Computer Science Education

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ncse20

Developing a kindergarten computational thinking assessment using evidence-centered design: the case of algorithmic thinking

Jody Clarke-Midura, Deborah Silvis, Jessica F. Shumway, Victor R. Lee & Joseph S. Kozlowski

To cite this article: Jody Clarke-Midura, Deborah Silvis, Jessica F. Shumway, Victor R. Lee & Joseph S. Kozlowski (2021) Developing a kindergarten computational thinking assessment using evidence-centered design: the case of algorithmic thinking, Computer Science Education, 31:2, 117-140, DOI: 10.1080/08993408.2021.1877988

To link to this article: https://doi.org/10.1080/08993408.2021.1877988

COMPUTER SCIENCE EDUCATION
2021, VOL. 31, NO. 2, 117–140
https://doi.org/10.1080/08993408.2021.1877988

Routledge
Taylor & Francis Group

Developing a kindergarten computational thinking assessment using evidence-centered design: the case of algorithmic thinking

Jody Clarke-Midura[a], Deborah Silvis[a], Jessica F. Shumway[b], Victor R. Lee[c] and Joseph S. Kozlowski[b]

[a]Instructional Technology and Learning Science, Utah State University, Logan, UT, USA; [b]College of Teacher Education and Leadership, Utah State University, Logan, UT, USA; [c]Graduate School of Teacher Education, Stanford University, Stanford, CA, USA

ABSTRACT

Background and Context: There is a need for early childhood assessments of computational thinking (CT). However, there is not consensus on a guiding framework, definition, or set of proxies in which to measure CT. We are addressing this problem by using Evidence Centered Design (ECD) to develop an assessment of kindergarten-aged children's CT.
Objective: To present a design case on the development of the assessment, specifically the algorithmic thinking (AT) tasks and to share validity evidence that emerged.
Method: We focus on the AT sub-component of CT and present the principled assessment design process using ECD.
Findings: Our operationalization of CT includes spatial reasoning as a sub-component. Pilot results showed an acceptable internal consistency reliability for the AT items and critical design decisions that contributed to validity evidence.
Implications: An important contribution of this work is the inclusion of spatial reasoning in our definition of early childhood CT.

ARTICLE HISTORY
Received 30 April 2020
Accepted 7 January 2021

KEYWORDS
Computational thinking; assessment; early childhood; algorithmic thinking; evidence-centered design

Introduction

Early childhood is increasingly seen as a time to introduce and foster computer science skills and computational thinking (CT). While this trend is not without historical precedent (e.g., Papert, 1980), there is currently a movement to produce computationally-themed materials and toys for kindergarten-aged children (Hamilton et al., 2020). In some cases, the design of these materials and toys are informed by research (e.g., Bers et al., 2014). In many other cases, the use and development of such materials and toys are driven by popular intuition. As the whole range of materials are beginning to see uptake in kindergarten and early childhood settings, we see a more pressing need for there to be assessments for this age group that measure CT.

CONTACT Jody Clarke-Midura jody.clarke@usu.edu Instructional Technology and Learning Science, Utah State University, Logan, UT, USA

Supplemental data for this article can be accessed here.

However, for a number of reasons, determining what to assess and how to do so is not as simple as it seems (Pellegrino et al., 2016). First, the field does not currently have an agreed upon definition of CT. Unlike assessments in disciplines like math (Clements et al., 2019), there is not yet consensus on what skills constitute CT. Second, even if an agreed-upon definition of CT was established, we still lack a clear understanding of the knowledge, skills, and abilities (KSAs) involved in young children's CT given some developmental constraints. For example, Kindergarten children are just learning how to read and write, meaning the expression of their CT capabilities do not lend themselves to many standard forms of computer-based or paper-and-pencil elicitation techniques. Third, and relatedly, interest in early childhood CT is so recent we also lack common curriculum practices to observe that could inform assessment design. For these reasons and more, it is easy to come unmoored because so much is unknown and in need of study. In these messy circumstances, it helps to put a stake in something, and for us that was the Evidence-Centered Design process (ECD, Mislevy & Haertel, 2006).

The present article applies ECD to the early stages of the design of an assessment of kindergarteners' CT. Specifically, our goal has been to design an assessment that can be used to assess CT as promoted in currently designed materials and toys for young children who are preliterate, which emphasize controlling and moving an agent through a grid-like space. Aligning our design process with a rigorous set of established steps for setting up our assessment as an argument through conducting domain analysis, domain modeling, and conceptual assessment framing, provided structure for assessment design. This approach supported making valid inferences about children's learning, based on iteratively specifying and testing their CT knowledge, skills, and abilities. ECD was a means of managing the uniquely messy circumstances of assessing young children's CT. This article presents (1) a design case that outlines specific challenges associated with the design, validation, and administration of early childhood CT assessments with special attention focused on fairness (i.e., equity) and (2) an explanation for how we dealt with these challenges in the design of an interview-based assessment of CT for kindergarten students.

In the sections that follow, we first lay out the scope of what is known about CT assessments, highlighting the critical issue of validity and describing the ECD approach to validity by design. Then, we describe our project, called Coding in Kindergarten, contextualizing our assessment design in the overall design process and aims of the project, including a summary of our procedures and sample. Then we turn to the central contribution of the paper and align the first three layers of ECD – domain analysis, domain modeling, and conceptual assessment framework – to our process of developing our Kindergarten CT assessment. Due to space limitations and our interest in illustrating our process and argument in sufficient depth, we focus on algorithmic thinking (AT), one sub-component of the larger CT construct. For AT, we describe (a) concepts and considerations that emerged from our domain analysis, (b) how these were articulated in terms of a design pattern, and (c) instantiated in the student, evidence, and task models in our assessment. Finally, we discuss how developing an early CT assessment using the iterative ECD process required us to engage with related important competency areas, such as spatial reasoning. We suggest some implications of this and other special considerations for designing assessments of kindergarten-level CT that bear on fairness and equity.

Current scope of assessing computational thinking

The computer science education community is growing rapidly and beginning to construct nuanced definitions of CT (e.g., Cutumisu et al., 2019; Grover et al., 2015; Shute et al., 2017; Tang et al., 2020) however, there lacks comprehensive research on how to measure CT. Part of the challenge is that CT has been conceptualized somewhat differently at different age ranges, and therefore different frameworks have been employed to both define and understand CT (e.g., Basu et al., 2020; Bers et al., 2014, 2019; Brennan & Resnick, 2012; Grover & Pea, 2013, 2018; Lye & Koh, 2014; Shute et al., 2017; Snow et al., 2019; Yadav et al., 2014). Partly due to these different conceptions and frameworks adopted to guide research on CT, current assessment tools for CT look very different from one another. Each has been created in a variety of formats, engage students in different tasks types, and have been developed to function in different educational settings. We aim to develop an assessment that can be used across different research projects studying the integration of CT into kindergarten classrooms. Specifically, to be used with young children who are preliterate or emerging readers.

Validity and reliability of computational thinking assessments

Assessment of CT has garnered enough interest and initial efforts from researchers, that we have begun to see systematic literature reviews on the topic (Cutumisu et al., 2019; Tang et al., 2020). After combing through well over 100 combined articles that empirically reported the use of assessments to measure CT, one of the most resounding shared findings was that the reliability and validity of the current CT assessments was in short order. For example, Tang et al. (2020) reported that of the 96 empirical CT assessment studies analyzed, only 45% reported reliability measures and only 18% reported validity evidence.

The role of validity in assessment design

Validity is a central issue in assessment design. Validity in assessments refers to the degree to which evidence and theory support the interpretations of test scores entailed by proposed uses of tests (American Educational Research Association, American Psychological Association & National Council on Measurement in Education [AERA, APA, NCME], 2014; Messick, 1989, 1995). Much like theory development, the process of validation involves accumulating relevant evidence to provide a sound scientific basis for the proposed interpretations of the assessment (American Educational Research Association, American Psychological Association & National Council on Measurement in Education [AERA, APA, NCME], 2014). However, establishing validity is an ongoing process and it may change as interpretations and uses develop or as new evidence is accumulated (American Educational Research Association, American Psychological Association & National Council on Measurement in Education [AERA, APA, NCME], 2014; Kane, 2006; M. T. Kane, 2013). It is important to stress that validity is not a property of the test but of the proposed interpretations and situated inferences of the scores (M. T. Kane, 2013). Thus, not only does validity ensure a test is measuring what it is intended to measure, it ensures inferences based on the outcome of a particular use of an assessment are appropriate.

