Promoting computational thinking skills: would you use this Bebras task?

Annalisa Calcagni[0000−0002−5729−8798], Violetta Lonati[0000−0002−4722−244X], Dario Malchiodi[0000−0002−7574−697X], Mattia Monga[0000−0003−4852−0067], and Anna Morpurgo[0000−0003−0081−914X]

Università degli Studi di Milano, Italy, http://aladdin.di.unimi.it

Abstract. Bebras, an international challenge organized on an annual basis in several countries (50 in 2016), has the goal of promoting informatics and computational thinking through attractive tasks. We analyzed Bebras tasks by considering the Computational Thinking (CT) skills each task promotes, starting from the operational definition of CT developed by the International Society for Technology in Education (ISTE) and the ACM-founded Computer Science Teachers Association (CSTA). We argue that such an approach is indeed well-suited to present Bebras tasks, especially with the goal to use them in curricular teaching: framing them as CT enablers helps in making explicit their educational potential, that can be appreciated also by teachers without a formal education in informatics and adapted to a wide range of ages and schools. We explored the viability of our approach by interviewing teachers of different kinds of schools. We propose to use these CT skills also as a tool to classify Bebras tasks, which results in a more uniform distribution of tasks, w.r.t. the one obtained by leveraging content topics.

Keywords: informatics and education, computational thinking, learning contests, Bebras

1 Introduction

The last decade has seen an increasing need for spreading the fundamental concepts of informatics to a vast audience of students, stemming from the belief that some basic concepts of the discipline should be taught even in the first stages of the educational systems. An important contribution to this goal is provided by informatics contests organized worldwide: they are indeed able to stimulate interest among pupils and teachers with different mixes of fun games and safe levels of competitiveness [3]. An initiative that proved to be particularly successful is the Bebras challenge[1] [6,11,10], organized since 2004 on an annual basis in several countries (50 in 2016), with about one and a half million participants in the last edition.

[1] http://bebras.org/

The Bebras community gathers annually to discuss a pool of new tasks to be proposed to pupils; from this pool the organizers in each country choose the tasks to set up the local contests. The tasks should be fun and attractive, be adequate for the contestants' age and the solution should take on average three minutes per task. Moreover, since the contest is aimed at a non-vocational audience, tasks should be independent of specific curricular activities and avoid the use of jargon. In fact, Bebras tasks focus on that part of informatics that should be familiar to everyone, not just computing professionals. This computing core is sometimes called *computational thinking* (CT) and its promotion is one of the key goals in Bebras, whose full name is indeed "International Challenge on Informatics and Computational Thinking" [8].

In the years since its inception, the Bebras community has developed hundreds of tasks. Most of them are proposed as interactive and/or open-ended questions. However, even when answers have to be chosen from a list, there is no unique way of getting to the solution. The tasks can be used to organize new contests, but they can also be the starting point for in-depth educational activities (a recent proposal is [8]). Potentially, their diversity represents a trove in which every teacher could find suggestions and insights for introducing a computational topic or reflecting on it. Indeed, the Bebras community equips the tasks with comments about their key points ("It's informatics"). While some of the Bebras countries (Switzerland, Lithuania, Singapore) provide additional material to teachers, normally related to the latest edition (booklets or websites with tasks, solutions, and an expanded version of their informatics context), in most cases teachers just discuss the tasks with their students in one session after the contest, and there is little evidence that teachers re-use older tasks and integrate them in their curricular teaching activities (see, for instance [12]).

On the contrary, the Bebras corpus could become a considerable resource to teach informatics and computational thinking, provided that tasks are made easy to retrieve and their content is clearly signposted [8]. This statement is consistent with a survey we conducted recently in our country (see Sect. 2).

