

Mathematical Logic and Deduction in Computer Science Education

Hashim HABIBALLA

*Department of Computer Science, Faculty of Science, University of Ostrava
30.dubna 22, 701 03 Ostrava 1, The Czech Republic
e-mail: hashim.habiballa@osu.cz*

Tibor KMET'

*Department of Computer Science, Faculty of Natural Sciences
Constantine the Philosopher University in Nitra
Tr. A. Hlinku 1, 949 74 Nitra, Slovak Republic
e-mail: tkmet@ukf.sk*

Received: June 2006

Abstract. Mathematical logic is a discipline used in sciences and humanities with different point of view. Although in tertiary level computer science education it has a solid place, it does not hold also for secondary level education. We present a heterogeneous study both theoretical based and empirically based which points out the key role of logic in computer science, computer science education and knowledge representation. We focus on the key contrast of semantics and syntax, the resolution principle as a leading inference technique (giving also interesting non-clausal generalization of the rule). Further we discuss the possibilities of inclusion the non-classical (many-valued) logics in education together with the original generalization of the non-clausal resolution rule into fuzzy logic. The last part describes partial results of the research concerning the secondary education in the Czech Republic especially in the mathematical logic field. The generalization of the presented ideas entails the article.

Keywords: computer science education, mathematical logic.

Introduction and Motivation

Logic is a well-established branch, in comparison with the other theoretical computer science disciplines, with deep tradition and its roots could be found in ancient history. Although the questions that logic encountered in the past were different, we could formulate some common issues namely the effort to simulate the human reasoning by exact (symbolic) way. Although a mathematician rather uses logic as a tool for expressing and proving properties of mathematical objects, for a computer scientist logic is a full member of theoretical computer science. Its main task is to provide formal (symbolic) framework for knowledge representation and deduction (Lukasová, 2003). As it is common in computer science, we also try to find effective algorithms solving this task. The symbolic approach means the ability to express the knowledge by “symbols” and it could

be perceived as a concurrent approach to the connectionistic one. We would like to provide the reader with some issues concerning logic and especially automated deduction in computer science education (CSE). The problem of automated deduction is the key role of logic in computer science. We will focus not only to the tertiary level of education, but also to the upper secondary level of education. Formal deduction forms a key issue concerning mathematical and computer science aspects of logic teaching. The main focus on the secondary level education is given to the logical semantics and the formal aspect is usually overlooked (Lukasová, 2003). We are trying at present to establish more mathematical topics into computer science education (CSE). Our main current interest is closely related with teaching *formal methods* of deduction on secondary level and tertiary level education. The experiment for such a teaching has been prepared and it will contain topics described in (Habiballa *et al.*, 2006). The methodology is inspired by our previous works (Habiballa, 2003; Habiballa, 2004). It is based especially on two formal methods – *tableau method* (Fitting, 1996) and the wide-used *resolution principle* (Bachmair, 2001).

Logic and Knowledge Representation in CSE

The classical (two-valued) logic is a formalism known for many years. We are able to communicate by the natural language and further we process this knowledge and deduce conclusions. The natural language is unfortunately too complex to be effectively processed by a machine. That's why logic provides a tool of expressing the knowledge in a relatively narrow class of symbols. On the other hand logic brings methods for the automated (algorithmical) solution of deduction.

On the secondary level education the classical logic is typically taught in the frame of mathematics. The lessons contain the propositional logic with semantically oriented explanation and basics of predicate logic. The main intent is mathematically focused with the accent to usage of the logic as a tool for establishing mathematical theories. The formal aspect of logic is suppressed, but such an approach is not suitable for CSE. Logic in computer science has a solid place and it provides an important theoretical background for applied disciplines like database technology, algorithmization etc. Its leading role in knowledge representation has even more significant impact. We should be aware of the fact that these key applications of logic use its formal (syntactical) aspects too. Both of these sides are reasons to consider the formal logic in secondary level education. We recommend the reader to study very interesting research performed in Israel concerning theoretical computer science education in secondary (high) school (Gal-Ezer and Harel, 1999; Armoni, 2004). The research is based on the experimental education consisting of several themes that we are used to encounter only on university level education in computer science. We would like to state especially the CM (computational model) unit providing students with theoretical model of computation like finite automata, push-down automata or Turing's machines and further the second paradigm module providing basics of logic programming. When we hear some voices requiring even the algorithmization teaching to be reduced on secondary level or theoretical computer science to be reduced

on tertiary level, it seems to be revolutionary project held by Israeli CSE researchers with positive results (Armoni, 2004). We would like to show that also logic (and especially its formal aspects) should have its place in secondary level curricula. But the recommendations mentioned could be even used for tertiary level education especially in introductory courses.