Mislevy (2007) has argued for the need to make the underlying principles of assessment design more explicit and to structure such designs around assessment arguments. He referred to this as "validity by design" where "as test creators, we are not carrying out validation activities but carrying out design activities structured in such a way that validity evidence emerges" (Mislevy, 2007, p. 467).

Evidence-centered design & validity (by design)

Assessments require a comprehensive framework for making valid inferences about learning. One such framework is Evidence-Centered Design (ECD), which provides rigorous procedures for linking theories of learning and knowing to observations and to interpretation (Mislevy & Haertel, 2006). The ECD framework is a systematic way to design assessments and involves constructing educational assessments in terms of evidentiary arguments (Mislevy & Haertel, 2006). Evidentiary argumentation is borrowed from Toulmin's argument schema (Toulmin, 1958) where the argument is constructed through a series of logically connected propositions that are supported by data and subject to alternative explanations (Mislevy & Riconscente, 2005). The ECD framework provides the structure for developing the argument, but it is up to the assessment designer to provide the content (Mislevy & Riconscente, 2005). This view of assessment as argument is central to validity arguments presented by Kane (2006), M. T. Kane (2013)) and the AERA/NCME Standards (American Educational Research Association, American Psychological Association & National Council on Measurement in Education [AERA, APA, NCME], 2014).

ECD is a multilayer approach comprised five layers: (1) domain analysis, (2) domain modeling, (3), the conceptual assessment framework, (4) assessment implementation, and (5) assessment delivery. In layers 1 and 2, the focus is on the purposes of the assessment, the nature of knowing, and structures for observing and organizing knowledge. In the third layer, assessment designers focus on the student model (what skills are being assessed), the evidence model (how are skills measured), and the task model (situations that elicit the behaviors/evidence). These aspects of the design are interrelated. The ECD method was developed to assess complex performances. For this reason, other groups (e.g., Basu et al., 2020; Snow et al., 2019) have used ECD to develop assessments that measure CT practices. While Snow et al. (2019) documented some of the challenges of using ECD such as time and cost, they have also shown the potential of ECD as a framework for developing an assessment for CT practices. In particular, they have illustrated the importance of design patterns in that they provide a framework that clearly outlines the focal KSAs and guidance on how to develop tasks to measure them. Building off the work of Snow et al. (2019), we used ECD to guide the design of our assessment.

The coding in kindergarten project

The larger National Science Foundation-funded research (Grant no. NSF #1842116) from which this article originates, the Coding in Kindergarten (CiK) project, is investigating how to both integrate computer science into kindergarten mathematics instruction using commercial screen-free coding toys and assess the students' CT. We are using commercial toys for two reasons. First, a number of early childhood education classrooms, including those with which we partner, have a "no screen-time" policy, making apps that have been thoughtfully designed (e.g., Scratch Jr.) ineligible for kindergarten classroom use. Some work has been done to develop high quality academic research-based toys, but teachers and schools can view those as a substantial material investment beyond what they can typically afford (e.g., Bers et al., 2014). That has made easily purchased and easily replaced commercial toys an attractive option for educators as a way to introduce coding in a playful and simple way.

Second, and in line with the project's computer science and mathematics integration, we hypothesize these toys, which all involve moving a robot-like entity through a grid space, will foster spatial reasoning skills (Sarama & Clements, 2009). For example, prior research on young children's spatial assembly skills with blocks suggests an important link between spatial and mathematical skills and that tangible block-building activities facilitate the use of spatial language in cooperative social settings (Verdine et al., 2014). Similarly, these types of coding toy contexts engage students in cooperative spatial orientation and measurement experiences (Shumway et al., 2019; see Figure 1).

The CiK project has three goals: to operationalize what CT looks like in kindergarten classrooms when students interact with coding toys; to develop curricula and resources for teachers to use coding toys in their classrooms; and, to develop an assessment that could be used across toys that measures kindergarteners' CT. Thus, we are not assessing the curriculum per se, but the kinds of CT skills that are afforded by using screen-free coding toys and other similar learning objects (e.g., the Robot Turtles educational board game produced by ThinkFun). By necessity, we have had to design and implement curriculum as we operationalize CT and develop our assessment. Our research involves working in kindergarten classrooms in public schools. While we developed and piloted curriculum around the toys in classrooms (Silvis et al., 2020), the focus of this paper is on the design of our assessments and the validity evidence that has emerged in the early design process. Nonetheless, it is important to acknowledge the concurrent design activities, including the curricula and how enacting it in classrooms informed our definitions of CT, our design patterns, our student models, task models, and evidence models.

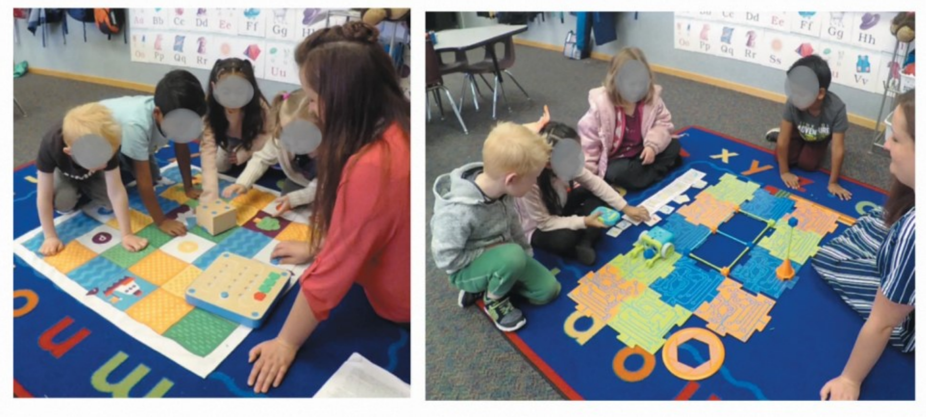


Figure 1. Group of Students and Teacher Interacting with Coding Robots.

Context of the research

We worked with four elementary schools in the Mountain West, United States. All four schools are classified as Title I; three are classified as rural, and those three also received funding to offer full-day kindergarten. Figure 2 presents the timeline of our work with the four schools, depicted as A, B, C, and D. In school A, we worked with three teachers and six classes. In schools B, C, and D we worked with one teacher and class per school.

Curriculum implementations involved a member of the research team, a former elementary or preschool teacher, teaching a 30-min lesson with a coding toy to small groups of 4–5 kindergarten students during their STEAM Centers (Science, Technology, Engineering, Art, and Mathematics activities that students visit in a rotation each week). Each lesson was observed and video recorded by a member of the research team. Assessment implementations involved a member of the research team working one-on-one with a child.

During *Phase 1*, Spring, Year 1, we piloted prototypes of curriculum tasks around two toys and then some assessment items. During *Phase 2*, we conducted two different types of pilots. Our first approach involved School A and was focused on piloting assessment items. The teachers, who had been using coding toys prior to working with us, had access to our curricula and resources and implemented coding lessons on their own. Members of the research team piloted assessment items with students. Our second approach focused on piloting both curriculum and assessments. In school B and C, members of the research team taught a sequence of six lessons around two or three different coding toys. At the end of the curriculum, researchers administered a version of our assessment with each student.

Data collection at School D was not completed at the time of writing this paper. For the purposes of this paper, we focus only on *Phase 2* in Figure 2, the highlighted period between early fall and late winter. We describe the *Phase 2* procedures and sample below. *Phase 3* (not pictured) involves scaling up curriculum implementation through professional development and continuing co-design of lessons using coding toys. While we expect teachers will deliver the lessons, the intended use of the assessment is as a tool for researchers who study early childhood CT. Although it will be available for teachers to use.

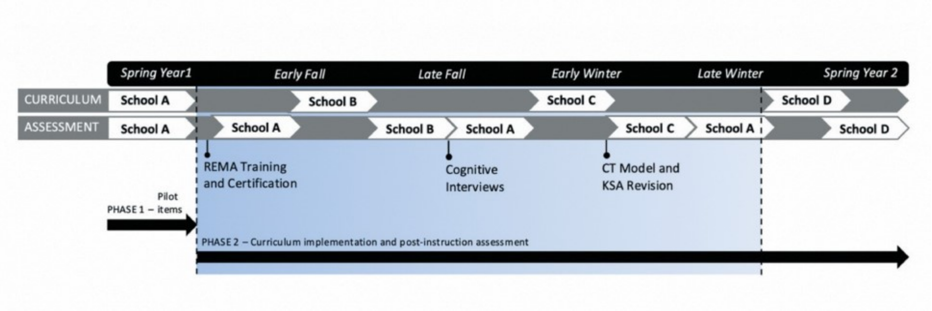


Figure 2. Timeline for Assessment Events Over a One-Year Period.

