Development of a Lean Computational Thinking Abilities Assessment for Middle Grades Students

Eric Wiebe[†]
North Carolina State University
Raleigh, NC USA
wiebe@ncsu.edu

Jennifer London
London & Assoc.
Raleigh, NC USA
londonje@gmail.com

Osman Aksit
Dhahran Ahliyya Schools
Dammam, Saudi Arabia
osmanaksit@gmail.com

Bradford W. Mott
North Carolina State University
Raleigh, NC USA
bwmott@ncsu.edu

Kristy Elizabeth Boyer
University of Florida
Gainesville, FL USA
keboyer@ufl.edu

James C. Lester
North Carolina State University
Raleigh, NC USA
lester@ncsu.edu

ABSTRACT

The recognition of middle grades as a critical juncture in CS education has led to the widespread development of CS curricula and integration efforts. The goal of many of these interventions is to develop a set of underlying abilities that has been termed computational thinking (CT). This goal presents a key challenge for assessing student learning: we must identify assessment items associated with an emergent understanding of key cognitive abilities underlying CT that avoid specialized knowledge of specific programming languages. In this work we explore the psychometric properties of assessment items appropriate for use with middle grades (US grades 6-8; ages 11-13) students. We also investigate whether these items measure a single ability dimension. Finally, we strive to recommend a "lean" set of items that can be completed in a single 50-minute class period and have high face validity. The paper makes the following contributions: 1) adds to the literature related to the emerging construct of CT, and its relationship to the existing CTt and Bebras instruments, and 2) offers a research-based CT assessment instrument for use by both researchers and educators in the field.

CCS CONCEPTS

• Social and professional topics → Computational thinking; Student assessment; K- 12 education

KEYWORDS

computational thinking, IRT, psychometrics, K-12 computer science education, middle grades, assessment

SIGCSE '19, February 27-March 2, 2019, Minneapolis, MN, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5890-3/19/02...$15.00
https://doi.org/10.1145/3287324.3287390

ACM Reference format:
E. Wiebe, J. London, O. Aksit, B. W. Mott, K. E. Boyer and J. C. Lester. 2019. Development of a Lean Computational Thinking Abilities Assessment for Middle Grades Students. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE'19), Feb. 27-Mar. 2, 2019, Minneapolis, MN, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3287324.3287390

1 INTRODUCTION

The recognition of middle grades as a critical juncture in CS education has, expectedly, led to the development of curricula and other interventions in both formal and informal educational settings [1-4]. An equal effort has gone into elective classes in areas such as robotics (e.g., [5]), maker-based activities (e.g., [6]), in addition to block-based programming (e.g., [7]). It follows that work has gone into developing assessments for the targeted knowledge and practices that these curricula and interventions address [8].

Part of the assessment challenge is that the goal of such interventions has, over the past ten years, shifted away from only a specific goal of teaching CS concepts and programming abilities, to developing a set of underlying abilities that has been termed computational thinking (CT) [9, 10]. The emergence of CT as a conceptual framework guiding curricular development and assessment [11, 12] has happened in parallel with a move to think more broadly as to which types of educational contexts might be appropriate for students to engage in CT-based activities. In particular, this has included a strong move to integrate CT activities into established STEM courses at the middle grades and high school level [2, 13-15]. Such approaches appropriately demand a strategy that assumes that CT ability is being developed across multiple formal and informal academic contexts [5]. Similarly, assessments that measure both what students bring to a particular intervention and how that intervention impacts their knowledge and abilities, to be effective, would need to be decontextualized to some degree from the particular application or academic subject area.

