Informatics in Education, 2021, Vol. 20, No. 4, 717–736
© 2021 *Vilnius University, ETH Zürich*
DOI: 10.15388/infedu.2021.30

A Comparison of Abstraction and Algorithmic Tasks Used in Bebras Challenge

Jiří VANÍČEK, Václav ŠIMANDL, Patrik KLOFÁČ

Department of Informatics, University of South Bohemia, České Budějovice, Czech Republic
e-mail: {vanicek, simandl, pklofac}@pf.jcu.cz

Received: February 2021

Abstract. This paper focuses on the analysis of Bebras Challenge tasks to find Informatics tasks that develop abstract thinking. Our study seeks to find which Bebras tasks develop abstraction and in what way. We analysed hundreds of tasks from the Czech contest to identify those tasks requiring participants to abstract directly or use abstract structures. Results show that an agreement among experts on stating which task is focused on abstraction is at a moderate level. We discovered that tasks focused on abstraction occur four to five times less frequently in sets of contest tasks than algorithmic tasks. Our findings proved that abstract tasks results compared with algorithmic ones did not differ in neither age nor gender group of contestants.

Keywords: computer science education, secondary school, Bebras Challenge, abstract tasks, task difficulty, gender comparison, age comparison.

1. Introduction, Theory of Abstraction

Computational thinking, originally used by Wing (2006), has become a keyword in the field of computer science (abbr. CS) education. During years many definitions of this term have appeared (Wing, 2010; Wing, 2014; Selby and Woolard, 2014; Denning, 2011; The Royal Society, 2012; Aho, 2012; Hu, 2011). According to Wing, computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent (Wing, 2010). There has not been a unity of opinion among authors about which concepts are covered by computational thinking. In this article we will proceed from the definition of Selby and Woollard (2014) according to whom computational thinking incorporates the concepts of automation, abstraction, decomposition, algorithmic design, evaluation and generalisation (Selby and Woollard, 2014). As Wing (2010) claims, the most important and high-level thought process in computational thinking is the abstraction process.

The term abstraction means different things to different people and in different contexts (Cetin and Dubinski, 2017). In general, it means the act or process of leaving out, out of consideration, one or more qualities of a complex object so as to attend to others; or the act or process of imaginatively isolating or considering apart common properties or characteristics of distinct objects (Gove and G. & C. Merriam Company, 1981). The concept of abstraction can be found not only in CS and mathematics (Cetin and Dubinski, 2017), science and social science (Barr and Stephenson, 2011), but also in art or music (Kramer, 2007). As Cetin and Dubinski (2017) state, in CS the most common meaning of abstraction of a concept is extraction, that is, the idea of considering common features of several examples and building a structure or category which has all of these features. Wing (2017) claims that abstraction is used in CS in defining patterns, generalizing from specific instances and parameterization. It is used to let one object stand for many. It is used to capture essential properties common to a set of objects while hiding irrelevant distinctions among them (Wing, 2017). Dagienė, Sentance and Stupurienė (2017) assert that in CS abstract thinking involves abilities like removing unnecessary details, spotting key elements of a problem and choosing a representation of a system.

Since there are a number of views on abstraction as a concept as well as a process, it would be interesting to research the presence of this part of computational thinking in an Informatics curriculum. However, in many countries, including the Czech Republic until 2021, as CS Education is not compulsory in secondary school, students are likely to be exposed to computational thinking and abstraction for the first time while participating in the Bebras Challege, which is the focus of our study.

2. Abstraction in Bebras Tasks

Bebras Challenge is a contest organized for schools to promote computational thinking. The contest is held online in over 50 countries around the world and a good deal of new tasks is developed for it every year (Bebras Challenge, 2016). So-called Bebras tasks focus on different parts of Informatics and development of computational thinking. Due to this direction, they should consist of tasks which develop abstraction and aim at the usage of abstract thinking.

The need for development of contest tests and new tasks lead to using so-called task types. According to the related features, tasks are put into groups which are given type names by the prevailing feature. New tasks are assigned to these types. Such types can be of different nature according to their determination: pupil´s age, presumed difficulty, an area of Informatics, a topic of Informatics education, a part of computational thinking, etc.

Various methods of classification have appeared during Bebras Challenge history, dating back to the year 2004 in Lithuania (Dagienė *et al.*, 2017). The oldest one, which we were able to find, comes from the year 2006 and it distinguished the following task types: general logic, ICT in everyday life, practical and technical issues, information comprehension, algorithms and programming, mathematics underlying computer science, history and trivia (Opmanis *et al.*, 2006, p. 511).

The first official classification, which was used for more than 10 years, was established by the Bebras Organizing Committee on the basis of research by Dagienė and Futschek (2008) who discussed criteria for a good task. That classification consisted of types ALG (algorithmization), INF (information comprehension and representation), STRUC (structures including graphs), PUZ (puzzles, logic, problem solving) and SOC (social, ethical aspects, everyday use of ICT).

In the year 2017 those criteria were altered to cover newly emergent tasks proportionally better on account of the fact that the original type such as SOC stopped being filled with new tasks. We present a description of this classification here as it is the official classification of the contest due to which it is possible to grasp the concept as for how much one can reconstruct abstraction from these individual descriptions:

- **ALP:** Algorithms and Programming (Algorithm, program, programming language, variables, loop, function, parameters, recursion, encapsulation, inheritance, objects, optimization, searching, sorting, computational complexity, operations AND, OR, NOT, etc.).
- **DSR:** Data, Data Structures and Representations (Information, binary and hexadecimal representations, string, integer, array, record, attributes, linked list, queue, stack, binary tree, character encoding, databases, data mining, flowcharts, fractals, graphs, hash table, etc.).
- **CPH:** Computer Processes and Hardware (Operating systems, parallel processing, peripherals, image processing, sound processing, grid computing, priorities, RAID array, registers, multithreading, deadlock, fetch-execute cycle, scheduling, memory, cloud computing, turing machine, etc.).
- **COM:** Communications and Networking (Client/server, computer networks, cryptography, e-commerce, encryption, parity bit, protocols, topologies, etc.).
- **ISS:** Interactions, Systems and Society (Classification, graphical user interface, design, interaction, computer use, robotics, virus, ethics, social issues, etc.) (Dagienė *et al.*, 2017).