Sample for this study

Data in this paper focuses on six classrooms within three Title 1 schools (A, B, and C). It is important to note that full day kindergarten is rare in this region. The three schools we worked with implemented full day kindergarten in order to provide opportunities for students demonstrating academic risk upon kindergarten entry. Participants' parents were asked to complete an optional demographic survey, but not all of them did. Eighty-nine students took versions of the assessment during this time period (females = 45, males = 44). Forty-seven identified as White, 14 as Latinx, 1 as Black, 4 as Asian/Pacific Islander, 2 as Native American, and 2 as Other. Ten students reported receiving ESL services, 9 reported receiving special education services, and 22 as receiving free and reduced lunch.

Procedures for the assessment administration

Research team members were trained and certified on the administration of the Research-Based Early Mathematics Assessment, a validated mathematics assessment designed for a similar age group (Clements et al., 2019). The assessors all taught the curriculum in Schools B and C and demonstrated good rapport and management skills with children. They had prior formal classroom teaching experience and advanced degrees in education. On average, we were allotted 20 minutes to assess each child. Standard materials included in the assessment task environment were: a small 3-D printed moveable agent; a series of paper-based grids bound in a flip book that served as the navigation plane for moving the agent; a collection of paper-based arrow code tiles for sequencing and some preset laminated code strips; a program organizer to contain the sequence; administrator pages, with item scripts and prompts; and scoring sheets (see Figure 3). All assessments were video recorded and double scored.

Applying ECD to kindergarten CT assessment

As mentioned above, the ECD model has five layers; for the purposes of this paper, we discuss the first three in relation to our design process. Likewise, our assessment targets

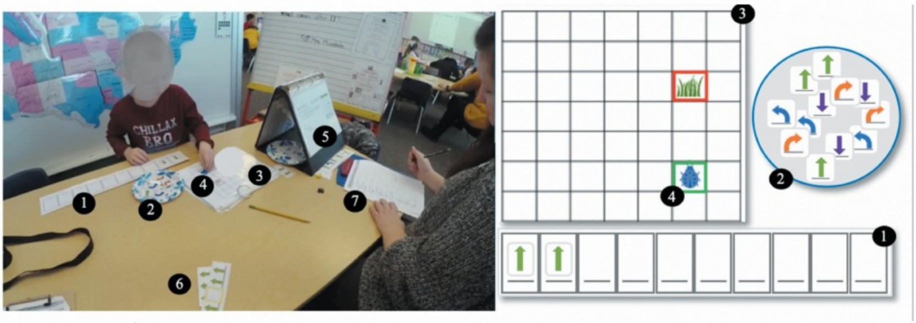


Figure 3. Student taking assessment. Left shows materials used in assessment tasks: (1) program organizer (2) arrow codes (3) grid pages, flip book (4) moveable agent (5) administration pages, with script (6) preset code strips (7) scoring sheets. Right shows child's-eye view of assessment materials.

multiple components of the CT construct, including algorithmic thinking, spatial thinking, debugging, and decomposition. Due to space limitations, we focus this paper on only the algorithmic thinking (AT) sub-component, and we document design changes that took place regarding this crucial component of CT over one year of an assessment design cycle. In what follows, we describe how our assessment and conceptualization of this construct developed and became further refined through ongoing domain analysis, domain modeling, and conceptual assessment re-framing.

Domain analysis

This layer of assessment design is focused on gathering information about the domain being assessed. At the core of this process is documenting "how this knowledge is acquired and used, as well as how competence is defined and how it develops" (Mislevy & Riconscente, 2005, p. 7). The information gathered is set up as arguments in the second layer, domain modeling. In this section, we summarize portions of our domain analysis: CT definitions and ways of measuring CT, AT, and critical aspects of AT assessment items (e.g., spatial thinking).

CT definitions

We started our domain analysis by looking at the published research on CT as well as assessments of CT (e.g., Angeli et al., 2016; Bienkowski et al., 2015; Brennan & Resnick, 2012; Shute et al., 2017; Snow et al., 2017, 2019; Sullivan & Bers, 2016; Weintrop et al., 2016). First, we culled information on definitions and frameworks for CT including the CSTA standards. We categorized the definitions based on context, concepts, and age.

Definitions of CT vary and are associated with the environments and domains in which particular research is situated. For example, studying CT the context of the Scratch programming environment, led Brennan & Resnick to identify concepts like sequences and practices like debugging as relevant for CT. Ehsan and Cardella (2017) framed CT within engineering education and identified CT competencies such as abstraction, algorithm and procedure, debugging/troubleshooting, pattern recognition, and simulation while observing first-grade students complete an engineering design task in an informal learning context. Weintrop et al. (2016) performed a review of CT literature and interviewed experts in mathematics and science in order to develop a taxonomic definition of CT consisting of four categories: (1) data practices (2) modeling and simulation practices (3) computational problem-solving practices and (4) systems thinking practices. When it comes to operationalizing CT- which factors? how many? how do they interact? what sorts of practices do they support?- context appears to matter a great deal.

In the domain of early childhood CS education, a number of researchers emphasize *coding and programming environments* as relevant for CT development, recognizing that other types of CT engagement beyond coding are possible (i.e. unplugged or story-based activities). Based in this emerging context for CT, we aligned our definition with the ways in which early childhood CT has been conceptualized and assessed, as a multi-factor construct (Bers, 2018; Relkin et al., 2020; Shute et al., 2017). Shute et al. (2017) developed a definition for CT as the "conceptual foundation required to solve problems effectively and efficiently" (p. 151). Included with their definition, they analyzed various CT concepts and categorized commonly identified CT skills into six main facets: decomposition, abstraction, algorithm design, debugging, iteration, and generalization. Similarly, Relkin et al. (2020) created the *TechCheck* which was specifically meant to assess young children's "various domains of CT described by Bers (2018) as developmentally appropriate for young children: algorithms, modularity, control structures, representation, hardware/ software, and debugging, with the exception of the design process" (p. 4). Though the operative dimensions may vary, a variety of CT assessments measure sub-constructs of CT and situate their measures in programming contexts.

Computational thinking assessments for young children: unique challenges

Currently, there is a lack of valid and reliable CT assessments for young children. One potential reason is because within the early-childhood literature, there has not been a consensus on a guiding framework, definition, or set of proxies with which to measure CT (e.g., Angeli & Valenides, 2019; Bers et al., 2019; Cittá et al., 2019; Martinez et al., 2015). For example, Angeli and Valenides (2019) measured CT of 50, five- to six-year-old children by individually assessing decomposition, AT, sequencing, and debugging and used these skills and practices to garner a holistic score of students' CT. Cittá et al. (2019) measured CT of 92, six- to ten-year-old students by having them take a paper-pencil test that assessed the " . . . students' ability to write and interpret an algorithm on paper in a closed environment represented by a chessboard" (p. 5). Bers et al. (2019) measured CT of 172, three- to five-year-old children by having them solve a robot-based challenge. The resulting product was assessed by the researcher using a Solve-Its checklist (Strawhacker et al., 2013). This checklist has progressive levels of complex programs, and as students include more or less complexity in their program, they receive more or less points. These three examples highlight how different CT assessments for young children measure different skills and practices of CT. The varying conceptions of what CT is and what skills and practices represent it, makes comparisons of CT learning challenging and demonstrates a need for a CT assessment aimed at generality. Based on converging literature, we defined CT as being comprised of the following facets: (a) algorithmic thinking, (b) decomposition, and (c) debugging. We added an additional skill (d) spatial reasoning, which is not part of most CT definitions but something we knew would be important based on the movement of the toys and the data collected thus far in the project.

A focus on the algorithmic thinking component of CT

There are different approaches and definitions to defining algorithms and programming in the literature. Most of them start out similarly by defining it as a step-by-step process to complete a task (e.g., CSTA CS standards, 1A-AP-08, 2017) and "design(ing) logical and ordered instructions for rendering a solution to a problem" (Shute et al., 2017, p. 12). These steps can be carried out with or without a computer. Algorithmic thinking (AT) can be thought of as the writing or reading of a sequence of steps to solve a problem, either computer-based or in an unplugged environment. Sometimes, this ability to read or write a program is measured directly (e.g., Cittá et al., 2019) and sometimes underlying skills such as sequencing (Bers et al., 2019; Martinez et al., 2015) or action-symbol correspondence (Muñoz-Repiso & Caballero-González, 2019) have been measured with young children. AT necessitates careful examination of items that would allow for a certain AT trajectory to be developed and assessed. This could include foundational skills like sequencing and action-symbol correspondence, but would not regard them as the sole indicators or algorithmic thinking. These foundational skills may be particularly critical to assess in young populations such as kindergarten.