The growth of interest in integrated CT interventions points to a need to develop CT assessments that can work in such diverse instructional contexts [16-18]. However, most assessments developed for either CS or CT at the middle grades level are still based on CS frameworks around the construction or analysis of coding artifacts [4, 8, 19, 20]. These assessments use a range of text-based, block-based, and pseudo-code, but still use coding artifacts based on CS concepts as the paradigm for analysis [21, 22]. It raises questions as to whether such instruments would be appropriate as a pre-test for students who have never had experience with developing code or for interventions where coding is not the basis of the CT curricular intervention [23]. For example, many interventions utilize unplugged activities that may be independent of specific code-based representational schemes and syntaxes [3, 9, 24]. In addition, many assessments are not only written with programming artifacts as the task, but also represent CS-centric conceptual frameworks, rather than CT frameworks [19]. In summary, an assessment—particularly a pre-assessment—should reflect core CT abilities being developed in the intervention but be free from specialized representational notations (e.g., programming languages) or knowledge that a student would not have been exposed to outside the intervention. Put another way, you may want to distinguish abilities related to underlying CT constructs apart from specialized knowledge of programming ability, and a code-centric assessment would not allow you to do this. Additional associated problems from a code-centric assessment might include both floor effects of pre-assessment scores and/or frustration on the part of students faced with what amounts to a foreign language.

The goal of the work presented in this paper is pragmatic. Leveraging both theoretical and empirical work done to date, we have set out to identify assessments associated with a current emergent understanding of key cognitive abilities underlying CT. These assessments should avoid the abovementioned concerns related to specialized knowledge related to programming languages or other allied CS knowledge. In doing this work, we will explore the psychometric properties of items appropriate for use with middle grades (U.S. grades 6-8; ages 11-13) students. We will also investigate whether these items represent a single dimension of ability. Finally, we will strive to recommend a set of items that has a relatively short administration (i.e., can be completed in a single 50-minute class period) and have high face validity for both teachers and students.

2 RELATED WORK

An important line of work to develop a CT assessment independent both of specific programming languages and curricular contexts has been led by Román-González and colleagues [25-27]. They chose to take a psychometric approach rooted in the CHC model of intelligence [28, 29]. Román-González [26] and a small number of other researchers (e.g., [30]) have conjectured a relationship between CT constructs such as abstraction, pattern generalization, algorithmic thinking, and conditional logic, and the CHC constructs of fluid intelligence, visual processing, and working memory. The design of their items were informed both by prior assessments developed around CS-centric programming tasks [31, 32], as well as other efforts developing programming-independent assessments [33, 34].

Using a 28-item version of his computational thinking assessment (CTt) [26] on a sample of 1,251 Spanish students, boys and girls from 24 different schools enrolled from 5th to 10th grade, Román-González established reliability as internal consistency with a Cronbach's Alpha = 0.793. Criterion validity was explored first by looking at correlations between the CTt and the PMA battery of cognitive tests, and then through a multiple linear regression through the CTt score. The CTt moderately correlated with the PMA spatial (r = 0.44) and reasoning ability (r = 0.44), and weakly correlated with verbal reasoning (r = 0.27). In addition, it was strongly correlated with the RP30 problem-solving test (r = 0.67) which is considered a proxy for fluid intelligence. The regression with these same subscales was also significant ($p < 0.01$), with spatial and reasoning ability as significant predictors. However, 73% of the CTt scores' variance was left unexplained. Román-González, Pérez-González, & Jiménez-Fernández [26] observed a near-normal distribution (*M*=16.38, .058 skewness, and -.446 kurtosis) among 1,251 Spanish students from 5th-10th grade. Items spanned a wide array of difficulty with later questions being hardest. The average success rate along the 28 items was p = 0.59 (medium difficulty); ranging from p = 0.16 (item 23; very high difficulty) to p = 0.96 (item 1; very low difficulty). In summary, the CTt was shown to have an appropriate distribution of item difficulty for middle grades students. In addition, the CTt seems to measure abilities related to fluid intelligence and spatial ability, however a majority of the differences in student performance ability on the assessment was not explained by these particular cognitive abilities.