In addition to this official classification, different authors suggested or used different classifications of Bebras tasks. Slovakian researchers suggested four task types for younger participants, such as digital literacy, programming, problem solving and data handling (Kala and Tomcsányiová, 2009); and later digital literacy, programming, algorithmic tasks and logical tasks (Budinská *et al.*, 2017).

None of the classifications, which are used in the Bebras community, have ever mentioned abstraction as a task type. The possible cause could be the fact that the contest is older than the term computational thinking; in the beginning, Bebras Challenge was characterized as a contest developing Informatics. Contest tasks were not applied to pedagogical terms and it was considered natural to classify tasks according to the prevailing field of Informatics that occurs in this contest. Only in the year 2015 did the statutes of the Bebras community, published by the International Bebras Committee, specify Bebras Challenge as an International Challenge on Informatics and Computational Thinking (Bebras Community, 2015); the same formulation is used by the current statutes (Bebras Community, 2020).

In the archives of contest tasks, both international ones and those used by every individual national contest, we are able to, for example, find and choose algorithmic tasks, but no tasks regarding abstraction. Abstraction has only recently been studied in the researches, such as a multinational study analysing performance data on Bebras tasks by comparing data from Italy, Finland, Lithuania, Australia, South Africa, Canada and Switzerland. That study was realized by Izu *et al.* in 2017; they used the following classification: data collection, data analysis, data representation, problem decomposition, abstraction, algorithms & procedures, automation, parallelization and simulation (Izu *et al.*, 2017). The abstraction type is described as reducing complexity to define the main idea and tasks belonging to this type are characterized as problems that ask for the creation of a formula, the distillation of broader ideas out of narrower concepts, tasks for finding rules that apply to a given problem, finding a pattern to model some behaviour or identifying essential facts about a structure or problem to verify correct answers (Izu *et al.*, 2017). As an illustrational example of abstraction, there is presented a graph-topological task of Chestnut animals, coincidentally Czech, with the international code 2015-CZ-01. The task is classified as abstraction type because "in order to match isomorphic graphs students are required to think of their structure at a more abstract level" (Izu *et al.*, 2017, p. 46).

The other case of using an abstract thinking type in relation to Bebras tasks research is Ternik *et al.* (2020). Their model of computational thinking skills includes six CT skills: Algorithmic thinking, Generalization, Decomposition, Evaluation, Abstraction and Modelling and Simulation. Their choice of types is almost in line with the definition of computational thinking by Selby and Woollard (2013). The proximity of composition between types of CT skills and CT thought processes in both sources can be considered as the basis for a statement that both classifications perceive abstraction similarly. The task Colouring Inn, which thematically relates to a problem of 4 colours, may serve as an example of tasks that include abstraction, as stated in Ternik *et al.* (2020). The reasoning behind the decision to classify this task as the abstract thinking type seems to be rather general: In order to solve the task, students have to identify key elements in the problem and remove unnecessary details (Ternik *et al.*, 2020, p. 35).

From the above mentioned facts, it emerges that it is difficult decision whether a Bebras task develops abstraction, not only due to the missing long-term experience of authors and the whole Bebras community with defining this task type, but also due to the missing specific examples which would help determine abstraction. This is in line with the findings of Ternik *et al.* (2020) which point out that the abstract thinking type is the most challenging to categorize.

3. Aim

Our main research aim is to ascertain which tasks can be considered abstract in Bebras Challenge in years 2014–2020 in the Czech Republic. While doing so, we distinguish between tasks focused on abstraction as a fundamental part and tasks containing abstraction besides other mental operations.

A part of the main research aim is to compare how abstract tasks are perceived by experts who have experience with Bebras tasks, and to what extent these tasks can be easily identified.

The additional aim is to compare frequency of abstract and algorithmic tasks.

Further aims are to compare tasks identified as abstraction ones in terms of difficulty to algorithmic tasks with regards to the age and gender of research participants and to compare the successfulness of solving abstraction tasks among genders.

The following research questions (RQ) were formulated on the basis of the research aim:

- **RQ1:** To what extent do experts agree on the classification of abstract tasks?
- **RQ2:** To what extent are abstract tasks offered in Bebras Challenge in comparison to frequency of algorithmic tasks?
- **RQ3:** Are tasks identified as those focused on abstraction at the same difficulty level as algorithmic tasks for all contestants ($RQ3_{all}$), younger contestants ($RQ3_{young}$), older contestants ($RQ3_{old}$), girls ($RQ3_{girls}$) and boys ($RQ3_{boys}$)?
- **RQ4:** Are tasks identified as tasks containing abstraction at the same difficulty level as algorithmic tasks for all contestants ($RQ4_{all}$), younger contestants ($RQ4_{young}$), older contestants ($RQ4_{old}$), girls ($RQ4_{girls}$) and boys ($RQ4_{boys}$)?
- **RQ5:** Is there a difference in the successfulness of solving tasks identified as those focused on abstraction between girls and boys at younger contestants ($RQ5_{young}$) and older contestants ($RQ5_{old}$)?
- **RQ6:** Is there a difference in the successfulness of solving tasks identified as those containing abstraction between girls and boys at younger contestants ($RQ6_{young}$) and older contestants ($RQ6_{old}$)?

4. Methodology

4.1. *Identification of Tasks Focused on Abstraction*

We concentrated on means of classifying an abstract task properly. To define abstract tasks, we did not find a suitable methodology which would not only repeat the basic definition of abstraction and characteristics of this concept. We decided to select abstract tasks on the basis of professional evaluation and discussion among experts, similarly to Ternik *et al.* (2020).

In our case, the team of experts consisted of three teachers' educators who have been working with Bebras tasks for a long period of time, who have also focused on task types according to the official Bebras classification and who understand the philosophy of the contest. All of them speak Czech and later they became authors of this article. At first we wanted to ascertain to what extent their point of view is similar regarding the identification of an abstract task.