In our review of standards and existing research, we initially defined algorithmic thinking in four parts: give a set of sequential instructions, correctly use the syntax of the coding system, create loops through loop procedures/control structures, and create functions. After three pilots of the AT assessment items (see Figure 2), we modified the aspects of assessment situations that can be varied to shift difficulty or focus, which we call variable features. After adjusting variable features in subsequent assessments, we were better able to observe what we thought kindergarten students could do.

As with the larger construct, the context in which we are operationalizing AT is relevant for measuring it. The context in which we are operationalizing and assessing AT is consistent with computational environments being widely adopted in early childhood settings: tangible coding materials that rely on a grid-based series of movement commands (e.g., Bers, 2018). While CT it is not exclusive to coding and programming, and not all coding and programming environments in early childhood involve grids and directional commands, this format is common enough- and the commercially available coding tools numerous enough- that we believe it is prudent to base early childhood CT assessment in this context. *However, circumscribing our assessment design around this particular context, introduced unexpected variables associated with a grid system and the development of spatial and directional thinking*. Rather than a limitation of our design or conceptualization of the constructs, we see this as a relevant finding for researchers and teachers who are trying to understand how CT operates and how to observe it in early childhood. We therefore analyzed the literature in this domain for its relevance to AT and CT.

Spatial thinking as a variable feature for AT items

Kindergarteners engage their spatial thinking to program toys and agents to move in a given space. Our curriculum tasks and AT assessment items were tethered to spatial thinking, and became an important variable feature of the AT assessment items. Hence, understanding student's knowledge of spatial relationships was particularly pertinent to our domain analysis. Spatial thinking entails understandings of space and objects' positions in space, reasoning with objects or representations in space, and operations on spatial relationships (National Research Council [NRC], 2006; Sarama & Clements, 2009). The K-12 mathematics curriculum relies on spatial thinking (National Research Council [NRC], 2006). The kindergarten geometry standards (Common Core State Standards Initiative [CCSSI], 2010) state, "describe objects in the environment using names of shapes, and describe the relative positions of these objects using terms such as above, below, beside, in front of, behind, and next to" (K.G.A.1). The standard emphasizes students' use of language to describe shapes and positions of shapes in space, but the critical underlying concepts are about developing students' spatial orientation. Spatial orientation is the understanding of different positions in space, and students first develop spatial orientation concepts in relation to their own position in space and later develop external-based reference systems using landmarks outside themselves (Sarama & Clements, 2009). In our programming context, students use their spatial orientation concepts when they move a toy or agent around a grid space. For example, in Figure 3, the student is holding the agent and ready to move the agent two movements forward on the grid to the grass.

Previous research suggests that mental rotations are difficult for young children and the ones who have more advanced mental rotation ability demonstrate higher levels of mathematical and computational thinking (Cheng & Mix, 2014; Cittá et al., 2019; Cuneo, 1985). Further, research also suggests young children progress through a specific spatial thinking learning trajectory (Sarama & Clements, 2009) and shift from certain egocentric to allocentric reference frames while programming (Clarke-Midura et al., 2021). In the Figure 3 example, the student is completing an easier task that involves two forward movements and is oriented in the same direction as the agent. The same task (two forward movements from the starting point to the grass) with the agent facing the student is more challenging and requires the student to consider the toys's frame of reference.

The grid-based system brings important challenges to our assessments, and it is important to consider the ways that students will be able to navigate movements within rows and columns on a grid. Our grids are two-dimensional arrays made up of rows and columns of squares. Student understanding of the arrangement of a grid, specifically the organized structure of the rows and columns of squares, requires the cognitive feat of spatial structuring. Spatial structuring is a form of abstraction that involves the ability to organize and coordinate a set of objects in space (Battista et al., 1998). In our context, kindergarteners are still developing spatial structuring concepts and understanding the organization of rows and columns in a grid is quite challenging until about the age of seven years old (Sarama & Clements, 2009). Hence, while the grid can provide an organizing structure to plan paths of movements for the toy or agent, young children may have difficulty understanding the grid's structure.

Domain modeling of AT

This layer of ECD is focused on diagramming or otherwise systematically planning how designers might create an assessment of the construct of interest. Modeling the domain means bringing the construct into alignment with the population, knowledge, behaviors, work products, and tasks (Oliveri et al., 2019). Along with operationalizing relevant attributes of the knowledge and population of interest that emerged from domain analysis, a major focus of articulating the domain model should be the evidence which supports the argument for the assessment's validity.

Design pattern for early childhood algorithmic thinking

As an approach to design modeling, design patterns (DPs) organize a set of decisions for developing assessments (Mislevy & Haertel, 2006). They consist of a series of defined attributes, such as focal knowledge, skills, and abilities (KSAs) and characteristic features of items, that structure an assessment. Design patterns rationalize an assessment's argument by describing how the construct of interest will be operationalized, elicited, and observed (Mislevy & Haertel, 2006). DPs also show how attributes of the design are interconnected and are a means of reconciling dependencies between nested design considerations (Oliveri et al., 2019). Domain modeling through the development of a design pattern took the form of the following attributes (see online supplemental material A for more detail on the DP).

Population and special considerations

In order to ensure fairness in assessments, designers must account for special considerations in a given target population (Oliveri et al., 2019). Examples of special considerations might be examinees' linguistic and cultural diversity, their prior knowledge, or other construct-relevant factors, like age or gender, that bear on making valid claims about test takers' performance. We were careful to create tasks that were not biased toward a particular socioeconomic status, culture, or prior background experiences. For example, we avoided holiday themes and activities that bias a particular group, and instead we used a bug theme among the physical materials and test items.

Our intended population is primarily preliterate, which demands certain considerations in assessment creation, such as interview-based delivery format, verbal instructions or observational assessment. Other issues that arise with young children is that they may or may not be familiar with typical symbols that seem customary to researchers, such as the rotate arrow (see Figure 3). In fact, these arrows may provoke different interpretations for children based on their previous, mundane experiences with the symbols (Silvis et al., 2020). For example, they may think the left rotate arrow means to make a rotate and forward movement, similar to what would be done in a car or walking around a corner, rather than make a stationary rotation.

We also subjected our measure to a time constraint of twenty minutes, knowing that there are limits on young children's attention and engagement that could influence their behaviors. In addition, screen-free and toy-free tasks were required, not only because teachers preferred these approaches, but also because we recognized that children have had variable exposure to programming contexts and different opportunities to learn to use coding toys. We considered children's' language development, both in terms of age (young children learning language) and linguistic background (English Language Learners), in the design of items and administration. Examples include using manipulatives (i.e., moving an agent on the grid and moving tangible arrows to a program organizer) and gestures (e.g., pointing to codes or a space on the grid), using prompts to support their responses, modifying questions for better comprehension, and eliciting nonverbal responses (e.g., yes/no questions, statements such as point to).

Assessment rationale

Building from definitions of CT that came out of our domain analysis, our initial working definition of AT contained four parts: give a set of sequential instructions; correctly use the syntax of the coding system; create loops through loop procedures/control structures; and create functions. Our original set of KSAs for AT were modeled after this definition. During early winter (see Figure 2), we found that this definition, while grounded in the literature on early CT, was inadequate for describing the kind of thinking kindergarteners exhibited when learning to code. For example, although our assessment is toy-free, the language of directional arrows is an inherent feature of most coding toys and programs geared towards children (Clarke-Midura et al., 2019). Thus, algorithms frequently take the form of an ordered sequence of directional arrow commands, the execution of which instructs an agent to navigate on a plane such as a map, grid, or number line.

Given the prevalence of this constraint in early childhood coding environments- and departing from broader conceptualizations of AT in the literature, we further refined the definition. For the purposes of our assessment, we define AT as developing and using logical and ordered sequences of instructions [see online supplemental material B]. By defining AT in this way, we retained sequencing at the center of the AT sub-component of the CT construct and ensured that assessment content would be context-relevant for the target population.

Focal knowledge, skills, and abilities (KSAs)

After piloting our assessment three times in the fall (two times with School A, one time with School B, see Figure 1), and after implementing the curriculum and observing what occurred under non-testing conditions, we modified our KSAs. Initially, the KSAs for AT were keyed to our four-part working definition, and the first two KSAs were that students could specify a short sequence of instructions (< 3) and a moderate length sequence of instructions (> 4). We subsequently unpacked and expanded our definition of AT into finer skills that were more reflective of the capabilities that students were demonstrating. One way we did this was to shift "length of sequence" from a variable skill to a variable task feature that could be modified to differentiate levels of AT. Another way was to more fully incorporate the requisite knowledge of one-to-one, movement-action correspondence into the KSAs for AT as a form of prior knowledge we were not (yet) measuring. We also articulated certain meta-computational concepts; test-taker knowledge of terms like "program", "code", and "instruction" had been implicit in student performance, but we now treated these as prior knowledge consequential for modeling and assessing AT.

