Development and Preliminary Validation of the Assessment of Computing for Elementary Students (ACES)

Miranda C. Parker
University of California, Irvine
Irvine, CA, USA
miranda.parker@uci.edu

Yvonne S. Kao
WestEd
Redwood City, CA, USA
ykao@wested.org

Dana Saito-Stehberger
University of California, Irvine
Irvine, CA, USA
dsaitost@uci.edu

Diana Franklin
University of Chicago
Chicago, IL, USA
dmfranklin@uchicago.edu

Susan Krause
University of Chicago
Chicago, IL, USA
sgkrause@uchicago.edu

Debra Richardson
University of California, Irvine
Irvine, CA, USA
djr@uci.edu

Mark Warschauer
University of California, Irvine
Irvine, CA, USA
markw@uci.edu

ABSTRACT

As reliance on technology increases in practically every aspect of life, all students deserve the opportunity to learn to think computationally from early in their educational experience. To support the kinds of computer science curriculum and instruction that makes this possible, there is an urgent need to develop and validate computational thinking (CT) assessments for elementary-aged students. We developed the Assessment of Computing for Elementary Students (ACES) to measure the CT concepts of loops and sequences for students in grades 3-5. The ACES includes block-based coding questions as well as non-programming, Bebras-style questions. We conducted cognitive interviews to understand student perspectives while taking the ACES. We piloted the assessment with 57 4th grade students who had completed a CT curriculum. Preliminary analyses indicate acceptable reliability and appropriate difficulty and discrimination among assessment items. The significance of this paper is to present a new CT measure for upper elementary students and to share its intentional development process.

CCS CONCEPTS

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

KEYWORDS

assessment, computational thinking, elementary education

ACM Reference Format:
Miranda C. Parker, Yvonne S. Kao, Dana Saito-Stehberger, Diana Franklin, Susan Krause, Debra Richardson, and Mark Warschauer. 2021. Development and Preliminary Validation of the Assessment of Computing for Elementary Students (ACES). In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432376

SIGCSE '21, March 13–20, 2021, Virtual Event, USA.

ACM ISBN 978-1-4503-8062-1/21/03.
https://doi.org/10.1145/3408877.3432376

1 INTRODUCTION

As reliance on technology increases in practically every aspect of life, all students deserve the opportunity to learn to think computationally from early in their educational experience. Support for computational thinking (CT) and computer science (CS) instruction at the elementary school level continues to gain momentum [5, 12, 34]. Researchers in the computing education field have been rapidly developing tools to teach elementary computing, including formal curricula like Scratch Act 1 [9, 24], coding platforms like Scratch Jr. [26], and commercial products like CodeSpark Academy [3].

As these efforts to teach CT and CS at the elementary level grow and reach maturity, there has been a corresponding increase in need for elementary-level CT and CS assessments. Assessment data is a critical aspect of evaluating a program's effectiveness. Assessments allow program developers to operationalize a program's intended outcomes and have a common point of reference for student performance across cohorts.

Thus, there is a demand for valid and reliable assessments to provide student benchmarks and to measure student gains in the area of CT at the elementary school level. Assessments also provide a means to analyze the effectiveness of curriculum. Researchers have developed and validated a variety of assessments for K-12 CT and CS in the last decade [21, 23, 31], but there has not been as much progress at the elementary level compared to middle and high school. Decker and McGill [8] conducted a literature review to catalog instruments to measure a variety of cognitive (i.e., content knowledge) and non-cognitive constructs (e.g., attitudes towards computing). Only 6% of instruments listed in that review were designed for elementary students (grades K-5).

In this paper, we present the development and initial psychometric analysis of the Assessment of Computing for Elementary Students (ACES), which seeks to help fill this gap in the computing education field. We address the following research questions: *How can we design a CT assessment for upper elementary students? What are the implications of using the assessment based on its psychometric properties?*

In the following section, we discuss existing CT assessments for elementary and middle school students and the challenge in adapting an assessment intended for middle school students to be used with upper elementary students. In Section 3, we provide information on the curriculum our assessment is designed to be used with and how that influenced our design of the assessment. Then, we discuss the development of the ACES, including the constructs it measures and how the items were created. In Sections 5 and 6, we present findings from our initial analysis of the ACES, from cognitive interviews and a pilot study with 57 4th grade students. We discuss the implications of these findings in Section 7. We conclude in Section 8, including steps for future work and discussing limitations.

