The educational tasks and objectives system within a formal context*

Ľubomír Antoni, Ján Guniš, Stanislav Krajči,
Ondrej Krídlo, Ľubomír Šnajder

Institute of Computer Science, Faculty of Science,
Pavol Jozef Šafárik University in Košice, Jesenná 5, 040 01 Košice, Slovakia.
lubomir.antoni@student.upjs.sk, jan.gunis@upjs.sk,
stanislav.krajci@upjs.sk, ondrej.kridlo@upjs.sk, lubomir.snajder@upjs.sk

Abstract. The educational objectives represent the precise statements of what we expect or intend students to learn as a result of education. We have conducted an analysis of the educational tasks and objectives system within a formal context with respect to the collected real data on an array data structure of five teachers in the field of computer science. We submitted a report and the corresponding concept lattice to each individual teacher and explored their additional feedback. In addition, we formulate the general observations and present the feasible set of tasks and objectives of an array data structure. The results are expected to annotate in the future formation of the curricular documents as supplement to the National Education Program in Slovak republic which is formulated concisely.

Keywords: formal context, educational task, system of objectives, concept lattice

1 Introduction

The scope of the computer science education in Slovakia is officially declared in The National Education Program of Slovak republic as the supreme curricular document. The Slovak National Education Program defines the main principles and general objectives on which education and training in computer science is based. The education of computer science at secondary schools in Slovakia includes five areas:

a) Theory of the information (numeral systems, coding, compression, etc.),
b) Information and communication technologies – ICTs (internet, computer networks, safety, etc.),

* This work was partially supported by the grant VEGA 1/0832/12 and by the Slovak Research and Development Agency under contract APVV-0035-10 "Algorithms, Automata, and Discrete Data Structures". We wish to express the special thanks to the teachers for their cooperation in this experiment and to the reviewers for their helpful remarks.

c) Procedures, problem solving and algorithmic thinking (algorithms, program, programming languages, etc.),
d) Principles of ICTs operations (software, hardware, architectures, etc.),
e) Information society (e-learning, licenses, risks, etc.).

Formal concept analysis [14] as a lattice theory allows us to explore the meaningful groupings of educational tasks (referred to objects) with respect to common objectives (referred to attributes) and it provides the visualization capabilities. The conceptual difficulties in mathematics education [32], or the integrated care pathways [30] are analyzed by formal concept analysis, as well. An extensive overview of the various application domains that include software mining, web analytics, medicine, biology and chemistry data is given by [29], [11]. Recently, the feasible attempts and generalizations are investigated in [1,5,7,22].

In this paper, our aim is to provide the system of objectives and tasks that is expected to fill in the gap of the National Education Program in Slovak republic. In general, the National Education Program is formulated concisely and we put emphasis in a long term to particularize other supplementary curricular documents and express the educational objectives more explicit in various areas. Therefore, we have focused on an algorithmic thinking area and chosen an array data structure as an educational content in which we have fruitfully applied formal concept analysis. Simultaneously in this area, we focus on algorithms including searching, sorting or text processing. In other countries, the national curricular documents and other standards define the educational objectives in the various levels of specification, see [36, 37].

2 Educational objectives of an array data structure

An array data structure, as a collection of indexed elements, plays an important role in the education of programming. An array or its equivalent as a kind of data type is implemented in the most of programming languages. The term is also used in a theoretical computer science as abstract data type.

We aim at specifying the particular and relatively precise objectives of an array data structure education in the algorithmic thinking area. Regarding our long-term cooperation with the teachers in the field, we declare some input set of objectives of an array data structure:

1) to specify an array as the structured homogeneous data type with elements denoted by single identifier,
2) to appoint the real examples of one-dimensional array data structure (e.g. rooms in a hotel, seats in a plane, etc.),
3) to interpret the notions of an array index (an array key) and an array element and to explain the difference between them,
4) to differ an array index type and an array element type,
5) to reason that an array index type is an ordinal type (numbers, characters, other enumerations),
6) to declare a variable of array,