Experts took into account the definitions of abstraction according to Cetin and Dubinski (2017), Wing (2017) and how to spot the use of skill to abstract (Dagienė *et al.*, 2017). At first, each expert decided on their criteria by which they divided tasks into strongly abstract, weakly abstract and without abstraction. After collective discussions those criteria were used for the description of the three following groups, into which experts divided each monitored task:

- ***ABS*** (in capital letters) – tasks focused on abstraction: abstraction is the most significant mental activity or part of computational thinking that needs to be used by contestants to solve the task; the solution of the task is based on abstraction; a pupil has to abstract in this task
- ***abs*** (in small letters) – tasks containing abstraction: abstraction occurs in the task; an abstract model is used here, however, abstraction in not the most significant mental activity leading to the solution; it is not necessary for pupils to form an abstract object on their own, nevertheless, they have to be able to orient themselves in it
- ***no abs*** – (no abstraction) in this task we did not discover a measure of abstraction which would require further study

In the following text we also use an expression *ABS+abs* which represents all abstract tasks, thus not only tasks focused on abstraction, but also tasks containing abstraction.

Those experts examined all tasks of the contest for 2 age categories – younger (the category called Benjamin, 11–13-year-old pupils) and older (Junior, 15–17-year-old pupils). Tasks were examined in two stages. In the first stage, each of the 84 tasks from the Czech edition of the contest in years 2017–2019 was independently classified by experts. After this stage, experts gathered and thoroughly discussed their classification of tasks, they resolved criteria and explained why a certain task in question belongs to the type *ABS*, *abs* or *no abs*. They eventually reached an agreement on classifying each task as one of the types.

In the second stage of examination, experts, being now aware of features of abstraction that lead to type definitions, again independently went through 114 tasks from the Czech edition of the contest in years 2014–2016 and the year 2020, which took place in the meantime. The second stage was finished in the same way as the first one; by a collective decision about classifying each contest task as one of the types *ABS*, *abs*, *no abs*.

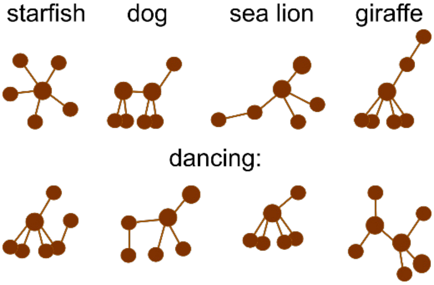
4.2. *Specific Examples of Abstract Tasks Used in Bebras Challenge*

4.2.1. *Type ABS*

First, let us demonstrate on specific examples which tasks were selected for the type *ABS*, in which abstraction is the dominant mental activity for the solution to the task. We will select one task from the older category and one task from the younger category. We state the pupils´ age to which these tasks were applied in the Czech edition of the contest, and also instructions of tasks in the same way as being used in the Czech edition.

Chestnut animals, 9–13 years, code 2015-CZ-01

Animals made of chestnuts came to life and danced wildly. Can you tell which is which?

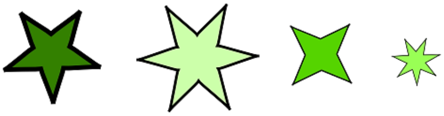


Argumentation: In this task for younger pupils contestants have to recognize the key factor which decides about an unambiguous match between graphs in the first and second line. They have to realize that the key factor is not the graph of the same shape, but the number of node or edges coming from one node.

This task also occurs as an example abstract task in Izu *et al.* (2017), a fact unknown to our experts.

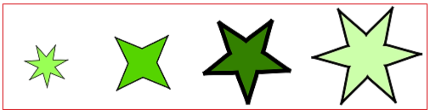
Non-ordered stars, 12–13 years, code 2014-CZ-05

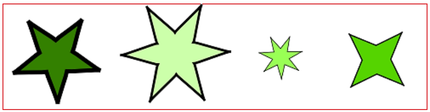
Lucy has 4 plastic stars. Every star has its own size, colour, thickness of contour and number of corners and she likes to order them by these properties. Stars in the picture below are ordered from the thickest contour line to the thinnest one.

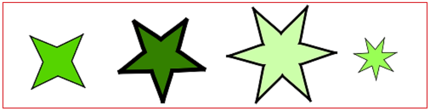


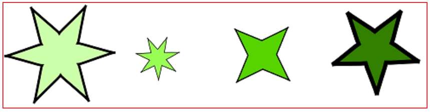
Order stars so that they are not ordered according to any of the described properties.

For example, they should be ordered neither from the lightest to the darkest one nor from the darkest to the lightest one.







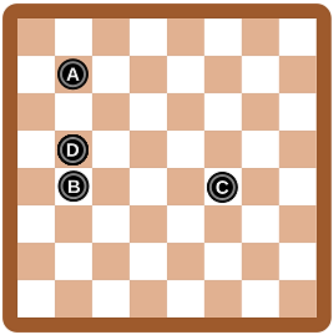


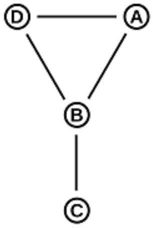
Argumentation: For each wrong answer from the set offered above, the contestant has to recognize which of the four properties is the essential one according to which the stars are ordered. It is even more challenging because the star sets are very similar.

Rows and columns, 15–17 years, code 2018-BE-03

Below on the left you see a picture of a game board with 4 pieces placed on it. We draw a diagram of this board, on the right of the picture, in the following way:

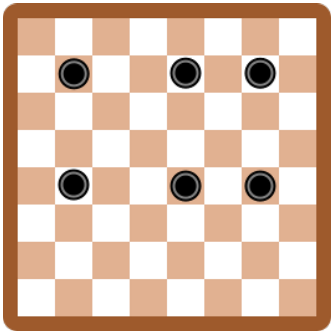
Draw a circle for each piece on the board. If two pieces are in the same row on the board or in the same column on the board, then draw a line between their circles in the diagram.