Syntax vs. Semantics – Key Problem

The key problem concerning right teaching of classical logic concerns to the essential need to explain the students the *difference between syntax and semantics*. The standard lesson focuses on the explanation of the logical connectives interpretation. Such an approach could satisfy the needs of mathematical education but could be hardly satisfying for CSE. We will present this issue by a simple example of deduction. Note that this type of logical tasks is perceived as suitable since abstract exercises are not motivating in contrast to these ones describing “real-life” situations. We will use only propositional logic in this example since its expressive power is sufficient for such a simple deduction and the simplicity has the didactic advantage.

EXAMPLE 1 (knowledge representation part). We would like to prove the knowledge about a court trial. We have three suspects – A , B , C and we know the following observations:

1. There are no other persons related with this case.
2. A never works alone.
3. C is not guilty.

We will use atomic propositions for the fact that A is guilty (A), B is guilty (B) and C is guilty (C). The first observation could be expressed by disjunction of the every atomic proposition. The second one could be symbolically represented by implication with A as antecedent and disjunction of B and C as consequent. The third one is simply a negation of C .

1. $A \vee B \vee C$, 2. $A \rightarrow (B \vee C)$, 3. $\neg C$.

The symbolic representation of the knowledge is only a half solution of the deduction problem. Before we follow the next step, we will formulate the specific notions to be taught in the logic lessons. At first the notions of *satisfiability*, *validity* and *inconsistency* should be explained and consequently the key notion of *logical consequence* should be well presented. These notions are mostly reduced to the notion of the tautology, which is not so important as the problem of consequence (the deduction bases on consequence). We follow with the semantical (unsuitable) solution.

EXAMPLE 2 (semantical solution part). If we work with semantics (interpretation) we should simply use the table method to prove (find) conclusion (logical consequence) of

Table 1
Semantic solution of deduction

A	B	C	$A \vee B \vee C$	$A \rightarrow (B \vee C)$	$\neg C$	A	$\neg A$	B	$\neg B$
0	0	0	0	1	1	0	1	0	1
0	0	1	1	1	0	0	1	0	1
0	1	0	1	1	1	* 0	/ 1	* 1	* 0 /
0	1	1	1	1	0	0	1	1	0
1	0	0	1	0	1	1	0	0	1
1	0	1	1	1	0	1	0	0	1
1	1	0	1	1	1	* 1	* 0	/ 1	* 0 /
1	1	1	1	1	0	1	0	1	0

premises. We find the models of the premises and compare the behavior of potential conclusions ($A, \neg A, B, \neg B$). The suspect C is not guilty as it is stated in the third observation (Table 1).

From the interpretation table we can observe that there are two models of the premises (marked with *). We mark potential conclusions with the symbol *, where it is also true and we mark it with the symbol / if it is false. It is clear that right logical consequence should be true in all models of premises. The table semantically proves that the proposition B is guilty is consequence, but we cannot infer any conclusion about A .

Above presented example shows a semantical way to prove the consequence. It is very important to *concentrate the attention of students on the complexity* of such an approach. In this case we have to create all possible interpretations of premises, which depends on the number of atomic propositions. Although semantically oriented table method is an algorithm deciding the problem of logical consequence, its time complexity is exponential w.r.t. the number of atomic propositions. It means if we will solve the similar court trial with 10 suspects we have to create about 1000 interpretations, for 20 suspects we will create over one million interpretations and so on. It is clear that even for a computer such a method is very hard and practically unusable.

Therefore it is very important to introduce *formal logical methods* in teaching. Formal (syntactical) methods in contrast with semantical ones do not require any interpretation of formulae. Instead of interpretation they work with the sound rules handling formulae only like symbols without any sense. These methods should keep the properties of soundness and completeness (these notions also to be explained). Several formal methods are known from the tertiary level education, but we recommend using two of them. The first one is the *tableau method* that is based on decomposition of an inconsistent formula into tree with closed branches. The method is relatively simple and straightforward. The second method – *resolution principle* – is interesting from another point of view. Many of practically successful applications of automated deduction are based on this principle, e.g., logic programming (PROLOG) or symbolic based expert systems. Formal methods can be also divided into direct and indirect methods. The first ones lead from premises

directly into logical consequence. The second ones use the fact, that the set of premises and the negated logical consequence is inconsistent and that's why we will conclude a contradiction (false formula). Indirect methods are more favorable for automated deduction since the objective of the process is the same for every logical consequence. It provides a possibility to find effective heuristics and algorithms reducing the complexity of the proving mechanism. The resolution principle in its clausal form works with clauses (disjunctive formulae with literals). It can be expressed in the following form.

Resolution Principle

$$\frac{C_1 \vee x \quad C_2 \vee \neg x}{C_1 \vee C_2}, \quad (1)$$

where C_1, C_2 are parts of clauses and x is a resolving atom.

EXAMPLE 3 (formal solution part). When trying to use the resolution principle for proving, we have to transform all premises into clauses. It could be simply done by the rewrite rules.

$$1. A \vee B \vee C \quad 2. A \rightarrow (B \vee C) \quad 3. \neg C$$

Premises 1. and 3. are clauses, but premise 2. is transformed into disjunction.