7) to read and to write out the array elements,
8) to manipulate with the array elements, to assign the array element to the other variables, to increment the array elements,
9) to appoint the common errors related to an array data structure (incorrect index type, overflow, incompatibility of the types),
10) to apply an array data structure in the simple issues (e.g. to store an array, to find the maximal value, to modify the elements of array, etc.),
11a) to apply an array data structure in a searching,
11b) to apply an array data structure in a sorting,
11c) to multiple access to the array elements,
11d) to apply an array data structure in a text processing,
11e) to apply an array data structure in a simple game programming,
12) to recognize the issues in which can be applied an array data structure effectively, to appoint the advantages and disadvantages of an array in comparison with other simple data structures (an access to elements, a space complexity).

The specified aims are enumerated by the revised taxonomy of Bloom [18] in order to classify statements of what we expect or intend students to learn as a result of education. The revised taxonomy focuses on four knowledge dimensions including factual knowledge (basic elements), conceptual knowledge (interrelationships among the basic elements), procedural knowledge (how to do something) and metacognitive knowledge (awareness and knowledge of one's own cognition). In general, an educational process consists of a motivation phase, a phase of the first acquisition, a fixation phase and a diagnostic phase. The phase of a systematization, a propedeutics or an application phase can be also involved.

We submitted the previous list of aims to the teachers in the secondary schools in Slovakia. The teachers were instructed to appoint the tasks which they usually apply in an educational process of an array data structure in programming. Teachers were not limited by the number of tasks and moreover, it was possible to add some additional aims (13, 14, ...) if they required. Having such instructions, every teacher was asked to fill in the following table:

Fig. 1. Table of tasks and aims

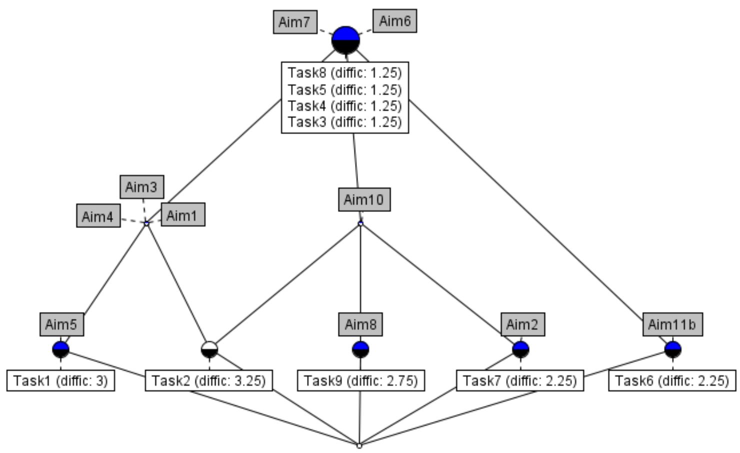
N.	task / aim	1	2	3	4	5	6	7	8	9	10	11a	11b	11c	11d	11e	12	13	···
0	2nd max. element				x		x	x			x	x		x					
1																		additional	aims
2																			
:																			

The zero row in the table is an example of task which we have added for an illustration. Particularly, we have appointed the task of finding the second largest element in an array data structure, which fulfills the aims $4, 6, 7, 10, 11a, 11c$.

3 Concept lattice of each individual teacher

We have obtained the data of five teachers who proposed overall 92 tasks, some of them equal with respect to the aims. Four teachers launched the additional educational aims including the applying of an array data structure as the parameter of the procedure, a dynamic array, two-dimensional arrays, the issue of indexing the first element, reasoning initializing errors (seven aims in overall). For each individual teacher we have constructed the concept lattice from the collected data using ConExp[1] software. We use a concept lattice with reduced labeling (labeled line diagram) regarding own objects and own attributes which is accessible to human reasoning. An example of a concept lattice shown to one of the teachers follows in Fig. 2. In effort to explore the task difficulty, we have assigned the degree to each individual aim from the set $\{0.25, 0.5, 0.75, 1\}$ depending on its dimension in Bloom revised taxonomy. A value of the task difficulty (computed as the sum of the degrees of involved aims) is shown in a concept lattice as a supplement of the particular task label (i.e. own object).

