2018-12-06

Developing an Online Computational Thinking Instrument for Elementary Students

Guanhua Chen
University of Miami, gxc186@miami.edu

Follow this and additional works at: https://scholarlyrepository.miami.edu/oa_dissertations

Recommended Citation
https://scholarlyrepository.miami.edu/oa_dissertations/2229

This Open access is brought to you for free and open access by the Electronic Theses and Dissertations at Scholarly Repository. It has been accepted for inclusion in Open Access Dissertations by an authorized administrator of Scholarly Repository. For more information, please contact repository.library@miami.edu.
UNIVERSITY OF MIAMI

DEVELOPING AN ONLINE COMPUTATIONAL THINKING INSTRUMENT FOR ELEMENTARY STUDENTS

By

Guanhua Chen

A DISSERTATION

Submitted to the Faculty of the University of Miami in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Coral Gables, Florida

December 2018
DEVELOPING AN ONLINE COMPUTATIONAL THINKING INSTRUMENT FOR ELEMENTARY STUDENTS

Guanhua Chen

Approved:

Ji Shen, Ph.D.
Associate Professor of Teaching and Learning

Jennifer Krawec, Ph.D.
Assistant Professor of Professional Practice of Teaching and Learning

Nam Ju Kim, Ph.D.
Assistant Professor of Teaching and Learning

Guillermo Prado, Ph.D.
Dean of the Graduate School

Moataz Eltoukhy, Ph.D.
Assistant Professor of Kinesiology and Sport Sciences
This study is part of a larger design study that iteratively improves a robotics programming curriculum as well as a computational thinking (CT) instrument. Its focus was majorly on CT assessment and particularly on an online CT instrument with logging functionality that can store a student's problem-solving process by recording interactions between a test-taker and the items with timestamps. The purposes of this research were to examine the psychometric properties of an online CT instrument, test if significant improvement in CT could be found by 200 5th graders who took a robotics programming course and the CT instrument as pretest and posttest, and explore the use of learning analytics methods, mainly convolutional neural networks (CNNs), to help interpret a student’s application of CT when solving a given problem effectively and efficiently.

Rasch testlet model was used to perform item response theory analysis on the CT instrument with six testlets. The results showed good reliability in measuring, adequate discrimination capacity of most of the items, and appropriate difficulty level in measuring CT of 5th graders. No statistically significant results were found regarding improvement in CT from pretest to posttest after the intervention, and possible reasons were listed and discussed. Regarding learning analytics, a CNN model was built, tweaked, and trained by student problem-solving process data from two items in the instrument to predict students' successfulness in solving the problems with good to excellent accuracy. And by
inspecting the trained model parameters, specific problem-solving patterns that inform the interpretation of CT use during the problem-solving process were identified and discussed.
ACKNOWLEDGEMENT

I would like to express the deepest appreciation to my advisor Dr. Ji Shen, who is also my committee chair since this work would not have been possible without his guidance and support. In the past five years, he has granted me all the freedom to explore my research interests while offering invaluable suggestions that inspired me to stay on course. With his insightfulness in research and passion for teaching in the STEM education field, Dr. Shen serves as an excellent role model that I aspire to be.

I have great pleasure in acknowledging my gratitude to my dissertation committee members Dr. Jennifer Krawec, Dr. Nam Ju Kim, and Dr. Moataz Eltoukhy for your guidance throughout this process. Your discussions, ideas, and feedback have contributed significantly to the completion of this dissertation.

I am thankful to all the colleagues and friends I have worked with within and outside of the University of Miami community. It was enjoyable to work with Dr. Lauren Barth-Cohen and Dr. Shiyan Jiang on the TREES project that your valuable input and creativity have led to significant improvement of this project and made the collection of my dissertation data possible. I am indebted to Dr. Jue Wang and Dr. Fangwen Wu, who had generously offered help when I got stuck on methodology problems when writing my dissertation. I am particularly grateful to Dr. Charles Xie, who possesses all the attributes of a genius scientist doing the cutting-edge research that I can imagine, for showing me the exciting future of doing research in the digital era and always inspiring me to move forward and learn new stuff. I had a great time working and studying with my fellow doctoral students Changzhao Wang, Maria Kolovou, Chunhong Zhou, and Hua Ran.
Thank you all for the stimulating discussions we had and the delightful moments we shared.

Last but not least, I would like to thank my parents for their unconditional love and spiritual support to whatever I pursue and my loving and supportive wife, Qin, for the enormous sacrifices she made in the past years to help me get to this point.
Table of Contents

List of Tables ..................................................................................................................... vii
List of Figures ................................................................................................................... viii
Chapter 1 ............................................................................................................................. 1
INTRODUCTION .............................................................................................................. 1
  1.1 Background of the Problem ....................................................................................... 1
  1.3 Overview of Chapters ................................................................................................ 4
Chapter 2 ............................................................................................................................. 5
LITERATURE REVIEW AND THEORETICAL FRAMEWORK .................................. 5
  2.1 Computational Thinking ........................................................................................... 5
    2.1.1 CT overview and importance to STEM fields ..................................................... 5
    2.1.2 Framework .......................................................................................................... 7
    2.1.3 CT in Everyday Reasoning ................................................................................. 9
  2.2 Computational Thinking Assessment ........................................................................ 10
  2.3 Summary ................................................................................................................... 14
Chapter 3 ........................................................................................................................... 15
METHODOLOGY ........................................................................................................... 15
  3.1 Robotics Curriculum ............................................................................................... 17
  3.2 CT Instrument Overview ......................................................................................... 17
  3.3 Prior Work ............................................................................................................... 19
    3.3.1 Pilot study 1 ...................................................................................................... 19
    3.3.2 Pilot study 2 ...................................................................................................... 20
    3.3.3 Pilot study 3 ...................................................................................................... 21
  3.4 Current Design ........................................................................................................ 22
  3.5 Reliability and Validity of the Instrument ................................................................. 22
  3.6 Scoring Process ....................................................................................................... 24
  3.7 Participants and Data Collection ............................................................................... 28
    3.7.1 Participants ...................................................................................................... 28
    3.7.2 Curriculum implementation ............................................................................. 28
    3.7.3 Data sources ..................................................................................................... 29
List of Tables

Table 1 ................................................................................................................................ 7
Table 2 .................................................................................................................................. 16
Table 3 .................................................................................................................................. 18
Table 4 .................................................................................................................................. 25
Table 5 .................................................................................................................................. 30
Table 6 .................................................................................................................................. 31
Table 7 .................................................................................................................................. 33
Table 8 .................................................................................................................................. 48
Table 9 .................................................................................................................................. 49
Table 10 ............................................................................................................................... 59
Table 11 ............................................................................................................................... 62
Table 12 ............................................................................................................................... 70
Table 13 ............................................................................................................................... 71
Table 14 ............................................................................................................................... 72
Table 15 ............................................................................................................................... 75
Table 16 ............................................................................................................................... 77
List of Figures

Figure 1. Programming interface of Choregraphe ............................................................ 16
Figure 2. The iterative process of coding some open-ended items (Chen et al., 2017) ... 24
Figure 3. Perceptron Structure Illustration................................................................. 39
Figure 4. Decision Boundary and Linear Separability.................................................. 40
Figure 5. An Example of Non-linearly Separable Classes with Class Boundary ......... 41
Figure 6. An Example of MLP with Three Layers ...................................................... 42
Figure 7. Iteration process of train an ANN ............................................................... 43
Figure 8. Comparison between a more complex model (left) that classifies training set well but lacks generalizability and a simple model (right) with less training accuracy yet better generalizability.............................................................. 44
Figure 9. Architecture of a CNN (Patel & Pingel, 2018)......................................... 46
Figure 10. Item Characteristic Curve for all the items based on the pretest data ....... 51
Figure 11. Map of Latent Distributions and Response Model Parameter Estimates from Pretest (Each “X” represent 1.6 cases) .............................................................. 52
Figure 12. Histogram (left) and Normal Q-Q Plot (right) of the data ....................... 53
Figure 13. Normal QQ Plot of the difference between pretest and posttest scores after ruling out possible idling students ................................................................. 54
Figure 14. Using Neural Network to Predict Performance from Process Data .......... 56
Figure 15. An Example of Action Frequency – Time Series Visualization ............. 58
Figure 16. Item 4.2 Screenshot ............................................................................. 59
Figure 17. Examples of Student Action Frequency - Time Series Visualization for Item 4.2................................................................. 60
Figure 18. Overfitting as Represented by the Gap between the Training Error and Test Error after 1000 Training Steps ............................................................... 61
Figure 19. Training Stopped at the Optimal Point .................................................... 62
Figure 20. The comparison of a correctly predicted successful case from the test set (left) and a similar case from the training set (right) ........................................... 63
Figure 21. Visualization of the Results on Applying a Filter Carrying Heavy Weights to both Students’ Action Frequency – Time Series Visualization from Figure 20 ....... 64
Figure 22. The comparison of a correctly predicted failure case from the test set (left) and a similar case from the training set (right) ....................................................... 65
Figure 23. Visualization of the Results on Applying a Filter Carrying Heavy Weights to both Students’ Action Frequency – Time Series Visualization from Figure 22 ........ 65
Figure 24. A predicted successful but failed case ..................................................... 66
Figure 25. Milestones of the problem-solving process of the false positive case ....... 67
Figure 26. Visualization of problem-solving processes where student took minimal actions to either solve the problem correctly (left) or incorrectly (right) ................... 68
Figure 27. Item 4.2 Screenshot ............................................................................. 69
Figure 28. Examples of Student Action Frequency - Time Series Visualization for Item 5.2................................................................. 70
Figure 29. Training Process of Pretest Data ............................................................ 71
Figure 30. False negative case 1 ........................................................................... 74
Figure 31. False negative case 2 ....................................................................................... 76
Chapter 1

INTRODUCTION

1.1 Background of the Problem

Computational thinking (CT), as addressed by Wing (2006) as a set of skills that could be applied in many settings, was advocated to be a component of our next generation’s fundamental skills besides reading, writing, and arithmetic. Wing (2006) defined CT as something “involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (p. 33). Since then, CT has been drawing widespread academic attention. Part of the reason is due to the ever-increasing popularity of using machines in solving practical and research problems in virtually all social sectors and disciplines. The goal of equipping our next generation with essential CT skills necessitates the development of both effective curriculum and valid assessment that directly address CT throughout K-12 levels.

Currently existing research on CT has focused on STEM courses that afford opportunities to encapsulate CT in specific tasks. Among those curricula, the most prominent ones are those that involve programming. It is intuitive to turn to programming, considering that the origin of CT stemmed from concepts in computer science in which programming is an integral component. According to Grover and Pea (2013), abstraction, systematic thinking, symbolic representation, algorithmic thinking, modularization, efficiency, and debugging are widely accepted as essential to CT (Aho, 2012; Barr & Stephenson, 2011; National Research Council, 2010; CSTA, 2011; Wing,
No matter which specific language is taught and utilized, programming offers ample opportunities for all of the CT components mentioned before.

Using programming to practice CT is entirely feasible for secondary or higher-level students. It is quite challenging, however, for elementary students. One challenge lies in younger children’s inexperience in programming in general and the relative difficulty of teaching programming to these students. Using visual programming environments (e.g., Scratch, Alice) may address this challenge because this kind of environments utilizes more intuitive visual commands with built-in drag-and-drop features, which is less abstract to understand and reduces the amount of effort in debugging grammatical errors. Another challenge for young children is that it is difficult to evaluate students' learning gains. Although researchers could gain understanding of a student’s CT by analyzing his/her artifacts (Bers et al., 2014; Boe et al., 2013; Brennan & Resnick, 2012; Moreno-León, Robles, & Román-González, 2015; Werner et al., 2012) as well as programming process (Koh et al., 2010; Werner, McDowell, & Denner, 2013), the baseline CT levels could not be easily evaluated before the programming language is introduced.

Another more fundamental problem needs to be addressed. If CT is considered one of the foundational skills, as compared to reading, writing, and arithmetic (the three Rs), it could be directly practiced and assessed just like the three Rs, with or without programming. In fact, the focus of CT has already gone through the process from programming to a broadened idea of computing literacy, meaning applying computing in other fields and everyday life, similar to language and mathematics literacy (Grover & Pea, 2013; National Research Council, 2010).
Considering the challenges discussed above, I am arguing that, especially for elementary students, it is critically important to develop instructional materials that teach CT using non-programming contexts and new CT assessments for participants with minimal programming experience. As an exploratory effort, the Transformative Robotics Experience for Elementary Students (TREES) project features an iterative process of building a robotics programming curriculum, and corresponding assessments with an emphasis on CT interpreted in the broad sense, and the team included Dr. Ji Shen, Dr. Lauren Barth-Cohen, and Dr. Moataz Eltoukhy.

1.2 Research Questions

The study aims to investigate 5th grader’s CT by an online instrument with logging functionality that enables problem-solving process analysis afterward. This assessment tool contains an interest survey and a CT instrument with both everyday reasoning and programming items. The same test was used to elicit student responses as both pretest and posttest. To explore how the instrument helps to understand a test-taker’s CT and how to improve the instrument based on data analysis, this study aims to answer the following research questions:

1. What are the psychometric properties of the CT test?
2. What are the differences of students’ performance from pretest to posttest?
3. How does test-takers’ problem-solving process help to identify and understand their CT application patterns?
1.3 Overview of Chapters

So far I have conveyed the idea of explicitly teach CT to solve problems and the necessity of assessing CT in addition to programming. I also mentioned the iterative nature of building a good instrument based on analysis from data collected through rounds of administration. I briefly summarize the topics for the remaining chapters.

Chapter 2 has two subsections: computational thinking, computational thinking assessment. In the computational thinking subsection, I introduce multiple definitions of CT proposed by different scholars. The current development of various curriculums that integrate CT into different subjects at different levels. In the computational thinking assessment subsection, I summarize the methods adopted by a variety of scholars to assess CT from the literature. I then conclude this chapter with strategies that inform the current study. In chapter 3, I introduce the robotics curriculum and the CT instrument first. I then describe the experience learned from previous runs and the design of the current study, data collection, and analysis methods. In chapter 4, I report the results and findings from the analysis as well as interpretations regarding each of my research questions. In chapter 5, I briefly summarize this study, conclude my significant findings with discussions, point out limitations and future research directions.
Chapter 2

LITERATURE REVIEW AND THEORETICAL FRAMEWORK

2.1 Computational Thinking

2.1.1 CT overview and importance to STEM fields

Since Wing (2006) directed scholarly attention to CT by proposing it as one of the fundamental skills that apply to everyone living in the 21st century, CT has been acknowledged as essential for future generations given the fast development of computing industry and widespread application of intelligent devices which ushered in the so-called big data era. Computing has already revolutionized or been proved to be able to effect significant changes in many fields such as manufacturing industry, business, scientific research, etc. This fact necessitates educational practitioners and researchers to strive for equipping our next generation the ability to manipulate computers successfully to cope with newly emerged challenges. The first step to achieving such a goal is to be able to understand how machines work to solve problems and how to communicate with it. CT, therefore, is deemed essential.

