involved in a project which aims to investigate execution models, abstract machines, and implementation techniques for building parallel logic programming systems efficiently on general purpose distributed systems. Prototype systems are being designed and developed to demonstrate and evaluate the efficiency and effectiveness of this approach in supporting parallel hypothetico-deductive systems and constraint satisfaction problems.

Some of this research is carried out under the aegis of the Knowledge Representation and Reasoning Unit (KRRU) (URL: `http://www.cit.gu.edu.au/~sattar/KRRU.html`). More information on computational logic related research and teaching activities can be found by following links from the home page of the School of Computing and Information Technology, Griffith University (URL: `http://www.cit.gu.edu.au/`).

• James Cook University

The TPTP Problem Library. The TPTP (Thousands of Problems for Theorem Provers) is a library of problems for automated theorem proving (ATP) systems, for the clause normal form (CNF) of first order logic. The TPTP is comprehensive and thus provides an overview of, and a simple unambiguous reference mechanism for, ATP problems. The TPTP problems are presented in a specifically designed, easy to understand format, and automatic conversion to other known ATP formats is provided. TPTP is the joint work of Geoff Sutcliffe at the Department of Computer Science, James Cook University, and Christian Suttner at the Institut für Informatik, Technische Universität, Munich. TPTP may be accessed from the WWW: `http://www.cs.jcu.edu.au/ftp/users/GSutcliffe/TPTP.html` or `http://wwwjessen.informatik.tu-muenchen.de/~tptp/`.

Prolog-D-Linda. Prolog-D-Linda is an embedding of the Linda paradigm into Prolog. The original motivation for embedding Linda into Prolog was the naturalness and ease with which it could be done. The system has developed from the original implementation (in muProlog) that used a single processor to run all the application processes and a centralised tuple space, to the current version (in SICStus Prolog) that supports applications running over an internet of processors with a distributed tuple space (hence Prolog-D(istributed)-Linda). Many aspects of the development path have been influenced by experience gained in the development of two non-trivial applications: Heterogeneous Parallel Deduction Systems and Parallel Genetic Algorithms. The viability of Prolog-D-Linda as a practical parallel logic programming language has been illustrated by the success of these applications. Prolog-D-Linda is available on the WWW: `http://www.cs.jcu.edu.au/ftp/users/GSutcliffe/Prolog-D-Linda/Prolog-D-Linda.tar.gz`

Heterogeneous Parallel Deduction Systems. A Heterogeneous Parallel Deduction System (HPDS) has multiple deduction components, each of which executes independently on a separate computer. Each component runs a different ATP system, but the components cooperate by distributing results to each other. The HPDSs developed so far have exhibited significant speed improvements when compared to corresponding sequential ATP systems. The speed improvements obtained are largely due to cross-fertilisation between the various components. The HPDSs are being built using the parallel programming language Prolog-D-Linda. The use of Prolog-D-Linda for implementation has made the HPDSs highly modular and easy to modify, thus making further development possible.

• Knowledge Systems Group

The Knowledge Systems Group (KSG) conducts research and teaching in knowledge representation and comprises researchers from three universities in the greater Sydney area: University of New South Wales (School of Computer Science and Engineering), Macquarie University (Department of Computing) and University of Western Sydney (Department of Computing). Its research ranges from the passionately theoretical to the intensely pragmatic. KSG has hosted, and will continue to host, distinguished local and international visitors from kindred research institutions. Current research projects include:

MACQUARIE UNIVERSITY

Assessing Theories of Action. In Reasoning about Action, a new, more systematic methodology has recently been developed that provides a framework for formally assessing theories of action [27, 12]. In the spirit of this new methodology this project aims to evaluate two classes of solutions to the frame problem that to a large extent have dominated the research in the area, namely, those that capitalize on the concept of minimal change, and those that exploit the concept of causality. The results of this investigation will highlight the strengths and weaknesses of the two classes, and could lead to the design of better hybrid solutions.

Software Requirements Evolution. The aim of this project is to model the process of changing software requirements using concepts and techniques from Belief Revision and Nonmonotonic Reasoning, with the intent of developing tools to support reasoning about requirements evolution. This project is undertaken in collaboration with the Joint Research Centre for Advanced System Engineering (JRCASE) at Macquarie University.

Temporal Databases. The ability to model the time dimension of the real world is essential in many computer applications such as banking, inventory control, airline reservation systems, etc. The main objective of this project is to investigate the utility of a temporal logic based approach for representing, manipulating and querying time-varying information, and to implement a prototype temporal database system with the purpose of providing feedback to the basic research underlying it.

Temporal Logic Programming. The concept of time is of central importance to an increasingly wide range of applications in computer science and artificial intelligence, including planning, ecological modelling, knowledge-based simulation, temporal reasoning and so on. Therefore many logics incorporating temporal notions are being developed and applied. This project is concerned with investigating the basis of logical representations incorporating temporal notions which can be directly executed.