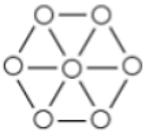


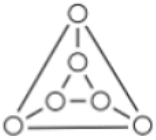


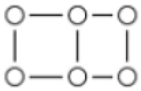
Letters have been placed on the pieces and the circles so you can easily check that the diagram is correct.

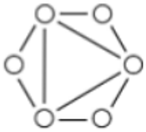
We have drawn a diagram in this way for the board with six pieces which you see below. Which of the four diagrams below was drawn?











Argumentation: In this task it is necessary to find the key elements in both data representations which are related to each other. Contestants should find out that the number of neighbours of the chosen node is the same as the number of pieces lying in the same row or column on the board.

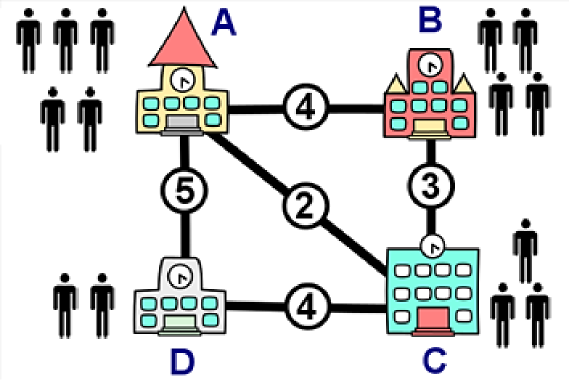
4.2.2. *Type abs*

In this part we will introduce two examples of tasks which were chosen for the type *abs* where abstraction occurs but not as a dominant mental process for the task solution.

Meeting, 11–15 years, code 2015-JP-03

There are four schools in a town. There is going to be a teacher´s meeting where a number of teachers are going to participate from each school. The picture shows placement of all four schools and all teachers participating in the meeting. The lines represent how schools are interconnected with bus lines and the numbers represent ticket prices among schools. One can travel from B to D through A or C.

Plan this meeting the way teachers would spend the least on tickets. Which school should the meeting be held in?



Argumentation: In this task contestants have to orient themselves in an abstract structure of the graph and discover a connection with attributes of the real world. Abstraction, however, is not dominant here – within an optimization task a suitable strategy is applied to solve the problem as well as combination abilities.

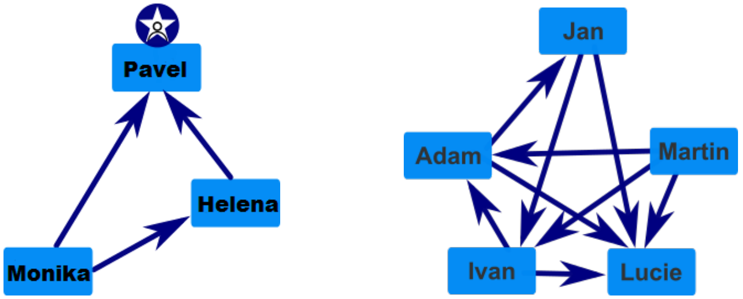
Superstar, 15–17 years, code 2019-DE-08

In a social network "TeeniGram" users can follow themselves and create groups of users. There is a so-called "superstar" in every group.

A superstar is someone who:

- is followed by all members of the group
- does not follow any other member of the group

For example, this group has the following members: Monika, Helena and Pavel (graph on the left). Monika follows Pavel and Helena; Helena follows Pavel; Pavel does not follow anyone. Pavel is the superstar in this group.



Another group has these members: Adam, Jan, Martin, Lucie and Ivan (graph on the right). Is there a superstar in this group?

A. Yes, Adam is the superstar in this group.
B. Yes, Martin and Ivan are the superstars in this group.
C. Yes, Lucie is the superstar in this group.
D. No, there is not a superstar in this group.

Argumentation: In this task one uses a graph which has already been created. It is necessary to orient oneself in this abstract structure and understand graph symbols. Contestants, however, do not create an abstract object by themselves, they do not abstract from the reality.

In this chapter we explained on examples the difference between the type *ABS* and *abs*. In relation to this we can say that the task about colouring the space with as a few colours as possible, which is presented by Ternik *et al.* (2020, p. 36) as a model example of abstract tasks, would belong to the type *abs*, not *ABS* according to our criteria.

4.3. *Finding Algorithmic Tasks*

In order to answer RQ3 to RQ6 we also need to extract the set of algorithmic tasks. This process is facilitated by the fact that the databank of Bebras Challenge has an Algorithm type (ALG or ALP). We consider this classification valid as it has been used for a long period of time and unwritten criteria for decisions about task types has been made by the whole Bebras community.

When we searched algorithmic tasks, we used the official classification of tasks for Bebras Challenge which is carried out by authors of tasks. All tasks, including their classification, are revised during annual Bebras task workshops. In this classification algorithmization is explicitly introduced as one of the task types (in years 2014–2019 under the type *ALG*, in the year 2020 as *ALP*). One task may belong to more than one of the total number of five previously described types. We assigned tasks to one of the two following groups:

- ***ALG*** – (algorithmization) the type *ALG*, or *ALP* occurs among task types which a task is assigned to

- ***no alg*** – (no algorithmization) a task is not assigned to the type *ALG*, or *ALP*

We did not include an analogous division into groups of “big algorithmization” and “small algorithmization” as in abstract tasks owing to the official classification not consisting of such groups of tasks. We would have to create these groups by ourselves. In that case it would not be relevant to adopt the classification of tasks from their authors and the Bebras community.

4.4. *Sample of Respondents*

The sample of research respondents answering RQ3 to RQ6 was selected from the contestants of the Czech Bebras Challenge in adequate age categories. All contestants who had not been explicitly excluded by the school coordinator of the contest (for example due to cheating) became participants. It can be said that this is a case of nonprobability sampling, more specifically convenience sampling (Creswell, 2015). The numbers of contestants are presented in Table 1.

We used SQL query over the contest databank to ascertain respondents´ answers. On the basis of SQL query we collected numbers of correct and incorrect answers from respondents to observed tasks.