1. $A \vee B \vee C$ (axiom), 2. $\neg A \vee B \vee C$ (axiom), 3. $\neg C$ (axiom), then we can construct a direct proof with some common simplification rewrite rules (e.g., if a clause contains occurrence of the same literal we may remove all redundant occurrences).

$$4. (\text{resolution on } A \text{ in 1. a 2.}): (B \vee C) \vee (B \vee C) \Rightarrow B \vee C.$$

$$5. (\text{resolution on } C \text{ in 4. a 3.} - \text{formula 3. completely resolved}): B.$$

We formally proved that B is guilty. Of course we can also show indirect proof (and we recommend it in teaching). The main advantage of formal methods should be explained to students – these methods have the complexity depending on the structure of a formula (which is more sensible for practical usage).

Automated Deduction Implementations in Teaching

We give the reader an interesting link to the implementation of the above-presented ideas. Libuse Pavliskova has developed very useful application and didactic material in the frame of her teacher studies (Pavlisková, 2003), which can be used for secondary and tertiary education. It presents semantical and formal methods for deduction together with its automatization through computer software. An important advantage is also the presence of motivating solved examples like the court trial (Fig. 1).

Traditional Clausal Form Resolution in Education

The resolution principle (rule) is a well known and probably the most used deductive (inference) approach to automated deduction in practice (logic programming, expert sys-

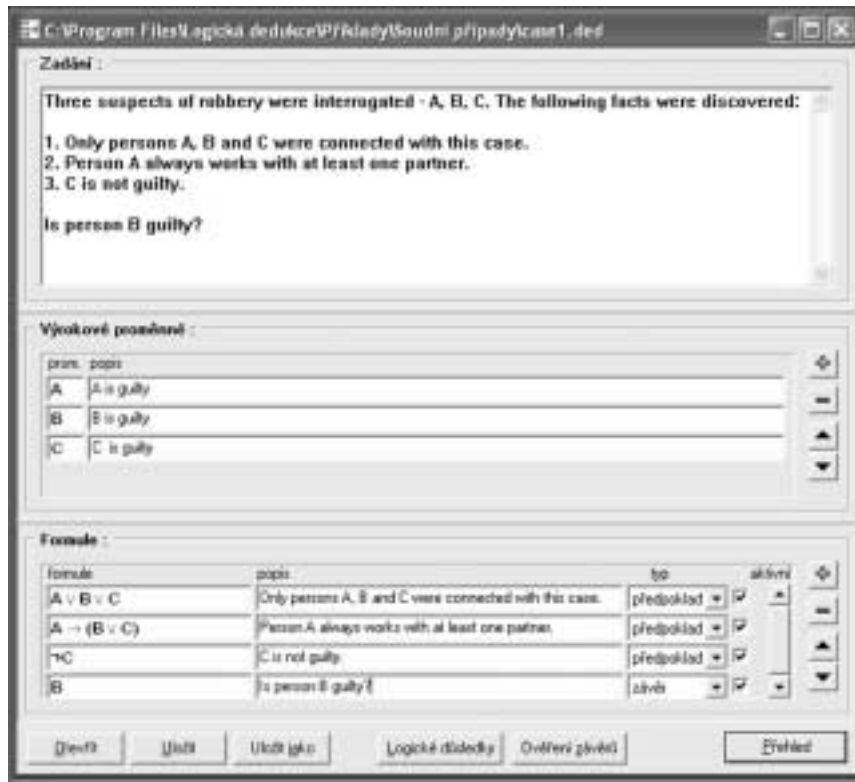


Fig. 1. Application “Logical deduction”.

tems, clausal form logic etc.). The rule in its clausal form works with clauses (disjunctive formulae with literals).

Standard view of resolution principle has several *educational* problems:

1. **The requirement of clausal form transformation** (it requires conjunctive normal form transformation in propositional logic and furthermore in non-propositional case it needs skolemization).
2. **No direct semantical explication of soundness** (it requires usage of some semantically oriented proof method).
3. **Transformation brings significant structure loss** (one information-keeping structure – a formula – is decomposed into several clauses).

Non-clausal resolution (mixture of syntax and semantics)

Resolution principle could be also generalized for the usage with general propositional logic formulae (not only clauses).

General resolution – propositional version

$$\frac{F[G] \quad F'[G]}{F[G/\perp] \vee F'[G/T]}, \quad (2)$$

where the propositional logic formulae F and F' are the premises of inference and G is an occurrence of a subformula of both F and F' . The expression $F[G/\perp] \vee F'[G/T]$ is the resolvent of the premises on G . Every occurrence of G is replaced by false symbol in the first formula and by true symbol in the second one. It is called F the positive, F' the negative premise, G the resolved subformula.

EXAMPLE 4 (general resolution with equivalence).