Despite the acknowledgment of CT as a desirable skill for future citizens, it has been proven complicated to achieve a universal definition and the work is far from been done (Aho, 2012; Barr & Stephenson, 2011; Cuny, Snyder, & Wing, 2010; Lu & Fletcher, 2009; National Research Council, 2010; Shute, Sun, & Asbell-Clarke, 2017; Wing, 2006; Wing, 2008). For example, according to Cuny, Snyder, & Wing (2010), “CT is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-
processing agent.” Aho (2012) deemed CT "to be the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms." Barr and Stephenson (2011) addressed CT "an approach to solving problems in a way that can be implemented with a computer." In the meantime, specific dimensions of CT were also mentioned, such as abstraction (Aho, 2012; Barr & Stephenson, 2011; National Research Council, 2010; Wing, 2006), algorithm and algorithmic thinking (Aho, 2012; National Research Council, 2010; Wing, 2006), modularization (Barr & Stephenson, 2011; National Research Council, 2010), data (Barr & Stephenson, 2011; Wing, 2006). CSTA (2011) generalized an operational definition of CT that categorized it into six dimensions: (1) formulating problems in means that machines can help to solve, (2) processing data in logical ways, (3) abstracting data away for representation purposes, (4) organizing solution into automatic algorithm, (5) achieving both effectiveness and efficiency for solutions, (6) generalizing similar problem-solving to apply to other situations. Although different definitions exist regarding what it actually is, CT seemed to be deeply rooted in problem-solving especially when machines are involved (Cunny et al., 2010; Barr & Stephenson, 2011; CSTA, 2011; Aho, 2012). And given the close relationship between the application of computing and STEM-related subjects, CT has been gaining increasing attention from the science education research community. Efforts have been made to define and integrate CT to specifically fit instruction in STEM-related fields (Sengupta et al., 2013; Weintrop et al., 2016). National Research Council (2013) listed CT in the Next Generation Science Standards as one of the eight core SEPs (Sciences and Engineering Practices) that play
essential roles in preparing students to master practical skills as well as deepening their understanding of science and engineering.

2.1.2 Framework

Given the little current consensus of what CT actually is and the pressing needs to assess and foster a student's CT, we proposed a five-component CT framework (Table 1) to guide our effort to integrate CT in our robotics curriculum and to create items for assessment purposes. This framework was adapted from the operational definition of CSTA (2011), which includes major agreed upon essential skills of CT and in the meanwhile operationalizable to be measured.

Table 1

*Five-component CT framework (Chen et al., 2017)*

<table>
<thead>
<tr>
<th>Component</th>
<th>Syntax</th>
<th>Data</th>
<th>Representation</th>
<th>Algorithmic Thinking</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formulating problems and solutions using machine recognizable syntax</td>
<td>Organizing and analyzing data</td>
<td>Representing problems and solutions through multiple external means such as a model and a formula</td>
<td>Conceptualizing and generating solutions through algorithms (a series of ordered steps)</td>
<td>Generating, revising, and evaluating solutions with the goal of achieving the most efficient and effective combination of steps and resources</td>
<td></td>
</tr>
</tbody>
</table>
Syntax. Syntax is the first component of our CT framework. We deemed it an essential dimension of CT since it serves as a means to bridge human thoughts regarding a problem to computer models. It sets the stage for human-computer communication and is directly related to formulating problems as well as solutions in a computer-representable way (Cuny, Snyder, & Wing, 2010). In the assessment process, this component could be measured by observing if students understand and able to use given commands and rules to represent problems and solutions.

Data. The Data component of our CT framework aims to describe the ability to understand the connotations of a piece of data and to process it in a logical manner (CSTA, 2011). Brennan and Resnick (2012) listed data as one of the seven CT concepts in their framework for CT assessment. Angeli et al. (2016) echoed this idea by constantly explaining such concepts to learners during the programming processes to help them build CT capacity. The ability to understand the meaning of individual pieces of data, to discern different kinds of data, and to perform operations on them is a necessary condition to solve problems involving CT successfully.

Representation. Representation is pervasive along the problem-solving process involving CT. In fact, computation could be described as a sequence of transformation of representations (Denning, 2010). CSTA (2011) approaches representation as a way to represent data in light of abstractions.

Algorithmic thinking. CT stresses problem-solving in computer recognizable ways (Aho, 2012; Cuny, Snyder, & Wing, 2010; Wing, 2006; Wing, 2008). Any such solutions could eventually be organized into executable ordered steps. Thus,
understanding of algorithm related concepts such as flow of control is important (Lu & Fletcher, 2009).

Efficiency. Computers were invented to manage complex computations and such problems in reality cost significant resources. Searching for more efficient solutions is, therefore, a constant objective for computer scientists just as Aho and Hopcroft (1974) claimed, “Perhaps the most important principle for the good algorithm designer is to refuse to be content” (p. 70).

It is worth mentioning that above five components of CT were delineated separately to guide our curriculum design and assessment development. In reality, those components are all intertwined with each other. For example, syntax could serve a form of representation to describe problems and solutions in computation; processing data logically could help with efficiency in solving a problem, representing the solution process forms an algorithm, etc. Also, to successfully solve a problem requires a student to master multiple skills rather than a single one and the skill set applied could be different from one student to another.

2.1.3 CT in Everyday Reasoning

Computational thinking is pervasive now in influencing research in both sciences and humanities, and changing how we think (Bunty, 2007; Wing, 2008). Given that problem-solving is at the core of CT, and since computational concepts offer new ways of representing theories and practices, it is natural to apply the same set of CT skills to cope with a wide range of problems (Bunty, 2007). CSTA (2011) directly address one essential CT component as “generalizing and transferring this problem-solving process to
a wide variety of problems.” Thus, CT should be able to be applied transferably to problems in fields other than computer science alone and even in some everyday reasoning settings. Brennan and Resnick (2012) identified certain CT concepts that they claim could be applied by multiple programming languages. Chen et al. (2017) showed some evidence of transferring student learning from a robotics curriculum stressing CT to solving problems in everyday reasoning settings. Koh et al. (2010) developed a visual evaluation tool (Computational Thinking Pattern Graph) that indicated transfer of student learning on CT from games to simulations regarding science. Repenning, Webb, and Ioannidou (2010) even argued that no CT could exist without transfer of student learning to STEM. In sum, if CT is deemed a foundational skill for every citizen living in the digital age, it could be applied in solving problems in different settings.

2.2 Computational Thinking Assessment

CT could not be successfully integrated into the K-12 setting without assessment of it; a valid assessment could not only reveal the effectiveness of a curriculum involves CT but provide information of what students have just learned (Grover & Pea, 2013). Given the importance of computer programming in the origin of CT and its close relationship to it when we consider CT from a linguistic angle (National Research Council, 2010), it is a natural way to consider using programming product when it comes to the assessment of CT. This is what a couple of literature regarding CT assessment did indeed. From their study of interactive media designers, Brennan and Resnick (2012) developed a CT framework with three key dimensions: CT concepts, CT practices, and CT perspectives. They then applied this framework to study the artifacts created by participants (age ranges from 8-17) using Scratch. Bers et al. (2014) implemented a
robotics program in three kindergarten classrooms with 63 kids enrolled (53 included in analysis). By rating every child-fashioned robot program after each instructional activity, they were able to evaluate students' level of understanding of CT based on their framework. Werner et al. (2012) developed an Alice program named the Fairy Assessment to analyze students' application of CT in solving problems in a gaming environment. In this study, two researchers granted scores based on student solutions to three in-game tasks and the analysis showed promising results for assessing CT at the middle school level. Moreno-León, Robles, and Román-González (2015) examined the effectiveness of using an open-source web application called Dr. Scratch, which used the Hairball plug-in developed by Boe et al. (2013) to automatically find undesirable practices in Scratch codes to help students and teachers inspect Scratch programs, to evaluate secondary level students’ Scratch codes and assist them gain CT skills and obtained positive results.

Also, since CT is in its nature is about problem-solving (see section 2.1.1), another strand of research on CT assessment is focused on analyzing problem-solving processes. This method requires the assistance of computer systems with log features. Koh et al. (2010) depicted students’ use of CT concepts by semantically analyzing their game/simulation creating process over time and then visualized the results using CT Pattern graph (CTP) as an evaluation tool. This study presented promising results in CT assessment in that it foreshadowed the possibility of automatic semantic analysis of problem-solving processes involving CT and represented a way to compare the CT profiles of a different student using different platforms. Werner, McDowell, and Denner (2013; 2013 March), in their attempt to convert original log files to more informative
meaningful actions from data collected from a project to develop CT of middle school students, mentioned the possibility of using log files to distinguish programming and problem-solving strategies as well as differences between group programming and programming and between closed-ended and open-ended problems, which is relevant to CT assessment using log files. In fact, the National Academy of Sciences organized a workshop consists of leading scholars from both learning sciences and computer science to discuss "different aspects of what participants thought about computational thinking."

In the published report (National Research Council, 2010), many scholars contended that there is a process or procedure nature resides in CT:

“Computational thinking was closely related to, if not the same as, the original notions of procedural thinking... Procedural thinking includes developing, representing, testing, and debugging procedures...

... computational thinking was primarily about process... other areas of science focus on physical objects, whereas computational thinking focuses on processes and abstract phenomena that enable processes...

... computational thinking is about rigorous analysis and procedures for accomplishing a defined task efficiently...

... computational thinking as “what humans do as they approach the world [that is, their framing, paradigm, philosophy, or language], considering processes, manipulating digital representations (and [meta] models),” and hence all humans engage in computational thinking to some extent already in their daily lives (p. 11-12).”

Some other researchers focus on CT assessment in terms of instruments consist of multiple questions, just as assessing reading ability by reading materials and answer questions and arithmetic ability by solving some problems (Chen et al., 2017; González, 2015; Gouws, Bradshaw, & Wentworth, 2013; Lee, Lin, & Lin, 2014). Among those endeavors, Bebras is widely used and becomes increasingly popular (Dagienè &
Sentence, 2016; Dagienė & Stupuriene, 2016; Lee, Lin, & Lin, 2014). Bebras is an international contest aiming to promote informatics and CT among students of all ages, which possesses a large pool of questions/tasks that could be selected for CT assessment purposes.

While those different approaches taken by scholars to assess CT looked at it from different angles and all have theoretical strongholds, they also have their pros and cons. For instance, merely analyzing student-created artifacts/programs omitted their probably different paths in solving given problems and requires human effort to rate, which is inefficient. Analysis of logged problem-solving process could appropriately provide a finer granularity to look into how a student applies specific CT skills along the problem-solving process, but this method requires a more advanced system with massive storage, strong computing power, and innovative research methods to deal with data. CT instruments consists of multiple-choice questions are efficient and could be easily scaled up if an online version exists. However, the analysis is often inadequate since successfully answering a question may involve multiple CT skills intertwined with each other, and the use of those skills are usually different from one student to another.

In sum, a CT assessment is desirable in that whatever forms it takes in administration, it should be efficient enough to quickly know a student’s ability to use CT in solving problems in terms of scores and insightful enough to know what a student does to solve problems.
2.3 Summary

CT, with its close relationship to computer science domain and transferrable nature to problem-solving in other fields, has been acknowledged as an essential 21st-century skill and attracted much scholarly attention to define, implement, and assess it in several educational settings. However, due to general lack of programming experience of elementary students as well as effective instruments, research on CT at the elementary level is still rare. And despite extensive efforts made in CT assessment by inspecting programming artifacts, analyzing problem-solving processes from computer-logged files, and forming a more traditional test consists of a set of individual questions, it is still a moot point to achieve both effectiveness and efficiency in evaluating a student’s current CT level. Guided by a framework of CT in light of existing literature and analysis results from collected data, this study strives to develop and improve an instrument that achieves both efficiency and insightfulness in CT assessment.
Chapter 3

METHODOLOGY

This study was part of a larger study that adopts principles of design-based research (Brown, 1992; Collins, 1992) to gain insights into CT and foster CT among elementary level students. This design study includes an iterative process of designing and improving a robotics programming curriculum material and CT assessment instrument suitable for upper elementary level students based on multiple rounds of implementation. Situated in the larger design study, this study is focused on validating an online CT assessment and inspecting representative items and student responses from that assessment. In this section, I will explain the details of the program, relevant prior studies regarding the assessment, data collection and analysis approaches, as well as the expected strengths and weaknesses of the study.

3.1 Robotics Curriculum

In 2015, our research team, consisting of educational researchers, robotics experts, and computer scientists, developed the first version of our robotics curriculum for 5th graders and completed a major revision based on feedback collected from the school that ran our curriculum for three rounds of pilot study and analysis results attributed to data collected from that school. This curriculum adopted NAO, a commercialized humanoid robot, as the vehicle for teaching robotics programming and fostering CT. And the platform used to program is Choregraphe (Figure 1), a visual programming environment with a virtual robot that programmers can drag-and-drop
commands in the form of boxes with input and output, connect boxes into ordered steps (algorithms) and test on both virtual and physical robots.

Figure 1. The programming interface of Choregraphe

The current version of our curriculum has six chapters (Table 2) which starts from introducing NAO and Choregraphe in general to familiarize students with the robot and programming environment. Following chapters are organized based on common functions afforded by NAO, such as voice recognition, tactile sensors, movement, and animation (dance). The curriculum stresses students to learn not only programming but also necessary CT skills and apply them in solving problems that could involve machines. There are three mini-projects in this curriculum, all of them (conversation, navigation, and dance) are related to creating solutions in some real-world scenarios.

Table 2

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Main Topics</th>
<th>Mini-projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction and Serial Execution: program the robot to stand up, say hello, and sit down in a row</td>
<td></td>
</tr>
</tbody>
</table>
2 Parallel Execution:

program the robot to say and wave hello simultaneously

3 Voice Recognition:

program the robot to recognize human voices and behave accordingly

4 Tactile Sensors:

program the robot to respond based on different sensors touched

5 Movement:

program the robot to move and turn on Cartesian Coordinates

6 Animation/Dance:

program the robot to dance and repeat dance movements using loops

3.2 CT Instrument Overview

Our current CT instrument is an internet-based assessment tool powered by Qualtrics (Snow & Mann, 2013). This instrument (for screenshots of specific items see Appendix A) has 17 items in total and 9 of them are multiple-choice questions and eight open-ended questions. Those individual items form 5 problem sets, and each set has 2-5 questions sharing the same context with that problem set. There are two contexts in general: everyday reasoning and robotics programming. Among those problem sets, problem set 2 and problem set 3 are addressing robotics programming specifically; problem set 2 is about programming a robotic arm with given command to draw shapes on a paper and problem set 3 is asking students to code a robot to move and behave
according to set rules. Problem set 1 (cooking scenario), problem set 4 (booking plane ticket scenario), and problem set 5 (doing laundry scenario) belong to everyday reasoning context. Each question targets one or more prominent CT dimensions based on our framework (section 2.1.2). And for all the open-ended questions that require students to interact with features afforded by the item on the webpage to reach an answer, student actions could be logged with time stamps. Table 3 shows how all the items from our assessment belong to which context and measure what component or combinations of elements of our framework.