Although our curriculum implementations and pilot assessments in late fall had begun to indicate that the ability to create functions and loops using control functions was beyond the skill level of kindergarteners we worked with, we preserved these as ceiling KSAs in our model and retained the corresponding tasks as "upper anchors" (Penuel et al., 2014, p. 79) in the assessment. We made this decision because we had only explored loops in classrooms with two of the toys to date. Admittedly, these two toys did not present loops and functions in a way that we found to be intuitive. However, Kibo, which we have not implemented in classrooms, presents loops as "repeat" blocks whose sequence begins with "Repeat begin" and ends with "repeat end". While research on Kibo has found that loops can be difficult for preschool-aged children (e.g., Elkin et al., 2018), we have not been able to conduct research on programming loops with Kibo in kindergarten classrooms. Further, Bers and colleagues, are exploring coding trajectories that they say start with sequences (order) and end with more complex patterns of sequences such as loops (Bers, 2019). Given that there is still much for the field to learn about coding in early childhood, we also plan to continue to explore how children engage with these concepts. Finally, in consideration of the fact that kindergarteners were in the process of learning to read and write, we realized that our working definition of AT foregrounded writing (specifying) code, but that reading and enacting programs were equally important. We therefore reorganized assessment items to align with these three activities in which students demonstrated the AT knowledge we were targeting. We provide a blueprint for how this was instantiated in our task design as part of the conceptual assessment framework (CAF).

Additional KSAs: the role of spatial reasoning

Another key insight from pilot assessments and curriculum implementations was that accounting for directional movements of an agent was central to warranting the assessment argument. In other words, alternative explanations of performance that hinged on students' directional and spatial knowledge emerged as a potential confound for our focal KSA. Teasing out the complex relationship between spatial and directional reasoning and disambiguating it from AT became a central design challenge.

One aim of assessment design patterning is to account for alternative explanations for performance. Design patterns are useful in this regard, because they provide a means of identifying additional KSAs implicated in the construct and determining whether these are collateral of, prerequisite for, or irrelevant to the construct (Oliveri et al., 2019). Whereas with older children, eliminating such sources of construct-irrelevant variance may have been an easier matter, young children's AT was entangled with their incomplete spatial knowledge of left and right, their varying interpretations of the semantics of arrow symbols, and their developing understanding of symbol-action correspondence.

We had to determine the relevance of spatial reasoning to the construct of interest. Was spatial reasoning an additional set of KSAs that needed explication in the design pattern? Or did we need to somehow incorporate it into our focal KSAs? Design patterns may identify additional "knowledge, skills, and abilities that are not part of the construct but depending on design choices and task features, may be required to perform the task" (Oliveri et al., 2019, p. 282). We decided to treat spatial reasoning in this way, as meaningfully relevant for solving the particular problems that arise in early childhood CT educational contexts. Our assessment includes a section of spatial reasoning items preceding the AT section, that, like the other components of CT has undergone revisions according to a series of design features. We discuss these in more detail following the kinds of work products and potential observations that we posited can provide evidence to support claims about student knowledge.

Potential work products and observations

Based on our definition of AT, we mapped KSAs onto student's potential work products and observable behaviors. We documented types of responses that could serve as indicators of their KSAs and evidence of latent proficiencies. For example, we expected that they could produce a program of three or more codes as well as programs with a turn. These programs, comprised of sequences of arrow cards arranged along a specially designed "program organizer," would serve as physical evidence of students' AT (see Figure 2).

Additional forms of evidence might be students' enactments of programs as they followed a pre-set program that we called a "code strip" (Figure 2). We knew from curriculum implementations and pilot assessments that enacting and simulating ready-made instructions was a primary mode through which students demonstrated their early knowledge of sequencing. We recognized that students learning to read English were likewise learning to read a directional arrow-based programming language and that to specify a sequence, they first needed to decode the symbol system. Enactments and think-aloud sessions with students provided means of accessing their ability to read and follow sequences. Furthermore, our focus on math integration suggested that counting the number of codes required to solve a problem served as another potential source of evidence of student's AT.

Variable features

We prototyped materials and piloted tasks that could elicit these potential work products and observable behaviors. Variable features are features of task sets that can be varied to change its level of difficulty, context (to facilitate new task generation), or granularity (Oliveri et al., 2019). Through iterative cycles of item redesign, piloting, and data analysis, we modified features of tasks to make them more or less difficult and to unpack AT. We identified variable features for the context of task sets: using a navigation plane such as a map, grid, or number line that the agent could move across; having a path that the agent takes with start and end points that correspond to the start and end of algorithms; manoeuvring an agent within the plane; positioning codes in a linear sequence; and embedding each task in a simple scenario representing a problem for the agent to solve. Within this task set, we identified that three key program features, when varied, controlled the difficulty of tasks. The first was starting orientation of the program. When the agent shared a starting reference frame with the student, the task was easier (Shown later in Figure 5, Tasks 2 & 3). Conversely, a starting orientation that was rotated ninety degrees or was facing the child (rotated 180 degrees) increased the level of difficulty (Shown later in Figure 5, Task 1). The second program feature we varied was the presence of turns in the sequence. Tasks that asked children to generate or to follow a set of instructions were more challenging if they included turns in the sequence. Furthermore, increasing numbers of turns increased the difficulty level. The third program feature that we used to change difficulty level was the distance traveled; longer sequences presented more challenging problems.

The conceptual assessment framework

The third layer in the ECD approach is the conceptual assessment framework (CAF). This layer is a blueprint of the assessment where decisions are made in regards to the statistical models, materials and delivery processes that will inform students' work products, and the scoring (Mislevy & Riconscente, 2005).

The CAF is comprised of three models: the student model, the evidence model, and the task model. These three models are fleshed out with the information in the design pattern and work together to provide technical details of the assessment. The CAF specifies how items can be varied to create families of items and how we update claims about students' proficiencies based on their work products or performances. The CAF also "serves as another place for examining the impact the assessment may have on test takers from different populations" and to "minimize inadvertent construct-irrelevant demands" (Oliveri et al., 2019, p. 291)

Student model: what are we measuring?

The Student Model answers the question of what the assessment is measuring. It is the latent proficiency variables in the statistical model used to accumulate evidence about the student. The Student Model variables are the link between students' performances on tasks and the claims we wish to make about their proficiencies (Mislevy & Riconscente, 2005). In the present study, the student model for AT is a simple model with one student variable. In the model, we will accumulate a number right or total score to characterize students' overall proficiency in AT.

The evidence model: how do we measure it?

The Evidence Model provides the technical details on how the assessment measures the Student Model. In the Evidence Model we update the Student Model based on observations of what children say or do. The Evidence Model has two components: evaluation and measurement. The evaluation component is linked to the Student Model and puts a value on (evaluates) the observations of what students say and do. This can be done in the form of a rubric, answer key, etc. The measurement component compiles the data from evidence component across the tasks. While "each piece of data directly characterizes some aspect of a particular performance, it also conveys some information about the targeted claim regarding what the student knows or can do' (Mislevy & Riconscente, 2005, p. 19). In the present study, observations of students' actions on each task (item) get scored as correct or incorrect based on our answer key.

The task model: in what situation(s) do we measure it?

The Task Model describes the kinds of situations that will elicit the observations of students' proficiency. The Task Model is linked to the work products, characteristic features, and variable features of the tasks described in the design patterns. The claims we want to make about students guide and shape the design decisions around the task features. Such decisions include the form of the work product, what materials are necessary and directives. Many important decisions are made at this layer that affect the inferences that will be made from the test. For example, it is important there are: a sufficient number of tasks to provide information about the construct, sufficient opportunities for students to show the broad range of KSAs necessary to assess the targeted construct, and opportunities for students to demonstrate the focal KSAs in ways that suit the intended context of use (Oliveri et al., 2019).