2 ASSESSING K-12 CS AND CT

2.1 Assessments for Elementary Students

The CSEdResearch.org database is regularly updated with new instruments that are published in the research literature. As of this writing, there are only two records for elementary school instruments that measure CS or CT knowledge. One record is for Project Quantum, which is a crowd-sourced bank of computing quizzes rather than a validated instrument [20]. The other instrument was developed by Project TREES and measures student performance on the six dimensions of CT jointly developed by the International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) [10]. This assessment contains a mix of multiple-choice and open-ended items and was field-tested with 5th grade robotics students [2]. Though a valid instrument, the Project TREES assessment did not align well with our constructs of interest.

2.2 Assessments for Middle School Students

There are more, but still a limited number, of validated assessments developed for middle school students (grades 6-8). The Computational Thinking test (CTt) [16, 23] consists of 28 multiple-choice items. The CTt assesses sequences, loops, interaction, conditionals, functions, and variables. All test items present students with a character that must complete a specific task (i.e., follow a path or draw a specific shape). Students must identify which set of instructions, sometimes written in block-based pseudocode, would accomplish the given task. The CTt requires approximately 45 minutes to complete.

The Computational Thinking Abilities - Middle Grades Assessment (CTA-M) [31] contains a mix of 19 items from the CTt and six multiple-choice tasks from the Bebras International Contest on Informatics and Computer Fluency [6, 7, 29]. The Bebras tasks do not require any programming background. Instead, computational concepts are presented in a story context. For example, students might be asked to sequence a series of pictures to create a smooth animation [19]. The CTA-M is designed to be completed within a 50-minute class period.

The Middle Grades Computer Science Concept Inventory (MG-CSCI) Assessment [21] is a 24-item multiple-choice assessment of CS concepts. The MG-CSCI uses the Scratch block-based programming language and is based on the Commutative Assessment [30]. It measures student understanding of variables, conditionals, loops, and algorithms.

2.3 Challenges of Adapting Middle School Assessments for Upper Elementary

At first glance, it seems reasonable to try to adapt a middle school CT or CS assessment for use in the elementary context. We considered this option to create a 4th grade CS assessment for our study of the IMPACT curriculum, described further below. However, this approach is not necessarily more straightforward than developing a new assessment. Middle school assessments have many items that are too difficult even for upper elementary students. Using a subset of the easiest items does not necessarily result in an appropriate distribution of items by content or by difficulty. It would have been necessary to develop new items to ensure adequate content coverage and to capture the lower end of 4th grade CS performance, especially on a pre-test. Thus, we opted to develop a new assessment, the Assessment of Computing for Elementary Students (ACES).

3 THE IMPACT CURRICULUM

The ACES was designed to be used in a randomized controlled trial comparing the CT performance of students who participated in a CT curriculum integrated into their English language arts classes with students with business-as-usual instruction. The curriculum to be tested is the IMPACT curriculum. The IMPACT curriculum is an adaptation of the Creative Computing curriculum that was developed by the ScratchEd Team at Harvard Graduate School of Education and Code.org. The IMPACT curriculum aims to introduce multi-lingual students in grades 3-5 to foundational CT concepts and practices. The curriculum provides engaging exploration and practice through inquiry-based processes, such as Use-Modify-Create[14] and TIPP&SEE [24]. The content is organized into the following five units:

- Unit 1: Introduction to CS and Scratch interface
- Unit 2: Algorithm, program, and sequence
- Unit 3: Events
- Unit 4: Loops
- Unit 5: Synchronization

In the first unit, students learn definitions of “computer science" and “program," as well as how to use the Scratch interface and run a program. In Unit 2, students learn about algorithms and learn to create scripts with several actions that must be run in the proper order. Students learn about events-based programming in Unit 3, such as that scripts are triggered when specific events occur (e.g. a button pressed or the mouse is clicked). In Unit 4, students learn how to use loops and compare scripts with and without loops to evaluate similarity; students are not, however, taught about nested

loops or infinite loops. In the last unit, students learn about synchronization in Scratch to coordinate actions between sprites, such as conversations.