Table 3

Assessment items targeting five CT components (Chen et al., 2017) and two contexts

<table>
<thead>
<tr>
<th>Components</th>
<th>Syntax</th>
<th>Data</th>
<th>Representation</th>
<th>Algorithmic Thinking</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Everyday Reasoning Items</td>
<td>1.1-1.2</td>
<td>1.1-1.2</td>
<td>1.1-1.2</td>
<td>4.1-4.2</td>
<td>4.1-4.2</td>
</tr>
<tr>
<td></td>
<td>1.1-1.2</td>
<td></td>
<td>4.1-4.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.1-1.2</td>
<td></td>
<td>5.1-5.2, 5.3a,</td>
<td></td>
<td>5.1-5.2</td>
</tr>
<tr>
<td></td>
<td>4.1-4.2</td>
<td></td>
<td>5.3b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.3a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robotics Programming Items</td>
<td>2.1-2.5</td>
<td>2.2-2.3</td>
<td>2.1-2.5</td>
<td>2.2-2.5</td>
<td>2.4-2.5</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1-3.2</td>
<td>3.1-3.4</td>
<td></td>
<td></td>
<td>3.1-3.4</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.1</td>
<td>3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our CT instrument has been evolving since three years ago and experienced three rounds of curriculum implementation.
3.3 Prior Work

3.3.1 Pilot study 1

The first pilot study was implemented in one class of around 20 5\textsuperscript{th} graders of a public elementary school. Robot programming was taught as a science course for one semester long with a two-hour session per week in Spring 2015. In this round of implementation, we introduced NAO to students and explored the ways to teach them how to program the robot and how to assess their learning. A robot expert of our team taught the whole course. We asked students to pair up during instruction and programming sessions. We collected on-site videos of certain focus groups, took field notes, and interviewed students in pairs by asking each group to create a robot program to solve a given problem. We also created some questions probing students' learning of robot programming based on our observation of the classroom instruction for the whole semester. This round of pilot study was exploratory, and after countless attempts by trial-and-error, our robot curriculum and assessment plan began to take shape.

First, our first version of the curriculum was compiled from the instructional materials for each session, which organized chapters by topics and served the basic structure for future versions of the curriculum.

Second, rather than assessing students' learning regarding programming in Choregraphe, we began to develop our first instrument evaluating CT. Compared with those 5\textsuperscript{th} graders' learning of a particular programming language (i.e., Choregraphe), we were more concerned with if they performed better in thinking computationally to solve problems.
3.3.2 Pilot study 2

The second pilot study was implemented in the same school as the first pilot study for about six months starting from Fall 2015. The entire fifth grade (6 classes and 125 students) participated this time. We held a three-day professional development workshop for six science teachers of the fifth grade of that school so that they could deliver the curriculum. There was one robotics programming session each week, and each session lasted for 45 minutes to 1 hour. Depends on the specialties of each class, teachers had the freedom to control the progress of their instruction and content to be covered within each session.

For this round of pilot study, we developed our first CT assessment in paper form. It had six problem sets and 23 items in total. Among those items, 15 were multiple-choice questions and eight open-ended questions. The pretest was administered in early Fall in 2015, and 121 students completed it. Due to confliction with standardized test schedule of that school and some tech issues, only two classes completed the curriculum as well as the posttest. By analyzing those collected student responses, we found that the CT assessment showed good psychometric properties in matching calculated student abilities to item difficulty levels and achieved good construct validity and excellent reliability (Chen et al., 2017). Comparison of pretest and posttest results showed that our instrument captured student improvement over time.

Although this paper-and-pencil version of CT instrument showed multiple desirable properties in measuring 5th-grade students' CT skills when solving problems, it has some drawbacks. First, administering a paper-based assessment is time-consuming and difficult to scale up. Second, it cost much time and effort to score each student's
answers, especially their responses regarding open-ended questions. Third, we could only see students' final answer on the paper without a clue of how they reached that answer (i.e., their problem-solving processes), which is important for CT assessment given CT’s problem-solving nature (section 2.2). We then decided to convert our physical assessment into a web-based version to overcome some difficulties as mentioned above.

3.3.3 Pilot study 3

The third pilot study was implemented in the same school as the previous two pilot study in May 2017 for five days, and each session lasted for about 2 hours. Six classes of 125 students of the 5th grade participated and the same six science teachers who taught in the second pilot study delivered the curriculum. The pretest was administered in the previous week of the curriculum and posttest in the following week. A total of 107 students completed both pretest and posttest.

In this round of study, our assessment was done entirely online. Based on feedback received and the data analysis result of pilot study 2, we removed some items that seemed too easy, tweaked some that were too wordy or causing confusions, and altered or added some that require students to interact with the item in their problem-solving process, and those interactions could then be recorded. This version of CT instrument was the same as the one that was used in this study (section 3.2.1). And after student pre/posttest were collected, we analyzed how students' performance improved in both everyday reasoning and programming context, how their motivation was related to their learning gain in terms of CT, and how did some features (e.g. gender, familiarity with problem settings, initial performance, etc.) factor into their learning improvement.
We also explored log data to identify some patterns on how students apply CT skills in the problem-solving process.

### 3.4 Current Design

The current CT instrument is virtually identical to the one administered in pilot study 3 (section 3.2.4) concerning problem set and item structure. We made further changes on wordings of some items based on feedback collected from our collaborating school students, teachers, and researchers. In order to prevent fatigue that may factor into student answering the questions, the instrument will randomize six problem sets for each test-taker. Each item is in a separate webpage. The log file for each item begins with the time a page for an item is loaded and ends when students hit and confirm the "next" button. For items with interactivity features, instructional videos were prepared to walk students through on how to utilize the built-in functions to solve the problem and the system for future analysis will save each interaction.

### 3.5 Reliability and Validity of the Instrument

The reliability and validity of the instrument were both considered in this study. Reliability stands for the consistency of a test in measuring a construct. There are three common ways of estimating the reliability of a test: (1) administration a test on different occasions, (2) utilization of a parallel test, and (3) inspection of internal consistency that items from the same test agree with each other (National Research Council, 1998). Due to the schedule constraints and lack of an established parallel test, the third approach was applied in this paper.
Validity indicates the degree to which a test reflects the construct it claims to measure and thus the credibility of conclusions obtained regarding that construct. Since the assignment of scores in a test is based on the operationalized performances derived from a theory of the construct, psychometrics treats construct validity as essential. According to National Research Council (1998), construct validity contains six aspects: (1) content, which refers to whether a test includes enough sample of behaviors that reflect the knowledge to be measured; (2) substantive, the cognitive processes as required in one item is performed by a test-taker; (3) structural, the creation of items and scoring criteria match the construct domain; (4) generalizable, the measuring of the intended construct in a broader sense rather than unrepresentative sub-contents; (5) external, the correlation between the test performance and some other measures; and (6) consequential, the possible positive and negative consequences of a test. Construct validity could not be securely established without a collection of various evidence that provides support for a combination of aspects aforementioned. Several aspects were already explored in our previous and current research project regarding CT. For example, scientists and robotics experts from our research team participated in creating the CT framework, assessed the instrument, and ensured essential components of CT were covered. This content validity was further addressed by students and teachers from our collaborating schools in the previous runs to make sure the wordings of the instrument are age appropriate. Substantive validity was examined by analyzing the recorded problem-solving process. Structural validity was taken care of from multiple rounds of discussion on newly created items and scoring rubrics. Generalizable validity intersects reliability and is particularly crucial for performance assessment. It is built in the nature
of our CT instrument, which involves both programming and everyday scenarios that we hope could capture a student’s transferring of what he learns from robotics programming to solve problems of other settings. Data collected from previous runs supported the existence of such generalizability (Chen et al., 2017). Construct validity was and will be continually scrutinized and interpreted by data collected from this run and future runs.

### 3.6 Scoring Process

Since student submissions will be automatically saved by Qualtrics and their answers to our CT test could be generated in table format. Scores for all multiple-choice questions and some open-ended questions could be calculated automatically in the form of 0s (incorrect answer) or 1s (correct answer). Item 2.5, 3.3, and 3.4 could not be automatically processed since they ask students to type in code sequences to solve problems, which makes it difficult to rely on machines alone to analyze submitted answers. For those three items, two trained graduate students will score student answers according to the procedure showed in Figure 2 to guarantee satisfactory interrater reliability. Table 4 shows the scoring rubric of item 2.5 that will be compared against during the coding process as an example.

![Figure 2](image.png)  
*Figure 2. The iterative process of coding some open-ended items (Chen et al., 2017)*
Table 4

Rubric sample for CT components of item 2.5, which asks students to draw a shape most similar to a triangle and generate a code sequence to have a robotic arm to complete that shape using the given commands that can only draw vertical and horizontal lines

<table>
<thead>
<tr>
<th>Operational Definition of CT</th>
<th>Expected Performance</th>
<th>Scoring</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formulating problems and solutions using machine recognizable syntax</td>
<td>The answer is written in codes instead of natural languages.</td>
<td>Command s in the solution obey the given form totally</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>step 1: MOVE 1, B.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>step 2: REPEAT 11 [1:1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command s in the solution obey the given form partially</td>
<td>MOVE 2, direction.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REPEAT 3 [2:5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command s in the solution are in natural language or do not obey the given form</td>
<td>Blank.</td>
<td>0</td>
</tr>
<tr>
<td>Representing problems and solutions through multiple external means such as a model and a formula</td>
<td>Shows ability to draw a triangle-like shape on the canvas</td>
<td>The drawn shape looks like a triangle with three sides (at least a slanted side).</td>
<td>2</td>
</tr>
</tbody>
</table>
The drawn shape has at least one slanted side.

The drawn shape has no slanted side.

Conceptualizing and generating solutions through algorithms (a series of ordered steps) The coded steps solve the problem.

Solution algorithm draws a shape that is the same as or similar to a drawn shape.

The coded steps solve the problem.
Generating, revising, and evaluating solutions with the goal of achieving the most efficient and effective combination of

The solution is successful with minimum complexity.

Problem solved without redundant codes

Solution algorithm draws part of the drawn shape.

Solution algorithm doesn’t draw anything of the drawn shape.
3.7 Participants and Data Collection

3.7.1 Participants

A public elementary school in a southeastern state adopted our curriculum for its 5th grade STEM course in Spring, 2018. The school purchased one NAO robot and installed Choregraphe in all the computers of the computer lab. Every student has access to Choregraphe when they are in the computer lab.

The School has 1183 students in total in 2018 (Gender: 48.7% female and 51.3% male; Ethnicity: 64.3% Hispanic, 28.3% White, 3.3% Black, 2.3% Asian, 0.1% Native Hawaiian/Pacific Islander, and 1.7% two or more ethnicities) and 200 students (10 classes) in the fifth grade.

3.7.2 Curriculum implementation

The curriculum was delivered by four 5th grade STEM education teachers of the participating school. A professional development session was held in 2017 to familiarize them with Choregraphe and robot programming, and the electronic copy of the curriculum was distributed to them at the beginning of the spring semester of 2018. A
graduate student went to the school to serve as a teaching assistant on Monday, Wednesday, and Thursday of each week whenever there was a session for the first five chapters and every day for the sixth chapter (animation/dance) since it is a more advanced section and the STEM education teachers requested more assistance.

In the Spring, 2018 semester, each 5th grader class out of 10 had a 45-50 minutes STEM session for every two weeks to deliver the robotics curriculum. The first two weeks were utilized to administer the CT assessment as the pretest and the last two as the posttest. Each class had roughly 8 hours work on the curriculum. And the time cost for each chapter of the curriculum ranges from one hour to two hours. Within each session, the standard procedure was to first introduce students with the topic of the session (e.g., voice recognition, tactile sensors, etc.) with a physical robot demo, followed by a step-by-step instruction to ask students duplicate the demo code on their own Choregraphe. After students prepared their program and tested it and passed on their own virtual robot, they began to either work on a mini-project (if there was one for that chapter and time allowed) or prepare their own version of the program that was related to the topic of the session. The teacher would then pick two to five (depending on the time left) student programs to run on the physical robot.

3.7.3 Data sources

Following sources of data were collected from the two administration of pretest and posttest. Both pretest and posttest were administered in one hour, and students typically took around 30 minutes to complete it.

1. Student answers of the CT instrument (section 3.2).
2. Log files of students' interactions with nine items (8 open-ended questions and one multiple-choice question) from which their problem-solving process could be reconstructed in light of timestamps of each action. Table 5 showed an example of what structured action data were available for analysis from item 2.5.

Table 5

<table>
<thead>
<tr>
<th>Actions</th>
<th>Details</th>
<th>Structured &quot;Activities&quot; List</th>
</tr>
</thead>
<tbody>
<tr>
<td>canvas click</td>
<td>Date, action type, coordinates, color</td>
<td>{&quot;Timestamp&quot;: Time (hh:mm:ss), &quot;Draw&quot;: {&quot;x&quot;: X, &quot;y&quot;: Y, &quot;color&quot;: color}}</td>
</tr>
<tr>
<td>text change</td>
<td>Date, action type, row number, new list of commands</td>
<td>{&quot;Timestamp&quot;: Time (hh:mm:ss), &quot;Write Code&quot;: {&quot;Row&quot;: row number of code change, &quot;Code&quot;: [row 1 code, row 2 code, ...]}}</td>
</tr>
<tr>
<td>check row</td>
<td>Date, action type, row number</td>
<td>{&quot;Timestamp&quot;: Time (hh:mm:ss), &quot;Check Row&quot;: {&quot;Row&quot;: checked row number}}</td>
</tr>
<tr>
<td>uncheck row</td>
<td>Date, action type, row number</td>
<td>{&quot;Timestamp&quot;: Time (hh:mm:ss), &quot;Uncheck Row&quot;: {&quot;Row&quot;: unchecked row number}}</td>
</tr>
<tr>
<td>add row</td>
<td>Date, command type</td>
<td>{&quot;Timestamp&quot;: Time (hh:mm:ss), &quot;Add Row&quot;: &quot;move/repeat&quot;}</td>
</tr>
<tr>
<td>delete row</td>
<td>Date, list of deleted row number</td>
<td>{&quot;Timestamp&quot;: Time (hh:mm:ss), &quot;Delete Row&quot;: [deleted row number]}</td>
</tr>
</tbody>
</table>

In total, 288 student responses were collected. Among those submissions, 171 were from pretest and 117 were from posttest. A total of 96 students completed both pretest and posttest.
3.8 Data Analysis

This study is majorly focused on quantitative analysis. Table 6 listed proposed research questions in this paper and their corresponding methodologies to be used. Detailed introduction of each method will be described in the following sections.