4.5. *Data Analysis*

4.5.1. *Experts´ Agreement on the Classification of Task Types*

To answer RQ1, we researched the extent of experts’ agreement when they were deciding about which task belongs to which type (ABS, abs, no abs). We used a method of analysis of categorical data for this purpose. We chose Fleiss kappa statistics due to the intentional selection of more than two raters. This method deals with statistical analysis of tables of frequencies which occur in description and analysis of relations among categorical variables (Hendl, 2015, p. 333). It is presented as an analogy of correlation analysis of continuous variables. The dependence of categorical variables specifically

Table 1

Numbers of contestants in the younger category and the older category

Year of the contest	11–13 years old		15–17 years old	
	Boys	Girls	Boys	Girls
2020	8861	9165	6366	6121
2019	16098	14789	6630	4532
2018	13197	12085	5915	4344
2017	12655	11714	6252	4466
2016	9123	8228	6477	4369
2015	7413	6935	5093	3622
2014	6034	5360	4626	3385

Table 2

Interpretation of kappa coeficient according to McHugh

Agreement	Kappa
slight	$\kappa < 0{,}20$
fair	$0{,}20 \leq \kappa < 0{,}40$
moderate	$0{,}40 \leq \kappa < 0{,}60$
substantial	$0{,}60 \leq \kappa < 0{,}80$
almost perfect	$0{,}80 \leq \kappa$

determines kappa coefficient which defines the level of agreement among raters, being corrected by chance agreement.

Kappa coefficient can range from –1 to +1, where 0 represents the amount of agreement that can be expected from random chance, while 1 represents perfect agreement between the raters (McHugh, 2012). Although the impartial criteria do not exist for evaluation of medium values, they are usually interpreted in the way showed in Table 2.

Evaluation of Bebras tasks made by three experts was recorded into a table calculator, it was modified into a contingency table and it was ultimately used to calculate kappa coefficient. To do this, the additional Excel tool XRealStats and instructions according to Zaiontz (2013) were used.

4.5.2. *Successfulness of Solving Abstract Tasks (RQ3–RQ6)*

To answer the research questions RQ3–RQ6, we formulated several research hypotheses for all participants, younger participants, older participants, girls and boys:

- Based on $RQ3_{all}$ / $RQ3_{young}$ / $RQ3_{old}$ / $RQ3_{girls}$ / $RQ3_{boys}$ we developed hypotheses $H3_{all}$ / $H3_{young}$ / $H3_{old}$ / $H3_{girls}$ / $H3_{boys}$: In the group, there is no difference between the proportion of correct answers in tasks focused on abstraction and the proportion of correct answers in algorithmic tasks.
- Based on $RQ4_{all}$ / $RQ4_{young}$ / $RQ4_{old}$ / $RQ4_{girls}$ / $RQ4_{boys}$ we developed hypotheses $H4_{all}$ / $H4_{young}$ / $H4_{old}$ / $H4_{girls}$ / $H4_{boys}$: In the group, there is no difference between the proportion of correct answers in all abstract tasks and the proportion of correct answers in algorithmic tasks.
- Based on $RQ5_{young}$ / $RQ5_{old}$ we developed hypotheses $H5_{young}$ / $H5_{old}$: Between the groups of girls and boys, there is no difference in the proportion of correct answers in tasks focused on abstraction (type ABS).
- Based on $RQ6_{young}$ / $RQ6_{old}$ we developed hypotheses $H6_{young}$ / $H6_{old}$: Between the groups of girls and boys, there is no difference in the proportion of correct answers in all abstract tasks (type ABS + abs).

Data analysis was carried out using the Student's t-test statistical method. As per King and Eckersley (2019), this parametric hypothesis test was specifically designed to work with smaller sample sizes. It allows one to test whether the mean of the population from which a sample was drawn differs from an expected value, or whether the means of two different populations differ (King and Eckersley, 2019).

5. Results and Discussion

5.1. *Experts' Agreement on Whether Tasks are Abstract (RQ1)*

The total number of 198 Bebras tasks was classified from years 2014–2020.

Results, which are shown in Table 2, are divided into stages. The row of Stage 1 introduces values from years 2017–2019 when experts had not been influenced by statements of other ones yet. The row of Stage 2 introduces values from years 2014–2016 and 2020 when experts decided about the type classification together after a mutual discussion about the selection of abstract tasks in Stage 1. We present the final kappa coefficient in Table 3 in 2 columns: *ABS+abs* (the experts´ agreement on the occurrence of abstraction in tasks) and *ABS* (the experts´ agreement on abstraction being the main part of a task).

The final kappa value about 0.5 showed a moderate agreement. This value is significantly higher than the value –0.1 in research of Ternik *et al.* (2020, p. 38). These results also indicate that experts struggled to agree on the classification of tasks focused on abstraction since the values in the column *ABS* in Table 3 are lower than the values in the column *ABS+abs*.

There is no significant increase of kappa value between Stage 1 and Stage 2; hence, it emerged that there was no further proximity of experts´ statements about which tasks develop abstraction and which tasks do not. This fact was manifested in both task types *ABS*, *abs*.

We interpret it that our team consisted of experts who had already known Bebras tasks since they had either worked with tasks in Bebras Challenge for a long period of time, they had been editors or administrators of those tasks, or they had participated in the task classification used in the Czech version of the contest. Therefore, they had already had a strong opinion in Stage 1 on which tasks developed abstract thinking.

Another cause of this difference might be the fact that in our research experts decided only between two situations (abstraction is present / is not present) and they only concentrated on one type of contest tasks. As for research of Ternik *et al.* (2020), experts had a wide range of task types, thus they did not concentrate purely on abstraction.

We were not able to conduct a similar experts´ evaluation of the second observed type of algorithmic tasks due to the fact that we took over a classification of task authors and editors of the international databank of Bebras tasks. Algorithmization is an official type of contest tasks and we did not want to risk a situation when experts, during a decid-

Table 3
Final kappa values

	ABS+abs	*ABS*
Stage 1	κ = 0.548	κ = 0.442
Stage 2	κ = 0.568	κ = 0.459

ing period, could be affected by the awareness of the way in which tasks were classified as particular types by their authors.