UNIVERSITY OF NEW SOUTH WALES

Belief and Ontology Revision. Belief change logics allow us to accommodate new information in static as well as dynamic worlds, to reason about hypothetical situations and, in general, perform commonsense reasoning in a consistent manner. Typical problem areas currently under investigation include: iterated belief change, propagation of conditional beliefs, abductive belief change, belief base dynamics, implementation of belief revision systems and ontology dynamics — addition and deletion of basic concepts (that an agent uses to model its world) as a result of change in the agent's linguistic repertoire.

Events and Actions. Research is focussed on two main areas. Firstly, the use of discrete event simulation methodologies with current reasoning about action methodologies. Secondly, the use of belief revision techniques and implementations in reasoning about systems dynamics (in collaboration with Mary-Anne Williams of the Department of Management at the University of Newcastle).

Robotics (in collaboration with Microsoft Research Institute). Within AI there has already been much research on the components of intelligence, such as planning, reasoning about action, knowledge representation, and natural language understanding and generation. We are now beginning to see work done to integrate these components into useful agents. This project will combine theoretical research from KSG on logics of actions with natural language processing from MRI into a real-world agent: a general purpose office robot which can act as a guide or a courier.

Theoretical Terms. Theoretical terms are terms which do not refer to observables. Yet, such terms form an integral part of most scientific theories. The problem, then, is to formally justify the use of such terms and to determine the senses in which such terms are used correctly.

UNIVERSITY OF WESTERN SYDNEY, NEPEAN

Logic Programming and Knowledge Representation. This project aims to extend the semantics of declarative logic programming to capture causality, priority, and defeasibility in knowledge representation, and to investigate the foundational relationships between logic programming and nonmonotonic reasoning.

Reasoning about Change. This project explores principles of state change in reasoning about action and knowledge systems update. The project studies the related issues of resolving the frame, ramification and qualification problems in reasoning about action and change from foundational and computational viewpoints.

More information is available from the KSG web pages: `http://www.cse.unsw.edu.au/~ksg/` or email: `norman@cse.unsw.edu.au`.

- **Macquarie University**
  see Knowledge Systems Group.

- **Monash University**

The Department of Computer Science has three projects in the field of computational logic.

Constraint Programming. Constraint programming and in particular constraint logic programming is an outgrowth of logic programming research. One major application of constraint programming is to solve complex combinatorial optimization problems. Because of the industrial importance of such problems, there is considerable commercial interest in this new language paradigm. One of the earliest constraint logic programming languages, CLP(R), was developed at Monash University in 1987. Since that time there has been a history of collaboration with Melbourne University. Current research is into the application of analysis and transformation techniques to improve the performance of CLP(R) and into automatic parallelisation of CLP languages.

Specification of Visual Languages. Following Chomsky, grammatical formalisms for specification of string languages have been widely studied. Less effort has been devoted to specification techniques for two-dimensional languages, such as mathematical notation, state-transition diagrams or house plans. One promising technique is the use of constrained multi-set grammars, in which productions specify rewriting of multi-sets of attributed symbols. Current research is into the automatic generation of sophisticated visual interfaces from constrained multi-set grammars, and into their formal properties and relationship with classical and linear logics.

Extracting programs from proofs. Since Gentzen's work in the 1940s many people have developed systems for extracting programs from proofs in formal logic. The present project aims to develop this in such a way as to be able, ultimately, to take proofs written in (mathematical) English and translate them (with as little human interaction as possible) into programs. The project blends techniques of formal logic with an analysis of mathematicians' behaviour.

More information about these projects can be obtained by contacting Prof. John Crossley (`jnc@cs.monash.edu.au`) or Dr. Kim Marriott (`marriott@cs.monash.edu.au`).

- **Queensland University of Technology**

The Intelligent and Cooperative Information Systems (ICIS) research centre runs a project on knowledge discovery. Logic is currently being applied to the information retrieval problem:

- model preference logic for reasoning with user preferences for information.
- plausible inference (both symbolic and probabilistic) is being investigated for realising information retrieval as an inference process between a document characterization and the user's request.
- information-based logics are being examined in order to develop an underlying theory for information discovery (e.g. on the internet).

ICIS is attached to the School of Information Systems at the Queensland University of Technology. For further information consult `http://www.icis.qut.edu.au/kdrg/`.

- **Royal Melbourne Institute of Technology**

The computer science department has an Artificial Intelligence Research Group with a number of researchers working on logics for knowledge representation, and on Lygon, a programming language for linear logic. There is work on description logics, conceptual graphs, logics for reasoning with uncertainty, and modal logics of belief, desire and intention for agent-oriented systems. There is also work on proof-theoretic analyses of various logics, including linear logic and other resource-sensitive logics.

Description logics are languages based on a subset of first order logic, with a syntax specialised for describing concepts and objects. They focus on inferences which use subsumption checking, in particular classification. The work in description logics involves theoretical work on extending the core of description logics to include the ability to represent and reason about composite objects, and the ability to represent and reason about defaults. This theoretical work has been applied to the application domains of agent-oriented systems, document management and information retrieval, and medical diagnosis.