Table 6

Research questions and corresponding methodologies

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. What are the psychometric properties of the CT test?</td>
<td>Item Response Theory analyses:</td>
</tr>
<tr>
<td></td>
<td>Rasch Testlet Model</td>
</tr>
<tr>
<td></td>
<td>Wright Map</td>
</tr>
<tr>
<td></td>
<td>Item Characteristic Curve</td>
</tr>
<tr>
<td>2. What are the differences of students’ performance from pretest to posttest?</td>
<td>1. Shapiro-Wilk test for normality</td>
</tr>
<tr>
<td></td>
<td>2. Paired t-test</td>
</tr>
<tr>
<td></td>
<td>3. Wilcoxon signed-rank test</td>
</tr>
<tr>
<td>3. How does test-takers' problem-solving process help to identify and understand their CT application patterns?</td>
<td>Learning analytics analysis:</td>
</tr>
<tr>
<td></td>
<td>Convolutional Neural Networks</td>
</tr>
</tbody>
</table>

3.8.1 Item Response Theory

IRT analysis is the key to answer research question 1. IRT is also known as the latent response theory whose purpose is ability assessment and was applied in the development of standardized test at the beginning (Lord & Novick, 2008). It then became an important method to assess items from instruments and characteristics of test-takers.
The benefit of IRT is that it possesses a strong theoretical background by inferring the probabilistic distribution of a test-taker’s successful response on items, and this focus on item-level information has the advantage that an examinee’s ability is invariant regarding the items used in calculating it (Baker, 2001; Fan, 1998). This approach is different from Classical Test Theory (CTT) which treats the raw score obtained from an examinee's responses to an instrument as the true score plus random error. Rather than building models to relate a person's ability to answer items correctly, CTT focuses on calculating item difficulty value (aka p-value of the item, which is the correct rate of a group of test-takers on that item) and item discrimination that is often the Pearson correlation between the item scores and the total scores, which makes student and item statistics depend on each other and thus considered as one major drawback of CTT (Fan, 1998, Lord & Novick, 2008).

In general, the purposes of applying IRT models to fit student test data are to estimate the parameters of the items and to estimate the examinees’ ability, given the true score of both are unknown (Baker, 2001). Among the models existing for estimating such measurement, the Rasch model (Rasch, 1960), with one ability parameter $\theta$ for each test-taker and one item difficult measure $b$ for an item, is a simple but powerful model that has been widely utilized. In this study, the Rasch model was used to gain insights into how student ability scores spread across the item difficulty scale for the overall assessment as well as five individual problem set.

Rasch model was first proposed by Georg Rasch in order to analyze test data from a probabilistic perspective (Baker, 2001; Rasch, 1960). The equation for the Rasch model is:
\[
P(x = 1|\theta, b) = \frac{1}{1 + e^{-(\theta - b)}}
\]

In the equation, \( \theta \) stands for person ability, and \( b \) is the item difficulty. \((\theta - b)\) is the logistic deviate (logit). As is shown by the equation, it expresses the probabilistic relationship between the ability level of a test-taker and the item difficulty. The higher a test-taker's ability as compared with the difficulty level of the item (i.e., a larger logit value), the lower the value of the denominator of the equation, which corresponds to a higher probability that this student will correctly answer the question, and vice versa.

Table 7 shows the logit differences and their corresponding probabilities for the Rasch model (Wilson, 2005).

**Table 7**

*Logits and Corresponding Probabilities for the Rasch Model (adapted from Wilson, 2005)*

<table>
<thead>
<tr>
<th>( \theta - b )</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4.0</td>
<td>0.018</td>
</tr>
<tr>
<td>-3.0</td>
<td>0.047</td>
</tr>
<tr>
<td>-2.0</td>
<td>0.119</td>
</tr>
<tr>
<td>-1.0</td>
<td>0.269</td>
</tr>
<tr>
<td>0</td>
<td>0.500</td>
</tr>
<tr>
<td>1.0</td>
<td>0.731</td>
</tr>
<tr>
<td>2.0</td>
<td>0.881</td>
</tr>
<tr>
<td>3.0</td>
<td>0.953</td>
</tr>
<tr>
<td>4.0</td>
<td>0.982</td>
</tr>
</tbody>
</table>
Two important assumptions need to be satisfied when applying simple Rasch model to analyze items from an instrument: 1) unidimensionality, which indicates that only one latent variable is measured by items from the same instrument; and 2) conditional independence, which assumes no correlation between any two items from the same instrument. In our case, unidimensionality could be satisfied since the composite score of the instrument is used as a measure of a student’s CT. However, given that our five sets of items, or testlets, are built in either everyday reasoning or programming scenarios and further in different contexts (such as controlling a robotic arm, doing laundry, etc.), it is likely that significant correlations exist among items belong to the same set with the same context and thus item local independence is violated — the idea that local item dependence accounting for significant correlations among questions after controlling the main Rasch dimension is called testlet effect, which could result in misestimation of item parameters and reliability if ignored (Lee, 2004; Wainer & Wang, 2000). This testlet effect is prevalent in tests of reading comprehension where certain items are sharing a common passage. Because of the probable existence of testlet effect in our instrument, simple Rasch model analysis could be misleading and thus inappropriate. Therefore, I chose to use the Rasch testlet model (Wang & Wilson, 2005) to perform the analysis. In the Rasch testlet model, the probability of correctly answering one item is:

\[ P(x = 1|\theta, b, \gamma_{nd}) = \frac{1}{1 + e^{-(\theta - b + \gamma_{nd})}} \]  

Where \( \theta \) stands for person ability and \( b \) the item difficulty as in equation (1). \( \gamma_{nd} \) is introduced as the random effect that accounts for the interaction between a test-taker and an item within a testlet and \( \gamma_{nd} \sim N(0, \sigma^2_{\gamma}) \) where \( \sigma^2_{\gamma} \) represents the testlet effects of
the set of items that the particular item belongs to (Wang & Wilson, 2005). As is shown in equation (2), the testlet effect is dealt with by the Rasch testlet model and a more precise set of item parameters and student abilities could be estimated.

Equation (2) is used for dichotomous items only, that is, the questions scored one if a student answers correctly and 0 incorrectly in our instrument, just like equation (3) indicates:

\[
P(x = 1|\theta, b, \gamma_{nd}) + P(x = 0|\theta, b, \gamma_{nd}) = 1
\]  

Equation (2) could then be represented as:

\[
\log \left( \frac{P(x = 1|\theta, b, \gamma_{nd})}{P(x = 0|\theta, b, \gamma_{nd})} \right) = \theta - b + \gamma_{nd}
\]  

In order to take polytomous items with three levels of difficulty and thus partial credit is given toward a correct response, equation (4) could be extended to equation (5) as:

\[
\log \left( \frac{P(x = j|\theta, b, \gamma_{nd})}{P(x = j - 1|\theta, b, \gamma_{nd})} \right) = \theta - b_j + \gamma_{nd}
\]  

Where \(x \in \{0, 1, \ldots, m\}\), and \(m\) is the maximum score for a particular item. Let:

\[
b_j = b + (b_j - b) \equiv b + \tau_j
\]  

Where \(b\) is the overall difficulty and \(\tau_j\) is the \(j\)th threshold parameter of that item (Wang & Wilson, 2005). Equation (5) could finally be represented as:

\[
\log \left( \frac{P(x = j|\theta, b, \gamma_{nd})}{P(x = j - 1|\theta, b, \gamma_{nd})} \right) = \theta - (b + \tau_j) + \gamma_{nd}
\]
Wang and Wilson (2005) used ConQuest, a generalized item response modeling tool (Wu, Adams, & Wilson, 1998) to estimate the Rasch testlet model parameters of an English test with 11 testlets. Based on the work Gelfand and Smith (1990) did to infer model parameters by sampling from the marginal posterior distribution using a Markov chain Monte Carlo (MCMC) simulation, they used marginal maximum likelihood (MML) with expectation-maximization (EM) algorithm to obtain estimated person and testlet parameters. In their model, a numerical method was used to iteratively calculate approximated log-likelihood based on parameter estimates from the previous circle and then maximize the expected log-likelihood. I ran the iterative maximum likelihood estimation process to estimate those parameters for the CT instrument items. I then calculated the chi-square goodness-of-fit index to see the agreement between the observed correction rate and the fitted item characteristic curve of the items and analyze the results. The EAP/PV reliability, which is analogous to Cronbach’s alpha in Classical Test Theory and measures the internal consistency among items, and person separation reliability, which measures whether the instrument is sensitive enough to differentiate high performers from low performing students, were both reported.

After item parameters and person abilities are estimated from the Rasch testlet model, Wright Map (Wilson, 2005) could be used to represent the probabilistic relationship between them on one graph sharing the same measurement scale. Wright Map uses logit as the unit of measure and places estimated person ability scores together with item difficulties to help better understand how the instrument performed in measuring the intended construct.
ConQuest was used in the analysis of the above models using the collected pretest responses from our CT instrument to estimate item parameters, student abilities, and related reliability measures as well as produce the Wright Maps. Matplotlib, a python visualization package, was used to generate the item characteristic curve for all the items and other charts as well as calculate the chi-square goodness-of-fit index for items based on estimated parameters.

3.8.2 Pre/posttest comparison

To answer research question 2, paired t-test was performed to see if students achieve statistically significant learning gain from pretest to posttest. Since paired t-test assumes the difference from pretest to posttest follows a normal distribution, Shapiro-Wilk test was used to test the normality of the data and Wilcoxon signed-rank test was used as an alternative to test if learning gain is significant.

3.8.3 Learning analytics

In order to answer research question 3. Specific learning analytics methods were used to analyze the log files. Two students may give precisely the same answer for an item. However, the paths they take to reach that answer may be very different. It is, therefore, meaningful to look at their problem-solving process in details. To learn student problem-solving patterns could shed light on how they leverage various CT skills in the problem-solving process. Learning analytics methods could be utilized to automatically mine such patterns from the recorded problem-solving sequence for each test-taker in no time. Grounded on big data collected from real-time interaction by students and questions embedded in a computer-supported system, these kinds of learning analytics methods
enable a machine to automatically produce knowledge on how a problem is solved, which in our case goes a long way to explain how CT is applied in solving a problem. Also, this combination of big data collection and intelligent analysis powered by learning analytics approaches could serve as a foundation to create automatic feedback systems, which could assist both teachers and students in the future. This paper explored the application of one kind of deep neural networks called convolutional neural network (CNN) to learn from students’ problem-solving process data and predict their correctness of answer with certain accuracy. One of the primary reasons for applying CNN models to logged problem-solving process data is that CNNs assume that proximate data points are closely related, and local patterns are relevant everywhere, which makes such models particularly useful for identifying patterns that help with classification tasks, given that actions close to each other temporally are also more relevant to each other, and specific orders of actions could be meaningful indicators of some output classes (e.g., success or failure in solving a particular problem). By tuning and training parameters of the built CNN model, problem-solving patterns emerge automatically that helps to interpret the CT application of different test-takers.

3.7.3.1 The Perceptron Model

Artificial neural networks (ANNs) was an idea inspired by research on artificial intelligence. It was first introduced by McCulloch and Pitts (1943) in a study that explored the use of a simplified neuron model in groups in order to apply propositional logic to explain how human brain perform complex tasks. This computational theory of mind and brain pioneered the creation of ANNs with various architecture. Among those ANNs, the Perceptron is the simplest in structure with only one layer and serves as the
foundation for building up neural networks with more layers. It was proposed by Rosenblatt (1958) and Figure 3 shows its architecture. Suppose data consist of training examples with inputs that could be represented as \( x (x_1, x_2, \ldots, x_n) \) and output as binary values, in other words, \( \text{class}(x)=1 \) if it is a positive example and \( \text{class}(x)=0 \) if negative. The Perceptron model could then be trained to perform linear binary classification. In the training process, inputs of every example are multiplied by their corresponding weights and added together to gain the weighted sum, and this sum is then applied to a step function such as \( h(x) \) in Figure 3 to obtain the predicted class label \( \text{class}(x) \).

**Figure 3. Perceptron Structure Illustration**

Initially, weights of the model are randomly assigned small values since a weight \( w_i \) shows the strength of the node \( x_i \) in deciding the label of a particular example and no assumptions is needed in that the model could learn to adjust its weights from mislabeled cases. Equation (8) is used to update the weights each time after a training instance \( x \) is fed and hypothesized label \( h(x) \) is calculated.

\[
w_i = w_i + \eta \cdot [\text{class}(x) - h(x)] \cdot x_i
\]  

(8)
In the above equation, $\eta$ is the learning rate indicating how much weight should be adjusted with respect to the distance between the predicted class label and the true label. As is shown in the equation, if $\text{class}(x) = \text{h}(x)$, meaning the instance is correctly classified, the weight $w_i$ does not need to be changed. If $\text{class}(x) = 1$ and $\text{h}(x) = 0$, the weight needs to increase in order to gain a larger weighted sum, which increases the possibility of output a positive class ($\text{h}(x) = 1$). And if $\text{class}(x) = 0$ and $\text{h}(x) = 1$, the updated weight will be lowered to approach a $\text{h}(x) = 0$. In sum, one input node $x_i$ and its connection to final output are either reinforced or weakened based on information obtained from the data until $\text{class}(x) = \text{h}(x)$ for all training examples. The endless number of decision boundary exists to classify linearly separable training examples (see Figure 4 for instance).

![Figure 4. Decision Boundary and Linear Separability](image)

Also, because the decision boundary for the Perceptron model is linear, it could not perform complex tasks such as classes represented by Figure 5.
A way to overcome the major drawback of Perceptrons that differentiates only linearly separable classes is to connect more units by links of weights into a more extensive network, and this method has been proven highly promising. Since this organization of single units into layers with weighted links originated from the research on the relationship between mind and body as well as on building of human intelligence into a machine, it resembles the flow of information in human mind to perceive the world through signals passing from one neuron to another. It is thus called ANN, and each unit in the network is called a neuron. Among the developing field of ANNs with various architecture, MLP is the most frequently mentioned example. Figure 6. Shows an instance of MLP with three layers. Just as the Perceptron model, this MLP model has an input layer and an output layer. It also has one more layer in between called a hidden layer. An ANN could have more that one hidden layer. In fact, many applications have
ANNs with hundreds of hidden layers, which contain more parameters to be trained and hence require more data to form a big enough training set correspondingly. An ANN with two or more than two hidden layers is called a deep neural network, which serves as the basis for deep learning.