5.2. *Count of Abstract and Algorithmic Tasks (RQ2)*

We evaluated 99 tasks in each age category from years 2014 to 2020. Detailed counts of tasks are provided in Table 4 and Table 5.

From the above mentioned findings, it emerges that the number of observed algorithmic tasks is much higher than abstract ones. In tasks focused on abstraction (ABS), this proportion occurred very significantly for algorithmic tasks, specifically 5:1 for younger contestants and 4:1 for older contestants. This proportion is in line with findings in research of Izu *et al.* (2017). The previously stated proportion amounted to even higher figures within this research – estimated from the chart 9:1 for younger contestants and 20:1 for older contestants.

Our research indicated the lower proportion in all tasks containing abstraction (*ABS+abs*) in comparison with algorithmic tasks, specifically 2:1 for younger contestants and 3:2 for older contestants in favour of algorithmic tasks.

5.3. *Successfulness of Solutions to Abstract Tasks (RQ3 to RQ6)*

Each data selection was tested first on normality by Anderson-Darling and Kolmogorov-Smirnov test. Since there was not a case of rejection of a hypothesis, that the population is normally distributed, the pairs of selections were then tested by F-test for homogeneity of variances. This was also not a case of a hypothesis of equality of variances being rejected.

Table 4

Count of tasks according to our classification in the younger category (11–13 years old)

	ABS	*abs*	*no abs*	Total
ALG	1	5	37	43
no alg	8	9	39	56
Total	9	14	76	99

Table 5

Count of tasks according to our classification in the older category (15–17 years old)

	ABS	abs	no abs	Total
ALG	2	9	36	47
no alg	9	12	31	52
Total	11	21	67	99

After those tests, a two-choice Student's t-test was performed on each pair of data selection. The null hypothesis was tested whether the means of the selections are equal. In this case, the null hypothesis of equality of mean values was not rejected. We interpret it that we did not ascertain a difference in the successfulness of solutions to abstract tasks in any comparison among selected respondents. Detailed calculated p-values are given in Table 6 and Table 7.

While comparing the successfulness of solutions to abstract tasks and algorithmic ones, the null hypothesis was rejected in not a single observed case. This means that those two task types were of the same difficulty to contestants, that is to say not only contestants of different age, but different gender too. Another view of results shows that authors of contest tests were able to select comparably difficult tasks of both observed

Table 6

Comparison of successfulness of abstract and algorithmic tasks

RQ	Description	p value	Rejection of the null hypothesis at a significance level of 0.05
Difference in the successfulness of solutions to tasks focused on abstraction *(ABS*) and *ALG* tasks			
$RQ3_{all}$	- all contestants	0.3059	No
$RQ3_{young}$	- younger contestants (11–13 years)	0.2224	No
$RQ3_{old}$	- older contestants (15–17 years)	0.9173	No
$RQ3_{girls}$	- girls	0.5811	No
$RQ3_{boys}$	- boys	0.8022	No
Difference in the successfulness of solutions to all abstract tasks (*ABS*+*abs*) and *ALG* tasks			
$RQ4_{all}$	- all contestants	0.8556	No
$RQ4_{young}$	- younger contestants	0.8694	No
$RQ4_{old}$	- older contestants	0.7704	No
$RQ4_{girls}$	- girls	0.8942	No
$RQ4_{boys}$	- boys	0.9463	No

Table 7

Difference between genders in the successfulness of solutions to abstract tasks

RQ	Description	p value	Rejection of the null hypothesis at a significance level of 0.05
$RQ5_{young}$	Difference between genders in the successfulness of solutions to *ABS* tasks at younger contestants	0.8571	No
$RQ5_{old}$	Difference between genders in the successfulness of solutions to *ABS* tasks at older contestants	0.8739	No
$RQ6_{young}$	Difference between genders in the successfulness of solutions to all abstract tasks (*ABS*+*abs*) at younger contestants	0.8846	No
$RQ6_{old}$	Difference between genders in the successfulness of solutions to all abstract tasks (*ABS*+*abs*) at older contestants	0.6289	No

types. No difference was found when comparing algorithmic tasks with the ones focused on abstraction and with all abstract tasks.

To a certain extent these results are in line with the findings from our former research which presented the fact that a task type does not fundamentally determine whether a certain task will be difficult or not, although abstraction was not included in observed task types (Vaníček and Křížová, 2014). These findings, however, contradict research of van der Vegt and Schrijvers (2019), according to whom tasks of category DSR (Data, Data Structures and Representations) seem to be easier than ALP (Algorithms and Programming) tasks while the combination of these types increases the difficulty even further. In addition, in our former research (Vaníček, 2016) structural tasks were more difficult; presence of a structure in a task made the task more difficult than without it.

When we compared the successfulness of solutions to abstract tasks between genders, no null hypotheses were rejected, neither in the younger category, nor the older one. This means that no differences between genders were found, neither at tasks with a solution based on abstraction, nor at all abstract tasks. Abstract tasks were of the same difficulty for girls as well as boys.

It does not emerge from the above mentioned findings that particular abstract tasks would be comparably difficult. As Fig. 1 indicates, the successfulness of contestants considerably differs in tasks focused on abstraction. According to Lonati *et al.* (2017), the task difficulty has several parts which may be divided into two main groups – intrinsic and surface. The necessity to abstract is therefore only one of the parts determining the overall task difficulty. Similarly, it does not emerge from our results that every particular task would be comparably difficult for girls and boys.