Fig. 2. Concept lattice of the participated teacher



In our feedback given to teachers, we have added the comments and some additional questions. The comments include the aims which were obtained in all tasks (Aim6 and Aim7 in Fig. 2), the tasks which contain the unique aims (for instance Aim11b is included only in Task6) and the following instructions how to read a concept lattice to help teachers analyze the results:

– tasks in the first row are the representatives of the teacher's system of tasks (there is no task with the superset of aims),
– the shaded labels linked to a node in the first row represent the attributes introduced uniquely in the task,
– the aims introduced by a particular task one can obtain by collecting the shaded labels on all paths leading up from the selected task node,

[1] Concept Explorer, version 1.3, website: http://conexp.sourceforge.net. Nevertheless, we have successfully tested some other formal concept analysis software tools, for instance FcaStone, Lattice Miner, ToscanaJ, FCART, as well.

– the tasks that involve a particular aim one can acquire by tracing task labels leading down from the selected aim node,
– the top element of a concept lattice introduces the aims obtained in all tasks,
– a task in a higher row of a concept lattice is appropriate for the first acquisition phase of an educational process (not compulsory),
– a task in a lower row of a concept lattice is appropriate for the fixation or systematization phase of an educational process (not compulsory).

We state that not all of the submitted aims were used by the engaged teachers (for instance Aim9 is not introduced in Fig. 2). In contrary, some of the teachers have added their additional aims. Namely, we have first analyzed the systems of tasks from the viewpoint of each individual teacher. Otherwise, not introduced aims would be figured at the bottom element of a concept lattice.

For each concept lattice, we have calculated the degree of tasks and aims system gradation level as the proportion of the number of task nodes (as own objects) in the longest path and the total number of the tasks in a concept lattice. This indicator shows how gradated are tasks of teacher's system. The smaller number indicates the more diversified system, the higher number expresses the more gradational system. The obtained results and concluding remarks follows:

– the minimal number of tasks was 9, the maximal 27 in teacher's set of tasks,
– the most frequently introduced aims are Aim6, Aim7 in order to declare a variable of array and to read/write out elements of an array,
– two systems contain the aim(s) introduced in every task,
– two systems include the set of equal tasks with respect to the aims (i.e., at least two tasks equal),
– the gradational level in the systems takes the values from 0.13 to 0.22,
– the average task difficulty in teacher's set of tasks takes the values from 1.97 to 4.09; lower value indicates that the set of tasks is appropriate more for beginners, the higher value expresses focusing on advanced students.

We were interested in a feedback of the teachers in relation with the obtained results. One of the teachers confirmed that his/her set of tasks was used for advanced students (the average task difficulty is 3.22). The teachers have explained the reasons to add some new aims, reported the tasks which they used to apply in a diagnostic phase, declared the most problematic aims for students, etc. These issues and some other recommendations will be still discussed with the teachers and other respondents in a formal and an informal way.

4 Attribute exploration of each individual teacher

Beside the concept lattice diagram one can examine the implications between attributes valid in a teacher's tasks and aims tables. For instance, the implication {Aim7, Aim8} $\rightarrow$ {Aim6} shows that the following rule holds in a table of tasks and aims: Aim6 is introduced in every task that includes Aim7 and Aim8 together. This means that the task focused on reading, writing out and manipulation with the elements of an array will also satisfy the aim of the declaration of an array data type. We have verified that this natural implication holds for the all five collected set of tasks (confidence of this association rule is 100%). On the other hand, the implication {Aim8} → {Aim7} does not hold in general, because there are some tasks (the counterexamples) focused on manipulation with the array elements, but do not read and write out the array elements (confidence of this association rule is 66%). Another natural implication {Aim4, Aim5} → {Aim3} means that every task focused on recognizing a difference between an array index type and an array element type will satisfy the aim of interpretation of an array index and array element. This association rule holds in the tasks of teachers with confidence 86%.