4 DEVELOPING THE ASSESSMENT OF COMPUTING FOR ELEMENTARY STUDENTS (ACES)

4.1 Constructs of Measurement

Of the five units in the IMPACT curriculum, we selected two constructs to measure in the assessment: sequences and loops. The IMPACT curriculum devotes one full unit to each of these constructs. We chose not to include the other three units in this assessment to make the measure more broadly applicable. Introduction to Scratch, Events, and Synchronization focus on constructs that are somewhat Scratch-specific. In contrast, Sequences and Loops are concepts that are part of the introductory computer science core [28].

To further define and operationalize the constructs of measurement, we turned to the learning trajectories for sequences and repetition, or loops, developed by the Learning Trajectories for Everyday Computing (LTEC) project [22]. Learning Trajectories are hypothesized paths of knowledge building that students can move through on their learning journey [27]. The exact path a student follows is influenced by the curriculum [1, 25], environment, and peers [11]. Nonetheless, they are useful tools for building curriculum and have been used extensively in mathematics [4]. Likewise, they can be used to create assessments, allowing designers to more concretely identify the individual learning goals involved in a complex subject, allowing for more targeted questions.

The LTEC learning trajectories were developed following a literature review to identify consensus learning goals related to each concept. The consensus learning goals were then assembled into trajectories that build from students' everyday, "unplugged" (i.e., offline, non-programming) knowledge to programming-specific applications of each idea. The learning goals in each trajectory are grouped into beginning, intermediate, and advanced ideas.

Because this assessment is intended for upper elementary students who may or may not have had prior programming instruction, we chose to focus on three of the "unplugged" learning goals:

(1) *Sequences*: Different sets of instructions can produce the same outcome.
(2) *Sequences*: The order in which instructions are carried out can affect the outcome.
(3) *Loops*: Instructions like "step 3 times" do the same thing as "step, step, step."

The first two goals are intermediate-level goals that belong to the sequences learning trajectory. The third goal is a beginning-level goal that belongs to the repetition (loops) trajectory.

4.2 Item Design

We had three major design goals for the assessment. First, we wanted upper elementary students to be able to complete the assessment well within one class period. Second, the assessment had to be computer scorable. Third, we wanted the assessment to be appropriate for students who did not have prior programming experience. This would allow the assessment to be used as a pre-test as well as a post-test. The assessment could also be used to compare the CT performance of students who received a programming-oriented CT intervention with a comparison group of students who did not.

Figure 1: An example of the "turn right" block created to be Scratch-like, but use more universal language

Table 1: Summary of the ACES items

#	Construct(s)	Prompt	Response
1	Sequences	Code blocks	Multiple-choice
2	Sequences	Code blocks	Multiple-choice
3	Sequences	Code blocks	Multi-select
4	Loops	Code blocks	Multiple-choice
5	Loops	Code blocks	Multi-select
6	Sequences+Loops	Code blocks	Multiple-choice
7	Sequences+Loops	Code blocks	Multiple-choice
8	Sequences+Loops	Code blocks	Ordering
9	Sequences	Bebras-style	Multiple-choice
10	Loops	Bebras-style	Multiple-choice

To accomplish the first and second design goals, we limited ourselves to writing ten close-ended questions. Question types included a mix of multiple-choice, multi-select, and one ordering task. We used the Sequence and Events assessments from the Scratch Act 1 curriculum as a starting point [9, 24], as it targets a similar age group and covers similar content. We used two strategies to ensure the assessment would be appropriate for programming novices. First, we used Scratch-like blocks, but changed some of the block names to use more universal language. For example, instead of using the "turn" blocks in Scratch, which use arrows to indicate direction and take an argument to specify the number of degrees to turn, we created simple "turn left" and "turn right" blocks to perform 90-degree turns (as seen in Figure 1). Second, we included two multiple-choice items adapted from Bebras tasks [7, 29] in order to present sequences and loops in non-programming contexts. For example, Question #9 from our assessment asks students to look at a place setting and identify the sequence used to set the items on the table (see Figure 2). We label these questions as being "Bebras-style" as they are inspired by Bebras tasks but are not taken directly from the set of available Bebras questions. Table 1 shows the overall structure of the ACES.