After having considered other proposals for the classification of the Bebras corpus (see Sect. 4), we decided to analyze it by means of the operational definition of computational thinking [1] developed by ISTE (International Society for Technology in Education) and CSTA (Computer Science Teachers Association). We first identify the CT skills that are mentioned in the definition and that are relevant when solving a Bebras task; for each of these skills we then give a description that shows which kinds of tasks can promote such skill; finally we analyse several tasks in order to detect the skills they promote. We argue that such approach is indeed well-suited to present Bebras tasks since it helps in making explicit their teaching potential. First, differently from classifications based on a taxonomy of informatics content, this approach is more suitable for identifying the cognitive skills involved in a task. Second, it can be appreciated also by teachers without a formal education in informatics (still the majority, in primary and non-vocational secondary schools): in fact the CT skill set uses terms and concepts which can be grasped even without a deep knowledge of informatics technicalities. Finally, CT skills can be adapted to a wide range of ages and schools, still maintaining their peculiarity, distinct from generic logical/analytical thinking.

We explored the viability of such approach by interviewing individually some teachers of different kinds of schools. Teachers were provided with a short description [4] of the CT skills mentioned above and then were requested to associate some Bebras tasks with those CT skills. More precisely, for each task and each CT skill, we asked teachers the following question *"Would you use this task to promote such skill?"*, and then discussed their answers with them, to understand their motivations. The outcome of such interviews seems to confirm the good potential of the approach (see Section 3).

We also propose to use this CT skills as a tool to classify Bebras tasks. We manually classified the pool of tasks prepared for the 2016 contest (120 tasks) and we verified that the resulting distribution among classes is more uniform than that obtained by leveraging content topics.

The paper is organized as follows: in Sect. 2 we describe how the CT skills mentioned in the ISTE/CSTA definition are related to Bebras tasks, and in Sect. 3 we reports the findings of our interviews with teachers. In Sect. 4 we apply this approach to obtain a classification of tasks according to the CT skills they promote, and in Sect. 5 we draw some conclusions.

2 Bebras and CT skills

After Bebras's last edition (2016), we conducted a survey among the teachers who had registered in the site of the contest in our country: an online questionnaire was filled in by 46% of those partecipating in the last edition (342 over 742 teachers); for most of them (71%) this was the first partecipation in Bebras (w.r.t. the previous edition, the number of participant teachers has more than doubled); in general we measured a strong appreciation for the initiative.

Among other questions, we were interested in understanding how teachers make use of tasks (or intend to) in their classes. When asked whether they intend to use Bebras tasks in their classes, 30.9% of the respondents answer affirmatively (in particular 20.3% say that they've already used them); 65% show interest in such a possibility but are uncertain (in particular 20.3% of the respondents choose "I would like to, but I cannot figure out how"); negative answers were marginal (3.6%).

A promising 43.7% of the respondents believe that Bebras tasks can be used for curricular activities (the subjects more frequently mentioned are mathematics, informatics, technology, depending on the grade and kind of school). Moreover, we asked which options would make Bebras tasks easier to re-use for teaching: besides some logistic issues concerning the access and form of online and printed materials, the most appreciated proposals are: to have examples of teaching units that use Bebras tasks (69.2%); to have tasks categorized/arranged according to the computational thinking skills they require and/or promote (51.5%); to have tasks categorized/arranged according to the informatic theme they refer to (46.4%).

Hence teachers are clearly eager to have descriptions of the tasks that could help them in finding the ones more suitable to their goals: both CT and informatics seem useful for them, but while all tasks traditionally have an "It's informatics" section designed to explain the informatic idea behind it, no effort is currently made by Bebras authors to identify and highlight the cognitive skills they require. Thus, we decided to focus on CT skills, which seem to us the core educational value of Bebras tasks, since they keep their computational peculiarity, while being accessible even without a deep knowledge of the more technical details of the discipline of informatics.

According to the operational definition proposed by ISTE and CSTA [1], CT is a problem-solving process that includes (but is not limited to) the following characteristics.

a) Formulating problems in a way that enables us to use a computer and other tools to help solve them.
b) Logically organizing and analyzing data.
c) Representing data through abstractions such as models and simulations.
d) Automating solutions through algorithmic thinking (a series of ordered steps).
e) Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources.
f) Generalizing and transferring this problem solving process to a wide variety of problems.

The first and last skills in the definition (a and f) are almost never addressed by a single Bebras task, mainly due to its brevity, even though the contest aims at helping the development of such skills in the long-term, by working on and solving several tasks. Instead, most Bebras tasks deal with organization, analysis, representation of data (skills b and c), algorithmic thinking (skill d) or design, analysis and implementation of algorithmic methods (skill e). Thus, we believe such a definition can be fruitful for analyzing and describing tasks.