Figure 6. An Example of MLP with Three Layers

The use of MLP to classify an instance $x = (x_1, x_2, \ldots, x_n)$ given $n$ classes ($y_1, y_2, \ldots, y_n$) is similar to that of the Perceptron model. This process is called forward propagation since features of $x$ are passed through links of weights to neurons of the next layer, and the calculation results of a neuron serve as the inputs for neurons of the next layer until output layer is finally reached. To illustrate this idea, the contribution of the $x_i$ from the input layer to the $j^{\text{th}}$ neuron of the hidden layer to the weighted sum of that neuron in Figure 6 is $w_{ij}^{(1)} \times x_i$, the superscript $(1)$ means weights from the input layer to the hidden layer. The weighted sum for the $j^{\text{th}}$ neuron in the hidden layer is then calculated by
$\Sigma_i w_{ij}^{(1)} \times x_i$. This weighted sum is further fed to the step function for the $j^{th}$ neuron to gain the calculation result of this neuron $a_j^{(1)}$, which equals to $h(\Sigma_i w_{ij}^{(1)} \times x_i)$. And $a_j^{(1)}$ serves as the input for calculation in the next layer. For example, the output of $y_k$ could be represented as $h(\Sigma_j w_{jk}^{(2)} \times a_j^{(1)})$ in Figure 6.

As long as the parameters of the MLP are appropriately set, it could be used to approximate any function that maps the input to the output correctly. The backpropagation algorithm proposed by Rumelhart, Hinton, and Williams (1986) was a groundbreaking piece that made possible the readjustment of parameters in ANNs to realize the mapping of input to output. This readjustment of model parameters from data is called the training process. It is achieved by an iterating process of calculating the distance between the neural network’s predicted results and the true labels of the inputs. As soon as the function is constructed to estimate the errors of prediction after each iteration, gradient descent could be applied to adjust the weight of neurons (Figure 7).

![Figure 7. Iteration process of train an ANN](image-url)
Since the backpropagation algorithm can constantly adjust weights to reach better classification accuracy, given enough time and proper learning rate, the trained model could become very complicated. Although it might perform well in predicting the labels of cases from the training set, it lacks the generalizability to apply to data that this model does not see in the training process (Figure 8). This is called overfitting. In order to prevent overfitting, a data set is typically split into a training set and a test set. Training set instances are to be used to train the model parameters, and after certain accuracy is achieved, the test set could be used to determine if the training needs to continue.

![Figure 8. Comparison between a more complex model (left) that classifies training set well but lacks generalizability and a simple model (right) with less training accuracy yet better generalizability](image)

### 3.7.3.3 Convolutional Neural Network (CNN)

CNN is a kind of ANNs that has been widely used in the computer vision domain. It belongs to MLP with certain variations (LeCun, 2015). CNN could identify and generalize patterns from images with specific labels and thus overcomes a significant drawback of other earlier image classification algorithms that need human knowledge at
the pre-processing stage. CNN uses convolution and pooling layers to extract and represent specific features of an image before feeding the outputs to fully connected layers of typical MLP for classification purpose (Figure 9). A convolution layer has several squared matrices with given length called filters to perform matrix multiplication and summation (convolution) when sliding over the input with a preset size of stride, and activation such as ReLU is applied on the convolution results to produce the output of the layer and then fed to the next layer. The filter square is sometimes called receptive field named after the process of how animal neuron cells perceive the world through vision. A pooling layer usually is after a convolution layer to further reduce the dimensionality of the inputs while attaining enough information to be passed to the next layer. This image processing method both reduces parameters needed to be trained and controls overfitting. Thus, CNN is particularly useful for addressing images since if each pixel of an image is to be treated as an input for a more traditional fully connected MLP, it could result in a significant network structure with a sea of parameters to be adjusted by data. Zeiler and Fergus (2014) created a way to visualize how intermediate layers of a CNN work in order to perform classification tasks. They found that layers of convolution have a hierarchical effect of generating patterns that could be useful in following classification layers. That is, from the detection of an edge, corner, etc. in previous layers to the generation of patterns showing a potent combination of features that are more distinctive.
The way how CNN works to use convolution layers to process areas within one receptive field of an image in that pixels are more relevant to each other due to their proximity could also be extended to analyze specific time series data, given that actions close to each other on a temporal scale are likely to be relevant, and this action-time relationship could be represented in an image to be fed to a CNN.

In this study, I built a CNN to fit student successfulness in solving some problems using their action data and explain their performance based on the extracted patterns from the neural network calculation. A student’s actions as logged by the system carry information of his or her thinking process to some extent in solving a given problem, and such information is potentially useful in helping identify how CT is applied in the problem-solving process. In order to apply CNN on those log files, students' problem-solving processes on a particular question were first converted to a visualization considers temporal order. By building and tweaking a CNN, it takes the visualized sequences (the actions in a time series manner) into consideration and fit it in ways that
lead to given outputs (student performance in terms of scores, calculated ability, label of identified clusters, etc. and in this study, student correctness in solving a particular question) (LeCun, Bengio, & Hinton, 2015; Schmidhuber, 2015). And by examining the intermediate layers of the classifiers of the trained network, the obtained information could reveal how patterns in a student’s problem-solving process could contribute to the given output (Yosinski 2015; Zeiler & Fergus, 2014). Inspecting those patterns helps to understand student correctness by looking at the process of how they approach a given problem to find a solution eventually, or not.
Chapter 4

FINDINGS AND INTERPRETATIONS

4.1. IRT Analysis Results

Table 8 shows the estimated item difficulty for all the items by the Rasch testlet model based on 117 pretest responses. Item 2.5, 3.3, and 3.4 are open-ended questions (see section 3.5) rated according to rubrics from a different component in our CT framework and each component is thus treated as a separate item.

Table 8

Item difficulty estimation results from the Rasch testlet model

<table>
<thead>
<tr>
<th>Item Index</th>
<th>Item Number</th>
<th>Estimated Item Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>-0.756</td>
</tr>
<tr>
<td>2</td>
<td>1.2</td>
<td>0.477</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
<td>-0.823</td>
</tr>
<tr>
<td>4</td>
<td>2.2</td>
<td>-0.337</td>
</tr>
<tr>
<td>5</td>
<td>2.3</td>
<td>0.016</td>
</tr>
<tr>
<td>6</td>
<td>2.4</td>
<td>0.939</td>
</tr>
<tr>
<td>7</td>
<td>2.5f</td>
<td>1.021</td>
</tr>
<tr>
<td>8</td>
<td>2.5r</td>
<td>-0.141</td>
</tr>
<tr>
<td>9</td>
<td>2.5a</td>
<td>1.327</td>
</tr>
<tr>
<td>10</td>
<td>2.5e</td>
<td>1.882</td>
</tr>
<tr>
<td>11</td>
<td>3.1</td>
<td>-1.055</td>
</tr>
<tr>
<td>12</td>
<td>3.2</td>
<td>-0.586</td>
</tr>
<tr>
<td>13</td>
<td>3.3f</td>
<td>0.282</td>
</tr>
<tr>
<td>14</td>
<td>3.3d</td>
<td>-0.567</td>
</tr>
<tr>
<td>15</td>
<td>3.3a</td>
<td>0.323</td>
</tr>
</tbody>
</table>
Student ability measures were also estimated and provided by the Rasch testlet model algorithm (Appendix B). IRT is concerned with the goodness-of-fit of a chosen item characteristic curve (ICC) model to the responses collected for an item (Baker, 2001). The chi-square goodness-of-fit index was then calculated for all 25 items based on student groups with different estimated ability level. Table 9 shows the calculated results. It demonstrates that most items showed satisfied goodness-of-fit that item parameter estimates match the observed proportion of correct responses of different student groups well.

Table 9

*Chi-square goodness-of-fit test results for all the items*

<table>
<thead>
<tr>
<th>Item Index</th>
<th>Item Number</th>
<th>$\chi^2$</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>9.146339</td>
<td>0.424</td>
</tr>
<tr>
<td>2</td>
<td>1.2</td>
<td>8.37567</td>
<td>0.497</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
<td>4.210833</td>
<td>0.897</td>
</tr>
<tr>
<td>4</td>
<td>2.2</td>
<td>3.330693</td>
<td>0.950</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>5</td>
<td>2.3</td>
<td>2.235789</td>
<td>0.987</td>
</tr>
<tr>
<td>6</td>
<td>2.4</td>
<td>18.80432</td>
<td>0.027**</td>
</tr>
<tr>
<td>7</td>
<td>2.5f</td>
<td>23.44578</td>
<td>0.005***</td>
</tr>
<tr>
<td>8</td>
<td>2.5r</td>
<td>5.774659</td>
<td>0.762</td>
</tr>
<tr>
<td>9</td>
<td>2.5a</td>
<td>19.95731</td>
<td>0.018**</td>
</tr>
<tr>
<td>10</td>
<td>2.5e</td>
<td>16.99441</td>
<td>0.049**</td>
</tr>
<tr>
<td>11</td>
<td>3.1</td>
<td>5.877775</td>
<td>0.752</td>
</tr>
<tr>
<td>12</td>
<td>3.2</td>
<td>3.017226</td>
<td>0.964</td>
</tr>
<tr>
<td>13</td>
<td>3.3f</td>
<td>4.749426</td>
<td>0.856</td>
</tr>
<tr>
<td>14</td>
<td>3.3d</td>
<td>9.327594</td>
<td>0.408</td>
</tr>
<tr>
<td>15</td>
<td>3.3a</td>
<td>4.611123</td>
<td>0.867</td>
</tr>
<tr>
<td>16</td>
<td>3.4f</td>
<td>8.103439</td>
<td>0.524</td>
</tr>
<tr>
<td>17</td>
<td>3.4d</td>
<td>14.7422</td>
<td>0.098*</td>
</tr>
<tr>
<td>18</td>
<td>3.4r</td>
<td>5.059128</td>
<td>0.829</td>
</tr>
<tr>
<td>19</td>
<td>3.4a</td>
<td>7.951384</td>
<td>0.539</td>
</tr>
<tr>
<td>20</td>
<td>4.1</td>
<td>8.948075</td>
<td>0.442</td>
</tr>
<tr>
<td>21</td>
<td>4.2</td>
<td>4.877319</td>
<td>0.845</td>
</tr>
<tr>
<td>22</td>
<td>5.1</td>
<td>16.13242</td>
<td>0.064*</td>
</tr>
<tr>
<td>23</td>
<td>5.2</td>
<td>16.09065</td>
<td>0.065*</td>
</tr>
<tr>
<td>24</td>
<td>5.3a</td>
<td>4.165021</td>
<td>0.900</td>
</tr>
<tr>
<td>25</td>
<td>5.3b</td>
<td>17.93244</td>
<td>0.036**</td>
</tr>
</tbody>
</table>

* p<.1, **p<.05, *** p<.01

Figure 10 visualizes the ICC for all 25 items and offers an intuitive understanding of the chi-square goodness-of-fit test results from Table 9. For example, item 2.4, 2.5e, 2.5a, 2.5f, 3.4d, 5.1, and 5.2 showed a significant difference between observed and expected sample frequency at various levels from p<.1 to p<.01. A look at their corresponding ICC curve showed that those items are more discriminating than what
Rasch model assumes (discrimination parameter=1.0 and the higher the parameter, the steeper the fitted curve). In fact, those items are the most difficult items (Table 8) that students from the highest ability group are more likely to answer correctly as shown on the figure. Another item of interest is 5.3b since it is not discriminating students from different ability groups at all. A possible reason is that this item is a multiple-choice question with three choices, so the guessing rate is moderately high (33%) that might confound the situation.

Figure 10. Item Characteristic Curve for all the items based on the pretest data
Figure 11 shows the Wright map that represents the probabilistic relationship between students with various estimated ability and items with different difficulty levels on one graph sharing the same measurement scale. The first column in Figure 11 represents the estimated student ability for the entire instrument. And column 2 to column 6 corresponds to estimated student ability for item sets 1 to item sets five respectively. The last column shows the distribution of difficulty levels of all items. Item difficulty levels ranges from -1.055 (item 3.1) to 1.882 (item 2.5e). According to this figure, item difficulty spreads along the scale and matches student ability overall as well as for each item set. The EAP/PV reliability of the instrument is 0.703, which serves as a satisfactory indicator of using this instrument to measure the same construct reliably. Person separation reliability is 0.978, meaning this instrument did a good job in differentiating students who performed well from those performed poorly.

*Figure 11. Map of Latent Distributions and Response Model Parameter Estimates from Pretest (Each “X” represent 1.6 cases)*
4.2. Pre/posttest Comparison

I collected 171 pretest responses and 117 posttest submissions, and 96 students completed both pretest and posttest by matching their id numbers they input at the beginning of the test and recorded test date. One-tailed, paired t-test could help to test if they significantly improved from pretest to posttest. Given that performing paired t-test requires the difference from pretest to posttest follows a normal distribution, a histogram showing the data distribution and a Normal Q-Q Plot were drawn (Figure 12). A Shapiro-Wilk test for normality was also performed, and the p-value is 0.051. Thus, the data could be considered to follow a normal distribution. The paired t-test of students' total scores is then performed, and no significant difference is found from pretest to posttest (t=0.46, df=95, p=0.678).

![Figure 12. Histogram (left) and Normal Q-Q Plot (right) of the data](image)

The histogram skews in a way to the right since the mode of the figure is located between 0 and 5 and there exists a long tail at the left end. I then suspected that some idling students' responses introduced difficulty to interpret the learning gain of all students. In fact, among all 171 pretest results, 12 students cost less than 900 seconds to complete the test, and 36 out of 117 posttest responses were completed under the same
threshold. I then removed those idling suspects from the list of students who completed both tests and checked the normality of their difference regarding scores from pretest to posttest for the remaining 68 cases (Figure 13). Since the Shapiro-Wilk test suggested a slightly non-normal distribution (p=0.035), an alternative Wilcoxon signed-rank test which does not assume normality for the data was then performed. And the difference is marginally significant (p=0.057). The matched-pairs rank biserial correlation, which could be interpreted as the strength of the relationship between the test condition and the dependent variable (Welkowitz et al., 2011), could serve as the effect size indicator. Since the scores from pretest to posttest are hypothesized to increase, the favorable rank sum is 1352.5 (61.2%) and the unfavorable rank sum is 858.5 (38.8%). According to Kerby (2014), the matched-pairs rank biserial correlation is then 22.4%, which indicates a small effect size.