5 COGNITIVE INTERVIEWS

Cognitive interviews were used to explore students' comprehension of the assessment [17, 32]. Cognitive interviews are often used in the development of surveys and assessments to evaluate the questions to improve them before the instrument is administered at scale [33]. These cognitive interviews, which are similar to think aloud-interviews [15], started with the interviewer introducing

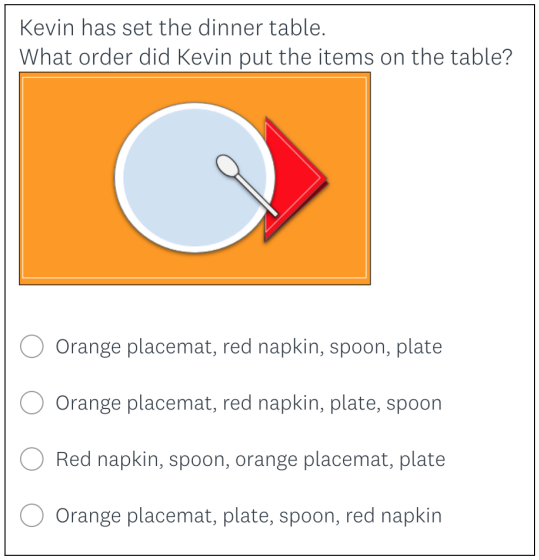

Figure 2: Question 9 of the ACES, based on the Bebras Challenge questions

themselves and the assessment. The interviewer would prompt the student to think-aloud as they read the questions and thought through the answers. However, students were also asked probing questions based on their responses. These questions could include asking students to further explain their selection, or could be focused on a feature of the question, such as a word, phrase, or code block.

We conducted four cognitive interviews with students in two different classrooms that had seen the IMPACT curriculum. These interviews were conducted one-on-one and virtually, using Zoom to record audio and video. The student's teacher was on the call for each interview. Each interview was approximately thirty minutes in length and each student completed the entire assessment during that time. One author conducted these interviews while taking notes. These notes were discussed with the other authors to agree on the changes needed to improve the assessment.

Through this process, we discovered questions that needed to be re-formatted to improve the user experience or revised to avoid confounding variables. Using the cognitive interviews, we found a number of *interface issues* that needed to be adjusted. For example, one question required students to scroll to see the answers and then scroll back to re-visit the question. After finding this in the cognitive interviews, we re-formatted the question to have it viewable on one screen and thereby eliminate the need to scroll to see different parts of the question. We also found instances of *confounding variables* that would affect performance on this CT assessment. One of the questions involved mathematical operations, including adding, multiplying, and subtracting. However, the cognitive interviews revealed that all three of these mathematical operations may be too much to ask of some upper elementary students. Performance on that question would confound mathematical skills and CT ability (specifically about sequences). As such, we revised the question to only use addition and subtraction.

These interviews also hinted at which questions may be more or less difficult than other questions and why. We will discuss these in conjunction with our item analysis in Section 7.

6 PILOT STUDY

6.1 Study Procedure

After edits were made to the assessment following the cognitive interviews, we piloted the ACES. This pilot study occurred with 57 4th grade students. These students came from five different classrooms of teachers that taught the IMPACT curriculum over the course of the school year. This served as their end-of-course assessment.

Students took the assessment on SurveyMonkey. Some teachers decided to virtually proctor the assessment synchronously by sharing their screen in Zoom with the ACES open in a web browser and having students follow along on their own devices. Teachers chose this method to encourage participation by students and help answer any clarification questions students had in a centralized fashion (i.e. the teacher did not have to answer the same question from students multiple times). Other teachers let their students take the assessment asynchronously. We sent reminders to teachers to encourage students to complete the assessment, and some teachers asked for a list of which of their students completed the assessment so they could remind the students that had not taken it yet.

6.2 Data Analysis and Results

After cleaning the data for complete and non-duplicate responses, we were left with 57 submissions of the ACES. On average, it took students 17 minutes to complete the assessment.

We analyzed the 57 responses using three different scoring mechanisms. Some questions on the ACES have only one correct answer and students can only select one answer ("multiple choice"). However, some questions have multiple correct answers and students can select multiple answers ("multiple select" or "multi-select"). Multiple choice questions were always graded the same, receiving one point for a correct answer. However, we saw multiple options for scoring the multi-select questions. The "Each Question" approach gave students one point for correctly answering every part of a question. For multi-select questions, this method meant if a student selected every right answer, and no wrong answer, they received one point. If they selected a wrong answer or didn't select a right answer, they received zero points. The "Each Item" approach gave students one point for each part of the question they answered correctly. This meant that if given five options to select from (A, B, C, D, and E) and two were correct (B and D) and three were incorrect (A, C, and E), then a student that answered B and E would receive three out of five possible points (one point for selecting one correct answer, and two additional points for not selecting two incorrect answers). The "Each Item Normalized" approach was similar to the "Each Item" approach except each question was normalized to have a maximum score of one, so fractional scores could occur. For the example in the "Each Item" approach, the student would have received a score of 0.6 for that question.