The authors conclude that the CTt provides a decontextualized assessment that compliments assessments designed to be more context-specific. It is interesting to note that many of the CTt items have block-based programming-like elements in them, raising concerns that this assessment would suffer from floor effects with students not familiar with block-based programming concepts. However, a recent study using the CTt did not find this effect with a population of students of whom a high proportion self-reported little or no prior programming experience [35]. Further work by Román-González and colleagues continued to explore the complimentary utility of CTt with other either programming-centric or more general assessments. In a convergent validity study, they found a high correlation between the CTt and Dr. Scratch [36] (r = 0.44) and with the CTt and a selected set of Bebras [33, 34] items (r = 0.52). They conclude that the correlational results point to a partial but not complete overlap between the three assessments, reflecting the designers' differing measurement goals. Thus the conclusion was that CTt and Bebras are measuring somewhat different abilities.

Bebras, it should be said, started not as an assessment, but as an international competition with the stated goal of raising awareness and interest in informatics (i.e., computing/computer science) education and career paths [33, 34]. It was designed to promote informatics learning in school by solving short, conceptually-based tasks that make up the heart of the competition. Since its beginnings in Lithuania in 2004, it has grown to the point that more than 1.3 million individuals across Europe and elsewhere participated in the competition during Bebras week in November 2015 [34]. Like the CTt, Bebras was designed to not depend on prior knowledge of programming/coding [37], but instead has puzzle-like problem-solving tasks readily accessible to students with no programming background. Though Bebras' roots are not as an assessment instrument (CS or otherwise), researchers have noted the ability to map its items to problem-solving constructs that underlie CS/CT, such as algorithmic thinking and working with structures and patterns [38]. For that reason, Bebras has increasingly been utilized in research and applications related to CT assessment [39].

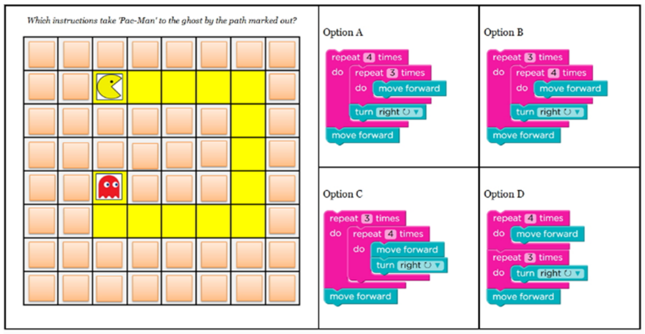
As part of this interest in utilizing Bebras tasks as part of CT assessments, IRT-based analyses have been conducted on Italian [40], Slovenian [41], and German [42] versions of the Bebras competition task set. In perhaps the largest-scale analysis of Bebras data, performance data on Bebras tasks from 115,400 students in grades 3-12 in seven countries were analyzed by Izu and colleagues [43]. They specifically looked at gender differences, where none was found in either participation or performance except at older grade ranges where there was higher male participation. They concurred with other studies that identified an apparent discrepancy between the tasks' estimated and perceived difficulty, finding that the Bebras versions had more "harder than expected" than "easier than expected" items. This has been noted as perhaps the result of the task set being designed as a competition and not an assessment [44]. As noted by the researchers conducting IRT analyses on Bebras tasks, this approach is particularly appropriate for assessing the difficulty of items if they are to be used as an assessment.

In summary, CTt and Bebras have emerged as two instruments with a mounting base of evidence linking them to core constructs underlying current conceptualizations of CT. Both instruments have set out to create items that are appropriate for middle grades students who have no prior experience with block-based or other programming languages. However, they originate from goals somewhat different from each other; while CTt was designed from the start as an assessment of CT, Bebras started as a task set for a competition and only more recently has been researched as an assessment tool. Another important distinction is that though the developers of the CTt and a small set of additional research has shown the instrument appropriate for use with students with no programming background, items such as the one shown in Figure 1 use what is, in effect, block-based programming code. In contrast, Bebras items use common, everyday visual metaphors in their puzzle-like items. It is worth reiterating that the CTt took inspiration from Bebras and has been shown to be correlated at a complementary rather than wholly overlapping level [25].