![Normal QQ Plot](image)

*Figure 13. Normal QQ Plot of the difference between pretest and posttest scores after ruling out possible idling students*
In conclusion, no significant improvement was observed from students' pretest to posttest by our instrument. There are some possible reasons for this finding. First, students might not treat this assessment seriously and invested less time and energy to complete it. This is perhaps a ubiquitous problem for such low-stake tests of this kind. And this idleness might be even more pronounced during the two weeks when posttest was administered in May: students were busy and tired by all kinds of tests. In fact, I collected only 117 posttest responses because our collaborating school did not have time to separate its computer room for the posttest for four classes. Second, every student received about 45 - 50 minutes of instruction for every two weeks, so the intervention might not be strong enough to effect a change. Another explanation is that the curriculum was not effective in this round of implementation due to some reasons such as incomplete delivery of the course content as a result of time constraint for each session. Also, possible validity issues of the instrument could factor into the null observation, which will be elaborated in the conclusion section.

4.3 Learning Analytics

In a common instrument with some questions measuring certain kinds of ability, if a student provides the correct answer to a question, he or she is considered likely to move up along the ability level continuum. On the one hand, this way of measurement is efficient especially for multiple-choice questions but lacks insightfulness. On the other hand, most questions either lack the information needed to delve into how a problem is solved or possess some information (open-ended questions, for instance) that requires human intelligence to decode. In this paper, I tackle both issues. I want not to merely look at a student's choice and compare it against the key but to dig more into how he
manages to apply CT to solve the problem efficiently. This is where learning analytics, with its various data mining and machine learning techniques, could help. This study explores the use of problem-solving process data to train a CNN to predict a student’s successfulness in solving given problems (Figure 14). By looking into the performance of the model on testing sets as well as the trained layer parameters for several items in our CT instrument, inference on item characteristics and student performances are analyzed and discussed.

\[\text{Figure 14. Using Neural Network to Predict Performance from Process Data}\]

\textit{4.3.1 Visualizing problem-solving process}

CNN is particularly useful in addressing data that could be represented as images in that pixels in one image is more relevant to nearby counterparts than those further away, so convolution and pooling layers could reduce dimensions by grouping pixels in one receptive field together. This assumption holds for time-series data such as a student's problem-solving process because actions close to each other concerning timestamp are likely to be relevant than those happen way earlier or later. To apply a CNN to a student's problem-solving process, I need first to convert our process data into
a visualization integrating time-series and actions. I then use the Action Frequency – Time Series visualization to count the action frequencies within continuous intervals to form a squared image. The number of intervals was set to be as the same as the number of extracted actions for an item to build a squared image. Each interval spans equally in terms of duration. One purpose of setting up interval in this way is to ensure that sequences of students who cost different time on the same item could be normalized and compared. Also, the actions close to each other may not be too far away from each other or clump together on the visualization. Figure 15. is an example where the horizontal axis stands for intervals from the loading of an item to the time this item is submitted, and the vertical axis represents actions extracted from the process data for this item that are related to CT. The color coding in such visualization is related to the frequencies of each square: the higher the number, the brighter the color. The frequencies and axis labels were added to illustrate the information presented by this figure and removed before fed into a CNN for training purposes. This method has its merits to represent counts of actions in a temporal manner that information of when and how a student manage to solve this problem is reserved to some degree. And actions lumped together on an image could be easily spotted by CNN and are likely to be informative and thus affect weight calculation in later layers. However, it lost dimensions such as the order of actions within the same time interval and transitions between two actions in two adjacent intervals. This simplification is necessary since it is unlikely to represent as many dimensions as needed within a 2-dimensional image, especially when time-series is considered an essential dimension that needs to be included anyway.
4.3.2 Item 4.2 analysis

Item 4.2 is a question to ask students to figure out a path that cost the least time to fly from Miami to LA, and possible paths with time and cost for each leg are given in a chart. Students answer this question by drag and drop and then order connecting stops. They could also use a button on the screen to calculate time and cost needed for a prepared path (Figure 16).
Figure 16. Item 4.2 Screenshot

Table 10 lists the relationship between specific observable actions and CT skills from our framework. Figure 17 shows two students’ Action Frequency – Time Series visualization from their pretest responses. From all the 171 pretest responses on this item, 119 (70%) were randomly chosen to form the training set and the other 52 (30%) testing set.

Table 10

<table>
<thead>
<tr>
<th>Observable Actions</th>
<th>Possible CT Categories Related</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times a student uses the calculation button.</td>
<td>Data, Representation, Algorithmic Thinking, Efficiency</td>
</tr>
</tbody>
</table>
Times a student calculates an existing route.  

Times a student calculates a non-existing route.  

Existing routes a student comes up with.  

Non-existing routes a student comes up with.  

...  

...  

<table>
<thead>
<tr>
<th>Action Frequency on Time Series</th>
<th>Action Frequency on Time Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Page</td>
<td>Load Page</td>
</tr>
<tr>
<td>Route: exist</td>
<td>Route: exist</td>
</tr>
<tr>
<td>Route: non-exist</td>
<td>Route: non-exist</td>
</tr>
<tr>
<td>Calculate</td>
<td>Calculate</td>
</tr>
<tr>
<td>Found: exist</td>
<td>Found: exist</td>
</tr>
<tr>
<td>Found: non-exist</td>
<td>Found: non-exist</td>
</tr>
<tr>
<td>Proceed: false</td>
<td>Proceed: false</td>
</tr>
<tr>
<td>Proceed: true</td>
<td>Proceed: true</td>
</tr>
</tbody>
</table>

**Figure 17.** Examples of Student Action Frequency - Time Series Visualization for Item 4.2

In order to address the problem of overfitting, the classification errors for both the training set and the testing set were plotted in Figure 18. As the training error regularly goes down when training continues, the test error goes down initially as well but drives up after a turning point, which signals that ideal model complexity is reached and early stopping is thus needed.
Figure 18. Overfitting as Represented by the Gap between the Training Error and Test Error after 1000 Training Steps

Figure 19 shows the training error and test error when early stopping is imposed, where test error reached around 30%, meaning approximately 70% examples were correctly classified. In fact, the test accuracy on the test set reached 69.2%. That is, among 52 instances in the test set, 36 were correctly classified. This model seemed to have improved by training given that 43.3% students actually answered this question in the pretest. However, the improvement is far from ideal, and it could be helpful to look into the correctly classified and misclassified cases to inform the interpretation of a student's use of CT in problem-solving and the amelioration of this item.
Table 11 shows the confusion matrix of the classification results on the test set. From this table, it could be seen that 11 students who successfully solved this problem were correctly predicted (true positive) and 25 who actually failed were also correctly classified (true negative). As opposed to those correctly classified cases, eight students who successfully solved the problem were predicted to fail (false negative), and another eight students who failed were anticipated to success (false positive).

Table 11

Confusion matrix of the prediction based on the test set (item 4.2)

<table>
<thead>
<tr>
<th>Actual Value</th>
<th>Predicted Value</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Successful</td>
<td>Successful</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Successful</td>
<td>Unsuccessful</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>Unsuccessful</td>
<td>Successful</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Unsuccessful</td>
<td>Unsuccessful</td>
<td></td>
<td>25</td>
</tr>
</tbody>
</table>

It is illustrative to look at a representative example as shown by Figure 20 to understand how the trained CNN correctly predicted a successful case. The left
visualization of this figure is a true positive case from the test set, and the right visualizes a successful problem-solver's process data from the training set. From the comparison, it could be seen that both students kept alternating between preparing routes and calculating to see the time and money cost for existing routes. It is intuitive for an instructor to infer that both students are likely to solve this problem since their actions reflect that they approach this problem in a somewhat systematic manner, that is, iterating possible solutions to compare and then decide. This pattern was reflected in the visualization as something looks like a staircase. And the same pattern was captured by the convolutional layer in the trained CNN as shown in Figure 21, where the "staircase" feature was noticed by one filter and visualized. This feature is connected with high weights to carry into later layers in deciding the classification result.

Figure 20. The comparison of a correctly predicted successful case from the test set (left) and a similar case from the training set (right)
Similar results were found for correctly predicted failures from the test set. Figure 22 shows two unsuccessful problem-solving cases from the training and test set respectively. Both students attempted various non-existing solutions and only one or no existing path. Also, both of them did not use the calculation function to obtain and evaluate possible solutions. They were more likely to fail to solve the problem, and this pattern was captured and shown by the similar convolution results with a highly weighted filter in the convolutional layer as shown in Figure 23.
Figure 22. The comparison of a correctly predicted failure case from the test set (left) and a similar case from the training set (right)

Figure 23. Visualization of the Results on Applying a Filter Carrying Heavy Weights to both Students’ Action Frequency – Time Series Visualization from Figure 22

It is even more interesting to look at some misclassified problem-solving processes. Figure 24 shows a false positive case which CNN output predicted successfulness while the student actually failed. Judging from the visualization, this student tried several routes either exist or not exist and calculated prepared routes three times and two of them exist. From the CT angle, this student showed some ability in
representing solutions and processing data logically, which resembles some successful cases from the trained CNN and was thus classified as successful.

Figure 24. A predicted successful but failed case

A closer look at the significant milestones along the problem-solving process (Figure 25) shows that this student seemed to start from randomly searching for solutions by trying and calculating a route does not exist to a more systematic way of calculating and comparing existing routes. This student even tried to submit an intermediate answer once during the process but canceled and then tried to examine one more route. Although he did not try the correct path for this item and provided a wrong answer, he showed some ability in representing possible solutions and comparing collected data to help make a decision as demonstrated by the visualization in Figure 24.
Most cases from the false negative category where CNN predicted unsuccessful while the student made it have visualizations that resemble the right image in Figure 26. This kind of figure shows that a student simply dragged and dropped a connecting stop to the designated area and then submitted the solution. Some students who adopted this problem-solving path might have thought through possible solutions in their mind and picked the correct choice, so their thinking processes could not be recorded as the log file. However, others might have just randomly chosen their options with a 33% guessing rate. The visualization could not differentiate those two kinds of problem-solving path and guessing rate compounded this issue. This issue is also likely a reason for the discrimination problem for item 4.2 observed in the ICC in the psychometrics analysis section that students with higher estimated ability performed worse than those who received lower ability scores. In fact, the ICC of item 4.2 is similar to item 5.3b for the reason that students from the high ability group performed less well. Both cases are likely to be affected by the relatively high guessing rate.

Figure 25. Milestones of the problem-solving process of the false positive case
Figure 26. Visualization of problem-solving processes where a student took minimal actions to either solve the problem correctly (left) or incorrectly (right)

The lack of information from the visualization could cause errors when the trained CNN is applied for classification purpose. In fact, 66 cases out of 171 responses have merely 2 or 3 actions recorded, which corresponds to students who just submitted without answering or dragged and dropped a connecting stop once and then submitted. This lack of informative patterns as plotted in their action frequency – time series image eliminated the possibility to gain a higher test accuracy by training the CNN model.

4.3.3 Item 5.2 analysis

Item 5.2 is a question in the context of doing laundry (Figure 27). Students are given six piles of loads and two washing machines. Each washing machine holds at most one full load. Pile 1 and pile 5 are two full loads, and other four piles are half loads. Also, pile 1-4 are dark loads and pile 4, and pile 6 are light loads that could not be mixed. Students need to provide a loading plan that follows the rule and uses the fewest cycles possible.
Table 12 lists some observable actions that could be extracted from process data and how they might contain information regarding CT skills included in our framework. Unlike many observable actions for item 4.2 that multiple CT components intertwined with each other, three actions in table 12 are more closely related to one dimension from the framework than others. Those actions are: (1) mixing dark and light piles in one washing cycle, indicating a failure to represent problems and solutions successfully; (2) adding more than one load to one washing cycle, meaning incorrect manipulation on data; (3) using more washing cycles than needed, showing inefficiency in solution.
Table 12

*Observable Actions and Their Relationships to CT (Item 5.2)*

<table>
<thead>
<tr>
<th>Observable Actions</th>
<th>Possible CT Categories Related</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back to the laundry basket</td>
<td>Data, Representation, Algorithmic Thinking, Efficiency</td>
</tr>
<tr>
<td>Mixed color</td>
<td>Representation</td>
</tr>
<tr>
<td>Put more than one load into a machine</td>
<td>Data</td>
</tr>
<tr>
<td>Use more washing cycle than actually needed</td>
<td>Efficiency</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Figure 28 shows two students’ Action Frequency – Time Series visualization from their pretest responses. As is the case for item 4.2, from all the 171 pretest responses on this item, 119 (70%) responses were randomly chosen to form the training set and the other 52 (30%) testing set.

*Figure 28. Examples of Student Action Frequency - Time Series Visualization for Item 5.2*

Figure 29 shows the relationship between training error and test error during the training process. Early stopping is imposed to maintain considerable model complexity as
well as classifying the testing set with reasonable accuracy. Item 5.2 is a difficult item that only 35 (20.5%) students answered it correctly.

Figure 29. Training Process of Pretest Data

The error rate on the test set when training stops is 3.8% (96.2% accuracy). In other words, for all the 52 instances from the test set, only 2 cases were incorrectly classified. Table 13 shows the confusion matrix of the prediction and the two misclassified cases are both false negative.

Table 13
Confusion matrix of the prediction based on the test set (item 5.2)

<table>
<thead>
<tr>
<th>Actual Value</th>
<th>Predicted Value</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Successful</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Successful</td>
<td>9</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Unsuccessful</td>
<td>0</td>
<td>41</td>
<td></td>
</tr>
</tbody>
</table>
Table 14 listed some examples of both incorrect and correct responses with both problem-solving process visualization and the convolution results of the visualization with a filter connected to later layers with high weights. These visualizations go a long way in explaining why the classification accuracy is high for this item. In general, students who correctly solved this problem continually made decisions and rarely acted against the rule as opposed to those who failed. And this lump of correct actions within a specified period could be "spotted" by CNN as a salient feature and then assign considerable weights to decide the classification results.

Table 14

*Examples of both incorrect and correct cases and their corresponding feature map under a filter in the convolutional layer*

<table>
<thead>
<tr>
<th>Incorrect Examples</th>
<th>Feature map under a filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem-solving process</td>
<td>Feature map under a filter</td>
</tr>
<tr>
<td>Action Frequency on Time Series</td>
<td>Action Frequency on Time Series</td>
</tr>
<tr>
<td>0 1 2 3 4 6 7 8 9 10 11 12</td>
<td>0 2 4 6 8 10 12</td>
</tr>
<tr>
<td>0 1 2 3 4 6 7 8 9 10 11 12</td>
<td>0 2 4 6 8 10 12</td>
</tr>
</tbody>
</table>
Correct Examples

<table>
<thead>
<tr>
<th>Problem-solving process</th>
<th>Feature map under a filter</th>
</tr>
</thead>
</table>

Action Frequency on Time Series

Feature map under a filter
Inspecting at the two false negative cases is useful to infer why our trained CNN predicted failure. Figure 30 shows the visualized problem-solving process of the first false negative case. This student followed the rule that color mix and overweight are not allowed but made two inefficient decisions.