ISTE/CSTA also propose a vocabulary of CT skills with a progression chart suggesting possible activities for different ages and subjects [2]. The intended goal of the vocabulary is "to unpack the operational definition by listing CT concepts implicit in the operational definition". The vocabulary lists and explains nine terms, giving example activities suitable for the age groups: Data Collection, Data Analysis, Data Representation, Problem Decomposition, Abstraction, Algorithms & Procedures, Automation, Simulation, Parallelization. This level of description, however, seems to enter in the explicit domain of the practice of informatics: as such, it could provide further enrichment for the "It's informatics" section which accompanies each task, but it is less useful to highlight their teaching potential. In general, different (and only partially overlapping) definitions of CT exist (a good recent survey can be found in [5], which discusses also frequent misconceptions of CT by primary teachers), but in this proposal we decided to focus on the operational ISTE/CSTA's definition as the one with the right granularity to emphasize the specificity of computational cognitive skills versus a generic analytical approach, while being accessible to teacher without any formal education in informatics.

In the following we illustrate the CS skills mentioned in the ISTE/CSTA operational definition by referring to Bebras tasks that can promote such skills.

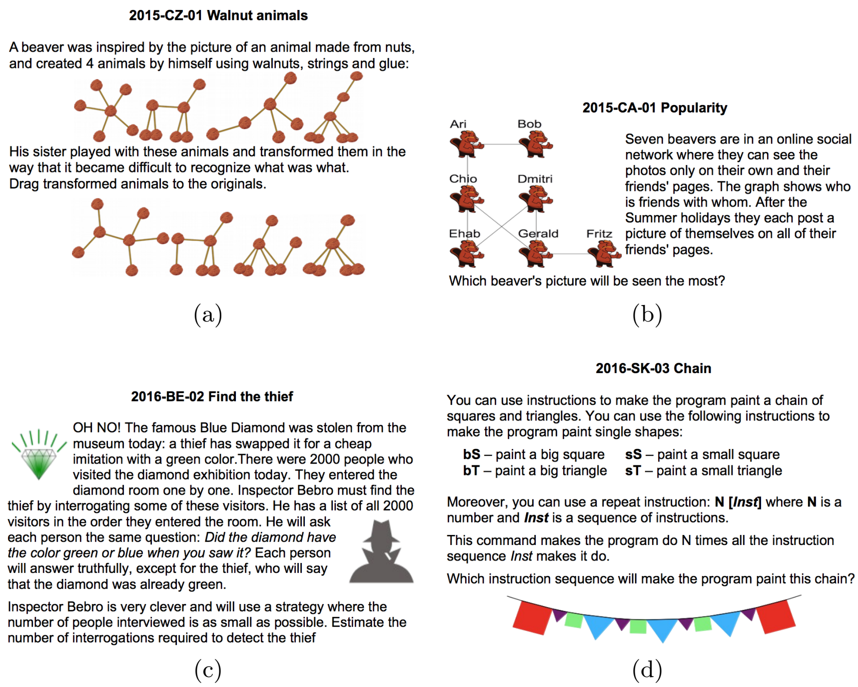

Fig. 1. Four tasks suitable to teach how to logically analyze data (a), how to represent information (b), how to identify strategies and analyse algorithmic solutions (c), and how to implement algorithmic solutions (d).

2.1 Logically organizing data

Typical tasks that promote this skill deal with ways to organize data according to given criteria, with (generally hidden) references to databases or set theory: they may ask to execute a query over a table of records representing a set of objects, to split a set of items into categories according to their characteristics, or to pick the misplaced object in a figure.

Other tasks for this skill focus on organizing data so that they enjoy relevant properties: that is the case for instance of cryptography, where we want a message not to be understandable even if eavesdropped, data compression, where we organize data in order make them easier to store or transmit, or correction codes, where we add bits to detect or recover possible errors in the representation of stored or transmitted data.