1. $a \leftrightarrow b \wedge \neg d$ (axiom).
2. a (axiom).
3. $b \vee d \rightarrow e$ (axiom).
4. $\perp \vee [T \leftrightarrow b \wedge \neg d]$ (resolved from (2), (1) on a) $\Rightarrow b \wedge \neg d$.
5. $\perp \wedge \neg d \vee T \vee d \rightarrow e$ (resolved from (4), (3) on b) $\Rightarrow e$.

Such a rule has several *educational advantages* in contrast to clausal form resolution:

1. *Soundness* of the general resolution is straight-forward and could be explicated to students in direct form.
2. The *structure* of original formulae is preserved.
3. The general resolution rule *does not require any transformation* of formulae. It means no other explication of these non-trivial procedures is needed.
4. There are possibilities to extend the rule for many valued logics (many-valued formalism based on standard residuated lattices or MV-algebras is sufficiently simple to be presented to students).

Many-Valued (Fuzzy) Logics in Education

We have addressed another interesting topic related with the teaching mathematical logic and deduction. The secondary level education rarely touches the presence of non-classical logics both in theoretical level and real-life. We often hear the notion “fuzzy logic” in relation with wide-used electrical appliances like washing machines. Of course this notion in the lay perception will not lead directly to the logical deduction. But this example could lead to the intuitive teaching of many-valued (fuzzy) logics. The two-valued (classical) logic is based on two truth values – true and false. Such an approach is not fully convincing for modelling of human thinking. People often use concepts that cannot be expressed only by full true or false. For example the concept of contented person could not be described by two logical values. We are contented in *some degree*. The most used approach for modelling of vagueness – fuzzy mathematics – use this intuitive tool as a many-valued interpretation. Fuzzy mathematics works with a fuzzy set that is based on membership degree to be a complex structure instead of the poor set $\{0,1\}$ (non-member, member of a set). The fuzzy logic uses this principle of interpretation to be also complex structures (e.g., where interpretation of a formula is classical two valued case – false (0), true (1); in fuzzy logic there is a possibility to interpret a formula by a member of complex structure mostly by the interval of real or rational numbers $[0,1]$). Then we can

express the contented person by some rational number (degree) (e.g., perfectly contented person in 1, almost perfectly contented person in 0.9, etc.). It allows us to model real-life a situation in soft way and that is why fuzzy logic is closely related with soft computing disciplines (evolutionary algorithms, neural networks, fuzzy modeling etc.).

Even we can come to believe that fuzzy logic is very simple formalism, we should be aware that the correct usage of fuzzy logics need to work proved structures as truth degrees. We usually work with standard algebras – structures with not only truth-value sets but also with fixed operators (conjunction, disjunction, implication) as functions. There are several such algebras with different properties (e.g., Gödel algebra, Goguen algebra, Łukasiewicz algebra). Fuzzy logics are generalizations of classical logic and therefore it is not surprising that not all laws and properties are preserved like in two-valued case. When using these algebras as truth value sets we call resulting logics by appropriate name (e.g., Gödel propositional logic). When trying to perform formal deduction we often use Łukasiewicz logic since it has several “nice” properties like holding law of double negation.

We will assume *Łukasiewicz algebra* to be:

$$L_{\mathbb{L}} = \langle [0, 1], \wedge, \vee, \otimes, \rightarrow, 0, 1 \rangle,$$

where $[0, 1]$ is the interval of reals between 0 and 1, which are the smallest and greatest elements respectively. Basic and additional operations are defined as follows:

$$\begin{aligned} a \otimes b &= 0 \vee (a + b - 1), & a \rightarrow b &= 1 \wedge (1 - a + b), \\ a \oplus b &= 1 \wedge (a + b), & \neg a &= 1 - a. \end{aligned}$$

Then we bound the new (fuzzy) logical connectives with the operators of the algebra. The interpretation of Łukasiewicz conjunction is bound with the operator \otimes , interpretation of Łukasiewicz disjunction is bound with the operator \oplus , interpretation of implication is bound with the operator \rightarrow and the negation is bound with \neg . The following sample calculations in Łukasiewicz algebra show also a little bit unreal behavior of its certain operators (Łukasiewicz conjunction and disjunction). Nevertheless other operators have reasonable interpretation (implication and negation). As stated above the main reason why choose Łukasiewicz algebra is that several essential laws comparable to two-valued logic are preserved.

EXAMPLE 5 (Łukasiewicz algebra).

$a = 0.5, b = 0.5 \Rightarrow a \otimes b = 0 \vee (0.5 + 0.5 - 1) = 0$ (in the logical meaning it states two “half” true propositions in conjunction lead to no truth at all).

$a = 0.5, b = 0.6 \Rightarrow a \rightarrow b = 1 \wedge (1 - 0.5 + 0.6) = 1$ (if consequent has the interpretation greater or equal than antecedent then the implication is fully true).

$a = 0.5, b = 0.3 \Rightarrow a \rightarrow b = 1 \wedge (1 - 0.5 + 0.3) = 0.8$ (if consequent has the interpretation less than antecedent then implication is less than 1).