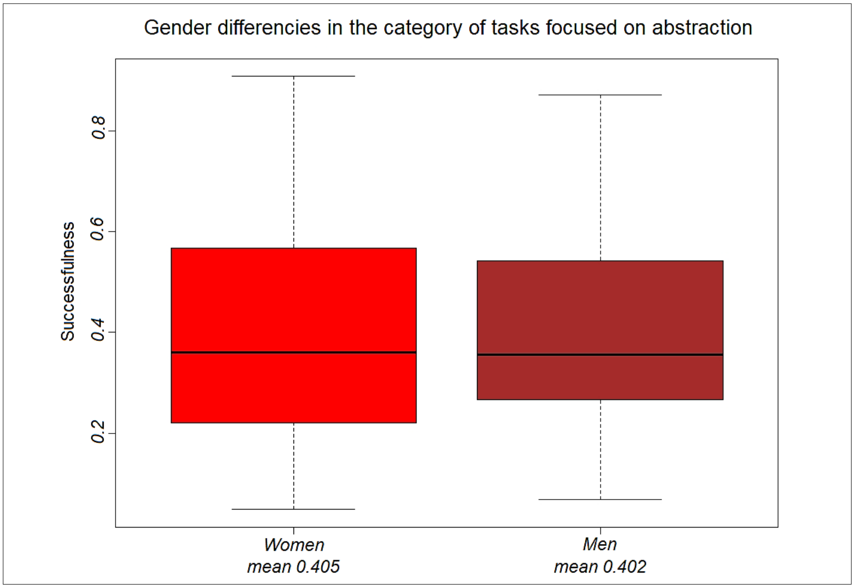


Fig. 1. Gender comparison of the successfulness of solving contest tasks focused on abstraction (ABS).

6. Conclusion

A reform of Informatics education is currently being in progress in the Czech Republic. New national informatics curricula have been authorized for elementary education. They have completely changed the orientation of Informatics as a subject from teaching using computers and digital literacy to Informatics content and development of computational thinking. Abstraction has therefore placed itself in the future curriculum, valid from September 2021, as one of the parts of computational thinking and Bebras tasks may play an important role as mediators of new educational contents. When we state which tasks develop abstraction, which way they do it and how much, it allows us to create more educational aimed materials for pupils and simultaneously education for teachers who need to acquaint themselves with these new terms. Furthermore, Bebras tasks are suitable for their aim at a specific problem and concept in a well arranged situation, a short period of time necessary for a solution and a form of inputting answers with an immediate service regarding its accuracy.

In case potential future researches would find themselves in need of the following data, we present international codes for tasks which were used in years 2014–2020 in the Czech version of Bebras Challenge and which were classified as focused on abstraction (*ABS*). For the older category of contestants: 2014-DE-03, 2014-TW-04, 2014-SP-02, 2015-CZ-02, 2016-CZ-09, 2016-CZ-04, 2017-CZ-03a, 2018-BE-03, 2019-BE-04, 2019-ID-02, 2020-TH-01. For the younger category of contestants: 2014-CZ-05, 2014-CZ-08, 2015-CZ-01, 2016-CZ-09, 2017-CZ-03, 2010-IT-01, 2018-MY-09, 2018-CZ-04, 2019-RO-01. References to particular wording of all abstract tasks as well as the others tasks are in the attachment on `https://www.ibobr.cz/papers/INFEDU2021.pdf`.

Our findings about Bebras tasks including only a small amount of abstract tasks may lead to seek bigger development of these tasks by the Bebras community, perhaps even revaluation of typology of tasks currently used by the community. It seems most likely that by creating a new task type “abstraction” may guide authors of tasks to think carefully about whether their tasks contain abstraction and how they might strengthen this part. Furthermore, it may lead to develop new task types which have not appeared in the contest yet and which may be connected for example with developing procedures during programming. Abstraction is strongly present in such procedures, yet they do not practically occur in the contest, presumably due to nonexistence of the software solution to such tasks.

We ascertained that abstract tasks are at the same difficulty level as algorithmic tasks and gender does not affect successfulness of solutions to abstract tasks. These findings are of great importance since some researches indicated that differences existed at the difficulty level of some task types according to the official typology of contest tasks. While creating abstract tasks, authors have to concern neither the gender aspect, nor greater difficulty level in comparison with other task types.

We understand limits of our research which were caused mainly by the selection of tasks for the Czech version of the contest. The occurrence of abstraction was not taken into consideration during the selection of tasks. Another limit is the choice of experts in task evaluation who were not completely independent. All of them had already been familiar with at least a few Bebras tasks and had participated in the organization of the contest, although they had never cooperated on preparations and realization of the contest in the view of creating contest tests or developing and editing tasks.

Some of our results show that abstraction and algorithmization are equivalent in some way. Abstraction might be a "precondition" for algorithmization. We neither comment nor discuss this since it is a different research question, which, however, could open an interesting new area for research.

Next research may deal with the question whether ordinary teachers would be equally able to recognize abstract tasks as our raters, or whether it is only a privilege of qualified experts. It might also be of interest to ascertain what kind of training teachers would need to be able to recognize this part of computational thinking in tasks. This would lead to greater independence of teachers during formation of school curriculum.

Acknowledgments

This research was covered from projects TAČR TL03000222 – The development of Informatics thinking by means of situational algorithmic problems; and GA JU 121/2016/S – Problem solving as one of key competencies in Mathematics and Informatics education.

References

Aho, A.F. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832–835. DOI: 10.1093/comjnl/bxs074

Barr, V., Stephenson, C. (2011). Bringing computational thinking to K-12: What is Involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54. DOI: 10.1145/1929887.1929905

Bebras Challenge (2016). A contest web. *University of Vilnius*. https://www.bebras.org/

Bebras Community (2015). Statutes. RC 3. *International Bebras Committee*.

Bebras Community (2020). Statutes. 1.0. *International Bebras Committee*. https://www.bebras.org/sites/default/files/BebrasStatutes-1_0.pdf

Budinská, L., Mayerová, K., Veselovská, M. (2017). Bebras task analysis in category little beavers in Slovakia. In: Dagienė, V., Hellas, A. (eds.) *ISSEP*. LNCS, Springer, Cham, 10696, 91–101. DOI: 10.1007/978-3-319-71483-7_8

Cetin, I., Dubinsky, E. (2017). Reflective abstraction in computational thinking. *The Journal of Mathematical Behavior*, 47, 70–80. DOI: 10.1016/j.jmathb.2017.06.004

Creswell, J.W. (2015). *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research, Enhanced Pearson eText with Loose-Leaf--Version-Access Card Package.* University of Nebraska, Lincoln.