The implications one can read off from the concept lattice with reduced labeling, it is sufficient to check whether the each attribute's node from an implication's conclusion is above (or equal to) the infimum of all attributes nodes from a premise. In addition, one can compute the Duquenne-Guigues basis of implications, which is optimal in terms of its size and includes a minimum cover of all valid attribute implications, for more information see [14], [6].

The knowledge acquisition method called attribute exploration is described in general by [14]. In our experiment, we generate Duquenne-Guigues basis of implications for each individual teacher. Then, for every implication (one by one) one can make a decision to accept or provide a counterexample. By providing a counterexample, we suggest to add a task with the combination of aims which was still lacking and is fruitful to include in an education. For instance, the attribute exploration process for one of the teachers is shown in Fig. 3. We introduce these implications from basis in which the premise is satisfied by at least one task in the table.

Fig. 3. The educational aim exploration

N.	question	answer	advised counterexample
1.	$\emptyset \to \{6,7\}$?	yes	
2.	$\{6,7,8\} \to \{10\}$?	yes	
3.	$\{5,6,7\} \to \{1,3,4\}$?	yes	
4.	$\{4,6,7\} \to \{1,3\}$?	yes	
5.	$\{3,6,7\} \to \{1,4\}$?	no	a task with $\{3,6,7,10\}$
6.	$\{2,6,7\} \to \{10\}$?	yes	
7.	$\{1,6,7\} \to \{3,4\}$?	yes	

First question in Fig. 3 indicates that all tasks cover the aims Aim6, Aim7. We have answered this question yes that means that we agree to preserve this implication in a teacher's set of tasks. In contrary, fifth question expresses that if a task has aims Aim3, Aim6, Aim7, then it also has aims Aim1, Aim4. We do not agree to preserve this implication and advise to add a new task having aims Aim3, Aim6, Aim7, Aim10. It is advised to distinguish Aim3 and Aim4. Actually, Aim3 is introduced in a task having, for instance, integers as an array index type and also as an array element type. Nevertheless, Aim4 requires differentiation of an array index type and an array element type to achieve this goal absolutely.

The separation of these two aims in at least one educational task is helpful to encourage students to understand an array data type. Hence, we advise in addition a counterexample of a task with aims Aim3, Aim6, Aim7, Aim10 by the educational aim exploration shown in Fig. 3. This task removes the undesirable implication from the basis of all implications. Remind that a counterexample must not contradict the implications we have confirmed so far (the rows 1,2,3,4 in Fig. 3). However, if the counterexamples are added into the table, the concept lattice is modified.

5 The summary results

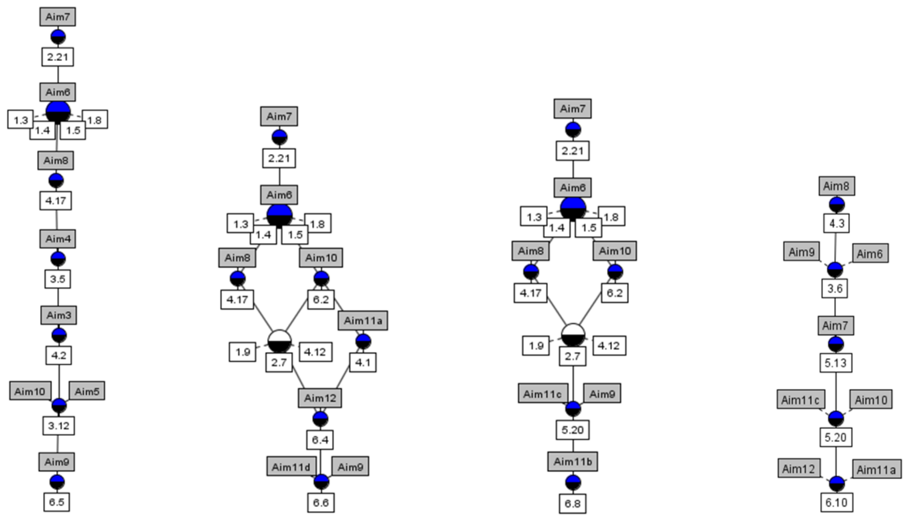
Regarding five teachers data and one additional set of 10 tasks proposed by two of the authors, we have analyzed 102 tasks and 23 educational aims obtained in this research. We have generated the summary concept lattice and found the following observations:

- 45 tasks (the first row in the summary concept lattice) are the representatives; i.e. every task includes the unique set of aims and there is no task that introduces the superset of these aims,
- 5 tasks (from 45 representatives) are such that every task includes the unique set of aims and there is no task that introduces neither superset nor subset of these aims,
- 3 aims (the first row in the summary concept lattice) are unique, i.e. the aim is introduced only by one task,
- the most frequent aims in general are Aim6 (80% of tasks), Aim7 (63%), Aim10 (51%), Aim8 (46%), Aim4 (35%) which represent the basic declaration, read, write out, manipulation and applications of an array data type in the simple tasks,
- the most frequent aims including applications of an array data type in the more difficult situations are Aim11c (22%), Aim11d (22%), Aim12 (20%) and Aim11a (19%); the applications in a sorting and a simple game programming are the least represented from these group of aims.

In effort to prepare the graduated sets of tasks, we have explored the longest paths extracted from the summary concept lattice with reduced labeling of all 102 tasks. Some of the longest paths are shown in Fig. 4. Every path contains the graduated system of tasks depending on the final task we want to achieve in conclusion. The object label, for instance 3.5, corresponds to the fifth task of third teacher. The set of tasks labeled 6.1 – 6.10 comes from the authors.

The paths have different lengths, because there are nodes in the summary concept lattice with reduced labeling, which do not contain neither task or aim label (own object or own attribute). Therefore, these nodes are omitted in the extracted longest paths shown in Fig. 4. Nevertheless, some of the paths can have the same length and can differ only in a small number of tasks. To capture two or more paths (which are similar in this sense) by one figure, we display also not linear cases (b), (c) in Fig. 4.

Fig. 4. The longest paths extracted from the summary concept lattice



(a) (b) (c) (d)

The longest paths are recommended to apply in a diagnostic phase of an educational process. For instance, the longest path (a) illustrates that if a student has a problem with Task6.5, we ask him/her to solve Task3.12. Moreover, if we have found that a student has a problem with Aim6 in Task6.5, we give him/her to fixation one task (or more) from the node which contains the set of equal tasks Task1.3, Task1.4, Task1.5, Task1.8. In contrary, if a student has no problem with Task6.5, we suppose that he/she will pass also the Task3.12. There is only one path including seven task nodes, however paths with six nodes appear in the summary concept lattice several times. The cases (b) and (c) were chosen to cover the most frequent aims by the combination of (a),(b),(c) cases. Moreover, the case (b) shows that if a student has a problem with Task6.4, we can choose either Task1.3 or Task 4.1 in order to cover the aim that was not fulfilled by a student. We can also extract some other (not compulsory the longest) paths starting with other initial aims and different initial tasks extracted from the summary concept lattice. The path (d) presents an example of the five nodes path starting with Task 4.3 in contrary with an initial task of the previously described paths.

As a conclusion, we propose to supplement four of the input objectives and to add two additional objectives (mainly for advanced students) into the input system of objectives as follows:

5) original form supplemented by: the first array index is not necessary 0 or 1,
8) original form supplemented by: find a presence of some value in a an array,
9) original form supplemented by: errors related to a clear of an array,
10) original form supplemented by: an array as a parameter of procedure,

13) to apply a dynamic array in the simple issues,
14) to apply a two-dimensional array in the simple issues.

Moreover, we present some interesting educational tasks which appear in the summary concept lattice mostly in the first row and one can advise them to apply in the educational process related to an array data structure. The formulations are shortened in comparison with the original author's texts.