These three models are connected and linked in an assembly model. The assembly model specifies how the Student Model, Evidence Model, and Task Model work together to generate sufficient evidence to form a valid assessment (Mislevy & Riconscente, 2005). Our plan to is use Item Response Theory (IRT), in particular, a Rasch model to fit the data. The Rasch model analyses will allow us to ensure the unidimensionality of AT and estimate the relative location of items and persons on a single scale (De Ayala, 2013). To date, we have not piloted our assessment with a large enough sample to conduct IRT on our AT data. However, we collected valid evidence on the internal structure of AT with 59 students in School A in late winter. Cohen's kappa was computed to test the agreement between two coders (i.e, the inter-rater reliability). Results showed that there was an almost perfect agreement between the two coders, Cohen's kappa = .99, p < .001. We conducted the Kuder-Richardson Formula 20 (KR20), an index of the internal consistency reliability for binary measurements, on our pilot measure of AT that contained 11 items. Our KR20 value was .82, which is in the acceptable range. Item analysis for this sample indicated a range of item difficulty from .13 to .83. Our goal is to include a range of tasks of varying complexity to discriminate students with different ability levels.

Developing the task models: three examples

Figure 4 presents our Student Model and example observations and task features. Below we present three different task models for our claim that students can specify a program using a sequence of codes [Figure 5]. In all of our items, we provide students with a grid that includes a green square indicating starting position and a red square indicating ending position (the green arrow indicates starting orientation for the assessor, but is not depicted on student grids). All items in our assessment involve an assessor interacting

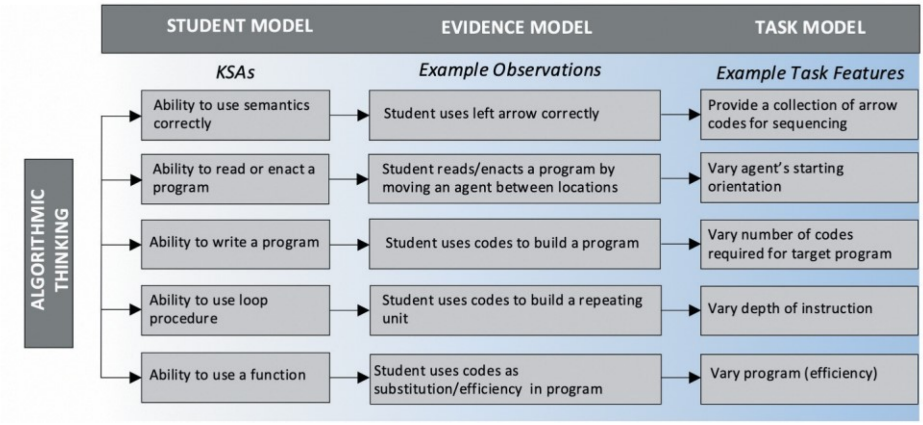
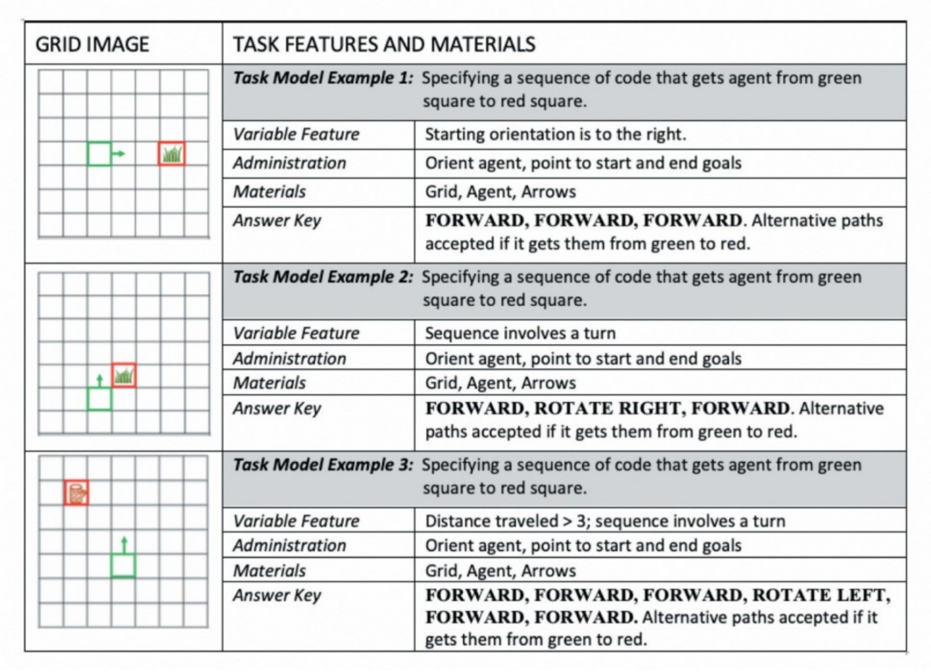

STUDENT MODEL
EVIDENCE MODEL
TASK MODEL
ALGORITHMIC THINKING
KSAs
Example Observations
Example Task Features
Ability to use semantics correctly
Student uses left arrow correctly
Provide a collection of arrow codes for sequencing
Ability to read or enact a program
Student reads/enacts a program by moving an agent between locations
Vary agent's starting orientation
Ability to write a program
Student uses codes to build a program
Vary number of codes required for target program
Ability to use loop procedure
Student uses codes to build a repeating unit
Vary depth of instruction
Ability to use a function
Student uses codes as substitution/efficiency in program
Vary program (efficiency)

Figure 4. Student, Evidence, and Task Models for AT.



GRID IMAGE	TASK FEATURES AND MATERIALS	
	Task Model Example 1:	Specifying a sequence of code that gets agent from green square to red square.
	Variable Feature	Starting orientation is to the right.
	Administration	Orient agent, point to start and end goals
	Materials	Grid, Agent, Arrows
	Answer Key	**FORWARD, FORWARD, FORWARD**. Alternative paths accepted if it gets them from green to red.
	Task Model Example 2:	Specifying a sequence of code that gets agent from green square to red square.
	Variable Feature	Sequence involves a turn
	Administration	Orient agent, point to start and end goals
	Materials	Grid, Agent, Arrows
	Answer Key	**FORWARD, ROTATE RIGHT, FORWARD**. Alternative paths accepted if it gets them from green to red.
	Task Model Example 3:	Specifying a sequence of code that gets agent from green square to red square.
	Variable Feature	Distance traveled > 3; sequence involves a turn
	Administration	Orient agent, point to start and end goals
	Materials	Grid, Agent, Arrows
	Answer Key	**FORWARD, FORWARD, FORWARD, ROTATE LEFT, FORWARD, FORWARD.** Alternative paths accepted if it gets them from green to red.

Figure 5. Task Model Examples.

with a student by reading a question and emphasizing materials or gesturing to the grid. In the following three examples, we accept alternative correct sequences; however, for the purposes of this paper, we provide only the most common (most efficient) response option. Please see online supplemental material D for an Assessment Administration example.

Task Model Example 1 varies the starting orientation for a short distance traveled. The task features for this model involve a starting orientation where the agent is facing towards the right from a students' perspective. For this particular task, the most efficient correct answer is three forward arrows (i.e. FORWARD, FORWARD, FORWARD). Task Model Example 2 varies the presence of a turn. For this particular item, the answer we are looking for is FORWARD, ROTATE RIGHT, FORWARD. Task Model Example 3 varies two features: traveling a distance greater than 3 squares and making a turn. For this particular item, the answer we are looking for is FORWARD, FORWARD, FORWARD, ROTATE LEFT, FORWARD, FORWARD.

Evidence to support arguments of fairness and validity

At the time of this writing, we are in what Kane (2010) has referred to as the *developmental stage* of validation. He writes:

> as an assessment program is being developed, the developers are expected to produce materials and procedures that support the proposed interpretations and uses and to make a case for the validity of the proposed interpretations and uses, and it is appropriate to talk about their efforts 'to validate' the claims being made. (Kane, 2010, p. 4)

While accumulating the relevant evidence to provide a sound scientific basis for the proposed interpretations of our assessment, we constantly ask *what is being claimed* and, given the evidence we have accumulated, *are the claims warranted for all students*? Using ECD provided some structure for us to map out evidence of how students' performances support the inferences that are made regarding students CT competency.

During the task design process, we made several decisions focused on assessing and identifying fairness (Oliveri et al., 2019). For example, we carefully selected the scenarios and vocabulary so that the context and content is accessible to all kindergarten students. We tried to provide opportunities for students to demonstrate the focal KSAs in ways that suit the intended context of use (i.e., kindergarten classrooms). We also considered possible alternative explanations that might account for the observed performances and scores on the assessment.

We collected validity evidence based on student responses through pilots and cognitive interviews with kindergarten students in public school classrooms, the population for which our assessment is intended. The cognitive interviews allowed us to identify any hidden assumptions or alternative plausible interpretations of the test scores. These data also allowed us to design counter arguments to our claims that the assessment is measuring what it is intended to measure. We gathered evidence of construct underrepresentation and expanded our definition of CT to account for spatial reasoning. We expanded our definitions of AT to account for the reading, writing, and enacting of programs. We also built counter arguments around construct irrelevant variance. In doing so, we modified our vocabulary and the language used in our tasks.