Conceptual graphs are a graphical notation for logic. A conceptual graphs workbench, Peirce, is being developed which supports conceptual graph theory. A major part of the research is development of classification techniques for logical formulae and the scalability of knowledge retrieval. The workbench is being used for intelligent knowledge retrieval of text where the text is modelled using conceptual graphs.

Work has been done previously by members of the department on *logics for reasoning with defaults*. In particular, a logical system was developed making use of prototypical specifications. This work was previously applied to medical diagnosis, and work is being done to integrate it with description logics. Simplified systems of logical reasoning with defaults under real-time constraints, but with interactive capabilities (so assumptions can be checked) are being explored in domains such as agent systems in dynamic worlds, e.g., soccer playing teams of agents.

There is joint work with the Australian Artificial Intelligence Institute (AAII) on developing modal logics of belief, desire, intention, and commitment. Such logics are used to formally model and specify reactive agent systems, particularly those similar to the dMARS system implemented at AAII. Current work involves investigating non-classical frameworks, modelling contextual and situational dependence of the attitudes, and use of dynamic and process logics for modelling plans.

There is work underway on proof-theoretic analyses of various logics, including linear logic, affine logic, relevant logic and various inference systems for classical and intuitionistic logic. A significant factor in such analyses are the completeness properties of search strategies, and the permutation properties of the proof systems, ie the ability to alter the order of the rules in a proof whilst maintaining the same premises and conclusion. Work is also underway on a framework for defining strategies which will allow a smooth integration of different search methods such as depth-first, breadth-first and iterative deepening search, and on computational models for forward-chaining reasoning.

The logic programming language Lygon has also been developed by members of the group. This is a language based on linear logic, which resulted from earlier work on proof-search strategies, and incorporates features such as global variables, mutual exclusion and a notion of state in a purely logical way. This language has been applied to problems such as bin-packing, graph search and concurrent programming tasks. Work is also underway on a declarative debugger for Lygon and other linear logic programming languages.

Work has been done previously by members of the department on the design, implementation and applications of deductive database systems, and in particular the Aditi system. Aditi incorporates a bottom-up execution model together with support for database functions such as indexing, joins, projections and query languages, as well as an interface to Nu-Prolog, to allow top-down processing if desired. The work on Lygon and Aditi is being undertaken in collaboration with the University of Melbourne.

Further information is available from the research group's web page: `http://www.cs.rmit.edu.au/ResearchPrograms/ResearchGroups/aigroup`.

- **University of Adelaide**

DEPARTMENT OF COMPUTER SCIENCE

Lattice-based Belief Revision. Chris Nowak has developed an important bridge between theory and recent practical work in lattice and order theory. Nowak works in the area of lattice-based belief revision and so combines order-theory and symbolic logic. He has developed a sound and complete logic to talk about agent beliefs that is claimed more computable than many other approaches to belief revision. The lattice theory group at TH Darmstadt in Germany led by Prof. Rudolf Wille have been collaborating with Nowak. There is a convergence between Prof. Berhard Ganter's work at TH Dresden on theory organisation and Nowak's conceptual reasoning framework.

Computational Design And Constraint Logic Programming. Andrew Burrow's work is involved with the Software Environment for Early stage Design support (SEED) project driven by a computational architect Dr. Rob Woodbury, formerly from CMU now at The University of Adelaide. His objectives are to re-engineer SEED using order-theoretic principles to store and retrieve partial building designs. In SEED, design is seen as the process of search, with a design being transformed by an architectural grammar and further constrained by functional constraints, imposed by a building specification; and aesthetic constraints, adjudicated by the human designer. The SEED project lends itself to a representation language called feature structures and these can be seen as a subset of conceptual graphs with desirable computational properties. The work also dove-tails into constraint logic programming. Indeed, the CLP(R) programming environment has been a principal delivery tool for previous versions of SEED. Importing constraint logic programming techniques is a natural way of structuring the design and case-store problem in SEED. Comparing an existing design with a previous case-store becomes a partial unification problem well-suited to constraint logic programming and solid modelling. The research is a good example of the application of formal approaches to the logical representation of constraints and practical problem solving.

Visit `http://www.cs.adelaide.edu.au/` for further information.

DEPARTMENT OF PHILOSOPHY

The following three research projects have been undertaken by Chris Mortensen, Reader in Philosophy, with the assistance of Steve Leishman.

Paraconsistent logic is a rapidly growing area which studies the phenomenon of inconsistency-tolerance. Various classes of paraconsistent logics have been identified, for example the relevant logics of Anderson and Belnap, and the C-logics of da Costa and the Brazilian school. But recently there has emerged another very natural class, arising from topology. While it is well known that the logic of the open sets on a topological space is intuitionist, it is not so well known that the logic of closed sets is paraconsistent. Furthermore, it is easy to construct both open sets and closed sets from a partial order (open = hereditary sets, closed = their complements). Hence, this project computed all the non-isomorphic posets (to size 6), computed the open sets and the closed sets, and then presented them as intuitionist and paraconsistent logics. (See [21] for further details.)