3 DATA AND METHODS

3.1 Test Items

The initial version of our assessment consisted of 43 items, which was the combination of 28 items from the CTt (Figure 1) and 15 items from the Bebras challenge (Figure 2). All 28 multiple-choice items of the CTt were included without making any modifications. Fifteen items were selected out of 18 items from the UK Bebras 2016 task set that were targeted for students of age 10 to 12 years old (referred as Juniors age group). Eleven of the 15 Bebras items were single-select multiple-choice, one item was multi-select multiple-choice, one item was drag and drop ranking-order type, one item required matching and lastly one item required students to enter a number. One of the Bebras items (Robot Exit) was modified to make it a multiple-choice question as it originally required an interactive drag and drop of instructional blocks guiding the robot. Both CTt and Bebras items were given to 15 middle school students before the main data collection to make sure that the questions are appropriate for this age group.



Which instructions take 'Pac-Man' to the ghost by the path marked out?

Option A

```
repeat 4 times
do  repeat 3 times
    do  move forward
    turn right ↻
move forward
```

Option B

```
repeat 3 times
do  repeat 4 times
    do  move forward
    turn right ↻
move forward
```

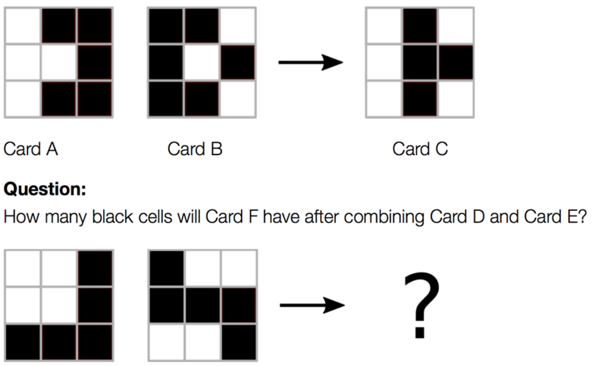
Option C

```
repeat 3 times
do  repeat 4 times
    do  move forward
        turn right ↻
move forward
```

Option D

```
repeat 4 times
do  move forward
repeat 3 times
do  turn right ↻
move forward
```

Figure 1: A sample item from CTt



Combining Card A and Card B, you get Card C:

Question:

How many black cells will Card F have after combining Card D and Card E?

Figure 2: A sample item from the Bebras Challenge

3.2 Data Sources

A Non-Equivalent Anchor Test (NEAT) design was used to maximize the data collected from our sample as it was infeasible to administer the 43-item assessment due to time constraints in the classroom. A NEAT design is one in which multiple versions of an exam are given; in this study there were 10 separate versions. All versions share a subset of anchor items (i.e., common items) and also contain a number of unique items. Anchor items are used to equate student scores on different versions of the assessment. The first set of 5 versions each contained 26 items, as that was the estimated number that could be completed in the allotted time (~45 min). The 26 items consisted of 17 items from the CTt (14 anchor, 3 unique) and 9 items from the Bebras challenge (7 anchor, 2 unique). The selection of anchor items from the CTt was based on the computational concepts (e.g., "basic directions and sequences", "simple functions") that each item addresses in the test. The CTt aims to assess seven computational concepts in total and each concept was addressed by four questions (Roman-Gonzalez et al., 2017, p. 681). We selected two questions addressing each concept as anchor questions, so each version of the assessment included 14 questions from the CTt that addressed all seven computational concepts defined by Roman-Gonzalez et al. [25]. The rest of the CTt items were then distributed across the 5 versions as items unique to each version. The selection of anchor items from the Bebras challenge was based on the difficulty level of the items (A = easy, B = moderate, C = difficult) set by the creators of the tasks. We selected three items from A level, two from B, and two from C as anchor items. Then, we distributed the rest of the items as unique items across the five versions in a way that each version included at least one C level item and one A/B level item. The second set of 5 shortened versions each contained 16 items, which consisted of 10 items from the CTt (7 anchor, 3 unique) and 6 items from the Bebras challenge (4 anchor, 2 unique), and did not include some items that were eliminated based on preliminary results. Since the items unique to each version had the highest amount of missing data, we decided to keep the unique items in the shortened versions and drop some of the anchor items in a way that it does not change our rationale for selecting those items. While the NEAT design is popular and efficient for deploying more items, it does result in a great deal of data that is missing by design.