For each scoring method, we found the difficulty and discrimination of each item (where an item could be a question or an answer choice depending on the scoring method), as well as the reliability

Table 2: Preliminary Validation Results of the ACES, using three different scoring methods

		Each Item (α = 0.686)				Each Question (α = 0.489)				E.I. Normalized (α = 0.549)			
	Item	Diff.	DI	PBC	Drop α	Diff.	DI	PBC	Drop α	Diff.	DI	PBC	Drop α
Q1	I1	0.89	0.05	0.08	0.693	0.89	0.05	0.06	0.532	0.89	0.20	0.10	0.592
Q2	I2	0.61	0.44	0.44	0.668	0.61	0.54	0.56	0.412	0.61	0.54	0.57	0.498
Q3	I3	0.73	0.45	0.32	0.680	0.50	0.44	0.43	0.475	0.75	0.24	0.36	0.537
	I4	0.82	0.24	0.38	0.673								
	I5	0.70	0.24	0.34	0.679								
Q4	I6	0.68	0.50	0.44	0.668	0.68	0.45	0.49	0.445	0.68	0.45	0.50	0.524
Q5	I7	0.98	0.05	0.18	0.684	0.32	0.42	0.49	0.445	0.76	0.26	0.58	0.490
	I8	0.96	0.10	0.20	0.683								
	I9	0.82	0.40	0.60	0.652								
	I10	0.34	0.48	0.48	0.664								
	I11	0.70	0.54	0.49	0.662								
Q6	I12	0.46	0.43	0.42	0.671	0.46	0.59	0.53	0.429	0.46	0.69	0.58	0.493
Q7	I13	0.50	0.59	0.44	0.669	0.04	0.00	0.05	0.507	0.54	0.18	0.42	0.520
	I14	0.89	0.05	0.16	0.688								
	I15	0.20	-0.14	-0.05	0.709								
	I16	0.59	0.49	0.44	0.669								
Q8	I17	0.79	0.50	0.65	0.645	0.79	0.55	0.62	0.381	0.85	0.37	0.61	0.471
	I18	0.86	0.30	0.47	0.666								
	I19	0.89	0.30	0.53	0.662								
Q9	I20	0.57	0.49	0.45	0.667	0.57	0.54	0.42	0.480	0.57	0.49	0.49	0.534
Q10	I21	0.36	0.18	0.03	0.710	0.14	0.32	0.31	0.484	0.65	0.17	0.26	0.548
	I22	0.91	0.15	0.23	0.683								
	I23	0.61	0.34	0.33	0.681								
	I24	0.73	0.19	0.22	0.690								

Diff. = Difficulty, DI = Discrimination index, PBC = Point-biserial correlation, Drop α = the resulting α if item were removed

of the assessment overall and if each item were removed. The results of these analyses can be found in Table 2. Results that are of concern (difficulty more than 0.8 or less than 0.2, discrimination and point-biserial correlation between -0.2 and 0.2, or a reliability increase if the item were dropped) are highlighted in the table. A difficulty of more than 0.8 indicates the item is easy, and can be read as "more than 80% of students answered this question correctly." Contrarily, a difficulty of less than 0.2 indicates the item is difficult, and can be read as "less than 20% of students answered this question correctly." The discrimination index (DI) is calculated by subtracting the average score of the lowest-performing third of students from the average score of the highest-performing third of students. A discrimination value between -0.2 and 0.2 indicates that performance on this question does not correspond with performance on the assessment overall, as there is not a large difference between the average score of the highest-performing and lowest-performing students. The point-biserial correlation (PBC) is a Pearson Product-Moment Correlation Coefficient between the scores on the question and the scores on the assessment overall. The PBC is a different way to assess discrimination. Cronbach's alpha (α) measures internal consistency of the assessment, or how closely related the items are. An α of 0.7 is sufficient for early-stage research, which is the case here, but an alpha of at least 0.8 would be necessary for use in a program evaluation or efficacy study [13, 18].

7 DISCUSSION

Based on our results in Table 2, Question 1 (Item 1), Question 7 (Items 14 and 15), and Question 10 (Item 21) were of particular interest and concern when considering revisions to the ACES.