Recent work leading to the award of a DSc to B. S. Niven has seen the application of logic to *ecology*. Niven formalised a general framework for ecology in first order logic, then applied it to the study of 27 species. Mortensen then reconstructed Niven's first order framework in PROLOG. This is an ongoing project which awaits proper funding. (See [20] for further details.)

Applying paraconsistent methods to control systems has produced some initial promising results. The idea has been to treat a malfunctioning machine by forming an inconsistent software model of the situation which incorporates elements both of the predicted behaviour and the observed behaviour. It has been shown that there are ranges of the parameters in which the inconsistent controller can return the system to well-functioning, albeit at a reduced level. This work is still proceeding, again at a slow pace.

For further information contact Chris Mortensen (`cmortens@arts.adelaide.edu.au`).

• **University of Melbourne**

A number of projects are currently being undertaken in the Department of Computer Science.

Constraint and Logic Programming. Programming language research in the department covers the study of semantics, design and implementation of high-level languages, as well as programming methodology. Important programming languages have their roots here, including MU-Prolog, NU-Prolog, CLP(R), Lygon and Mercury.

Analysis, Transformation and Debugging of Programs. High-level programming languages often have properties that allow the automatic generation of an efficient program from a less efficient version. This is the idea behind program transformation. Most useful transformations, however, require a preceding program analysis to discover pertinent facts about a program's runtime behaviour.

Abstract interpretation is a language-independent theory for static analysis of programs. This theory plays a prominent role in much work in the group, since most sophisticated implementation ideas require static analysis. The work contributes to the theory generally, as well as its applications in functional and logic programming, including sophisticated program transformation.

Techniques are also being developed to dramatically reduce the time and effort spent on debugging. Traditional debugging techniques are based on following the steps of program execution. With high-level languages such as logic programming languages, the sequence of execution steps can be very complex.

Declarative debugging is an interactive technique that pinpoints errors using programmers' responses to queries. The program, as a logical description of what is to be computed, is held up against the programmer's replies. By asking the programmer questions or using a formal specification, the system can identify precisely where in a program a bug is located. The research into declarative debugging has improved the original algorithms for debugging Prolog, and extended the ideas to functional and object-oriented languages.

Constraint Programming. Though constraint programming is still the subject of intensive research, it has considerable industrial usage, since it is clear that this emerging technology solves difficult problems quicker and with less development time than other approaches. Constraint programming is being used for solving difficult planning problems in large corporations such as manufacturers Michelin and Dassault, the French railway authority SNCF, airlines Swissair, SAS and Cathay Pacific, and Hong Kong International Terminals, the world's largest privately-owned container terminal.

Constraint logic programming (CLP) is the most developed of the constraint programming paradigms. CLP languages are a natural combination of logic programming and constraint programming, both of which deal with relations. In a short time, CLP has become industry's preferred constraint programming tool. CLP is ideal for interactive mathematical modelling work and for expressing complex combinatorial optimisation problems. The founding work on CLP languages was completed at the University of Melbourne and Monash University in 1987. One of the first CLP language implementations, CLP(R), which deals with constraints over real linear arithmetic, was developed at Monash around this time. Collaboration with Monash University on the application of analysis and transformation techniques to improve the performance of CLP(R) continues.

Logic Programming Techniques. Though the logic programming paradigm has been in use for over twenty years, the programming methodology continues to be enriched. Work on skeletons and techniques aims at systematic development of logic programs for both novices and experts. Related to this are higher order styles of programming.

Experimentation continues with a variety of techniques more familiar in lazy functional, object-oriented, imperative or relational database languages. Many of these have been adapted to logic programming, sometimes using extensions to Prolog. Programming methodology has motivated other extensions to Prolog, including types, modes and new pruning operators.

Lygon. Lygon is a logic programming language that is based on linear logic. However, a programmer does not need to know linear logic to be able to use Lygon. The language's features include a clean declarative notion of state and the ability to express problems involving concurrency. In addition, the availability of use once predicates allows simple solutions to problems such as graph manipulation. The design of Lygon and its prototype implementation, which allows experimentation with this promising new programming language paradigm, is joint work with the Royal Melbourne Institute of Technology.

Mercury. Mercury is a new, purely declarative programming language intended to make application programmers more productive. It provides the support that teams of programmers need when building large programs. Mercury's expressiveness allows problems to be described much more succinctly than in C, Ada and other imperative languages.

Mercury's efficient implementation avoids the speed penalties associated with previous logic programming languages. Mercury's strong type, mode and determination systems pinpoint almost all program errors at compile time. Its design includes a new and relatively simple execution model that takes advantage of the information these systems provide to generate very efficient code. A prototype compiler which uses this execution model to generate portable but fast C code has been developed.

Although Mercury programming philosophy differs considerably from that of constraint programming, current work investigates the possibility of producing a potentially very powerful tool using the Mercury approach to data structures and the CLP approach to constraints.

Deductive Databases. Current relational database systems are limited in their expressibility, making application development difficult and expensive. Deductive database systems avoid this problem by supporting a more expressive language allowing complete applications to be written using the database language.