This skill is promoted also by tasks where data structures are used to organize data for processing. For instance, a task may concern the use of binary search trees to find quickly any element in a set, without considering the whole set; similar examples can be done for heaps, queues, stacks, and so on. However, if a data structure is used to represent an intrinsic property of the data (e.g., graphs for binary relations, or trees for hierarchical relations), the task will be most suitably related to the skill *Representing information*.

2.2 Logically analyzing data

Many tasks for this skill are perceived as “logical problems”, since they require logical inference, deductive reasoning, and drawing conclusions about the data presented in the task.

Other tasks ask to check whether the data of the problem satisfy certain properties; often such properties are not straightforward but some reasoning, accurate observations (e.g., recognizing patterns), or a systematic approach are needed to come to the right conclusion. This kind of tasks are well represented by “Walnut animals”, proposed by Czech delegates in 2015 and depicted in Figure 1(a): to solve the task pupils need to abstract from the specific features of animals and consider only their structure, then pairs of isomorphic animals can be matched by analyzing their properties like the number of nuts, or the number of connections from each nut.

2.3 Representing information

Typical tasks for this skill deal with the digital representation of numbers, images, colors, and sets of objects in general, or the visual representation of data with diagrams like histograms or charts.

Other tasks introduce data structures to represent relevant properties of the data, e.g., graphs for binary relations, or trees for hierarchical relations. For instance, consider “Popularity”, proposed by Canadian delegates in 2015 and depicted in Figure 1(b) where a graph is used to represent and visually show the friendship relation of people in a social network.

2.4 Algorithmic thinking

Algorithmic thinking deals with transforming an intuitive idea into a form suited to be processed automatically. Hence it allows, for instance: to design a systematic method to tackle a problem, starting from an intuitive approach; to transform an intuitive idea of how to accomplish a task into a step-by-step procedure that achieves the goal; to give a synthetic description of a situation or a process, by detecting and exploiting its patterns and regularities; to start from a set of examples or an informal description and formalize a rule that can be applied in general; and so on.

Thus, typical tasks that promote this skill require to manipulate data following a formal procedure (i.e., a sequence of ordered steps or moves) or a set of rules or primitives. Tasks may require to execute some procedure or to compute or recognize its output; to apply some transition rules to a system in a given configuration; to predict the final state of a process described by a diagram (e.g., the transition diagram of a finite state automaton); to decompose a problem into components; to combine primitive operations in order to compute a result or accomplish a task; to systematically enumerate or examine all the possible cases that can occur in a given context; and so on.

Notice that when the focus of the task is on implementing (analyzing, or devising, respectively) an algorithmic solution, then the task should be most suitably related to skill *Implementing algorithmic solutions* (*Analyzing algorithmic solutions*, or *Identifying strategies*, respectively).

2.5 Identifying strategies

This skill concerns problem solving and in particular finding a suitable algorithmic strategy. Typically, formulating a solution algorithmically (so that it could be automated) is not sufficient, and tasks that promote this skill usually require to devise a non-trivial idea to address the problems they present.

The tasklet "Find the thief", proposed by Belgium delegates in 2016 and depicted in Figure 1(c), is a good representative for this task since, in order to estimate the number of interrogations required to detect the thief, students need to understand that examining all visitors sequentially is too time consuming and they need to address the problem with an original approach (in this case, binary search). Thus, the task would be a good choice to promote the ability of devising strategies to solve problems. Notice that, dealing the task also with complexity issues, it could be used to promote the next skill, too.

2.6 Analyzing algorithmic solutions

This skill is promoted by tasks concerning global characteristics of the considered algorithm, like correctness or complexity. Thus, tasks for this skill may require to examine an algorithm (or, more generally, a computation method) in order to understand its semantic, predict its overall behavior, determine its invariant properties, estimate how many resources it will consume. Moreover, other typical tasks are those inspired by optimization problems, in that they require to compare and evaluate different approaches in order to find the best one.

An example is given again by "Find the thief" (Figure 1(c)): if a teacher wants to introduce computational complexity, s/he can start from this task whose solution relies on binary search and requires a complexity analysis in order to estimate the right number of interrogations needed.