We then used concurrent equating to put all items from all test versions onto the same scale, allowing all of our administrations of smaller numbers of tests to be considered together and with equal contribution to the model. When the sample size reached 160, a preliminary Rasch analysis was conducted for the purpose of identifying items which were not performing well or as expected; 3 items were eliminated based on negative point-biserial correlation or very low variance. The remaining rounds of data collection were then conducted with the most problematic items removed. In addition, a subset of 21 students took the assessment twice in order to inform additional discussion of test-retest reliability. In total, 309 students completed the assessment.

3.3 Results

3.3.1 Preliminary Item Fit Analysis. To help select the best items to include in a final assessment, a form of Item Response Theory (IRT), Rasch analysis, was used to both look at the performance of individual items tested and the relationship of student responses on individual items with other items [45]. Rasch measures a sample of respondents' (i.e., students) performance on each question as a function of a latent trait θ. A model of each item, an Item Characteristic Curve (ICC), maps the probability (y-axis) that an individual would answer correctly against the individual's proficiency (θ; x-axis). One parameter for an item's ICC is its difficulty (β), where a higher β would indicate that a student would need to have a higher θ in order to have the same probability of answering it correctly. The discrimination parameter (α) indicates how effectively an item can discriminate between high and low-ability students. A high α would create a steeper sloped ICC, whereby small changes in θ would mean relatively larger shifts in probability of correct response. The approach here is to use these parameter characteristics, at the individual item level and collectively (i.e., across all items), to select a set of items with a range of difficulty and high discrimination, and which all behave statistically in a homogeneous fashion (e.g., have similar amounts of response variability).

With a goal of reducing the length of the test to the best performing items, our general approach was to use Rasch analysis [45] to investigate bad fit first for causes such as data entry errors or miskeys, and then eliminate items that still showed poor fit to the IRT model. We followed the recommendations of Linacre [46], for the order and priority of assessing items based on these statistics. Items with negative point-biserial correlations with either overall score or θ were removed first, next were violations of outfit and infit, then finally for mean-square and standardized mean square fit. Items that were very high or very low difficulty tended to be eliminated as they tended not to differentiate high and low ability participants well; this had the effect of trimming the correct response probability distribution somewhat. However, the resulting distribution was in line with the person ability distribution. In total, 17 items were eliminated via these methods.

Finally, as noted above, we re-administered the test to a small subset of 21 students that had taken the test during the prior semester. We found that one Bebras item, Bebras_Q2, showed a great deal more improvement than all other items. At time 1, 10% of participants answered correctly but at time 2, 43% answered correctly. Other than this item, items were reliable between administrations, with a tendency towards slight improvement at time 2. Because it was impossible to determine if there was an event that influenced a collective response to Bebras_Q2 or if the pattern was due to exposure effect, we decided to eliminate Bebras_Q2.

In summary, we eliminated 18 total items based on these criteria, and the final IRT model was refit to the data. Using the above-described techniques the remaining items, by process of elimination, demonstrated stable patterns of response across students and were of a range of difficulty that was both appropriate for middle grades students, but would still allow discrimination between levels of ability.

3.3.2 Analysis of Dimensionality. With the goal of evaluating whether the remaining items were measuring a single dimension of ability, a Rasch analysis was conducted with a final sample size of 308, after one student was eliminated for having all missing data after item reduction. Item parameters were estimated using the marginal maximum likelihood (MML) with an expectation-maximization (EM) algorithm using XCalibre software (Version 4.2; Assessment Systems, 2014). Items that were omitted by individuals were calculated as being incorrect. After the IRT model was fit to the data, maximum likelihood estimation (MLE) was used to calibrate the items.

