Journal of Universal Computer Science, vol. 12, no. 9 (2006), 1405-1410
submitted: 31/12/05, accepted: 12/5/06, appeared: 28/9/06 © J.UCS

Computer Science, Logic, Informatics Education

Katalin Pásztor Varga
(Eötvös Loránd University, Hungary
pkata@ludens.elte.hu)

Magda Várterész
(Debrecen University, Hungary
varteres@inf.unideb.hu)

Abstract: Our aim is to discuss what, when and, how deep logic should be taught in the computer science education in connection with the so called "Bologna process". We survey the spread of logic in the computer science education. We draw special attention to the process resulting by 1987 in a comprehensive school in the international logic research called "Computer Science Logic". It includes the investigation of research problems arising among others on such fields as programming theory, data- and knowledge base theory and, artificial intelligence as well. New directions have emerged during the problem solving, the earlier disciplines of classical logic like lambda calculus, type theory, model theory, modal logic, temporal logic has come back into the main scope. The results of these researches have a great impact on the development of computer science. The research results and the ever increasing role of logic should be obviously reflected by the education. We explain our conception concerning this issue as well.

Keywords: computer science, logic, education
Categories: F.4.0, F.4.1

1 Historical remarks

In 1988 the *Magna Charta Universitatum* issued on the occasion of the 900-th anniversary of the foundation of Bologna University summarized the mission of the universities in the modern society. In 1988, initiated by the French minister of education the *Sorbonne Declaration* set the mutual recognition of each other's diploma as a joint aim of the four signatory countries. Many other countries including Hungary accepted these ideas and expressed their willingness to join this process.

In order to emphasize their commitment the education ministers from 29 of these countries met in Bologna and undertook in a joint declaration (the Bologna Declaration, 19th June 1999) to establish a *European area of higher education* by 2010. The aim of the *Bologna Process* is to make the higher education systems in Europe converge towards a more transparent system which whereby the different national systems would use a common framework based on three education cycles - Degree/Bachelor, Master and Doctorate. Since then the involved countries have organized many high level important meetings (cf. Prague, 2002; Berlin, 2005) to revise the short term objectives of the process and to clarify some questions of the implementation.

As for the participation of Hungary in the *Bologna Process*, in 2001 the Hungarian Rectors' Conference committed itself to the change for the linear (two-cycle) higher education system and established a local *Hungarian Bologna Committee* to carry out the necessary planning and organizational tasks. The revision and reconstruction of informatics curricula have been realized under the supervision of a special (informatics) subcommittee of the Hungarian Bologna Committee. One of the possible majors, the "*B.Sc. in software engineering*" has been founded by the Faculty of Informatics of the Debrecen University.

Although the software engineering B.Sc. major can be considered in the Hungarian educational system as a direct successor of the former (six semester) "programmer mathematician" major and the developed curricula are strongly based on the long-term local experiences concerned with the more than 30 year maintenance of that major such changes in the education system always provides us the opportunity to make a big revision of the actual goals and contents. This revision has been carried out carefully concerning the role and content of mathematical logic in informatics education. Below we give an outline of the main points of our motivation and present our remarks and conclusions concerning the updated requirements and teaching program as well.

2 The logic in the 20th century and the computer science

At the beginning the century the axiomatization of the logic by David Hilbert (1921) made it possible to consider the logic as an axiomatized theory. In this approach one could prove theorems by the usual methods of mathematics. However, there weren't algorithms supporting the construction of deductions. The first significant break-through was due to G. Gentzen (1935) with the development of the natural (deduction) technique and the sequent calculus. With this, a special syntactic toolkit of the automatic theorem proving was in fact created. J. Herbrands model theoretic result (1930) was the provability of unsatisfiability over Herbrand universes which made it possible the rewriting of the first-order decision problem in a form of propositional formula by expansion over the Herbrand universe. Thus, a theorem proving problem can be reduced to the examination of formulae by a toolkit of propositional logic.

In the 1950's, when the computers became accessible, using Herbrands results M. Davis and H. Putnam elaborated the first effective computer algorithm for theorem proving. At the same time A. Newell, C. Shaw and H. Simon designed the ,"Logic Theorist" (LT) and the "General Problem Solver" (GPS) systems evoking a big impact on the contemporary artificial intelligence. The "Logic Theorist" used the so called "British Museum" algorithm[1] based on the axioms and inference rules given by B. Russell and A. Whitehead in the "Principia Mathematica".

The investigation of the unsatisfiability problem of the canonical CNF formulas and the generalization of the propositional resolution rule for first-order formulas led to the idea of the propositional logic resolution (basic resolution). J. Robinson defined the notion of the first-order resolvent in her 1965 paper, where on the base that if the basic resolvent exists, then the used two clauses have such basic instances, where a

[1] This is a low performance blind horizontal search method.

complementary basic literal pair appears. She recognized that the condition of the existence of a basic resolvent is that the original literal pair were unifiable. This way she created the first-order resolution calculus. Later many other people dealt with the development of various resolution strategies hoping the simplification of the implementation. The main strategies are semantic and linear resolution (complete calculi), linear input resolution and unit resolution (well implementable but non-complete calculi) (C.L. Chang and R.C.T. Lee 1973).