- Propose the way how to denote the parking places in front of a hotel. How are the train carriages enumerated? How would you denote the overall and final results of six teams in the television knowledge contest?
- We have observed GPS data containing ten altitudes on our tourist route. Write a program to print out the altitudes on a reverse route.
- Imagine that you have received SMS from your friend. Write a program to count the number of words in your text message.
- A musical instrument, like a piano, can be simulated by a computer program. Some of the keys will have assigned a particular tone frequency. Write a program to play a tone when the particular key is pressed.
- Consider the starting sequence of children names and the final shift of Ferris wheel as the input. Write a program to make a list of the children names in the sequence in which they will get out Ferris wheel.
- Write a program to generate twelve random values expressing the number of your website visits in a particular month. Draw a histogram, highlight the maximum and minimum and show an average value as a horizontal line.

6 App Inventor concept lattice

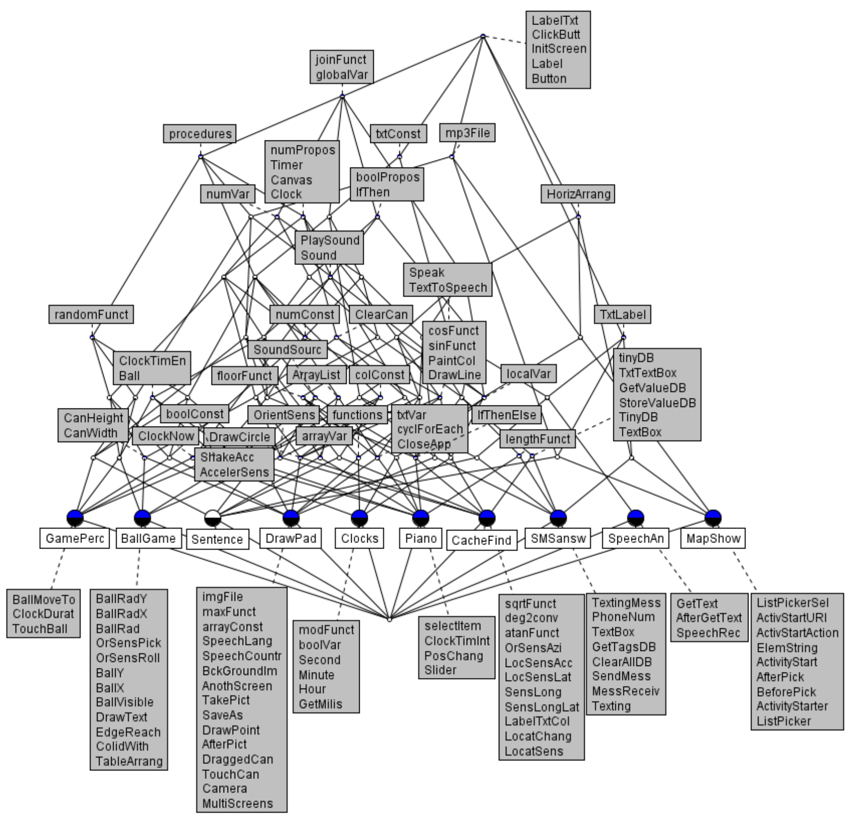
We have fruitfully applied formal concept analysis as a powerful tool in a simultaneous analysis that involves the teaching of programming skills in an open-source web application App Inventor 2. The tutorial website[2] provides materials in the form of learning cards for building the basic applications, but one of the authors of this paper has prepared the set of ten complex educational tasks which in summary cover 129 elements (components and their elements, event handlers, call, set instructions, get instructions, data structures, etc.) available at the present time. The added value includes the proposal of the introductory set of complex tasks and its further modification in effort to teach and learn the different target groups. The talented lower secondary school's pupils participate in our optional university courses and the teachers of secondary schools attend the didactic workshops at our university. Our results are concerned with the inclusion of the programming language elements (available at the App Inventor website at present) in the complex educational tasks and the effort to extract the appropriate tasks for the different types of an educational process. The formal context contains 10 tasks as the set objects and 129 App Inventor programming elements as the set of attributes.