We first tested whether the 1-parameter Rasch model was greatly violating the assumption of a reasonably stable item discrimination (alpha parameter) by examining item plots and by comparing model fit to a 2 parameter (2PL) model ([47], pg. 141). The item plots showed that the discrimination, indicated by the slope of the traces, was stable for almost all items. Furthermore, a comparison of AIC and BIC statistics (see Table 1) indicate that there is no meaningful difference between the 1PL (Rasch) and 2PL models, therefore the 1PL model is suitable for these data.

Based on our use of a NEAT design, we chose to compare model fit for a unidimensional vs. a two-dimensional model that was comprised of CTt items in one dimension and Bebras items in the other. The mirt package in R [48] was used for this task using item parameters from the XCalibre program output as starting values. The same estimation techniques were used (MML-EM) to calculate IRT model fit statistics for each confirmatory model (see Table 1) and the results showed a non-meaningful improvement of about 1%, indicating the more parsimonious 1PL, 1 factor model is preferable [47].

In summary, each instance of analysis, the unidimensional Rasch model was roughly equivalent or superior [48]. Under the new model, there were no violations of item fit statistics, and a person ability-item difficulty map indicated that there was good coverage of person ability levels by the assessment.

Model	-2LL	Relative Change	AIC	BIC
1PL,1 Factor	4626.892	Reference	4628.893	4632.623
2PL,1 Factor	4799.596	0.005	4651.596	4744.848
1PL,2 Factor	4699.452	-0.016	4703.453	4710.913

Table 1: Comparison of 1PL (Rasch) and 2PL Models

4 CONCLUSIONS AND FUTURE WORK

One of the main purposes of this study was to reduce a longer, preliminary hybrid Román-González – Bebras CT assessment into a short and well-performing assessment that could be administered in a reasonable time period. The conjecture by Román-González that CTt and Bebras items could be used together in a complimentary fashion seems to have been borne out based on our IRT-guided process of item reduction. Our final recommended set of 25 items contain both CTt (n=19) and Bebras (n=6) items.

This CT assessment instrument is designed to be used as a pre-assessment for students who may not have had any previous experience programming. While there might have been concerns that some of the CTt items may have used block-based programming representations requiring prior programming experience, prior literature [35] and this analysis seems to indicate otherwise. There is reason to be conservative in this conclusion until further research is conducted.

The instructional implication of a short, easily administered CT pre-assessment is the ability for both teachers and researchers to gauge the initial ability of students to engage in tasks (programming or otherwise) that require the current, generally recognized CT skill set. When used as a pre and post-assessment instrument it has the potential to provide insight as to the efficacy of the intervention or instruction to develop CT ability. The CTt has demonstrated sensitivity to relatively short interventions [35], but more research is needed to explore how this new 25-item instrument responds.

Another motivation for continued validation work on this set of 25 items is the NEAT data collection design. Though necessary and appropriate for the logistical constraints of this data collection, it resulted in some sparsity of responses. A new round of data collection with just the final 25 items, randomly administered across participants will help alleviate data sparsity and address any ordering effects. While the results of this study suggest that the hybrid assessment is unidimensional, further research focusing on the shortened exam alone would allow for a deeper look into dimensionality. Finally, while some of the prior research has pointed to the strong conceptual and theoretical relationship of the CTt and Bebras instruments to current CT frameworks, more work will be needed to continue to explore the criterion validity of this new instrument against the CHC model of intelligence, programming ability, and other related measures of CT.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation through Grants CNS-1138497 and DRL-1640141. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

REFERENCES

[1] Goode, J. and Chapman, G. *Exploring Computer Science.* University of Oregon, Eugene, OR, 2016.

[2] Lee, I., Martin, F. and Apone, K. Integrating computational thinking across the K-8 curriculum. *ACM Inroads*, 5, 4 (2014), 64-71.

[3] Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. and Settle, A. 2014. Computational Thinking in K-9 Education. In *Proceedings of the Proceedings of the Working Group Reports of the 2014 on Innovation and Technology in Computer Science Education Conference.* ACM, 1-29.

[4] Zur-Bargury, I., Parv, B. and Lanzberg, D. 2013. A nationwide exam as a tool for improving a new curriculum. In *Proceedings of the Proceedings of the 18th ACM conference on Innovation and technology in computer science education.* ACM, 267-272.