Graded fuzzy propositional calculus

Novák (1999) assigns grade to every axiom, in which the formula is valid. It will be written as

$$a/A,$$

where A is a formula and a is a syntactic evaluation. We will need to introduce several notions from fuzzy logic, in order to present the fuzzy graded calculus.

Inference rule

An n -ary inference rule r in the graded logical system is a scheme

$$R: \frac{a_1/A_1, \dots, a_n/A_n}{r^{evl}(a_1, \dots, a_n)/r^{syn}(A_1, \dots, A_n)}, \quad (3)$$

using which the evaluated formulas $a_1/A_1, \dots, a_n/A_n$ are assigned the evaluated formula $r^{evl}(a_1, \dots, a_n)/r^{syn}(A_1, \dots, A_n)$. The syntactic operation r^{syn} is a partial n -ary operation on F_J and the evaluation operation r^{evl} is an n -ary lower semicontinuous operation on L (i.e., it preserves arbitrary suprema in all variables).

Evaluated proof, refutational proof and refutation degree

An evaluated formal proof of a formula A from the fuzzy set $X \mathrel{\subset\sim} F_J$ is a finite sequence of evaluated formulas

$$w := a_0/A_0, \quad a_1/A_1, \dots, a_n/A_n, \quad (4)$$

such that $A_n := A$ and for each $i \leq n$, either there exists an m -ary inference rule r such that

$$a_i/A_i := r^{evl}(a_{i_1}, \dots, a_{i_m})/r^{syn}(A_{i_1}, \dots, A_{i_m}), \quad i_1, \dots, i_m < n$$

or

$$a_i/A_i := X(A_i)/A_i.$$

We will denote the value of the evaluated proof by $Val(w) = a_n$, which is the value of the last member in (4).

EXAMPLE 6 (many-valued logics). For educational purposes we can start with the simple but illustrative ‘‘sorites paradox’’. The well-known paradox shows the weakness of two-valued logic. We can model (using predicate logic) the situation stated in the following two sentences:

1. The heap without a stone is small. 2. If a heap with n stones is small then this heap with one added stone is also small.

When using the classical logic we can model this situation by two formulae (with the predicate):

1. $small_heap(0)$, 2. $small_heap(n) \rightarrow small_heap(n + 1)$.

It is clear that by these two formulae we will prove that every heap (with any number of stones) is small, which is nonsense (paradox following from two-valued truth).

Using Lukasiewicz predicate logic (generalization of classical predicate logic) we are able to model formulae with truth degree from interval $[0,1]$:

1. $small_heap(0)$ – true in the degree 1,
2. $small_heap(n) \rightarrow small_heap(n + 1)$ – true in the degree 0.999.

Formula 2. (implication) is in contrast with classical logic holding not fully but only in the degree 0.999. This fact allows us to avoid the paradox. Due to the degree 0.999 every application of the implication will lead to the inference of the fact that heap with one added stone is small in the degree decreased by 0.001. If we would like to prove the formula $small_heap(1000)$ (heap with 1000 stones is small) we follow the following sequence (I means interpretation of the implication according to the operator in previously defined algebra):

$$\begin{aligned} I(smallest_heap(0)) &= 1, \text{ then we will use that in equation,} \\ 0.999 &= I(1 \wedge 1 - 1 + smallest_heap(1)) \Rightarrow I(smallest_heap(1)) = 0.999 \\ 0.999 &= I(1 \wedge 1 - 0.999 + smallest_heap(2)) = 0.001 + I(smallest_heap(2)) \\ &\Rightarrow I(smallest_heap(2)) = 0.998 \end{aligned}$$

...

$$I(smallest_heap(1000)) = 0.$$

It means that we proved a heap with 1000 stones is not small in any degree higher than 0.

Fuzzy logic has also another interesting applications (used in industry, decision-making, etc.) based in linguistic expressions. The idea of these expressions lies in the formalization of natural language expressions that we often use in real-life like “something is very small”, “something is rather big”. We can model such expressions by fuzzy sets and use them in linguistic variables included in “IF – THEN” type complex formulae. These formulae can form large knowledge bases that can be used to control various processes, to decide hard problems according to the formulations of IF – THEN rules given by experts. For example we can model the process of car driving:

IF traffic lights changed to yellow AND car speed is *small* AND distance from crossing is *small* THEN press down brake by *medium* power.

IF traffic lights changed to yellow AND car speed is *medium* AND distance from crossing is *very small* THEN press down accelerator by *big* power.

There are a lot of such real applications for example Linguistic Fuzzy Logic Controller designed on University of Ostrava. Such popular examples of fuzzy logic usage may be used in education also on secondary level.

The authors have proposed also generalization of the general resolution rule for Lukasiewicz Predicate Logic with evaluated syntax and such an approach is also applicable to mathematical and computer science education (Habiballa, 2005). The rule could be with simplification stated as follows (of course we should also define the most general unifier, other notions for refutational proofs in fuzzy logic).