A second generation deductive database system called Aditi is currently being developed. The query language used for this system is an extension of Mercury. Mercury supports strong types, so database programs are generally robust and less prone to error. Aditi supports function symbols, recursion and explicitly locally stratified negation. Unlike most prototype deductive database systems which have been built, Aditi is a disk based database system. It is designed to support transaction processing, hypothetical reasoning and many well-known optimisation strategies. Aditi fully exploits parallel facilities provided by the underlying machine. The current implementation is targeted to shared memory multiprocessor machines running UNIX. The Aditi project also has participation from the Royal Melbourne Institute of Technology

Further information is available from the department web site: `http://www.cs.mu.oz.au/`. (See also [2] from which some of this material has been taken.)

- **University of New South Wales**

Past work in the School of Computer Science and Engineering has included foundational research on machine learning in a logic programming framework as part of Claude Sammut's MARVIN system. This was an important influence on the development of Inductive Logic Programming (ILP) by Muggleton and others. This discipline rests on the view of induction as the inverse of deduction. Therefore automation of induction or learning can be approached by inverting deductive inference, most notably forms of resolution theorem-proving.

A current project aims to develop methods for discovery of knowledge in databases that cannot be discovered by traditional machine learning methods. This is to be achieved by coupling inductive logic programming with deductive database technology. The main focus of the work so far has been been with the machine learning part. The next phase will involve integration of this work with deductive databases.

A major focus of this project is to deliver the benefits of our theoretical results to practical application. To that end, we are developing software tools for data mining that provide a range of machine learning methods for exploratory data analysis. This is novel in that all the tools are integrated into a logic programming environment that permits one machine learning algorithm to invoke another, depending on the domain.

This work was the subject of two invited talks by Sammut, one at the International Workshop on Multistrategy Learning held in the USA in May 1996 and the second at the Japanese Workshop on Inductive Logic Programming in September 1996. Part of the novelty of this software is that it includes a method for unsupervised learning based on inverse resolution

(IR). The algorithm constructs a lattice of concepts or common patterns found in the data tuples. This is then searched to find the IR operator which most compresses the data. The process is iterated until no further compression can be gained. This work was presented at the First International Workshop on ILP for KDD attached to ICML 96 in Italy in July, 1996.

Algorithms have also been developed for extracting mode, type, and functional dependency information from data. This information is extremely useful in reducing the search phase. Bayesian heuristics have been developed for learning from only positive data and from both positive and negative in the presence of noise. A technical report describing these results is under preparation. Theoretical results have been obtained about learnability from multiple sources of inaccurate data and about the relationship between various generalization and specialization strategies for learning in the identification in the limit framework. This project is about to lead to industrial links with a large Australian company. Current personnel are Claude Sammut, Arun Sharma, John Shepherd, Eric McCreath and Michael Bain. For sample publications, refer to [3, 4, 15, 26] and for further information, consult `http://www.cse.unsw.edu.au/`.

see also Knowledge Systems Group.

- **University of Newcastle**

Dr. Mary-Anne Williams of the Information Systems Group in the Department of Management is involved in a number of logic related projects. These include:

- investigating the relationship between belief revision and possibilistic logic and the use this may have for transmutations of information systems.
- the use of belief change techniques in reasoning about action (in collaboration with the Knowledge Systems Group).
- developing an operational system capable of determining explanations and making predictions based on default logic and belief revision.
- developing an intelligent information management toolkit for performing nonmonotonic reasoning and belief revision (as part of the Reasoning with **C**hanging **IN**formation — CIN — project in collaboration with Grigoris Antoniou, Griffith University).

Further information can be obtained by contacting Mary-Anne Williams (`maryanne@frey.newcastle.edu.au`) or visiting `http://frey.newcastle.edu.au/~is/maryanne.html`.

- **University of Queensland**

DEPARTMENT OF COMPUTER SCIENCE

Logical approaches for Knowledge Sharing. Formal descriptions of domain models in a logic typically embed task-specific assumptions in their representations, hindering their reuse for problem-solvers other than those for which they were originally captured. This work is concerned with the application of various logical techniques to the sharing of formal domain theories between different problem-solvers.

To this end, the role of contexts as formalised objects in knowledge representation languages has been investigated including the relationships between various modal and meta-theoretic treatments. A specific investigation has been undertaken into the effects on context-naïve formulations of subsumption lattices in the framework of an order-sorted first order logic extended with multiple sort partial orders and (an essentially multi-modal) context mechanism.

Concentrating on sharing theories between the configuration and diagnosis tasks, an interesting application for the circumscription schema has been observed for providing a common semantic foundation for the two tasks, particularly when viewed as variants on finite model generation. CLP($\mathcal{FD}$) is shown to provide an effective mechanisation of the model generation tasks for configuration and diagnosis for a restricted class of constraint-based specifications for devices and components, demonstrating an interesting application of constraint logic programming techniques as well as being an exercise in designing a knowledge representation language.

Current papers are available at `http://www.cs.uq.edu.au/~nirad/research.html`.