2.7 Implementing algorithmic solutions

Tasks that promote this skill may be referred to as programming or coding tasks since the focus is on the implementation of algorithms according to a formal syntax defined in the task. If the algorithm to be implemented is not straightforward, then the task will also belong to skill *Algorithmic thinking* or *Identifying strategies.*

Tasklet "Chain", proposed by Slovak delegates in 2016 and depicted in Figure 1(d) is apt to teach this skill: a simple programming language is provided in the text of the task and a small program has to be implemented to answer the question. Actually, the multiple choice form of the question allows to find the right answer by simply executing four programs; however, from a didactic point of view, the tasks is perfect to introduce programming to young pupils.

3 Key informant interviews

To explore the suitability of using CT skills to present Bebras tasks and make them easier to retrieve as a teaching resource, we used the *key informant technique*, *i.e.,* we interviewed eight teachers of different kinds of schools, selecting them for their first-hand knowledge, expertise and reflective practice in teaching. Half of them were quite expert of Bebras tasks, in that they had participated in the Bebras challenge in the past or had contributed in the preparation of tasks themselves; the others instead knew Bebras only a little and are not expert of computing education, but are very much concerned about curricular issues, professional development of teachers, or teaching methodologies; some of them had participated in our teacher training courses in computer science education.

Teachers were first provided with a short description [4] of the CT skills presented in Sect. 2 and were then requested to associate some Bebras tasks with those CT skills. More precisely, for each task and each CT skill, we asked them the following question *"Would you use this task to promote such skill?"*, emphazising that each task could be associated with more skills. To express their answers, we suggested that they fill out a double entry table (task/skill), where the answers might range between 0 and 3: 0 = 'absolutely not'; 1 = 'more no than yes'; 2 = 'more yes than no'; 3 = 'absolutely yes'.

We then (qualitatively) interviewed them and asked them to discuss their answers with us, in order to understand their motivations. The interviews were loosely structured, mainly relying on three wide issues.

- Are the (descriptions of the) skills clear? Are there terms or expressions that you do not understand or about which you are not sure or that are ambiguous?
- Why do you relate a certain task to a certain skill? (In particular we delved into those cases where the association between a task and a skill was unexpected for us).
- Do you think that if tasks were presented with this approach (that is considering the computational thinking skills they can promote), the educational value of the tasks would emerge more clearly? That is, if presented this way, would it be easier for a teacher to use them in the curricular educational activity, as a curricular resource?

During the interviews, we first noticed that despite the effort of removing (or reducing to the minimum) computer science technicalities, our description of skills still revealed some computer science implied meanings that teachers are not familiar with, and hence needed to be clarified, especially for teachers of primary schools. For instance, when we use terms like *problem* and *solution* we implicitly think of *computational* problems and solutions, while in primary schools a problem is what we would call an *instance of a problem*, and vice versa a computational problem would be seen as a class of similar problems in the primary school meaning. Similarly, for primary school teachers with no formal computer science education, a *solution* is simply an answer to an issue, while when we write "analysing a solution" we are usually thinking of a computational solution for a (computational) problem, *i.e.,* an automatic method to find the correct answer to any instance of the problem. Other expressions that ran into a similar misunderstanding are *representing data* and *organizing data*. Indeed, the digital representation of data as usually meant in computer science is not common knowledge for primary school teachers. When a task deals with some formal/symbolic representation of data, teachers realize that, to understand and tackle the task, one needs to rearrange or reformulate the data somehow and hence they associate the task with the "organizing data" skill. Also the terms *implement* and *coding* were not broadly familiar and needed some explanation. Despite the need to clarify these terms, in general we got confirmation of our hypotheses. We agreed on most associations between tasks and CT skills the teacher highlighted, which seems to confirm that the definitions of skills are understandable and their use to describe tasks is feasible. Teachers seem to appreciate the use of CT skills to analyze Bebras tasks since "make the underlying skills of a task explicit helps in choosing more consciously what to work on and how". The CT skill lens seems to foster the identification of the educational potential of tasks; indeed, often teacher highlighted associations between tasks and skills that we did not expect, but they were usually able to support their association with a convincing reasoning, or with clear examples (for instance envisaging original ways to use the tasks in the classroom, in order to promote a skill that was not directly addressed by the question of the task itself). From the interview we also got some new ideas and insights. We noticed that tasks that required some kind of analytic thought were often associated by teachers with "algorithmic thinking". For instance, solving a task by using the technique of *decomposition* or *step-by-step* reasoning was often associated with algorithmic thinking even when no algorithm or formal procedure or rule was involved. This could be acceptable at the primary school level but, for older kids, algorithmic and analytical thinking should be distinguished more neatly by teachers, in order to appreciate the true computational thinking value of a task, and to promote it in general education.