*Figure 30. False negative case 1*
Table 15 listed three stages of the problem-solving process for this student. In Stage 1, this student correctly put P1 and P5 into the first two cycles of machine 1. However, he then put P2 and P3 into the third washing cycle of machine 1, while a more efficient solution is to put both piles into the first washing cycle of machine 2. He seemed to realize this problem and corrected it in stage 3. But this process is treated by CNN as an unsuccessful case since this process is different from most other correct cases as shown in Table 14.

Table 15

*Stages of the problem-solving process for false negative case 1*

<table>
<thead>
<tr>
<th>Stage 1</th>
<th>Stage 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram of Stage 1" /></td>
<td><img src="image2.png" alt="Diagram of Stage 2" /></td>
</tr>
</tbody>
</table>
Figure 31 shows the second false negative case. The student moved piles a lot judging from the frequency of actions and made a few mistakes. This image resembles incorrect cases as shown in table 14 and our trained CNN then predicted a failure.

Table 16 gives the significant stages of how this student solved the problem. He did an excellent job at the end of stage 1 to put all dark piles obeying all rules but began to make a less efficient arrangement by loading P4 and P6 to cycle 3 of machine 1 (stage 2). He then realized the problem and made some complex modifications to solve this
problem (stage 3 and 4) correctly. All the actions against the rules were intermediate. However, they were represented in the visualization and caused the trained CNN to mislabel it.

Table 16

*Stages of the problem-solving process for false negative case 2*
Chapter 5

SUMMARY, CONCLUSION, AND FUTURE DIRECTIONS

5.1 Summary

This study is part of a larger design study that iteratively improves both a robotics programming curriculum and an assessment instrument evaluating students’ learning of CT: data collected and analyzed, findings discovered, conclusions drawn, and lessons learned could inform the future rounds of iteration. This paper put its focus on the instrument used in the study. To be more specific, three research questions guided the work of this study. Those questions are:

1. What are the psychometric properties of our CT test?
2. What are the differences between students’ performance from pretest to posttest?
3. How does test-takers’ problem-solving process help to understand their CT effectively and efficiently?

From several analyses based on data collected from 5th grade students of a collaborating elementary school, the instrument administered as both pretest and posttest showed satisfactory psychometric properties: different testlets within the instrument seemed to be assessing the same latent ability (CT in this case); item difficulty were widely distributed, and the range matched the stretch of students' estimated ability level; most items showed competency in discriminating students from higher ability groups from those from lower ability groups. Students did not improve much from pretest to
posttest as reflected by score increase and possible reasons were proposed and discussed in the results chapter. A test-taker's utilization of a CT skill or combination of CT skills in solving a problem as logged by computer systems has specific patterns that could be represented in some way and captured by a carefully trained neural network. And the extracted features could be used to predict a student's successfulness in solving that problem with certain accuracy in no time and possibly extend to other classification tasks.

5.2 Conclusion

This study first explored the use of IRT models to analyze a CT instrument. Although some CT instruments were applied to measure CT of students from different age groups, those studies stressed more on how students perform in terms of a score and implications drawn from the score, or comparison and association of the score to other possibly related test scores (González, 2015; Gouws, Bradshaw, & Wentworth, 2013; Lee, Lin, & Lin, 2014). In contrast, this study contributed to current literature of using instrument items to assess CT, in light of IRT methods, by: (1) inspecting if items of an instrument developed from a CT framework consistently measures the same construct and obtaining reliability measure; (2) displaying evidence of the instrument in discriminating students from different ability groups; (3) introducing a probabilistic model of a student answer to a particular question so the item difficulty is not dependent upon the ability level of students. IRT analysis results suggested that items within the instrument are reliably testing the same construct. Most items showed satisfactory goodness-of-fit level that estimated item parameters matched observed data well, which testified the use of the Rasch testlet model. Items did not pass the chi-square goodness-
of-fit test were those with the highest estimated difficulty. This observation, together with what the ICC for those items suggested, revealed that the discrimination coefficient for those items was higher than what the Rasch model assumes. Those difficult items needed to be considered to change to reach a more appropriate difficulty level as a whole that test-takers of different ability level could be more clearly differentiated. Guessing rate for multiple-choice items with three items or less could affect goodness-of-fit significantly, which suggested that such items needed to be avoided or modified. The Wright Map visualization indicated that in general, the estimated difficulty of all items match student estimated ability distribution well, which shows the feasibility of using such an instrument to assess CT of 5th graders appropriately. Although the administration of the same instrument as posttest did not show significant improvement from the pretest on the students' part, it could be due to multiple reasons. First, students did not improve regarding CT after the intervention. This could be true since every student cost only 45-50 minutes every two weeks and around 8 hours in total for the whole semester on the curriculum. Second, students were not taking this course for a grade, and such low-stake test might be treated without seriousness, and this situation was further worsened by the lengthiness of descriptions of context and question of some items as well as the overlap of administration of the posttest with students' standard testing period. More interactive and motivational features could be added to the current assessment to ease such adverse effect for test-takers. Possible questions exist regarding the validity of the instrument itself could affect the pre/posttest comparison from which null result was obtained. For example, CT dimensions often inextricably intertwined in many items that could induce difficulties during the test from the content and substantive perspective. Students
probably take distinct cognitive processes in solving problems; since different paths entail unique combinations of CT, intended contents and thinking processes of an item could not be guaranteed. Newly created items and scoring rubrics, although iteratively discussed and agreed upon by our eclectic team, could further complicate the issue as mentioned earlier. The scoring of some items could also incur problems concerning the structural aspect of construct validity. For instance, the binary scoring of multiple-choice items and scoring of open-ended questions with three rating scales might not be ideal for corresponding items which might test the application of multiple CT dimensions as well as various thinking processes that require more well-thought scoring rubrics. In sum, to tackle such possible problems and better establish the construct validity of the instrument, more data need to be collected and analyzed, and problem-solving process data are particularly useful for inspecting content coverage, identifying cognitive processes involved, and creating more comprehensive rubrics.

CT instruments consist of mainly multiple-choice questions are easy and efficient to administer, scale and scored. However, it is less informative when it comes to CT given its problem-solving nature: the problem-solving process could not be reflected merely by the answer choice a student provides. Another frequently used method in assessing CT is to teach students certain programming languages and then assign programming tasks to them to complete. Through the analysis of their programming artifacts, certain patterns a student leverages his CT skills in producing such programs could emerge (Bers et al., 2014; Brennan & Resnick, 2012; Werner et al., 2012). Those studies generated an in-depth understanding of how CT factored into student-generated programming products. However, given the complexity of currently existing CT
frameworks (Brennan & Resnic, 2012; Chen et al., 2017; CSTA, 2011; Perković et al., 2010; Sengupta et al., 2013), human intelligence is needed in coding those artifacts. As a result, such analysis costs more time and effort than that of the instrument approach. To tackle this issue, Moreno-León, Robles, and Román-González (2015) explored the possibility of automating the analysis of Scratch projects produced by students to help them learn CT. Nevertheless, another potential problem of such research still exists, that is, CT assessment relies on specific programming languages. Of course, CT and programming are closely related, but they are not equal. In fact, CT emphasizes the transfer of problem-solving to other problems (CSTA, 2011). That means students need to abstract away some procedures or rules specific to a programming language to solve similar problems in a broader sense. It is not to say learning one programming language is not helpful to foster CT, but this approach renders it difficult to find a benchmark for student's CT from the very beginning if students generally possess little or no programming experience. This is particularly the case when teaching elementary level students to learn CT. Different opinions exist and will probably continue to define what CT is, but problem-solving nature of CT is admittedly acknowledged. A way to understand and assess how CT is applied is to look at a student's problem-solving process. However, it seems more demanding to analyze such a problem-solving process than a student's programming product, not mentioning the difficulty to build a system that could log actions happen during the problem-solving process. Started from the purposes of assessing CT in an available form to 5th graders and an efficient manner, the instrument used in this study was initially a paper-and-pencil based version. And it is soon converted to a web-based instrument for easiness of administration and scoring.
Some items were then modified to be interactive, and logging functionality was added to record a test-taker's activity during the problem-solving process because it is at the core of CT. To make use of the log files to gain insight into a student's CT while keeping scoring of such an instrument efficient enough, learning analytics approaches were explored in this study to take advantage of powerful storage and computational power offered by modern technologies. This study demonstrated that the problem-solving process recorded by the instrument could be visualized and analyzed automatically with the help of a CNN. A well-trained CNN could be used to predict a student's successfulness in solving a problem based on process data. And by inspecting the adjusted parameters of the trained model, how CNN perceived the features on an action frequency – time series map could be inferred, and patterns that heavily affected later layers in deciding the final output of CNN could emerge. In the analysis of item 4.2, successful problem-solvers' orderly iteration between organizing new path and testing the path by calculation, which demonstrated their CT ability to represent routes from information given in the question and compare data from the calculation, were captured by the trained CNN as strong evidence of successes. In contrast, failed problem-solvers' inability to obtain information from the question to represent existing routes and lack of skills in data processing were also captured by the trained CNN and perceived as evidence of possible failures. CNN was also useful to identify some special cases by presenting misclassified problem-solving processes. In the item 4.2 analysis, a false positive case student showed certain CT ability in forming existing paths and comparing calculated data but finally failed. This example demonstrated that looking at a student's problem-solving process is more informative than merely assigning a score based on his
final answer. However, CNN was not able to accurately identify students who succeeded from those who failed if both took a minimal number of actions. Those confusions were caused by the uninformativeness attributes to the relatively simple solution space: students had a 33% guessing rate by randomly dragging and dropping a stop. This problem could be addressed by adding more possible routes and create a more complex solution by manipulating data. Compared with item 4.2, item 5.2 is tidier in that the solution of it requires more actions so the action frequency – time series map could represent more information to feed the CNN, certain student actions are majorly related to only one CT component from our framework, which makes it easy to infer the meaning of a student's action, and students with good CT solve this problem straightforwardly without any violations, so the paths show highly identical patterns. Those features were favorable to train a CNN model to spot possible patterns. The testing accuracy reached 96.2% indeed. The only two misclassified cases were all false negative cases since both students violated efficiency rule during their problem-solving process.

In sum, effectiveness and efficiency could not be easily reached when assessing a student CT. However, with the help of computer systems and learning analytics methods, our instrument showed the ability to achieve both. Also, it is difficult to create an item that is readily discriminating, possesses appropriate difficulty, and penetrating enough to reveal the ability it intends to measure. However, in light of IRT and learning analytics analyses, its potential problems could be identified and fixed.
5.3 Limitations and Future Directions

There are some limitations exist in this study that successful addressing of those issues serves as valuable guidance for our future rounds of improvement in both the robotics curriculum and CT instrument.

When it comes to the psychometric analysis, some items showed higher discrimination power than the Rasch model assumes. And yet some other items were strongly influenced by guessing success, which is also not considered in the Rasch model. Therefore, imposing the Rasch model analysis was inappropriate for those questions. A modified model with more parameters that take discrimination and guessing into consideration might produce better fitting results for the observed data.

Another possible problem of psychometric analysis is that although it could estimate item parameters based on collected data, it sometimes is still obscure to take advantage of them to improve the item. For example, a psychometric analysis may tell us that an item is difficult for most of the students given its high estimated difficulty level. But why so? And what can we do to tweak it as we need? This is an essential problem since it is directly related to substantive validity as discussed in section 3.5 and this is where process data could help. Process data in this study was used only to perform learning analytics analysis so far in this study and revealed promising results to infer some characteristics of the item (guessing rate of item 4.2, for instance). A more systematic analysis of the process data has the potential to enrich psychometric analysis result.
CNN demonstrated its utility in capturing visualized patterns based on students' problem-solving process and channeling such features in predicting their successfulness. However, the visualization of a student's problem-solving process on a 2D image necessitates the sacrifice of some information in such problem-solving process. In this study, time-series is one axis and action frequency within each time interval is the another, so the order of transition and corresponding time between two actions were omitted. Other data visualization methods might be needed to represent as much information as possible in the future to elevate the accuracy of classification by neural networks.

Another problem of our analysis is that probability was not considered in our CNN model. The output of the current model is either successful or unsuccessful in solving the problem. However, it makes more sense to consider an output with the probability that a student has the potential to succeed in solving the problem with a probability. This is helpful to identify some boundary cases and thus avoid false classification in some cases. Also, partial credit could be considered to assign to a student based on the probability to gain a more accurate estimation of his ability in terms of scores. Besides, the output of the CNN model is not necessarily successfulness in solving a problem. It could be other classes that are of interests. For example, we can train a CNN model to output the percentage of different CT skills used during the problem-solving process for a student. Of course, this needs more training data and content knowledge on what actions in the problem-solving process are considered related to which component in the CT framework. But this is definitely a future direction that I can explore. A trained CNN could also be applied to identify student pattern changes from pretest to posttest.
Regarding curriculum implementation for future study, since the comparison between the pretest and posttest results showed no significant improvement on students' CT score and ineffectiveness of the intervention is a possible reason, we might need to revise our curriculum further to fit the needs of students from our collaborating schools in fostering CT. Our previous rounds of implementation were either semester long with 45 minutes to 1 hour per week or 2 hours per session per day for five consecutive days. Students did have enough time to follow a teacher's step by step instruction and then practice on their own. However, for the current run, time is tight, and a student could not get enough practice before moving into a new chapter. As a result, we may need to arrange our curriculum materials according to the unique situation to strike a balance between teacher instruction and student practice if the school continually adopts such a schedule.

In conclusion, students living in this big data era are supposed to manage CT in order to collaborate with machines in the future, and at the same time, big data storage and analytics methods offered by computers could be utilized to assist them in learning better.
REFERENCES


Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. *Acm Inroads, 2*(1), 48-54.


APPENDIX A

Item Screenshots

Problem set 1

A cooking robot only recognizes special codes in preparing your recipe. And the code for different ingredients are listed below:

CC=Chocolate Chips
LE=Large Eggs
BT=Butter
SG=Sugar
FL=Flour
BS=Baking Soda
ST=Salt
CN=Chopped Nuts
VE=Vanilla Extract

To ask the robot to begin to prepare, we need to specify the quantity and the ingredients. The unit here is quarts for all items. For example, both of the following codes: "2CC0.75FL" and "0.75FL2CC" represent 2 quarts of chocolate chips and 0.75 quarts of flour.