Dagienė V., Futschek G. (2008). Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In: Mittermeir R.T., Sysło M.M. (eds) *Informatics Education - Supporting Computational Thinking*. ISSEP. LNCS, Springer, Berlin, Heidelberg, 5090. DOI: 10.1007/978-3-540-69924-8_2

Dagienė V., Sentance S., Stupurienė G. (2017). Developing a Two-Dimensional Categorization System for Educational Tasks in *Informatics. INFORMATICA*, Vilnius University, 28(1), 23–44. DOI: 10.15388/Informatica.2017.119

Denning, P.J. (2011). Ubiquity symposium: What have we said about computation? Closing statement. *Ubiquity*. 1–7. DOI: 10.1145/1967045.1967046

Gove, P.B., G. & C. Merriam Company (1981). *Webster's third new international dictionary of the English language, unabridged*. Volume I: A-G. Merriam-Webster, Springfield, Mass.

Hendl, J. (2015). *Přehled statistických metod: analýza a metaanalýza dat*. Portál, Praha.

Hu, C. (2011). Computational thinking: what it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (ITiCSE '11)*. Association for Computing Machinery, New York, NY, 223–227. DOI: 10.1145/1999747.1999811

Izu, C., Mirolo, C., Settle, A., Mannila, L., Stupurienė, G. (2017). Exploring bebras tasks content and performance: a multinational study. *Inf. Educ*. 16(1), 39–59.

Kala , I., Tomcsanyiova, M. (2009). Students' attitude to programming in modern informatics. In: *Proceedings of the 9th WCCE 2009*, Education and Technology for a Better World.

King, A.P., Eckersley R.J. (2019). Inferential Statistics II: Parametric Hypothesis Testing. In: *Statistics for Biomedical Engineers and Scientists*, 91–117. DOI: 10.1016/B978-0-08-102939-8.00014-1

Kramer, J. (2007). Is abstraction the key to computing? *Communications of the ACM*, 50(4), 36–42. DOI: 10.1145/1232743.1232745

Lonati, V., Monga, M., Malchiodi, D., Morpurgo, A. (2017). How presentation affects the difficulty of computational thinking tasks: an IRT analysis. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research,* 60–69.
`https://aladdin.di.unimi.it/materiali/pubb/2017_koli.pdf`

McHugh M.L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica.* 22(3), 276–282. DOI: 10.11613/BM.2012.031

Opmanis, M., Dagienė, V., Truu, A. (2006). Task types at "Beaver" contests. In: *Information Technologies at School*, 509–519.

Selby, C., Woollard, J. (2013). Computational thinking: the developing definition. In: *Special Interest Group on Computer Science Education (SIGCSE)*. ACM, Atlanta.

Selby, C., Woollard, J. (2014). *Refining an understanding of computational thinking*. University of Southampton, Southampton. `https://eprints.soton.ac.uk/372410/`

Ternik Ž., Todorovski L., Nančovska Šerbec I. (2020). Assessing the Agreement in the Bebras Tasks Categorisation. In: Kori K., Laanpere M. (eds) *Informatics in Schools*. Engaging Learners in Computational Thinking. ISSEP. LNCS, Springer, Cham, 12518. DOI: 10.1007/978-3-030-63212-0_3

The Royal Society. (2012). *Shut Down or Restart? The Way Forward for Computing in UK Schools*. The Royal Society, London.

Vaníček, J., Křížová, M. (2014). Kritéria obtížnosti testových otázek v informatické soutěži. In Lovászová, G. (ed.) *DidInfo'14*. Univerzita Mateja Béla, Banská Bystrica, 191–199.

Vaníček, J. (2016). What Makes Situational Informatics Tasks Difficult? In: Brodnik A., Tort F. (eds) *Informatics in Schools*: Improvement of Informatics Knowledge and Perception. ISSEP. LNCS, Springer, Cham, 9973, 90–104.

Vegt, W.V., Schrijvers, E. (2019). Analyzing Task Difficulty in a Bebras Contest Using Cuttle. *Olympiads in Informatics*, 13, 145–156. DOI: 10.15388/ioi.2019.09.

Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. DOI: 10.1145/1118178.1118215

Wing, J.M. (2010). *Computational thinking—What and why*. Carnegie Mellon University, Pittsburgh, PA.
`https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf`

Wing, J.M. (2014). Computational thinking benefits society. 40th Anniversary Blog of *Social Issues in Computing*. `http://socialissues.cs.toronto.edu/index.html%3Fp=279.html`

Wing, J.M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14. DOI: 10.17471/2499-4324/922

Zaiontz, Ch. (2013). *Fleiss' Kappa. Real statistics using Excel.*
`https://www.real-statistics.com/reliability/interrater-reliability/fleiss-kappa/`

J. Vaníček is an associate professor and the head of the Department of Informatics in the Faculty of Education at the University of South Bohemia in České Budějovice in the Czech Republic. His area of interest is Informatics education in primary and lower secondary schools and early age programming. He is an author of 7 textbooks about information technology and programming. Between 2017 and 2020 he was a head of the strategic project PRIM developing new Czech national informatics curricula. He has been organizing Bebras Challenge for 13 years and he is a representative of the Czech Republic in the International Bebras Committee.

V. Šimandl is an assistant professor in the Department of Informatics in the Faculty of Education at the University of South Bohemia in České Budějovice in the Czech Republic. His research activities focus on the area of Informatics education in lower secondary schools and programming. He has been organizing the Czech Bebras Challenge for 7 years.

P. Klofáč is an assistant professor in the Department of Informatics at the University of South Bohemia in České Budějovice. His main research interest is in the area of robotics education in lower secondary schools. He has been task tester of the Czech Bebras Challenge for 2 years and participated in a project focused on creation new Bebras tasks classification.