4 Categorization of Bebras tasks using CT skills

Bebras tasks categorization is also important for the contest itself in that it helps authors span on different topics/skills when they produce tasks and it helps national organizers in putting up a varied contest covering as many informatics topics and computational thinking skills as possible. In fact tasks categorization is an issue in the Bebras community since the beginning. The classification proposed in 2008 [7] turned out to be too coarse to be applied to a variety of tasks; for example Table 1 shows how 120 tasks were classified in the 2016 edition (in fact only 92 of them were indeed classified, since in the other cases the authors of the tasks did not mark them, and we did not complete the classification).

Other categorizations based on the informatics content were proposed first in [12] where four components of informatics education are considered (digital literacy, programming, problem solving, and data handling) and more recently in [13] where a hierarchical classification is suggested based on Schwill's master ideas [14] (algorithmization, structuring, formalization); a classification starting from the Bloom Taxonomy of cognitive skills is presented in [10].

Finally, in [9] a two-dimensional classification is proposed: the first dimension is based on informatics knowledge and proposes five informatics "domains", the second one is based on five CT skills. The five informatics domains, described by means of several technical keywords (*e.g.,* bubble sort, binary tree, *etc.*), were also proposed and tentatively used to classify the 2017 tasks during their creation and discussion, with the vast majority of tasks falling in the first two categories. We also categorized 120 tasks from the 2016 edition within these five domains; the results are given in Table 2. The second dimension, including abstraction, algorithmic thinking, decomposition, evaluation, and generalisation as CT skills, is indeed in the same direction we are proposing here. As such, however, it is very high level, with some categories (especially abstraction) that risk to be too ubiquitous to be useful. We chose not to apply such a classification on our own, in order to avoid misinterpreting the original authors' intention. We instead applied our own scheme to classify the same 120 tasks from the 2016 edition, with the results shown in Table 3, also showing cases in which two (or more) categories were used together, in bold the number of times a category was used alone.

5 Conclusions

Bebras tasks are a considerable teaching resource, but unfortunately they are mostly underused beyond the contest.

We analyse and describe Bebras tasks by using the operational definition of computational thinking and identifying seven fudamental CT skills, concerning the organization, analysis, representation of data, algorithmic thinking, and the design, analysis and implementation of algorithmc methods.

Such an approach applies to a wide range of ages and schools and can be described with terms and concepts that do not rely on a strong or wide knowledge

Table 1. Classification of 120 tasks from 2016 edition according to [7]

Topics and concepts	No. Tasks
Algorithmic thinking	67
Information comprehension	27
Structures, patterns and arrangements	16
Puzzles	16
Social, ethical, cultural, international, and legal issues	1
Using computer systems	3

Table 2. Classification of 120 tasks from 2016 edition according to [9]

Topics and concepts	No. Tasks
Algorithms and programming	87
Data, data structures and representations	37
Computer processes and hardware	15
Communication and networking	3
Interactions, systems and society	2

Table 3. Classification of 120 tasks from 2016 edition: number of tasks and category pairs; bold figures count the occurrences of a category alone

CT skill	No. Tasks	OD	AD	RD	AT	ID	AS	IS
Organizing data (OD)	17	**4**	8	2	3	0	0	0
Analyzing data (AD)	31		**11**	8	3	1	0	0
Representing data (RD)	25			**8**	0	0	7	0
Algorithmic thinking (AT)	40				**19**	0	5	9
Identifying strategies (ID)	11					**5**	5	0
Analyzing strategies (AS)	20						**3**	0
Implementing a solution (IS)	15							**6**

of computer science, hence also teachers without a formal education in informatics can understand it and relate it with their curricular teaching activity. Moreover, differently from other classifications based on a taxonomy of informatics topics, this approach also helps in detecting the cognitive skills involved by a task, thus it would make their educational potential more explicit.