Already in the sixties another demand for the application of the logic arose in connection with the program analysis and synthesis. This meant that the properties of a program were described by logical formulas (axioms) and, one tried to answer the questions concerning the correct functioning of the program. The formal base of this approach was primarly the Hoare logic [Hoare 69]. The books [Manna, 71], [Manna, 74b], [Pnueli, 81] contain the most important results. Very useful tools are the so called question-answer-systems. The answer was a supposed consequence/inference of the formulas describing the properties of the program. These systems were endowed by a special technique which could additionally generate certain sort of answer concerning the consistence of the theorem provided the theorem itself had been proven. Here the second-order logic and even the temporal are used in formalization (C. Green, J. McCarthy 1961, Z. Manna 1974, F. Kröger 1987). The books [Cliff, 80], [Hoare, 85], [Loeckx, 87], [Goos, 03] illustrate that this is an important and developing application area of logic.

At the beginning of 1970's, using the incomplete linear input strategy R.A. Kowalski and A. Colmerauer developed the theorem proving system Prolog for the Horn clauses. The statements of a logic program are definit first-order Horn clauses. The investigation of the abilities of the Prolog system implied considerable researches. On the basis of the principle of the model given by ground atoms the development of the notion of least Herbrand model and, a certain treatment of negation became possible. It is a known result that an ordering preserving mapping defined on a complete lattice always has a fixpoint. A Herbrand interpretation is a subset of the set of ground atoms over the Herbrand universe (i.e. a subset of the Herbrand base), so the set of all Herbrand interpretations is a power set of the Herbrand base. It results that the set of all Herbrand interpretations is a complete lattice with two lattice operators and the subset relation.

A direct consequence function assigned to a logic program over this net is ordering preserving, thus it has a least fixpoint. It has been proven that this is the least Herbrand model of the logic program. In connection with it the fixed point extension of the first-order logic (S. Abiteboul, Y. Vianu, 1989) was defined playing a significant role at the DATALOG approach of relational databases and knowledge bases.

The investigation of logical deduction systems had a great impact on the development of model theory, primarily on the examination of the questions of axiomatization (H. Ebbinghaus, J. Flumm, 1999). The appearance of the functional programming languages, then the development of parallel and concurrent programming technologies lead to the introduction of new tools in the programming theory. One of the tools of such kind is the λ-calculus due to A. Curch in 1932. The λ-calculus has been used for the unified treatment of different order (null-, first-, higher)

logics (P.B. Andrews, 1986). The evolution of languages involved the evolution of various tools as π-, μ-calculus etc.

Since the sixties the toolkits of AI, programming theory, knowledge base theory have included such non-classical logical tools that the various versions of temporal logic, modal logics, many-valued logics, relevant logics, non-monotone logic, fuzzy logic.

3 The role of logic in the informatics education

At the beginning logic was only taught as a descriptive technique. By the sixties the applications of logic supposed the knowledge of large areas of logic. This implied that the basic topics of mathematical logic got included into the advanced courses syllabi. By the seventies, the logic became as a tool on that fields of informatics as the artificial intelligence, the theory of relational databases, the program analysis and synthesis. Since the end of seventies the foundations of logic has became part of the first year lectures and the teaching of those fields which play significant role in the applications has been shifted to the upper classes.

In the nineties began the publication of logic books for informatics students and experts: [Bergmann, 77], [Richter, 78], [Gallier, 87], [Schönig, 87], [Börger, 89], [Fitting, 96], [Nerode, 97], [Huth, 04] etc. Even the book of the present authors published in 2003 [Pásztor-Varga] can be added to the above list. At the same time there were published big summarizing handbooks as well [Barr, 81], [Bledsoe, 84], [Boyer, 88], [Abramsky, 92], [Gabbay, 93], [Agostino, 99], [Krantz, 02].

4 What, when, how many?

In the present phase of the development of the informatics it is quite clear outlined the knowledge of which part of the logic is important for a graduated computer expert. In the basic curriculum and training should be taught the propositional and first-order logic in a language oriented approach: language, alphabet, syntax, semantics, important rewriting rules, semantic consequence notion, the theorem proving problem, the decision problem, the semantic solution of the theorem proving, the syntactic treatment of the logic, the principle of deduction systems. The resolution calculus and an arbitrary method connected by the Hilbert system as the most important two of known deduction systems should be included. It should be made understand the notions of correctness and completeness. This way we provide the necessary foundation to the introduction of logics (temporary, Hoare, unity, fixpoint, and description logic [Küsters, 01]) used in teaching of various fields of computer science. This can in certain extent prepare the teaching of data- and knowledge bases, programming theory and artificial intelligence. This is a good base to introduce non-classical logics. After such a logic foundation it will be possible to introduce the teaching of Prolog as a functional programming language.