[2] http://www.appinventor.org/

Exploring own attributes, the resulting concept lattice and its attribute labels shown in Fig. 5 give an information about the elements introduced uniquely by a particular task. As conclusion, we recommend the following methodology:

- a task with a high ratio of the own elements and the low total number of elements is advised to use in a first acquisition phase of an education,
- a task with a low ratio of the own elements and the low total number of elements is recommended to use in a fixation phase of an educational process,
- a task with a low ratio of the own elements and the high total number of elements is suggested in a systematization or diagnostic educational phase,
- a task with a high ratio of the own elements and the high total number of elements is the least appropriate for an educational process, because it brings many new elements without their introduction in a more simple task.

Fig. 5. A concept lattice of App Inventor 2 programming in education



7 Conclusion

We have investigated the educational tasks and objectives of five real teachers giving lessons in computer science. Extracted paths from the summary concept lattice seem to be based on the similar idea as in the learning paths from knowledge space theory introduced by Doignon and Falmagne [13]. The peer instruction is a learning method in which the results can be applied, as well.

The learning process of students in computer science is also concerned in the work of Uta Priss [31, 32]. The tools developed for learning and teaching in combination with curricula and teaching practices are aiming at actual project weSPOT at TU Graz with applying the formal concept analysis [33]. The triadic version of formal concept analysis [3, 15] seems to be fruitful for analyzing the concordance of the teachers in our future work.

References

1. C. Alcalde, A. Burusco, R. Fuentes-González, I. Zubia, The use of linguistic variables and fuzzy propositions in the L-fuzzy concept theory, Comput. Math. Appl. 62 (8) (2011) 3111-3122.
2. L. Antoni, S. Krajči, O. Krídlo, B. Macek, L. Pisková, On heterogeneous formal contexts, Fuzzy Sets Syst. 234 (2014) 22–33.
3. R. Bělohlávek, C. Glodeanu, V. Vychodil, Optimal Factorization of Three-Way Binary Data Using Triadic Concepts, Order 30 (2013) 437–454.
4. R. Bělohlávek, Sup-t-norm and inf-residuum are one type of relational product: unifying framework and consequences, Fuzzy Sets Syst. 197 (2012) 45–58.
5. R. Bělohlávek, Ordinally equivalent data: A measurement-theoretic look at formal concept analysis of fuzzy attributes, Int. J. Approx. Reason. 54 (9) (2013) 1496–1506.
6. K. Bazhanov, S. Obiedkov, Optimizations in computing the Duquenne-Guigues basis of implications, Ann. Math. Artif. Intell. 70 (2014) 5–24.
7. P. Butka, J. Pócs, J. Pócsová, On equivalence of conceptual scaling and generalized one-sided concept lattices, Inf. Sci. 259 (2014) 57–70.
8. P. Butka, J. Pócs, J. Pócsová, Representation of fuzzy concept lattices in the framework of classical FCA, J. Appl. Math. (2013) Article ID 236725, 7 pages.
9. P. Butka, J. Pócs, Generalization of one-sided concept lattices, Comput. Inform. 32 (2) (2013) 355–370.
10. I. P. Cabrera, P. Cordero, G. Gutiérez, J. Martinez, M. Ojeda-Aciego, On residuation in multilattices: Filters, congruences, and homorphisms, Fuzzy Sets Syst. 234 (2014) 1–21.
11. C. Carpineto, G. Romano, Concept Data Analysis Theory and Applications, J. Wiley, 2004.
12. M. E. Cornejo, J. Medina, E. Ramírez, A comparative study of adjoint triples, Fuzzy Sets Syst. 211 (2013) 1–14.
13. J. P. Doignon, J. C. Falmagne, Knowledge Spaces, Springer Verlag, 1999.
14. B. Ganter, R. Wille, Formal Concept Analysis Mathematical Foundation, Springer Verlag, 1999.
15. J. Konečný, P. Osička, Triadic concept lattices in the framework of aggregation structures, Inf. Sci. 279 (2014) 512–527.