[5] Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C. and Shoop, R. Developing Computational Thinking through a Virtual Robotics Programming Curriculum. *ACM Trans. Comput. Educ.*, 18, 1 (2017), 1-20.

[6] Martin, L. The promise of the Maker Movement for education. *Journal of Pre-College Engineering Education Research (J-PEER)*, 5, 1 (2015), 4.

[7] Rodger, S. H., Hayes, J., Lezin, G., Qin, H., Nelson, D., Tucker, R., Lopez, M., Cooper, S., Dann, W. and Slater, D. 2009. Engaging middle school teachers and students with alice in a diverse set of subjects. In *Proceedings of the Proceedings of the 40th ACM technical symposium on Computer science education (SIGCSE)*. ACM, 271-275.

[8] Bienkowski, M., Snow, E., Rutstein, D. W. and Grover, S. *Assessment design patterns for computational thinking practices in secondary computer science: A first look*. SRI International, Menlo Park, CA, 2015.

[9] Grover, S. and Pea, R. Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42, 1 (2013), 38-43.

[10] Wing, J. M. Computational Thinking. *Communications of the ACM*, 49, 3 (2006), 33-35.

[11] K12CS *K–12 Computer Science Framework*. 2016.

[12] NRC, N. R. C. *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. The National Academies, Washington, DC, 2011.

[13] Buffum, P. S., Martinez-Arocho, A. G., Frankosky, M. H., Rodriguez, F. J., Wiebe, E. N. and Boyer, K. E. 2014. CS principles goes to middle school: learning how to teach Big Data. *Proceedings of the 45th ACM technical symposium on computer science education (SIGCSE '14)*. ACM, 151-156.

[14] Jona, K., Wilensky, U., Trouille, L., Horn, M., Orton, K., Weintrop, D. and Beheshti, E. 2014. Embedding computational thinking in science, technology, engineering, and math (CT-STEM). *Future Directions in Computer Science Education Summit Meeting*.

[15] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. and Wilensky, U. Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25, 1 (2016), 127-147.

[16] Grover, S. 2017. Assessing Algorithmic and Computational Thinking in K-12: Lessons from a Middle School Classroom. In *Emerging Research, Practice, and Policy on Computational Thinking. Educational Communications and Technology: Issues and Innovations*. Springer, 269-288.

[17] Grover, S., Cooper, S. and Pea, R. 2014. Assessing computational learning in K-12. In *Proceedings of the 2014 conference on Innovation & technology in computer science education (ITTICSE '14)*. ACM, 57-62.

[18] Shute, V. J., Sun, C. and Asbell-Clarke, J. Demystifying computational thinking. *Educational Research Review*, 22 (2017), 142-158.

[19] Brennan, K. and Resnick, M. 2012. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*. AERA.

[20] Denner, J., Werner, L. and Ortiz, E. Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, 58, 1 (1// 2012), 240-249.

[21] Tew, A. E. and Guzdial, M. 2011. The FCS1: a language independent assessment of CS1 knowledge. In *Proceedings of the Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM, 111-116.

[22] Weintrop, D. and Wilensky, U. 2015. Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. In *International Computing Education Research Conference* (*ICER '15)*. ACM, 101-110.

[23] Taylor, C., Zingaro, D., Porter, L., Webb, K. C., Lee, C. B. and Clancy, M. Computer science concept inventories: past and future. *Computer Science Education*, 24, 4 (2014), 253-276.

[24] Curzon, P., McOwan, P. W., Plant, N. and Meagher, L. R. 2014. Introducing teachers to computational thinking using unplugged storytelling. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*. ACM, 89-92.

[25] Román-González, M., Moreno-León, J. and Robles, G. 2017. Complementary tools for computational thinking assessment. In *Proceedings of international conference on computational thinking education (CTE 2017)*. The Education University of Hong Kong, 154-159.

[26] Román-González, M., Pérez-González, J.-C. and Jiménez-Fernández, C. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior* 72 (2016), 678-691.