General Resolution for Fuzzy Propositional Logic

$$r_{GR}: \frac{a/F[G_1, \dots, G_k], \quad b/F'[G'_1, \dots, G'_n]}{a \otimes b/F[G/\perp] \nabla F'[G/T]}$$

where F and F' are premises and G resolved subformula and \otimes is the Lukasiewicz multiplication.

EXAMPLE 7 (evaluated proof in fuzzy propositional logic using resolution).

1. $a \leftrightarrow b \wedge \neg d/1$ (axiom with evaluation degree 1).
2. $a/0.7$ (axiom with evaluation degree 0.7).
3. $b \vee d \rightarrow e/0.75$ (axiom with evaluation degree 0.75).
4. $\perp \vee [T \leftrightarrow b \wedge \neg d]/0.7 \otimes 1$ (resolved from (2), (1) on a) $\Rightarrow b \wedge \neg d/0.7$.
5. $\perp \wedge \neg d \vee T \vee d \rightarrow e/0.7 \otimes 0.75$ (resolved from (4), (3) on b) $\Rightarrow e/0.45$.

Automated Non-Clausal Resolution Implementations in Teaching

The authors has already developed also a tool for automated theorem proving based on general resolution (Fig. 2) (Fuzzy Predicate Logic Generalized Deductive system) – enriched application for classical logic [6]. The application could be used for presentation of the rule and also several inference strategies developed especially for this formalism.

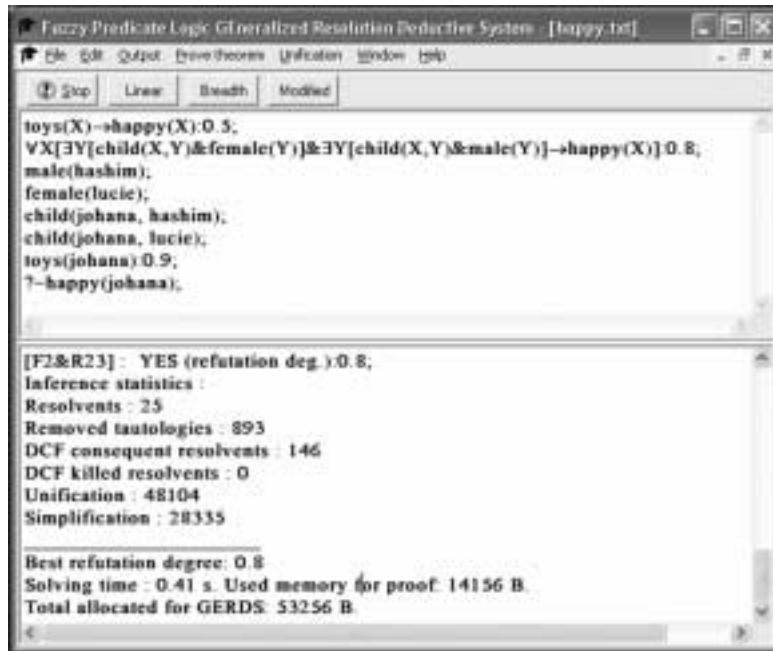


Fig. 2. Fuzzy Predicate Logic Generalized Deductive System.

The future extensions consist of presentation of the *tree of resolution proof visualization*, improvement of object-oriented implementation, free distribution of the package and also some “real-life” applications.

Current State of Logic Education in Czech Republic

We have already presented the theoretic principles of effective teaching of logic together with interesting generalisation of the resolution principle gaining from syntactic-semantic properties. At the end we would like to shortly summarize the results of a research concerning the current state of high school education. It was performed through a web questionnaire sent to all secondary schools in the Czech Republic. The research concerning the actual content of teaching computer science was established on University of Ostrava in 2003. We have used the research instrument of on-line questionnaire. Link to this questionnaire was sent to 863 secondary level schools in the Czech Republic (according to the database of Institute for Information in Education of the Czech Republic); total amount of available responses was 132 from the entire Czech Republic. Although the research has wider focus we present only the results concerning the logic education.

Fig. 3 shows the structure of institutions that responded to the questionnaire.

The general secondary schools are the institutions primarily preparing for tertiary level education (mainly university studies). Other schools in the sample include mainly technical schools with various study programmes (electrical, chemical, telecommunication etc.) preparing for direct fulfillment in practice. Trade schools educate the secondary level professionals in economic field and other vocational schools include rather the humanities (pedagogical, social-juridical, nursing).

We will analyze only the results for general secondary schools (total number of 55) since these schools form the most consistent and the most frequent type of secondary schools in the research. We should observe that relatively high amount of schools teach at least the basics of propositional logic, which typically concern to semantics (symbols are bound with its interpretation).