16. S. Krajči, A generalized concept lattice, Logic J. IGPL, 13 (2005) 543–550.
17. S. Krajči, The basic theorem on generalized concept lattice, in: V. Snášel, R. Bělohlávek (Eds.), Proceed. of the 2nd Intern. Conf. CLA 2004, pp. 25–33.
18. D.R. Krathwohl, A revision of Bloom's taxonomy: an overview, Theor. Pract. 41 (4) (2012) 212-218.
19. O. Krídlo, S. Krajči, M. Ojeda-Aciego, The category of L-Chu correspondences and the structure of L-bonds, Fund. Inform. 115 (4) (2012) 297-325.
20. J. Medina, M. Ojeda-Aciego, Multi-adjoint t-concept lattices, Inf. Sci. 180 (2010) 712–725.
21. J. Medina, M. Ojeda-Aciego, On multi-adjoint concept lattices based on heterogeneous conjunctors, Fuzzy Sets Syst. 208 (2012) 95–110.
22. J. Medina, M. Ojeda-Aciego, Dual multi-adjoint concept lattices, Inf. Sci. 225 (2013) 47–54.
23. J. Medina, M. Ojeda-Aciego, J. Ruiz-Calviño, Formal concept analysis via multi-adjoint concept lattices, Fuzzy Sets Syst. 160 (2009) 130–144.
24. J. Medina, M. Ojeda-Aciego, A. Valverde, P. Vojtáš, Towards biresiduated multi-adjoint logic programming, Lect. Notes Artif. Intell. 3040 (2004) 608–617.
25. J. Medina, M. Ojeda-Aciego, P. Vojtáš, Multi-adjoint logic programming with continuous semantics, Lect. Notes Artif. Intell. 2173 (2001) 351–364.
26. J. Medina, M. Ojeda-Aciego, P. Vojtáš, Similarity-based unification: a multi-adjoint approach, Fuzzy Sets Syst. 146 (2004) 43–62.
27. J. Pócs, Note on generating fuzzy concept lattices via Galois connections, Inf. Sci. 185 (2012) 128–136.
28. J. Pócs, On possible generalization of fuzzy concept lattices using dually isomorphic retracts, Inf. Sci. 210 (2012) 89–98.
29. J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, G. Dedene, Formal concept analysis in knowledge processing: A survey on applications, Expert Syst. Appl. 40(16) (2013) 6538–6560.
30. J. Poelmans, G. Dedene, G. Verheyden, H. Van der Mussele, S. Viaene, E. Peters, Combining business process and data discovery techniques for analyzing and improving integrated care pathways, in: P. Perner (Ed.): ICDM 2010, Springer-Verlag Berlin Heidelberg, 2010, pp. 505–517.
31. U. Priss, Using FCA to Analyse How Students Learn to Program, in: P. Cellier, F. Distel, B. Ganter (Eds.), Proceedings of the 11th International Conference ICFCA 2013, Dresden, Germany, Springer Berlin Heidelberg, 2013, pp. 216–227.
32. U. Priss, P. Riegler, N. Jensen, Using FCA for Modelling Conceptual Difficulties in Learning Processes, in: F. Domenach, D. I. Ignatov, J. Poelmans (Eds.), Contrib. to the 10th Intern. Conf. ICFCA 2012, Leuven, Belgium, 2012, pp. 161–173.
33. A. Protopsaltis, P. Seitlinger, F. Chaimala, O. Firssova, S. Hetzner, K. Kikis-Papadakis, P. Boytchev, Working Environment with Social and Personal Open Tools for inquiry based learning: Pedagogic and Diagnostic Frameworks, Int. J. Sci. Math. Technol. Learn., in press.
34. L. Šnajder, Inquiry approach in learning selected computer science concepts, in: SMEC 2012 : Science and Mathematics Education Conference : Teaching at the heart of learning, Dublin (Ireland), Dublin City University, 2012, pp. 199-204.
35. L. Šnajder, J. Guniš, Inquiry based learning of selected computer sciences concepts and principles, ICTE Journal 1(1) (2012) 28–39.
36. CS Principles: Computation in Action Preview Curriculum, http://csta.acm.org/Curriculum/sub/CurrResources.html
37. The national curriculum in England, framework document, 2013, 224 pages, http://www.education.gov.uk/nationalcurriculum