[27] Román-González, M., Pérez-González, J.-C., Moreno-León, J. and Robles, G. Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior*, 80 (2018), 441-459.

[28] Alfonso, V. C., Flanagan, D. P. and Radwan, S. *The impact of the Cattell-Horn-Carroll theory on test development and interpretation of cognitive and academic abilities*. Guilford Publications, City, 2005.

[29] McGrew, K. S. CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research. *Intelligence*, 37, 1 (2009), 1-10.

[30] Ambrósio, A. P., Xavier, C. and Georges, F. 2014. Digital ink for cognitive assessment of computational thinking. In *Proceedings of the 2014 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1-7.

[31] Basawapatna, A. R., Koh, K. H. and Repenning, A. 2010. *Using scalable game design to teach computer science from middle school to graduate school*. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*. ACM, 224-228.

[32] Werner, L., Denner, J. and Campe, S. 2012. *The Fairy Performance Assessment: Measuring Computational Thinking in Middle School*. In *Proceeding of the 44th ACM technical symposium on computer science education (SIGCSE '12)*. ACM, 421-426.

[33] Dagienė, V. and Futschek, G. 2008. Bebras international contest on informatics and computer literacy: Criteria for good tasks. In *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*. Springer, 19-30.

[34] Dagienė, V. and Sentance, S. 2016. It's Computational Thinking! Bebras Tasks in the Curriculum. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. Springer, 28-39.

[35] Aksit, O. *Enhancing Science Learning through Computational Thinking and Modeling in Middle School Classrooms: A Mixed Methods Study*. Dissertation, North Carolina State University, Raleigh, NC, 2018.

[36] Moreno-León, J. and Robles, G. 2015. Dr. Scratch: A web tool to automatically evaluate Scratch projects. In *Proceedings of the workshop in primary and secondary computing education (WiPSCE '15)*. ACM, 132-133.

[37] Blokhuis, D., Millican, P., Roffey, C., Schrijvers, E. and Sentance, S. *UK Bebras Computational Thinking Challenge 2016*. University of Oxford, Oxford, UK, 2015.

[38] Barendsen, E., Mannila, L., Demo, B., Nata, Grgurina, A., Izu, C., Mirolo, C., Sentance, S., Settle, A., Gabriel, S. 2015. Concepts in K-9 Computer Science Education. In *Proceedings of the 2015 ITiCSE on Working Group Reports*. ACM, 85-116.

[39] Dagienė, V., Stupurien, G. and Vinikien, L. 2016. Promoting Inclusive Informatics Education Through the Bebras Challenge to All K-12 Students. In *Proceedings of the Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*. ACM, 407-414.

[40] Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A. and Torelli, M. 2015. How Challenging are Bebras Tasks?: An IRT Analysis Based on the Performance of Italian Students. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, 27-32.

[41] Gujberova, M. and Kalas, I. 2013. Designing productive gradations of tasks in primary programming education. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education*. ACM, 108-117.

[42] Hubwieser, P. and Muhling, A. 2014. Playing PISA with Bebras. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*. ACM, 128-129.

[43] Izu, C., Mirolo, C., Settle, A., Mannila, L. and Stupuriene, G. Exploring Bebras Tasks Content and Performance: A Multinational Study. *Informatics in Education*, 16, 1 (2017), 39-59.

[44] Dagienė, V., Mannila, L., Poranen, T., Rolandsson, L. and Derhjelm, S. 2014. Students' performance on programming-related tasks in an informatics contest in Finland, Sweden and Lithuania. In *Proceedings of the 2014 conference on Innovation; technology in computer science education*. ACM, 153-158.

[45] Fischer, G. H. and Molenaar, I. W. *Rasch models: Foundations, recent developments, and applications*. Springer Science, 2012.

[46] Linacre, J. M. *Winsteps®*. Winsteps.com, 2018.

[47] de Ayala, R. J. *The Theory and Practice of Item Response Theory*. Guilford Press, New York, 2009.

[48] Chalmers, R. P. mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software*, 48, 6 (2012), 1-29.