Question 1 is a multiple-choice question on sequences. The question presents students with three connected code blocks, each "saying" a different part of a conversation. The question asks students to select what will be said last. In the cognitive interviews, one student did not answer this correctly. In the interview, the student noted that they answered what *they* would say last "in real life" if they were having that conversation, which was different from what the code says last. This indicated that the question might be easily confounded with social norms. Our analysis confirmed concerns with this question, as Question 1 is consistently highlighted in Table 2 for being relatively easy and having low discrimination. The reliability of the assessment also increases if that question were removed. All of these indicate that Question 1 needs to be improved.

Question 7 is a multiple-select question on sequences and loops. The question shows students an animation of a ladybug on a grid moving towards a star. The question also includes a looping (repeat) code block with code blocks to move the ladybug inside the looping block. The question asks students to complete the code to move the ladybug on top of the star. In the cognitive interviews, one student read the answer choices incorrectly, mistaking a "move left" block as a "move right" block. Move blocks are relative to the direction in

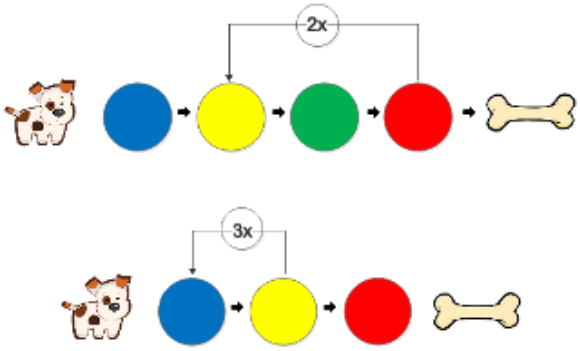

Figure 3: Items 21 and 22 on the ACES, indicating repeating dots in a pattern

which the sprite is pointed, so for a ladybug to move right on the screen, that may instead require a "move left" block if the ladybug is upside down. According to our pilot study, only 4% of students answered this question entirely correctly. The item that had the greatest difficulty, I15, was the item that required students to move relative to the sprite, not relative to the screen. As such, spatial skills were likely being compounded with their computational skills for this item.

Question 10 is a non-programming, Bebras-style question. The question displays an image of a dog with a grid of colored dots, including a "start" and "end" dot. The question asks students to identify which pattern of dots would lead the dog from the start to the end. When interviewing students, this was a question that required reformatting. Due to the size and orientation of the images, it required students to scroll back and forth between the image of the dog and grid and the images in the answer choices. However, the version in the pilot studies adjusted for this. Another pattern noticed in the interviews was with one particular answer choice, I21. Three of the four answer choices involved an arrow that pointed back to to a previous dot, indicating dots would be repeated in the pattern. However, as seen in Figure 3, I21 was the only one that involved an arrow that pointed back to two dots prior. In the interviews, students were confused about whether that meant those two dots were repeated (red-yellow-red) or if the dot between them would be repeated too (red-yellow-green-red). While only 36% of students answered this item correctly, that is still above our 20% threshold. However, the discrimination index and point-biserial correlation coefficient both indicate that this question does not predict performance on the assessment overall. Additionally, the reliability of the ACES as a whole would improve if this item were dropped or altered.

In terms of the multiple methods to score the assessment, the "Each Item" approach had the highest reliability with $\alpha = 0.686$. While this is below the acceptable threshold, adjustments on certain items or questions could make this reliability higher, and thus acceptable. However, this higher reliability could mostly be attributed to the higher number of items. With the other two approaches, the scores were out of ten, where each question received one point. The "Each Item" approach meant that scores were out of 24, where each item received one point. It is a side-effect of the way Cronbach's alpha is calculated that it is easier to get a higher reliability with more items.

8 CONCLUSIONS

In this paper, we present the development process of a new CT assessment for upper elementary students. The ACES builds on existing CT assessments for different grade levels. The ACES also caters to the sequences and repetition, or loops, sections of published learning trajectories. The questions are primarily block-based, but also include non-programming, Bebras-style questions to assess sequences and loops outside of a programming context. We present findings from cognitive interviews and a pilot study, discussing the initial reliability and argument for validity of the assessment. We found that, while certain questions and items were below our threshold values in terms of difficulty and discrimination, all issues could be supported by findings from the cognitive interviews. As such, we have now revised the assessment accordingly.