Courses focusing on database theory or artificial intelligence require not only fundamental logical knowledge. The future doctoral studies should also be founded here. We have looked up the relevant curricula of some authoritative oversee

universities concerning the logic in the master education and we have concluded that the most important fields are

- the theoretical background and methods of theorem proving,
- the problems of formalization and axiomatization (the problems of axiomatized theories),
- logic programming and its theoretical background (fixpoint logic, Herbrand model of a program)
- λ-calculus in the frame of functional programming,
- non-classical logics (temporal logic, many-valued logic, fuzzy logic, description logic etc).

All of the above topics are important to teach in the frame of a master program. At the same time the teaching of several topics (cf. non-classical logic) at undergraduate level are also justified at least as an elective course.

References

[Abramsky, 92] S. Abramsky, D. M. Gabbay and T. E. S. Maibaum, eds., Handbook of Logic in Computer Science, vols. 1-4, Oxford University Press, Oxford 1992-95.

[Agostino, 99] M. D. Agostino, D. M. Gabbay, R. Hähnle and J. Possega, Handbook of Tableau Methods, Kluwer, 1999.

[Barr, 81] A. Barr and E. Feigenbaum, eds., Handbook of Artificial Intelligence, vols. 3, Heuristic Press, Stanford, 1981.

[Bergmann, 77] E. Bergmann and H. Noll, Mathematische Logik mit Informatik-Anwendungen, Springer-Verlag Berlin New York, 1977.

[Bledsoe, 84] W. Bledsoe and D. Loveland, Automatic Theorem Proving after 25 years, American Mathematical Society, Providence, 1984.

[Boyer, 88] R. S. Boyer and J. S. Moore, eds., A Computational Logic Handbook, Academic Press, Boston, 1988.

[Börger, 89] E. Börger, Computability, Complexity, Logic, North-Holland, Amsterdam, 1989.

[Cliff, 80] B. J. Cliff, Software development. A rigorous Approach, Prentice Hall, 1980.

[Fitting, 96] M. Fitting, First-Order Logic and Automated Theorem Proving, second ed. New York, Spinger Verlag, 1996.

[Gabbay, 93] D. M. Gabbay, C. J. Hogger and J. A. Robinson, Handbook of Logic in Artificial Intelligence and Logic Programming, Oxford University Press, Oxford, 1993.

[Gallier, 87] J. H. Gallier, Logic for Computer Science, John Wiley, 1987.

[Goos, 03] G. Goos, J. Hartmanis and J. van Leuwen, Verification, Theory and Practice, LNCS 2772, Springer, 2003.

[Hoare, 69] C. A. R. Hoare, An Axiomatic Basis for Computer Programming, C.ACM 12, 1969.

[Hoare, 85] C. A. R. Hoare and J. C. Shepherdson, eds., Mathematical Logic and Programming Languages, Prentice-Hall, 1985.

[Huth, 04] M. Huth and M. Ryan, Logic in Computer Science - Modelling and Reasoning about Systems, Cambridge University Press, 2004.

[Krantz, 02] S. G. Krantz, Handbook of Logic and Proof Techniques for Computer Science, Birkhäuser, Boston, 2002.

[Kröger, 87] F. Kröger, Temporal Logic of Programs, Springer, 1987.

[Küsters, 01] R. Küsters, Non-Standard Inferences in Description Logics, LNAI 2100, Springer, 2001.

[Loeckx, 87] J. Loeckx and K. Sieber, The Foundations of Program Verification, Wiley, 1987.

[Manna, 71] Z. Manna and R. Waldinger, Toward Automatic Program Synthesis, C. ACM 14, 1971.

[Manna, 74a] Z. Manna, Mathematical Theory of Computation, McGraw Hill Book Co., 1974.

[Manna, 74b] Z. Manna and A. Pnueli, Axiomatic approach to total correctness of programs, Acta Inform., 3, pp. 243-264, 1974.

[Nerode, 97] A. Nerode and R. A. Shore, Logic for Applications, Springer-Verlag New York, 1997.

[Pásztor-Varga, 03] K. Pásztor Varga and M. Várterész, Mathematical Logic (an Application Oriented Approach), (in Hungarian), Panem Kiadó, Budapest, 2003.

[Pnueli, 81] A. Pnueli, The temporal logic of programs, Proc. 18th FOCS, pp. 46-57. Springer, 1981.

[Richter, 78] M. M. Richter, Logikkalküle, In teubner Studienbücher Informatik, 1978.

[Robinson, 65] J. A. Robinson, A machine oriented logic based on the resolution principle, J.ACM 12., no. 1. 23-41, 1965.

[Schönig, 87] U. Schönig, Logik für Informatiker, Spektrum Akad. Verl. Heildelberg Berlin Oxford, 1987.