1. What does “2CC3FL0.5BS” mean?
   - 2 quarts of flour, 3 quarts of baking soda, 0.5 quarts of chocolate chips.
   - 2 quarts of chocolate chips, 3 quarts of flour, 0.5 quarts of baking soda.
   - 2 quarts of chocolate chips, 3 quarts of salt, 0.5 quarts of baking soda.
   - 2 quarts of flour, 3 quarts of chocolate chips, 0.5 quarts of baking soda.

2. Which of the following codes instruct the robot to prepare 3 quarts of butter, 2.5 quarts of flour, 0.5 quarts of sugar, and 0.05 quarts of vanilla extract?
   - 0.05VE2.5FL3BT0.5SG
   - 0.5SG3BT2.5FL0.05VE
   - 3BT2.5FL0.5SG0.05VE
   - All of the above
A research team is developing a robotic arm that can draw shapes according to input commands. Here is the format of the command that tells the arm to draw a line:

\[ \text{\textasciitilde} \text{MOVE } n, \text{ direction.} \]

“\( n \)” is a variable that can be any whole number between 1 and 10 (steps); “direction” can be any of the following choices:

- F (Forward)
- B (Backward)
- L (Left)
- R (Right)

For example, the following command “\( \text{\textasciitilde} \text{MOVE } 2, L \)” means that the pen will draw a line of 2 steps to the left.

1. What is the meaning of the following command?
   - MOVE 3, B.
     - Move the pen forward to draw a line of 3 steps.
     - Move the pen backward to draw a line of 3 steps.
     - Move the pen left to draw a line of 3 steps.
     - Move the pen right to draw a line of 3 steps.

2. Where is the pen head if the robotic arm finishes running the following commands:
   - MOVE 5, F.
   - MOVE 6, B.
   - MOVE 10, L.
   - MOVE 10, R.
     - The original location.
     - 5 steps forward to the original location.
     - 5 steps left to the original location.
     - 10 steps right to the original location.

3. Now you have a new command “\( \text{\textasciitilde} \text{REPEAT } x \)” The variable \( x \) represents the number of times the previous command will be repeated. What would be the result of running the following code?

\[ \text{\textasciitilde} \text{MOVE } 1, F. \]
   - The robotic arm will draw a line of 1 step forward.
\[ \text{\textasciitilde} \text{REPEAT } 2. \]
   - The robotic arm will draw a line of 2 step forward.
   - The robotic arm will draw a line of 3 step forward.
   - The robotic arm will draw a line of 4 step forward.
4. Drag and drop the commands in the left column below into the "steps" box in an order to make the robotic arm produce a cross on the paper. You do **NOT** need to use all the commands. The dimensions, directions, and starting points of the cross are specified in the picture below.

>`->MOVE n, direction.  (n = number of steps moved, direction = F, B, L, or R)

Now please drag and drop the commands in the left column below into the "steps" box in an order to make the robotic arm produce a cross on the paper.

<table>
<thead>
<tr>
<th>Commands</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-&gt;MOVE 1, R. </code></td>
<td></td>
</tr>
<tr>
<td><code>-&gt;MOVE 1, R. </code></td>
<td></td>
</tr>
<tr>
<td><code>-&gt;MOVE 1, F. </code></td>
<td></td>
</tr>
<tr>
<td><code>-&gt;MOVE 1, F. </code></td>
<td></td>
</tr>
<tr>
<td><code>-&gt;MOVE 1, B. </code></td>
<td></td>
</tr>
<tr>
<td><code>-&gt;MOVE 1, B. </code></td>
<td></td>
</tr>
<tr>
<td><code>-&gt;MOVE 1, L. </code></td>
<td></td>
</tr>
<tr>
<td><code>-&gt;MOVE 1, L. </code></td>
<td></td>
</tr>
</tbody>
</table>
5. Programming this robotic arm to draw a triangle is challenging. Please show the path/pattern that you think resembles a triangle the most by clicking and then change the color of the squares below (you can re-click to uncolor it). The flag \( \text{\textbullet} \) stands for the initial location of the pen head, please start drawing from here.

**Video Instruction**

Your pattern:

\[
\begin{align*}
& F \\
& L \\
& B \\
& R
\end{align*}
\]

Now please write a code sequence using the following 2 commands to draw the pattern above.

- `->MOVE n, direction.` \( (n = \text{number of steps moved; direction = F, B, L, or R}) \)
- `->REPEAT x [start; end].` \( (x = \text{times of previous command repeated in the same code box, [start; end]} = \text{specify the start and the end number of previous commands that you want to repeat}) \)

For example, `REPEAT 3 [2:5]`, stands for repeating step 2 to step 5 for 3 times.

Please show your code sequence:

- [Add Move]
- [Add Repeat]
- [Delete Row]
A team of computer scientists are programming a humanoid robot. Here are some codes they are experimenting with.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start.</td>
<td>(This code tells the robot to start, ready for action.)</td>
</tr>
<tr>
<td>End.</td>
<td>(This code tells the robot to end program and rest.)</td>
</tr>
<tr>
<td>Say ...</td>
<td>(This code tells the robot to say something the programmer inputs.)</td>
</tr>
<tr>
<td>Walk ?</td>
<td>(This code tells the robot to walk forward for the number of steps the programmer specifies.)</td>
</tr>
<tr>
<td>Turn L.</td>
<td>(This code tells the robot to turn Left.)</td>
</tr>
<tr>
<td>Turn R.</td>
<td>(This code tells the robot to turn Right.)</td>
</tr>
</tbody>
</table>

1. What does the following code sequence tell the robot to do?

   - Start
   - Turn L
   - Walk 5

   - Turn left and then walk forward for 5 steps.
   - Turn left and then walk in any direction for 5 steps.
   - Walk forward for 5 steps and then turn left.
   - Walk in any direction for 5 steps and then turn left.
2. A team member wrote the following code:

-> Start.
-> Walk 30.
-> Walk 30.
-> Turn L.
-> Walk 30.
-> End.

Which will be the robot's moving path if we run the code?

- [ ]
- [ ]
- [ ]
3. The team is experimenting with having the robot do multiple things at the same time using a new command [And]. For example, the following code makes the robot walk while saying "Hello" at the same time.

`->Start.  
->Walk 5, [And]  
->Say "Hello"  
->End. 

Below are the codes for your reference:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-&gt;Start.  </code></td>
<td>(This code tells the robot to start, ready for action.)</td>
</tr>
<tr>
<td><code>-&gt;End. </code></td>
<td>(This code tells the robot to end program and rest.)</td>
</tr>
<tr>
<td><code>-&gt;Say .... </code></td>
<td>(This code tells the robot to say something the programmer inputs).</td>
</tr>
<tr>
<td><code>-&gt;Walk ?.</code></td>
<td>(This code tells the robot to walk forward for the number of steps the programmer specifies).</td>
</tr>
<tr>
<td><code>-&gt;Turn L.</code></td>
<td>(This code tells the robot to turn Left.)</td>
</tr>
<tr>
<td><code>-&gt;Turn R. </code></td>
<td>(This code tells the robot to turn Right.)</td>
</tr>
</tbody>
</table>

Write a code sequence so that the robot walks the path shown below, but also says "I am turning" while turning.

![Diagram of a robot path with 10 steps moving in a zigzag pattern](image)

Please show your code sequence (one command for each line):

1. `->`
4. Write a code sequence to remotely control the robot 🦀 in the picture below to move to the square with 🟢 and then move to the destination 🟠. When it is moving, the robot has to avoid rocks 🌫. When it reaches the square with 🦀, the robot has to stop and say "I am half way." Initially, the robot is facing the direction as indicated by the red arrow ⬅️. The distance between two neighboring squares is 2 steps.

Please show roughly the path/pattern that you plan to draw by clicking and then change the color of the squares below (you can re-click to uncolor it).
Problem set 4

1. You are trying to buy an airplane ticket for your trip from Miami to LA. The airline website gives you a map (shown below) that tells you:

- What flights are available
- How much each flight costs
- How long each flight takes
- How long the waiting time at each airport

The web site also gives you a “flight calculator” (under the map) that tells you how much your trip will cost and how long it will take. Using the flight calculator to help you, pick the path that will save you the most money.

- $100, 65 mins
- $120, 80 mins
- $105, 70 mins
- $110, 70 mins
- $60, 55 mins
- $40, 25 mins

Calculate this route!
2. If you want to save time instead of money, which path would you like to choose?
Problem set 5

You are asked to do all of the laundry in your household. You have one and a half piles of dark clothes (P1 and P2). Your sibling has one half pile of dark clothes and one half pile of light clothes (P3 and P4). Your parents have one pile of dark clothes and half a pile of light clothes (P5 and P6).

1. There are two important rules about doing laundry in your house:
   * You cannot mix dark and light clothes in the same cycle.
   * The machine can hold at most one pile of laundry for each cycle.

Your sibling put together a plan for doing all of the laundry and says that this follows the rules and will also use as few laundry cycles as possible. Do you agree that your sibling’s plan follows the rules and uses as few laundry cycles as possible?

Do you agree with your sibling’s plan?
- Yes, I agree with this plan.
- No, this plan doesn’t follow the rules. I have a better plan.
- No, this plan doesn’t use the fewest cycles possible. I have a better plan.
- No, this plan doesn’t follow the rules and it also doesn’t use the fewest cycles possible. I have a better plan.
Please show us your plan (please drag and drop the loads in the box below to the corresponding cycle):

[Video Instruction]
2. What if you have 2 washing machines that can run simultaneously. All other conditions still apply: the dark and light clothes cannot be mixed; each machine holds one pile of laundry at a time; use the fewest cycles possible. Fill out the loading plan table below.

Video Instruction

Your plan (please drag and drop the loads in the box below to the corresponding cycle):
3. You are tired of sorting out clothes and decide to design a "smart clothes feeder" that can feed your clothes automatically into the washing machine. Here is a flow chart you came up with. Answer the questions below:

![Flow chart image]

a. Below is the table which lists 12 pieces of cloth to be washed. If your "smart clothes feeder" starts to pick one piece of cloth in order, at which cloth number it will start washing? (Tip: you can hang your mouse over the table to see the total weight up to the specific item you put your mouse on.)

<table>
<thead>
<tr>
<th>Cloth</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>1.5</td>
<td>3</td>
<td>2</td>
<td>2.5</td>
<td>3.5</td>
<td>2</td>
<td>1.5</td>
<td>3</td>
<td>2.5</td>
<td>1.5</td>
<td>2</td>
<td>3.5</td>
</tr>
</tbody>
</table>

- Total weight: 0
  - 5
  - 9
  - 10
  - 11

b. Your friend points out that your flowchart does not consider sorting clothes based on color. If you are going to add a "color sorting" step based on your flowchart, which place below would you choose to add?

![Flow chart image]
APPENDIX B

ConQuest Code for Rasch Testlet Model

datafile final_score2.dat;
format responses 1-25;
set constraints=cases, warnings=no;
score (0.1)(0.1)(0.1) () () () items (1);
score (0.1)(0.1)(0.1) () () () items (2);
score (0.1)(0.1)(0.1) () () () items (3);
score (0.1)(0.1)(0.1) () () () items (4);
score (0.1)(0.1)(0.1) () () () items (5);
score (0.1)(0.1)(0.1) () () () items (6);
score (0.1)(0.1)(0.1) () () () items (7);
score (0.1)(0.1)(0.1) () () () items (8);
score (0.1)(0.1)(0.1) () () () items (9);
score (0.1)(0.1)(0.1) () () () items (10);
score (0.1)(0.1)(0.1) () () () items (11);
score (0.1)(0.1)(0.1) () () () items (12);
score (0.1)(0.1)(0.1) () () () items (13);
score (0.1)(0.1)(0.1) () () () items (14);
score (0.1)(0.1)(0.1) () () () items (15);
score (0.1)(0.1)(0.1) () () () items (16);
score (0.1)(0.1)(0.1) () () () items (17);
score (0.1)(0.1)(0.1) () () () items (18);
score (0.1)(0.1)(0.1) () () () items (19);
score (0.1)(0.1)(0.1) () () () items (20);
score (0.1)(0.1)(0.1) () () () items (21);
score (0.1)(0.1)(0.1) () () () items (22);
score (0.1)(0.1)(0.1) () () () items (23);
score (0.1)(0.1)(0.1) () () () items (24);
score (0.1)(0.1)(0.1) () () () items (25);
model itemstep;
export parameters >> data_old.prm;
export reg_coefficients >> data_old.reg;
export covariance >> data_old.cov;
estimate fmethod=montecarlo, nodes=4000, iterations=600;
quit;
datafile final_score2.dat;
format responses 1-25:
set constraints=cases, warnings=no;
score (0.1)(0.1)(0.1) () () () items (1);
score (0.1)(0.1)(0.1) () () () items (2);
score (0.1)(0.1)(0.1) () () () items (3);
score (0.1)(0.1)(0.1) () () () items (4);
score (0.1)(0.1)(0.1) () () () items (5);
score (0.1)(0.1)(0.1) () () () items (6);
score (0.1,2) (0.1,2) () (0.1,2) () () () items (7);
score (0.1,2) (0.1,2) () (0.1,2) () () () items (8);
score (0.1,2) (0.1,2) () (0.1,2) () () () items (9);
score (0.1) (0.1) () (0.1) () () () items (10);
score (0.1)(0.1) () (0.1) () () () items (11);
score (0.1)(0.1) () (0.1) () () () items (12);
score (0.1,2) (0.1,2) () (0.1,2) () () () items (13);
score (0.1)(0.1) () (0.1) () () () items (14);
score (0.1,2) (0.1,2) () (0.1,2) () () () items (15);
score (0.1,2) (0.1,2) () (0.1,2) () () () items (16);
score (0.1)(0.1) () (0.1) () () () items (17);
score (0.1,2) (0.1,2) () (0.1,2) () () () items (18);
score (0.1,2) (0.1,2) () (0.1,2) () () () items (19);
score (0.1)(0.1) () (0.1) () () () items (20);
score (0.1)(0.1) () (0.1) () () () items (21);
score (0.1)(0.1) () (0.1) () () () items (22);
score (0.1)(0.1) () (0.1) () () () items (23);
score (0.1)(0.1) () (0.1) () () () items (24);
score (0.1)(0.1) () (0.1) () () () items (25);
model item=step:
import init_parameters << data_old.prm;
import init_reg_coefficients << data_old.reg;
import anchor_covariance<< data_old.cov;
estimate !method=montecarlo, nodes=4000, iterations=600;
show! estimate=latent >> result.shw;
show cases! estimates=eap >> result.eap;