DEPARTMENT OF COMPUTER SCIENCE, SVRC

The SVRC (Software Verification Research Centre — Australia's only Commonwealth Special Research Centre in the computing area) has substantial interests in computational logic, particularly as it applies to machine assisted reasoning about mathematical models of software and computing systems. Recent research activities are fairly comprehensively described on our web pages `http://www.cs.uq.edu.au/svrc` but in brief:

Qu-Prolog 4 is an extended Prolog designed specifically for rapid prototyping of computational logic algorithms, particularly algorithms involving rich mathematical notations — such as interactive theorem provers. Basically, whereas Prolog data objects are ground terms of a first-order, quantifier-free vocabulary, Qu-Prolog data objects are arbitrary terms of a higher order vocabulary which provides for quantification over object variables. The formal syntax of Qu-Prolog is then a (first order, quantifier-free) metalanguage in which symbolic computation on the data objects can be implemented using logic programming. After ten years research and development, Qu-Prolog is fairly mature as a vehicle for rapid prototyping of single-user systems, and current research is focussed on improvements to facilitate rapid prototyping of multi-user, distributed systems. As well as the Ergo provers mentioned below, Qu-Prolog is the implementation language of a multi-user, distributed repository manager being developed as an upgrade to our Cogito 1 formal development environment.

The Ergo family of interactive theorem provers (Ergo 4 in use, Ergo 5 under development) is implemented in Qu-Prolog. Ergo 4 is the theorem prover of the Cogito 1 formal development methodology and environment, which has undergone substantial industrial trialling. Ergo is also the theorem prover for other research projects including a refinement tool project and the CARE project which supports a separation of skills in formal development projects, so that design engineers can focus on design and theorem proving can be confined to specialists having appropriate skills. Currently Ergo is a single-user system. Future work will exploit current Qu-Prolog improvements to support multi-user, distributed theorem proving.

DEPARTMENT OF PHILOSOPHY

The Department of Philosophy at the University of Queensland has played a major role in the logic community in Australia (in particular, regarding philosophical logic), having a strong tradition in logic. Recently, research activities have been centred on non-classical logic, in particular, paraconsistent logic. Paraconsistent logic is, in nature, theoretical/philosophical, yet it has been shown to be important in computing related areas such as artificial intelligence and logic programming as well. Current research activities in the department include: paraconsistent logic and its applications, paraconsistency in nonmonotonic reasoning, paraconsistent logic and non-classical logics such as temporal logic, probability and paraconsistent logic and its applications in natural language, paraconsistent belief revision, and ontology and the analysis of conditionals. The logic courses that are taught in the department are: *classical and non-classical logics*, and *classical and non-classical metatheory* at the undergraduate level, and advanced topics of many kind at the honours level. There is also a logic honours programme that involves the computer science, mathematics, and philosophy departments. Research collaboration with other departments such as computer science and mathematics, as well as the Knowledge Representation and Reasoning Unit at Griffith University is currently active. For more details about logic related activities in the department, see the department home page: `http://www.uq.edu.au/philosophy/home.html` or contact Prof. Graham Priest (`ggp@lingua.cltr.uq.oz.au`).

- **University of Sydney**

The following projects are being undertaken by Dr. Wayne Wobcke (`wobcke@cs.su.oz.au`) in the Department of Computer Science.

Situation Semantics. This research is based on the idea that *scripts*, representations of stereotypical sequences of events, correspond to *situations*, parts of the world extended in space and time. The problem of inference using scripts is treated as one of plan recognition. Plan recognition is the inference, from a set of observations, of statements describing the plans an agent could be executing. Two theories have been developed: a monotonic theory in which plan recognition is formalized as deduction and a nonmonotonic theory in which a consequence operation is derived from a preferential ordering on the possible plans.

Situation semantics has also been used to provide an analysis of subjunctive conditionals in which the connection between the antecedent and consequent of a subjunctive conditional is an informational relation between types of facts of the kind that can be represented in a planning or inheritance hierarchy. The 'paradoxes' of conditional implication (the failure of strengthening the antecedent, transitivity and contraposition) are explained in terms of the use of defaults and exceptions in such hierarchies.

Belief Revision. This research investigates the connections between belief revision, indicative conditionals and nonmonotonic reasoning. The theory of Alchourrón, Gärdenfors and Makinson has been generalized to an account of iterative belief change operations on belief states modelled as ranked epistemic entrenchments. This is used as the semantics for a logic of indicative conditionals compatible with both extended AGM postulates and the Ramsey test, while avoiding Gärdenfors' triviality result. The extensions of a default theory are characterized in terms of a unique 'most conservative' ranked epistemic entrenchment in which beliefs are entrenched as low as possible. The additional assumptions of independence of default instances and uniqueness of names are proposed to handle first order default theories. Together with Simon Dixon of Flinders University, algorithms for implementing AGM belief change operations have been developed and implemented over the language of first order logic with equality using a resolution theorem prover.

Rational Agent Architectures. Recent work involves the formalization of the dynamics of agents' representational states in rational agent architectures. More specifically, this involves the development of a logic of intention and belief based on situation semantics and a theory of intention revision based on AGM belief change. Currently under investigation is the representation of an agent's plans using the formalism and the specification of such plans using temporal logic. This research is in collaboration with Abdul Sattar from Griffith University and Anand Rao from the Australian Artificial Intelligence Institute.