These efforts are guiding us towards our larger evaluation of our assessment where we will move to what Kane calls the *appraisal stage* (Kane, 2010). In this stage, we will calibrate our assessment and gather sufficient data to conduct a "critical evaluation of the extent to which the proposed interpretations and uses are plausible and appropriate" (Kane, 2010, p. 4).

Discussion

A central contention motivating this article is that there is a growing need to develop valid assessments of CT for kindergarten. At the same time, it is a challenging undertaking because so much has yet to be specified, both in terms of the types of instruction that are emerging and in the constructs that we hope to eventually measure. The goal of this paper has been to present a case of our efforts to use ECD to develop, test, and refine assessment items for the construct of AT with kindergartners. Design cases are useful for others interested in assessment as it can provide some procedural guidance for how to use ECD for the design of assessments in computer science education. At the same time, cases are useful for surfacing domain-specific tensions and challenges. It is to three of those challenges we now turn our discussion.

Challenge 1: assessing a nebulous construct

Typical assessment design processes presume that some image of how a target construct is instantiated can serve as a blueprint for the assessment development process. However, that image is lacking for early childhood CT. Because research does not yet converge on precisely what to assess and how to assess CT in early childhood, we started our project by operationalizing CT as best as we could give existing literature and standards documents. Through the iterative process of designing and piloting assessment items with kindergartners who have participated in some introductory toy-based CT instruction, we have necessarily adjusted both our items and our understanding of the underlying AT construct. The KSAs we articulated changed such that our initial targets became upward boundaries (e.g., looping) and more specific KSAs that were developmentally appropriate were articulated and added (e.g., spatial reasoning). Moreover, we expanded our KSAs to be inclusive of reading and enacting existing algorithms rather than focus exclusively on the production of algorithms. Because we iterated, we were able to begin to produce what we believe to be a more accurate image of what kindergartner's AT looks like when instantiated in these sorts of grid-based navigation tasks.

Challenge 2: expanding our definitions of CT

Many of the newly emerging tools and educational toys that are being used for kindergarten CT instruction involve navigating an agent through two-dimensional grid space (Clarke-Midura et al., 2019). Taking into account the current lack of consensus in the field as to what fully constitutes AT, and by extension, CT, it has still been an important realization for us that spatial reasoning plays a major role in young children's CT. Thus, for our assessment development process, we have found it appropriate to expand the scope of CT to include spatial reasoning. For kindergarteners, this involves both working with different spatial orientations – for instance, understanding right and left when an agent is facing the same direction as the child and when it is facing a different one – and spatial reasoning – such as when a student is trying to move an agent through the 2D grid space and needs to differentiate between rotations and translations. We came to this realization in our early pilot tests of our CT assessment items and began seeing systematic errors that students were making. By creating and testing new, related items, we have been able to confirm that children's spatial thinking plays an important role in these kinds of common CT tasks.

Challenge 3: assessing preliterate children

Given that the target population we are trying to assess is preliterate, our assessment protocol needed to be in the form of one-to-one oral administration. This puts it in the same style as other established assessments for early childhood (e.g., REMA, Clements et al., 2019) in terms of use of manipulatives, interview assessment, and prompts to aid task comprehension. There are advantages and drawbacks of this approach, with implications for scaling up administration. Although the assessment is intended as a research tool, teachers who may want to administer items would require brief training and time for one-on-one interviewing of their students (similar to how most assessments administered in kindergarten classrooms). We also recognize that children's developing oral and verbal abilities may influence how they interpret and respond to items, and we have taken steps such as cognitive interviewing a small subset of children to maximize clarity of scripts and prompts. Despite these constraints, we believe interview style assessment is most appropriate for Kindergarten children, because they are preliterate.

While we tried to make the assessment items accessible to young children, we did discover that their different levels of syntactic and semantic knowledge could affect performance as well. Arrows, which are commonly used as individual code instructions for this age group, are not necessarily transparent as a representational convention. Moreover, the importance of sequencing codes in a consistent manner, such as individually from left to right, and with a one-to-one mapping of a code onto a single increment of movement, cannot be taken for granted. To better understand students' AT, we revised and included items that ask children to read and enact existing programs rather than only write a program (specify a sequence of codes). By using items that provide a window into students' AT in terms of reading, enacting, and specifying a sequence of codes, we were better able to see what kindergarten students were able to do and which skills they were still developing.

Conclusion

Assessment design is influenced by its purpose, the context in which it will be used, and practical constraints such as resources and time (Pellegrino et al., 2016). In this paper, we presented how we are using ECD to develop an assessment of kindergarten-aged children's CT. In particular, we focused on our early design processes and decisions around operationalizing AT for kindergarten-aged students. We shared some of the realizations that we have made about AT for this population, such as the role that spatial reasoning plays in how AT is currently instantiated with newly developed coding toys and learning materials. While we are designing our assessment to be usable in conjunction with a broad set of instructional resources, we believe that our operationalizations are useful for others who may choose to develop their own assessment instruments for comparable populations. Up until now, there has not been a consensus on a guiding framework, definition, or set of proxies in which to measure CT in early childhood (e.g., Angeli &

Valenides, 2019; Bers et al., 2019; Cittá et al., 2019; Martinez et al., 2015; Snow et al., 2019) nor has there been consistent inclusion of spatial reasoning in definitions of CT.

The process we have described for AT item development is comparable to what we have been undertaking with other facets of CT, such as debugging and decomposition (see online supplemental material C). Our next steps are to finish refinements on our instrument, perform trials with more students, and apply IRT in order so to generate evidence that the assessment can effectively fill the niche it is designed to fill. It is our hope that as this work progresses, and other groups also pursue important work with this age group, that we will soon be in a stronger position for both characterizing and measuring the CT of young children.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Science Foundation grant #1842116. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation or Utah State University.

Notes on contributors

Jody Clarke-Midura is an Associate Professor of Instructional Technology and Learning Sciences at Utah State University. Her research investigates learning and assessment in STEM+C education, in both formal and informal settings. Her current research projects focus on K-5 computer science (CS) education and broadening participation in CS.

Deborah Silvis is a Postdoctoral Researcher for the Coding in Kindergarten project in Instructional Technology and Learning Sciences at Utah State University. Her research focuses on young children's engagement with technology as seen through sociotechnical and socioecological perspectives on learning designs.

Jessica F. Shumway is an Assistant Professor of Mathematics Education at Utah State University. She investigates instructional practices and learning technologies that foster mathematics learning in preschool and elementary school classrooms.

Victor R. Lee is an Associate Professor at Stanford University’s Graduate School of Education. His work addresses multiple facets of STEM education, both in formal and informal settings. Current major interest areas include data science education, maker education, and elementary computer science education.

Joseph S. Kozlowski is a doctoral student at Utah State University where he works on the Coding in Kindergarten (CiK) research project. His research interests include mathematics education, mathematical creativity, and the interconnectedness of computational and mathematical thinking.

ORCID

Jody Clarke-Midura http://orcid.org/0000-0001-5434-0324
Deborah Silvis http://orcid.org/0000-0001-5139-9048
Jessica F. Shumway http://orcid.org/0000-0001-7655-565X

Victor R. Lee http://orcid.org/0000-0001-6434-7589 Joseph S. Kozlowski http://orcid.org/0000-0003-4163-2955

References

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education [AERA, APA, NCME] (2014). *Standards for educational and psychological testing*. APA

Angeli, C., & Valanides, N. (2019). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior, 105*, 1–13. https://doi.org/10.1016/j.chb.2019.03.018

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, *19*(3), 47–57. https://www.jstor.org/stable/jeductechsoci.19.3.47

Basu, S., Rutstein, D., Xu, Y., & Shear, L. (2020, February). A principled approach to designing a computational thinking practices assessment for early grades. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (912–918).

Battista, M. T., Clements, D. H., Arnoff, J., Battista, K., & Van Auken Borrow, C. (1998). Students' spatial structuring of 2D arrays of squares. *Journal for Research in Mathematics Education*, *29*(5), 503–532. https://doi.org/10.5951/jresematheduc.29.5.0503

Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge. https://doi.org/10.4324/9781315398945

Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, *6*(4), 499–528. https://doi.org/10.1007/s40692-019-00147-3

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Bers, M. U., González-González, C., & Armas-Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, *138*, 130–145. https://doi.org/10.1016/j.compedu.2019.04.013

Bienkowski, M., Snow, E., Rutstein, D. W., & Grover, S. (2015). *Assessment design patterns for computational thinking practices in secondary computer science: A first look (SRI technical report)*. SRI International. http://pact.sri.com/resources.html

Brennan, K., & Resnick, M. (2012). Using artifact-based interviews to study the development of computational thinking in interactive media design. *Paper presented at annual American Educational Research Association meeting*, Vancouver, BC, Canada.