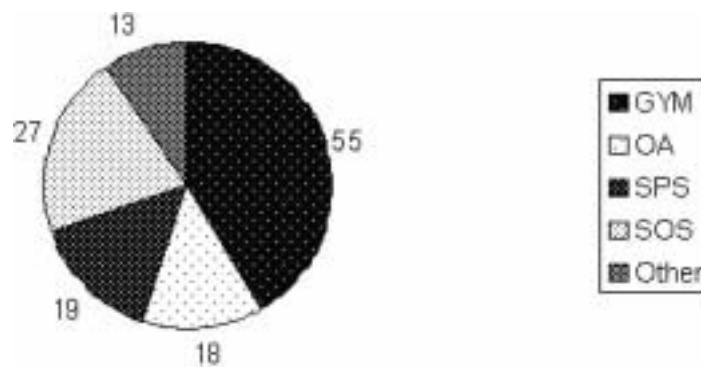


Fig. 3. Secondary schools structure in the research (GYM – general secondary schools, SPS – technical schools, OA – trade schools, SOS – vocational schools, Other – training schools).

Table 2
Basic principles

Basic Principles Teaching Logic on Secondary Schools	
Issue	Number (Percentage) of Schools
Proposition and truth value	42 (76%)
Propositional logic symbols	40 (73%)
Equivalence of formulae	30 (56%)

Then we can distinct the second sort of topics taught at lower level. Student are taught the key formula properties, but still in semantic level.

Table 3
Advanced semantics

Advanced Semantic Notions	
Issue	Number (Percentage) of Schools
Boolean functions	26 (47%)
Satisfiability, tautology, contradiction	23 (42%)

Further we follow with topics that have syntactic basis.

Table 4
Syntax

Syntactic Notions	
Issue	Number (Percentage) of Schools
Propositional Formula Construction	22 (40%)
Normal Forms	17 (31%)

All these sorts have relatively high extent in schools, but the topics addressed in the first part of the article are taught at very low level (formal deduction and methods).

We should observe not surprising results, which show that the teaching of key topics of logic from computer scientist’s point of view is at very low level in secondary school system in the Czech Republic. The principal notion of deduction is taught only at 2 schools from 55, even deduction forms the leading role in the usage of logic in practical “real-life” application (logic programming, expert systems based on logical knowledge representation). So wide used resolution principle, that is the most expanded method implemented in deductive inference systems, is taught only in one school. All these topics (as we tried to show in brief previous examples) can be explicated in relatively intuitive

Table 5
Formal deduction

Formal Deduction and Methods	
Issue	Number (Percentage) of Schools
Tableau method	6 (11%)
Syntactic trees	4 (7%)
Decidability and satisfiability	3 (5%)
Deduction	2 (4%)
Formula complexity	2 (4%)
Resolution principle	1 (4%)
Non-classical logics	1 (2%)

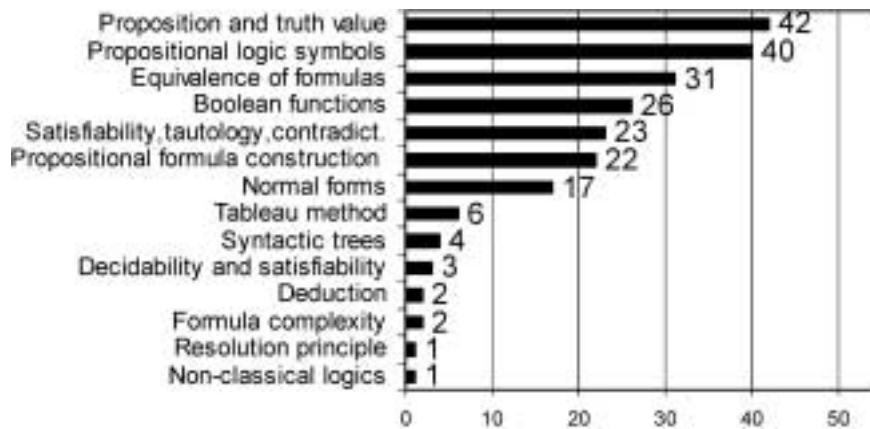


Fig. 4. Logic in general secondary schools (number of schools).

manner (free of formalization known from tertiary level education). There is also important fact that 77% of teaching logic has place in the frame of mathematical education (47 of 58, counting occurrences in all grades in one school).

Conclusions

We have presented a brief survey into the possibilities and advantages of computer science based education mathematical logic. These ideas are part of the research performed on University of Ostrava trying to improve secondary education in the Czech Republic. We have already published several papers in the national didactic journals and proceedings, which described in detail topics, have to be included into the secondary education. Israeli CSE researchers already successfully implement some of the similar ideas in the Second Paradigm module of the high-school curricula that include theoretical and practical teaching of deductive systems through logic programming (Gal-Ezer and Harel,

1999). Our approach should emphasize theoretical learning of the inference method principles, including non-clausal resolution and also give a short introduction into the theory and application of the many-valued logics.