- **University of Technology, Sydney**

Algorithms and Languages. The Algorithms and Languages Group within the School of Computing Sciences is working on the exploitation of shape theory in computing. Shape theory provides a means of symbolic reasoning about data structures with very little information about how the data is stored, e.g. in a tree or an array. Among other things, we are developing automated analyses for detecting shape errors (e.g. array-bound errors) and for automatic parallelisation of programs. Currently, the group has six members (two staff, one post-doctoral fellow, and three post-graduate research students) and extensive international links. They are willing to take on more postgraduate students. For further information, consult `http://linus.socs.uts.edu.au/~shape`.

Logic of Knowledge. Dr Ron van der Meyden of the School of Computing Sciences is conducting research on applications of the logic of knowledge in distributed and multi-agent systems. The focus of the work is on algorithmic aspects of the compilation of knowledge-based programs [19, 17, 18] and on proof theories for their verification [16]. Copies of papers are available at URL `http://linus.socs.uts.edu.au/~ron`. The work is supported by a UTS Internal Research Grant.

- **University of Western Australia**

The Department of Computer Science has a number of academics with interest in computation logic: Associate Professor C. P. Tsang (applications of constructive logic and linear logic in databases, program synthesis), Dr. Kym MacNish (non-monotonic logic, temporal logic, logics for artificial intelligence) and Dr. Lyndon While (linear logic and functional programming).

Logic related undergraduate courses:

Logic Programming, 3rd year semester unit for Computer Science major students. Topics covered include 1st order model theoretic logic, Herbrand model, Horn clauses, resolution, unification, completeness, soundness, negation by failure, design of Prolog, application of Prolog.

Computational Logics, 4th year semester unit for honour students. This unit is not given every year. Topics include modelling of computational process, intuitionistic logic, constructive logic, linear logic, program synthesis etc.

Logics for artificial intelligence, 4th year semester unit for honour students. This unit will be given in 1997 for the first time. Topics include non-monotonic logic, default logic, temporal logic and their applications to AI.

Completed PhDs in the logic related areas: Dr. Dorota H. Kieronska, "A System for the Synthesis of Concurrent Programs: An Algorithmic approach to State Graph Constructions and Transformations" (passed in April 1991), Dr. Geoff Sutcliffe, "A Linear Deduction System with Integrated Semantics Guidance" (submitted and passed in 1992), Dr. Nick Lewins, " Practical Solution-Caching for Prolog: An Explanation-Based Learning Approach", (submitted and passed in 1993) and Dr. P. Rajagopalan, "A Type Theory Approach to the Specification and Synthesis of Database Models" (submitted and passed in 1994).

Current PhD projects: Dong-Tsan Lee (linear logic and databases), J. Herrman (constructive logic and program synthesis), E. Ong (linear logic and functional programming) and C. Cordingley (non-monotonic logic and machine learning).

For a sample of papers in the last five years related to computational logic, refer to [11, 13, 14, 22, 23]. For further information, see `http://www.cs.uwa.edu.au/`.

- **University of Western Sydney**
  see Knowledge Systems Group.

- **University of Wollongong**

The Department of Mathematics conducts research on illative combinatory logic, lambda calculus, weak implicational logics, BCK algebras and generalised type theories (approx 95 papers). An application resulting from this work has been an automated theorem prover for subintuitionistic logics implemented by A. Dekker called Brouwer 0.7.9. This is being extended to include full classical logic and various subsystems. This always gives a proof or guarantee of no proof.

Logic is taught as part of the undergraduate curriculum and occasional courses are offered at the honours and graduate levels. A recently graduated MSc Hons student, Ramsey Rizkalla, completed a dissertation entitled "Theorem proving algorithms using lambda calculus and combinatory logic". Visitors/collaborators over the last four years include H. Barendregt, J. R. Hindley, J. Seldin, M. Dezani, H. Ono, Y. Komori, S. Hirokawa, R. Meyer and A. Dekker.

For sample publications refer to [5, 8, 7, 6]. Contact Martin Bunder (`Martin_Bunder@uow.edu.au`) for further information.

Acknowledgements

The editor would like to thank the following people for their contributions to this article: Mike Bain, Peter Bruza, Martin Bunder, Kyle Dean, Peter Eklund, Norman Foo, Michael Georgeff, Aditya Ghose, Rajeev Goré, Barry Jay, Rex Kwok, Kim Marriott, Ron van der Meyden, Chris Mortensen, Lee Naish, Abhaya Nayak, Mehmet Orgun, Pavlos Peppas, Anand Rao, Nirad Sharma, Harald Søndergaard, John Staples, Geoff Sutcliffe, Koji Tanaka, C. P. Tsang, Wayne Wobcke and Yan Zhang.

References

[1] *Logique et Analyse*, vols. 137–140, 1992. Special double issue devoted to Contempory Logical Research in Australia edited by Chris Mortensen and Graham Priest.