The limitations for this work include a lack of validity generalization and hypotheses, rather than proof, of cognitive processes. Our evidence for validity presented in this paper is restricted to the context in which the assessment was given. We do not have evidence for validity generalization, and thus researchers seeking to use this assessment in other settings should be aware of this limitation. Also, while the combination of cognitive interviews and the pilot study helped narrow in on questions that needed to be revised, we can only hypothesize why our pilot students chose the answers they did. We did not give the students spatial ability or mathematics assessments, and can not verify or concretely claim that performance on certain questions were confounding with spatial or math skills. We also did not conduct analyses to account for between-class differences which could arise from our student ample being from five different classrooms and teachers.

The ACES can accessed at www.impactconectar.org. While the ACES is already available for use, there is still an ongoing process to further develop and validate the assessment. Our development and pilot study occurred in Spring 2020. We have additional piloting planned for the 2020-2021 school year. We will be using this assessment to evaluate the IMPACT curriculum, and thus using it as a pre- and post- assessment. This data will help us further construct an argument for validity of this assessment with a larger set of students. We also plan on growing the ACES to include more units of the curriculum and other aspects of the learning trajectories. Although the Assessment of *Computing* for Elementary Students only currently covers loops and sequences, we plan on expanding to include other aspects of CT to live up to the namesake. Ideally, the ACES will become a modularized assessment, such that each aspect of CT can be measured individually or in combination with each other.

The ACES is part of a larger project focused on computational thinking for elementary students, with a specific aim of supporting English language learners. We are currently work on translating the assessment into Spanish and Chinese in order to offer the ACES in our students' primary languages. Piloting of the translations will begin in Spring 2021.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation through Grants #1923136 and #1660871 and by the United States Department of Education through Grant #U411C190092. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the United States Department of Education.

REFERENCES

[1] M. T. Battista. 2011. Conceptualizations and issues related to learning progressions, learning trajectories, and levels of sophistication. *The Mathematics Enthusiast* 8, 3 (2011), 507–570.

[2] Guanhua Chen, J. Shen, Lauren Barth-Cohen, Shiyan Jiang, X. Huang, and M. Eltoukhy. 2017. Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Comput. Educ.* 109 (2017), 162–175.

[3] CodeSpark. [n.d.]. *CodeSpark Academy.* https://codespark.com/

[4] J. Confrey, A. P. Maloney, and A. K. Corley. 2014. Learning trajectories: A framework for connecting standards with curriculum. *ZDM - International Journal on Mathematics Education* 46, 5 (2014), 719–733.

[5] Paul Curzon, Tim Bell, Jane Waite, and Mark Dorling. 2018. *Computational thinking.* Springer.

[6] Valentina Dagienė and Gerald Futschek. 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.

[7] Valentina Dagienė and Sue Sentance. 2016. It's Computational Thinking! Bebras Tasks in the Curriculum. In *Informatics in Schools: Improvement of Informatics Knowledge and Perception*, Andrej Brodnik and Françoise Tort (Eds.). Springer International Publishing, Cham, 28–39.

[8] Adrienne Decker and Monica M. McGill. 2019. A Topical Review of Evaluation Instruments for Computing Education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19).* Association for Computing Machinery, New York, NY, USA, 558–564. https://doi.org/10.1145/3287324.3287393

[9] Computing for ANyOne (CANON) Lab. [n.d.]. *Scratch Act 1.* https://www.canonlab.org/scratchact1modules

[10] International Society for Technology in Education and Computer Science Teachers Association. [n.d.]. https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

[11] D. Hammer and T. Sikorski. 2015. Implications of complexity for research on learning progressions. *Science Education 99* 3 (2015), 424–431.

[12] K-12 Computer Science Framework Steering Committee et al. 2016. K-12 computer science framework.

[13] Charles E. Lance, Marcus M. Butts, and Lawrence C. Michels. 2006. The Sources of Four Commonly Reported Cutoff Criteria: What Did They Really Say? *Organizational Research Methods* 9, 2 (2006), 202–220. https://doi.org/10.1177/1094428105284919 arXiv:https://doi.org/10.1177/1094428105284919

[14] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. Computational thinking for youth in practice. *Acm Inroads* 2, 1 (2011), 32–37.

[15] Jacqueline P Leighton. 2017. *Using think-aloud interviews and cognitive labs in educational research.* Oxford University Press.