We follow these principles:

- teaching of mathematical logic should be performed not only in the frame of mathematical education, but also in the frame of computer science education; such education should have *interdisciplinary* aspect to be effective (it means mathematics and computer science have to cooperate);
- mathematical education should inhere the basics – language, semantics, connectives, interpretation, tautologies;
- computer science education should give advancement - syntax vs. semantics, knowledge representation, problem tasks from “real-life”, deduction, formal deduction and methods, resolution, many-valued logics;
- we should use existing ICT tools to make the learning interesting for students (logic programming, visual inference engines with prepared examples).

References

- Armoni, M., Gal-ezer, J. and Tirosh, D. (2005). Solving problems reductively. *Journal of Educational Computing Research*, **2**(32), 113–129.
- Armoni, M. and Gal-ezer, J. (2004). On the achievements of high school students studying computational models. In *Proceedings of ITICSE*, Leeds, UK.
- Bachmair, L. and Ganzinger, H. (2001). Resolution theorem proving. In *Handbook of Automated Reasoning*. MIT Press.
- Dvořák, A., Habiballa, H. *et al.* (2003). The concept of LFLC 2000 – its specificity, realization and power of applications. *Computers in Industry*, **3**(51), 269–280.
- Fitting, M. (1996). *First-order Logic and Automated Theorem Proving*. Springer-Verlag.
- Gal-ezer, J. and Harel, D. (1999). Curriculum and course syllabi for a high-school program in computer science. *Comp. Sci. Education*, **2**(9), 114–147.
- Gal-ezer, J. and Zur, E. (2004). The efficiency of algorithms - misconceptions. *Computers and Education*, **42**(3), 215–226.
- Habiballa, H. (2006). Non-clausal resolution theorem proving for fuzzy description logic. In *Proceedings of SOFSEM'06*. MatFyz Press, Praha (to appear).
- Habiballa, H. and Kmet', T. (2004). Theoretical branches in teaching computer science. *International Journal of Mathematical Education in Science and Technology*, **6**(35), 829–841.
- Habiballa, H., Kmet', T. and Pavlisková, L. (2006). Logic (I, II, III). *Matematika-Fyzika-Informatika (Journal of UCMF)*, **07**(15) (in Czech – to appear).
- Habiballa, H. (2003). Didactic experiment in teaching theoretical computer science. *Technology of Education*, **3**(11), 6–12 (in Czech).
- Habiballa, H. (2005). Non-clausal resolution in fuzzy predicate logic with evaluated syntax (background and implementation). In *Proceedings of International Conf. the Logic of Soft Computing IV*. Ostrava.
- Lukasová, A. (2003). *Formal Logic in Artificial Intelligence*. Computer Press, Brno (in Czech).
- Novák, V., Perfilieva, I. and Močkoř, J. (1999). *Mathematical Principles of Fuzzy Logic*. Kluwer.
- Pavlisková, L. (2003). Deduction principles in propositional logic. Diploma thesis, University of Ostrava (in Czech).

H. Habiballa 1976 graduated at the Department of Computer Science, University of Ostrava, Czech Republic in computer science and technology – information systems (MSc – 1999); further he holds advanced degree in computer science education (EdD – 2003) at the Constantine the Philosopher University, Nitra, Slovak Republic and he defended doctoral thesis (PhD – 2004) in computer science education at the same institution. Further he defended rigorous thesis and second dissertation in information systems – logic (PhD – 2007). He works as an assistant professor at the Department of Computer Science, University of Ostrava and also participates as a research assistant in the Institute for Research and Applications of Fuzzy Modeling. His interests include research both in theoretical computer science and computer science education (several national and international grant projects). He is author of about 100 publications (articles in reviewed journals, proceedings of conferences, chapters in monographs, teaching texts etc.).

T. Kmet' full professor of mathematics, Constantine the Philosopher University in Nitra; interests in mathematical modelling and didactics of computer science.

Matematinė logika ir dedukcija informatikos kurse

Hashim HABIBALLA, Tibor KMET'

Matematinė logika įvairiais būdais naudojama gamtos ir humanitariniuose moksluose. Informatikos mokymas labai svarbus aukštajame moksle, tačiau vidurinėje pakopoje taip nėra. Straipsnyje pateikiamas įvairiarūšis teorinis ir empirinis tyrimas, kuris atkreipia dėmesį į logikos svarbą informatikoje, informatikos mokyme ir žinių atvaizdavime. Straipsnyje aptariami semantikos ir sintaksės klausimai, problemų sprendimo principas kaip išvadų konstravimo būdas (pateikiant įdomius taisyklių apibendrinimo būdus). Taip pat aptariama galimybė įtraukti neklasikinę (daugia-reikšmę) logiką į mokymą. Paskutinėje dalyje pateikiami daliniai tyrimai, atlikti Čekijos Respublikos viduriniame lavinime, būtent, matematinės logikos srityje. Straipsnyje itin pabrėžiamas pateiktų idėjų apibendrinimas.