[2] *Department of Computer Science Handbook*. University of Melbourne, 1996.

[3] M. Bain, C. Sammut, A. Sharma, and J. Shepherd. ReDuce: Automatic structuring and compression in relational databases. In *Proc. of ML-Net Workshop on Data Mining with Inductive Logic Programming*, July, 1996.

[4] G. Baliga, S. Jain, and A. Sharma. Computational limits on learning from multiple sources of inaccurate data. *SIAM Journal of Computing*, 1996. To appear.

[5] M. Bunder. Arithmetic based on church numerals in illative combinatory logic. *Studia Logica*, 47:129–143, 1988.

[6] M. Bunder. Lambda terms definable as combinators. *Theoretical Computer Science*, 1996. To appear.

[7] M. Bunder, H. P. Barendregt, and W. Dekkers. Systems of illative combinatory logic complete for first order propositional. *Journal of Symbolic Logic*, 58:769–788, 1993.

[8] M. Bunder, J. R. Hindley, and J. P. Seldin. On adding further forms of (χ) to weak equality in combinatory logic. *Journal of Symbolic Logic*, 54:590–607, 1989.

[9] M. P. Georgeff and A. S. Rao. The semantics of intention maintenance for rational agents. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1995.

[10] D. Kinny, M. P. Georgeff, and A. S. Rao. A methodology and modelling technique for systems of bdi agents. In W. V. de Velde and J. Perrame, editors, *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'96*. Springer Verlag, 1996. LNAI, Volume 1038.

[11] D.-T. Lee and C. P. Tsang. Solving the database update problem using linear logic. *Australian Computer Science Communications*, 18(2):131–138, 1996.

[12] F. Lin and Y. Shoham. Provably correct theories of action (preliminary report). In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 349–354, 1991.

[13] C. MacNish. Knowledge without modality: A simplified framework for chronological ignorance. In D. Pearce and D. Wagner, editors, *Logics in AI: Proc. 3rd European Workshop on Logics in Artificial Intelligence*, pages 25–35. Springer-Verlag, 1992. LNAI 633.

[14] C. MacNish, A. Galton, and J. Gooday. QSQR: Quantitatively scoped qualitative reasoning. In E. Sandewall and C. Jansson, editors, *Proc. 4th Scandinavian Conference on AI*, pages 30–37. IOS Press, 1993.

[15] E. McReath and A. Sharma. Extraction of metaknowledge to restrict the hypothesis space for ILP systems. In X. Yao, editor, *Proceedings of*

the Eighth Australian Joint Conference on Artificial Intelligence. World Scientific, 1995.

[16] R. v. d. Meyden. Axioms for knowledge and time in distributed systems with perfect recall. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 448–457. IEEE Press, 1994.

[17] R. v. d. Meyden. Constructing finite state implementations of knowledge-based programs with perfect recall. In *Proceedings of the PRICAI'96 Workshop on Foundations of Intelligent Agents*. Springer LNCS, to appear, December 1996.

[18] R. v. d. Meyden. Finite state implementations of knowledge based programs. In *Proc. of the Conf. on Foundations of Software Technology and Theoretical Computer Science*. Springer LNCS, to appear, December 1996.

[19] R. v. d. Meyden. Knowledge based programs: On the complexity of perfect recall in finite environments. In *Proc. of the Conf. on Theoretical Aspects of Rationality and Knowledge*, pages 31–50. Morgan Kaufmann, March 1996.

[20] C. Mortensen. Symbolic ecology I. Technical Reports in Symbolic Reasoning TR-SRS-4/95, Australian National University, 1995.

[21] C. Mortensen and S. Leishman. Computing dual paraconsistent and intuitionist logics. Technical Reports in Automated Reasoning TR-ARP-9/89, Australian National University, 1989.

[22] P. Rajagopalan and C. P. Tsang. Prototyping database functions using type theory. In *Proc. IEEE TENCON-93*, pages 327–330, 1993.

[23] P. Rajagopalan and C. P. Tsang. Representation and manipulation of data collections using generic types. *Australian Computer Science Communications*, 17(2):172–179, 1995.

[24] A. S. Rao. Agentspeak(l): BDI agents speak out in a logical, computable language. In W. V. de Velde and J. Perrame, editors, *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*. Springer Verlag, 1996. LNAI, Volume 1038.

[25] A. S. Rao. Decision procedures for propositional linear-time belief-desire-intention logics. In M. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents Volume II — Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages (ATAL-95)*. Springer-Verlag, 1996. Lecture Notes in Artificial Intelligence.

[26] C. Sammut. Using background knowledge to build multistrategy learners. In R. S. Michalski and J. Wnek, editors, *Proc. of the Third International Conference on Multistrategy Learning*, pages 11–16, 1996.

[27] E. Sandewall. *Features and Fluents*. Oxford University Press, 1994.

[28] G. Tidhar, A. S. Rao, and E. A. Sonenberg. Guided team selection. In *Proceedings of the International Conference on Multi-Agent Systems (ICMAS-96)*, 1996.