We gathered feedbacks about this approach by interviewing teachers and applied it also to classify Bebras tasks. We will also collect feedbacks by building a website where teachers will be able to retrieve tasks according to their potential in promoting CT skills. We are currently working to link such approach to the recommendations of the Italian Ministry of Education for non-vocational schools.

Finally, and overall, we believe the awareness of the importance of a CT-based classification can improve the way tasks will be written in the future. In particular we suggest to expand the Bebras task templates with a new "It's computational thinking" section, containing for each class an articulated answer to the question "Is this task suited to teach this CT skill?".

Acknowledgments

We would like to thank the whole Bebras community for the great effort spent in producing exciting tasklets and the key informant teachers.

References

1. International Society for Technology in Education & Computer Science Teachers Association: Operational definition of computational thinking for K-12 education. http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf (2011)
2. International Society for Technology in Education & Computer Science Teachers Association: Operational definition of computational thinking for K-12 education. http://www.csteachers.org/resource/resmgr/472.11CTTeacherResources_2ed.pdf (2017)
3. V. Lonati, M. Monga, A. Morpurgo, M. Torelli: What's the fun in informatics? Working to capture children and teachers into the pleasure of computing. In: Kalaš, I. and Mittermeir, R.T. (ed.) Informatics in schools: contributing to 21st century education. Proc. of the Int. Conf. on Informatics in Schools: Situation, Evolution and Perspectives. LNCS, vol. 7013, pp. 213–224. Springer-Verlag (2011)
4. Violetta Lonati, Dario Malchiodi, Mattia Monga, Anna Morpurgo: Bebras as a Teaching Resource: Classifying the Tasks Corpus Using Computational Thinking Skills. In: Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE 2017, Bologna, Italy, July 3-5, 2017. p. 366. ACM (2017)
5. Corradini, I., Lodi, M., Nardelli, E.: Conceptions and misconceptions about computational thinking among Italian primary school teachers. In: Proceedings of the 2017 ACM International Computing Education Research (ICER'17) (Aug 2017), to appear
6. Dagienė, V.: Supporting computer science education through competitions. In: Proc. 9th WCCE 2009, Education and Technology for a Better World, 9th WCCE 2009, Bento Goncalves (2009)
7. Dagienė, V., Futschek, G.: Bebras international contest on informatics and computer literacy: Criteria for good tasks. In: ISSEP 2008. LNCS, vol. 5090, pp. 19–30 (2008)
8. Dagienė, V., Sentance, S.: It's computational thinking! Bebras tasks in the curriculum. In: ISSEP. LNCS, vol. 9973, pp. 28–39 (2016), http://dx.doi.org/10.1007/978-3-319-46747-4_3
9. Dagienė, V., Sentance, S., Stupurienė, G.: Developing a two-dimensional categorization system for educational tasks in informatics. Informatica 28(1), 23–44 (2017)
10. Dagienė, V., Stupuriene, G.: Informatics education based on solving attractive tasks through a contest. KEYCIT 2014 - Key Competencies in Informatics and ICT pp. 97 – 115 (2015)
11. Haberman, B., Cohen, A., Dagienė, V.: The beaver contest: attracting youngsters to study computing. In: Proceedings of the 16th annual joint conference on Innovation and technology in computer science education. pp. 378–378. ACM (2011)
12. Kalos, I., Tomcsanyiova, M.: Students' attitude to programming in modern informatics. INFORMÁTICA NA EDUCAÇÃO: teoria & prática 12, 127–135 (2009)
13. Pohl, W., Hein, H.W.: Aspects of quality in the presentation of informatics challenge tasks. In: ISSEP 2015 local proceedings. pp. 21–22 (2015)
14. Schwill, A.: Fundamental ideas of computer science. Bulletin-European Association for Theoretical Computer Science 53, 274–295 (1994)