The Knowledge Engineering Review, Vol. 10:2, 1995, 209–211

Logic Programming and Education

ROSA MARIA BOTTINO, PAOLA FORCHERI AND MARIA TERESA MOLFINO

Istituto per la Matematica Applicata, Consiglio Nazionale delle Ricerche, Via De Marini 6, 16149 Genova, Italy (email: {bottino, forcheri, molfino}@ima.ge.cnr.it)

1 Introduction

Logic programming originated from the discovery that a subset of predicate logic could be given a procedural interpretation which has been first embodied in the programming language Prolog. The unique features of logic programming make it appealing for numerous applications in artificial intelligence, computer-aided design and verification, databases and operations research, as well as to explore parallel and concurrent programming. In the beginning, educationalists were attracted by logic programming, and Prolog in particular, for a number of reasons. Some hoped that the roots of the language in logic would make it a suitable medium to develop logic-deductive abilities; some believed that Prolog would offer many problem solving opportunities; some saw Prolog as suitable for the building by learners of database models of various school subjects. Also, logic programming was considered as an introductory programming paradigm both in secondary and university courses in computer science, as well as in other disciplines (such as cognitive science, artificial intelligence, and so on). Interest in respect of the educational applications of logic programming is confirmed by the conferences and the workshops organised in recent years in this field. Accordingly, we set up a workshop on "Logic Programming and Education" as one of the events of ICLP 94, the eleventh international conference on logic programming, one of the two major annual conferences which report recent research results in logic programming. The purpose of the workshop was to bring together people who are strongly motivated and experienced about the role and purpose of logic programming, and Prolog in particular, both in secondary school and at university level. The workshop was held in Santa Margherita Ligure, Italy in June 1994. It consisted of eleven refereed papers and one invited lecture.

2 Background

Interest in logic programming for educational purposes stems from an Imperial College project, initiated by Bob Kowalsky in 1980. The project, "Logic as a computer language for children", showed that forms of logic programming could be taught to secondary school students with interesting results (Kowalsky, 1984). Numerous research groups, mainly but not restricted only to Europe, were inspired by the project to investigate classroom uses of Prolog (see, for example, Nichol et al., 1988; Scherz et al., 1986). In spite of the initial enthusiasm for its educational potential, the efforts made to introduce Prolog in classroom practice often did not achieve the expected results because, as a rule, after the first simple examples, teachers and students found the language much too difficult to learn and use. For these reasons research was undertaken to analyse the misconceptions and problems shown by beginners in learning Prolog (Brna et al., 1990) and research projects were set up to develop suitable supports and environments for the learning of this language. Moreover, in a number of research projects the role of Prolog changed. This language is considered as an implementation tool for building classroom products, such as modelling tools, simulations, tutoring systems, and so on. All these different aspects have been looked at in the workshop. We strove to maintain in this multi-perspective approach a common set of characteristics which we consider very important: being set in school, considering teachers' as well as researchers' point of view, discussing the results of classroom experiments.

3 Main themes of the workshop

The presentations can be grouped into the following topics: logic programming in the teaching/learning of other subjects; teaching/learning of logic programming; Prolog as a tool for building teaching/learning environments.

3.1 Logic programming in the teaching/learning of other subjects

The exploration of concepts using declarative programming would appear to be a useful tool to improve learning. In particular, an experiment using a complete logic programming system (Prologb) within an introductory course in logic and discrete structures for computer science majors was presented (Neff). In the paper, inductive set definition was shown to be a useful bridge between discrete mathematics and logic programming. The pilot experiment was discussed.

Another interesting paper described an experimental approach to teaching AI at university level, based on logic programming (Lees). The author described the design of an appropriate set of software tools aimed at providing support for exploratory practical work in AI. Indications for further research were also given, on the basis of the experiment.

An experiment regarding the use of Prolog for getting high school students to develop individual expert systems in a course on AI and expert systems was also presented (Scherz, Haberman & Ragunis). The educational advantages of the proposal were discussed; students' attitude towards the experiment were analysed with respect to interest raised and difficulties faced.

3.2 Teaching/learning of logic programming

It is well known that logic programming concepts are difficult to learn for novice Prolog programmers, and steps are consequently being taken to develop effective teaching methods and tools.

As to the methods, an interesting course on Prolog for graduate students specialising in Computer Science in Education was presented, centred on concept-learning (Avrahami).

As to the tools, attention was devoted to the representation of the execution of Prolog programs. In particular, a new textual tree trace notation which explicitly represents the matching of clauses against goals, and distinguishes several types of goal failure (Taylor, du Boulay & Patel) was discussed. Such textual tree trace is intended to combine novel features with the useful ones of previous tracers. A tutor for Prolog-debugging which provides Prolog programs containing bugs to be found and corrected by the learner (Del Soldato) was also presented.

3.3 Prolog as tool for building teaching/learning environments

Several authors consider Prolog a useful language to develop educational systems.

The most interesting presentation made in the workshop within this approach discussed two intelligent tutoring systems for teaching geometry: TALC, Tuteur d'Aide Logique a la Construction, and GEOSPECIF, Specifications Geometrique, (Allen, Desmoulins & Trilling). Via these examples, the authors aimed to demonstrate that new programming constructs can grow out of efforts to design and build educational systems.

4 Invited lecture

The lecture centred on issues connected with the development of a more integrated and complete view of learning and teaching Prolog. The lecture was by P. Brna, of Lancaster University. In particular, a synergy between work in the general area of the psychology of programming and system designing was suggested. In this respect, various problems were analysed: how to develop Prolog programming environments that aid the novice-to-expert transition; how to incorporate

misconception detection tools into Prolog environments; how to incorporate higher order aspects of designing, explaining and understanding; how to exploit the power of visual programming and program visualisation.

The issues touched in this lecture were about supporting a richer model of learning to program—both in terms of the "scope" of the model (incorporating the different activities associated with programming: building, debugging, explaining, and understanding programs) and the "lifetime" of the model (the length of time over which the novice can use the environment).

5 Conclusions

This workshop constituted a valid backing to broaden discussion around the potential of logic programming for education. The audience consisted of teachers and researchers interested in the educational applications of logic programming and related cognitive problems. The numerous questions delved into many important issues: to experiment with Prolog to create a meaningful framework for students' work; to define and build valuable tools to facilitate the learning of logic programming; to explore the possibility of constraint logic programming in teaching the main concepts of logic programming; to evaluate the opportunity of inserting Prolog into multiparadigm languages; to analyse the use of tracers as tools for making the teaching of Prolog more effective; to discuss the potential of Prolog as a language for building educational software.

References

Bottino RM, Forcheri P and Molfino MT (eds), 1994. *Proceedings of ICLP'94 Post-Conference Workshop on Logic Programming and Education* GE, Italy: IMA-CNR.

Brna P, Pain H and du Boulay B (eds), 1990. "Special Issue of Instructional Science on teaching, learning and using Prolog", *Instructional Science* **19** (4/5).

Kowalski R, 1984. "Logic as a computer language for children" In: M Yazdani (ed) *New Horizons in Educational Computing* pp. 121–144. Ellis Horwood.

Nichol J, Briggs J and Dean J, (eds), 1988. *Prolog, Children and Students*. Kogan Page.

Scherz Z, Maler O and Shapiro E, 1986. "Learning with Prolog—A new approach", *Journal of Computers in Mathematics and Science Teaching* **6**(1) 31–37.