Brennan, K., & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking*. In Proceedings of the 2012 annual meeting of the American educational research association (Vol. 1, p. 25). Vancouver, Canada.

Cheng, Y. L., & Mix, K. S. (2014). Spatial training improves children's mathematics ability. *Journal of Cognition and Development*, *15*(1), 2–11. https://doi.org/10.1080/15248372.2012.725186 .

Cittá, G., Gentile, M., Allegra, M., Arrigo, M., Contri, D., Ottaviano, S., Reale, F., & Sciortino, M. (2019, November). The effects of mental rotation on computational thinking. *Computers & Education*, *141* Article number 103613. https://doi.org/10.1016/j.compedu.2019.103613

Clarke-Midura, J., Kozlowski, J., Shumway, J.F., and Lee, V.R. (2021). How young children engage in and shift between reference frames when playing with coding toys. *InternationalJournal of Child-Computer Interaction*. https://doi.org/10.1016/j.ijcci.2021.100250

Clarke-Midura, J., Lee, V. R., Shumway, J. F., & Hamilton, M. M. (2019). The building blocks of coding: A comparison of early childhood coding toys. *Information and Learning Sciences*, *120*(7/8), 505–518. https://doi.org/10.1108/ILS-06-2019-0059

Clements, D. H., Sarama, J., Wolfe, C. B., & Day-Hess, C. A. (2019). *REMA—research-based early mathematics assessment*. Kennedy Institute, University of Denver.

Common Core State Standards Initiative [CCSSI]. (2010). *Common core state standards for mathematics*. http://www.corestandards.org/Math/

Computers Science Teachers Association. (2017) . *K-12 computer science standards*. Creative Commons Attributions.

Cuneo, D. O. (1985). Young children and turtle graphics programming: Understanding turtle commands. *Paper presented at the Biennial Meeting of the Society for Research in Child Development*. Toronto, Ontario, Canada, April 25.

Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, *28*(6), 651–676. https://doi.org/10.1007/s10956-019-09799-3

De Ayala, R. J. (2013). *The theory and practice of item response theory*. Guilford Publications.

Ehsan, H., & Cardella, M. E. (2017). Capturing the computational thinking of families with young children. *Paper presented at the 124th ASEE Annual Conference & Exposition*. Columbus, OH.

Elkin, M., Sullivan, A., & Bers, M. U. (2018). Books, butterflies, and 'bots: Integrating engineering and robotics into early childhood curricula. In *Early engineering learning* (pp. 225–248). Springer, Singapore.

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. In S. Sentence, E. Barendsen, & C. Schulte (Eds.), *Computer science education: Perspective on teaching and learning in school* (pp. 19–38). Bloomsbury Academic.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, *25*(2), 199–237. https://doi.org/10.1080/08993408.2015.1033142

Hamilton, M., Clarke-Midura, J., Shumway, J. F., & Lee, V. (2020). An emerging technology report on computational toys in early childhood. *Technology, Knowledge and Learning*, *25*(1), 213–224. https://doi.org/10.1007/s10758-019-09423-8

Kane, M. (2006). Validation. In R. Brennan (Ed.), *Educational measurement* (4th ed., pp. 17–64). Greenwood Publishing.

Kane, M. (2010). Validity and fairness. *Language Testing*, *27*(2), 177–182. https://doi.org/10.1177/0265532209349467

Kane, M. T. (2013). Validating the interpretations and uses of test scores. *Journal of Educational Measurement*, *50*(1), 1–73. https://doi.org/10.1111/jedm.12000

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12. *Computers in Human Behavior*, *41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

Martinez, C., Gomez, M. J., & Benotti, L. (2015). A comparison of preschool and elementary school children learning computer science concepts through a multilanguage robot programming platform. *In Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education,* (159–164), Vilnius, Lithuania.

Messick, S. (1989). Validity. In R. L. Linn (Ed.), *Educational measurement* (3rd ed., pp. 13–104). American Council on education and Macmillan.

Messick, S. (1995). Standards of validity and the validity of standards in performance asessment. *Educational Measurement: Issues and Practice*, *14*(4), 5–8.

Mislevy, R. J. (2007). Validity by design. *Educational Researcher*, *36*(8), 463–469. https://doi.org/10.3102/0013189X07311660

Mislevy, R. J., & Haertel, G. D. (2006). Implications of evidence-centered design for educational testing. *Educational Measurement: Issues and Practice*, *25*(4), 6–20. https://doi.org/10.1111/j.1745-3992.2006.00075.x

Mislevy, R. J., & Riconscente, M. M. (2005). *Evidence-centered assessment design: Layers, structures, and terminology* (PADI Technical Report 9). SRI International. https://padi.sri.com/downloads/TR9_ECD.pdf

Muñoz-Repiso, A. G. V., & Caballero-González, Y.-A. (2019). Robotics to develop computational thinking in early childhood education. *Comunicar*, *27*(59), 63–72. https://doi.org/10.3916/C59-2019-06

National Research Council [NRC]. (2006) . *Learning to think spatially: GIS as a support system in the K–12 curriculum*. The National Academies Press.

Oliveri, M. E., Lawless, R., & Mislevy, R. S. (2019). Using evidence-centered design to support the development of culturally and linguistically sensitive collaborative problem-solving assessments. *International Journal of Testing*, *19*(3), 270–300. https://doi.org/10.1080/15305058.2018.1543308

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books Inc.

Pellegrino, J. W., DiBello, L. V., & Goldman, S. R. (2016). A framework for conceptualizing and evaluating the validity of instructionally relevant assessments. *Educational Psychologist*, *51*(1), 59–81. https://doi.org/10.1080/00461520.2016.1145550

Penuel, W. R., Confrey, J., Maloney, A., & Rupp, A. A. (2014). Design decisions in developing learning trajectories-based assessments in mathematics: A case study. *Journal of the Learning Sciences*, *23*(1), 47–95. https://doi.org/10.1080/10508406.2013.866118

Relkin, R., Ruiter, L. D., & Bers, M. U. (2020). *TechCheck*: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology, 29*, 482–498. https://doi.org/10.1007/s10856-020-09831-x .

Sarama, J., & Clements, D. H. (2009). *Early childhood mathematics education research: Learning trajectories for young children*. Routledge.

Shumway, J. F., Clarke-Midura, J., Lee, V. R., Hamilton, M. M., & Baczuk, C. (2019). Coding toys in kindergarten. *Teaching Children Mathematics*, *25*(5), 314–317. https://doi.org/10.5951/teacchilmath.25.5.0314

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Silvis, D., Lee, V., Clarke-Midura, J., Shumway, J., & Kozlowski, J. (2020). Blending everyday movement and representational infrastructure: An interaction analysis of kindergarteners coding robot routes. In M. Gresalfi & L. Horn (Edited by.), *Proceedings of International Conference of the Learning Sciences (ICLS)* 2020 Nashville, TN: International Society of the Learning Sciences.

Snow, E., Rutstein, D., Basu, S., Bienkowski, M., & Everson, H. T. (2019). Leveraging evidence-centered design to develop assessments of computational thinking practices. *International Journal of Testing*, *19*(2), 103–127. https://doi.org/10.1080/15305058.2018.1543311

Snow, E., Tate, C., Rutstein, D., & Bienkowski, M. (2017). *Assessment design patterns for computational thinking practices in exploring computer science*. SRI International.

Strawhacker, A., Sullivan, A., & Bers, N. U. (2013, June). TUI, GUI, HUI: is a bimodal interface truly worth the sum of its parts? In Proceedings of the 12th International Conference on Interaction Design and Children (pp. 309–312), New York, NY.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *27*(1), 3–20. https://doi.org/10.1007/s10798-015-9304-5

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education 148,* 103798. https://doi.org/10.1016/j.compedu.2019.103798

Toulmin, S. (1958). *The uses of argument*. Cambridge University Press.

Verdine, B. N., Golinkoff, R. M., Hirsh-Pasek, K., Newcombe, N. S., Filipowicz, A. T., & Chang, A. (2014). Deconstructing building blocks: Preschoolers' spatial assembly performance relates to early mathematical skills. *Child Development*, *85*(3), 1062–1076. https://doi.org/10.1111/cdev.12165

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, J., Touille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1), 5. https://doi.org/10.1145/2576872