[16] Marcos Román-González. 2015. Computational thinking test: Design guidelines and content validation. In *Proceedings of EDULEARN15 Conference.* Barcelona. https://doi.org/10.13140/RG.2.1.4203.4329

[17] Kristen Miller, Valerie Chepp, Stephanie Willson, and Jose-Luis Padilla. 2014. *Cognitive interviewing methodology.* John Wiley & Sons.

[18] Jum Nunnally and Ira H Bernstein. 1994. *Psychometric Theory.* The McGraw-Hill Companies.

[19] Bebras International Challenge on Informatics and Computational Thinking. [n.d.]. *Task Examples.* https://www.bebras.org/?q=exam

[20] Project Quantum. [n.d.]. *Project Quantum – A Collection of Computing Quizzes.* https://diagnosticquestions.com/quantum

[21] Arif Rachmatullah, Bita Akram, Danielle Boulden, Bradford Mott, Kristy Boyer, James Lester, and Eric Wiebe. 2020. Development and validation of the middle grades computer science concept inventory (MG-CSCI) assessment. *Eurasia Journal of Mathematics, Science and Technology Education* 16, 5 (2020). https://doi.org/10.29333/ejmste/116600

[22] Kathryn M. Rich, Carla Strickland, T. Andrew Binkowski, Cheryl Moran, and Diana Franklin. 2017. K-8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (Tacoma, Washington, USA) *(ICER '17).* Association for Computing Machinery, New York, NY, USA, 182–190. https://doi.org/10.1145/3105726.3106166

[23] Marcos Román-González, Juan Carlos Pérez-González, and Carmen Jiménez-Fernández. 2017. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior* 72 (jul 2017), 678–691. https://doi.org/10.1016/j.chb.2016.08.047

[24] Jean Salac, Cathy Thomas, Chloe Butler, Ashley Sanchez, and Diana Franklin. 2020. TIPP&SEE: A Learning Strategy to Guide Students through Use - Modify Scratch Activities. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20).* Association for Computing Machinery, New York, NY, USA, 79–85. https://doi.org/10.1145/3328778.3366821

[25] J. Sarama, D. H. Clements, J. Barrett, D. W. Van Dine, and J. S. McDonel. 2011. Evaluation of a learning trajectory for length in the early years. *ZDM - Mathematics Education* 43 (2011), 667–680.

[26] ScratchJr. [n.d.]. *ScratchJr.* https://www.scratchjr.org/

[27] M. A. Simone. 1995. Reconstructing mathematics pedagogy from a constructivist perspective. *Journal for Research in Mathematics Education* 26, 2 (1995), 114–145.

[28] Allison Elliott Tew and Mark Guzdial. 2010. Developing a Validated Assessment of Fundamental CS1 Concepts. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (Milwaukee, Wisconsin, USA) *(SIGCSE '10).* Association for Computing Machinery, New York, NY, USA, 97–101. https://doi.org/10.1145/1734263.1734297

[29] Jiří Vaníček. 2014. Bebras Informatics Contest: Criteria for Good Tasks Revised. In *Informatics in Schools. Teaching and Learning Perspectives*, Yasemin Gülbahar and Erinç Karataş (Eds.). Springer International Publishing, Cham, 17–28.

[30] David Weintrop and Uri Wilensky. 2015. Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. In *ICER 2015 - Proceedings of the 2015 ACM Conference on International Computing Education Research.* Association for Computing Machinery, Inc, 101–110. https://doi.org/10.1145/2787622.2787721

[31] Eric Wiebe, Bradford W. Mott, Jennifer London, Kristy Elizabeth Boyer, Osman Aksit, and James C. Lester. 2019. Development of a lean computational thinking abilities assessment for middle grades students. In *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education.* Association for Computing Machinery, Inc, 456–461. https://doi.org/10.1145/3287324.3287390

[32] Gordon B Willis. 2004. *Cognitive interviewing: A tool for improving questionnaire design.* sage publications.

[33] Gordon B Willis. 2015. *Analysis of the cognitive interview in questionnaire design.* Oxford University Press.

[34] Cameron Wilson, Chris Stephenson, and Mark Stehlik. 2010. *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age The Association for Computing Machinery The Computer Science Teachers Association The Computer Science Teachers Association, Member of ACM's Education Policy Committee Association for Computing Machinery.* Technical Report. http://www.acm.org/Runningonempty/http://csta.acm.org/Runningonempty/