Connecting computational thinking and mathematics: an examination of linguistic opportunities and challenges

by
Abbie Glickman

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Physics
(Honors Scholar)

Presented April 14, 2021
Commencement June 2022

AN ABSTRACT OF THE THESIS OF

Abbie Glickman for the degree of Honors Baccalaureate of Science in Physics presented on April 14, 2021. Title: Connecting computational thinking and mathematics: an examination of linguistic opportunities and challenges.

Abstract approved: _____

Rebekah Elliott

In this paper, I used a linguistic lens of translanguaging and semiotics to examine how K-12 mathematics teacher candidates (TCs) create meaning around computer science (CS) and mathematics concepts in a computational thinking (CT) setting. I defined the term "disciplinary repertoire" as a mirror to the concept of a linguistic repertoire. I addressed the following questions. 1) What are the linguistic affordances and challenges that K-12 mathematics TCs face as they engage in a CT setting? 2) How do the teacher candidates navigate those affordances and challenges: (a) With each other? and (b) Using tools they are provided by computational thinking modules? This paper focused on one pairing of mathematics TCs as they engaged in basic CS modules on a platform called BlockPy. From the data, two main themes arose: 1) the disciplinary repertoires of the TCs granted them both affordances and challenges; and 2) the TCs created meaning together by translanguaging and coordinating their disciplinary repertoires. These results indicate that it could be advantageous to explicitly acknowledge the disciplinary repertoires of TCs such that they have opportunities to foreground their knowledge and address potential challenges.

Key Words: computational thinking, computer science, mathematics, translanguaging, education

Corresponding e-mail address: glickmaa@oregonstate.edu

Connecting computational thinking and mathematics: an examination of linguistic opportunities and challenges

by
Abbie Glickman

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Physics
(Honors Scholar)

Presented April 14, 2021
Commencement June 2022

Honors Baccalaureate of Science in Physics project of Abbie Glickman presented on April 14, 2021.

APPROVED:

_____

Rebekah Elliott, Mentor, representing College of Education

_____

Jennifer Parham-Mocello, Committee Member, representing Department of Electrical Engineering and Computer Science

_____

Elise Lockwood, Committee Member, representing Department of Mathematics

_____

Cory Buxton, Committee Member, representing College of Education

_____

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, Honors College. My signature below authorizes release of my project to any reader upon request.

_____

Abbie Glickman, Author

Acknowledgements

I would like to thank my thesis mentor, Dr. Rebekah Elliott, for all of her guidance and support throughout this process. Each of our conversations have been an opportunity for me to learn and grow as a researcher. At every step she has shown enthusiasm for my understanding and found ways to introduce me to aspects of the field. I am lucky to have found such a kind and thought-provoking mentor.

I would also like to extend my thanks to the members of my committee, Dr. Elise Lockwood, Dr. Jennifer Parham-Mocello, and Dr. Cory Buxton. Each of them provided key guidance, especially in the formative stages of my process. I am grateful for their expertise and for the opportunities they've provided for me to learn more about education research.

I would also like to acknowledge the funding and support of the URSA Engage program, through which I started working on this project. My gratitude also goes out to the members of the research team that helped design the modules and collect the data that I used in this paper.

Lastly, I want to say thank you to the friends and family that have supported me throughout this process and beyond. I appreciate my dad, Steven Glickman, and grandmother, Gail Glickman, for taking the time to read through my thesis and offer their suggestions. I also want to thank my mom and step-dad, Rina and Jason Redrup, for caring so deeply about everything that I pursue. Thank you to Thomas Knudson who listened to all of my long explanations and excited tangents throughout this project. And to all my close friends who have been there for me, especially while working through this pandemic, I owe so much to you.

Table of Contents

Introduction

What is the issue at hand?

Computational thinking (CT) is increasingly important in K-12 educational settings. While it has direct ties to computer science (CS), CT is important across multiple STEM fields and academic disciplines. These connections are important, but not necessarily simple to address, especially when the aim is to leverage commonalities between CS and other academic fields. For instance, mathematics students learn algorithms in their classrooms, which is a central theme in CT. However, when they are asked to move this algorithmic thinking to CS settings, they encounter a unique set of opportunities and challenges.

Who is involved?

Specifically, there is value in addressing these opportunities and challenges in pre-service K-12 educators because if they are prepared to incorporate CT into their curriculum the payoff of their understanding will extend to their students (Yadav et al., 2017). With this in mind, this study examines the joint participation of K-12 mathematics teacher candidates (TCs) in CT modules. I investigate the ways that TCs construct meaning with one another and interact with mathematics and CS concepts in a computational setting. The data for this study are drawn from a larger study of five TC groups participating in a five-day CT module during their second graduate-level, mathematics methods course. The data I analyze are from one pairing of TCs during the first day of the module.

The research team from the broader study developed a set of basic computational learning modules on BlockPy. BlockPy is a block-based, as opposed to scripted, interface that integrates the coding language Python with a structure that removes some of the complexity of

its syntax. Within the five days, the TCs were introduced to concepts such as setting a variable, constructing a For loop, and designing code to draw basic shapes. Each day they were guided through the modules by the research team, responded to reflection questions in their small group, and participated in whole-group discussions that challenged them to draw connections between the computational elements and their mathematical background.

What is my research question or aim?

Through my analysis of the data, I identified several common themes – in particular the difficulty in choosing how to represent problems, as well as moments where the language the participants used to communicate with each other across disciplines presented challenges and opportunities. My aim is to provide a lens by which to view these themes and to examine specific examples of its application in the context of the research I described above. One of the ideas I will elaborate on throughout this text is the ways we can borrow translanguaging and mirror discussions around multilingual students as we approach the data in this research. Translanguaging is defined as "the development of a speaker's full linguistic repertoire without regard for the watchful adherence to the socially and politically defined boundaries of named (and usually national and state) languages" (Otheguy et al., 2015, p. 281). Multilingual students translanguage in classroom settings, pulling from elements of their linguistic repertoire as they engage in material and communicate with others. A linguistic repertoire is a person's set of linguistic devices used in meaning-making and communication. In this paper I will discuss linguistic repertoires and elaborate on what I will call "disciplinary repertoires". The disciplinary repertoire replicates the linguistic repertoire, in that it functions as the set of meaning-making and communicative devices specifically geared towards learning and problem solving within the

various academic disciplines. This includes how students build representations and how they interpret or use technical language within academic fields.

The discussion of linguistics and disciplinary repertoires is important to address in this research because it may help us recognize the affordances and challenges faced by the TCs in the CT setting. This leads me to my research questions:

1. What are the linguistic affordances and challenges that K-12 mathematics teacher candidates face as they engage in a computational thinking setting?
2. How do the teacher candidates navigate those affordances and challenges:
    a. With each other?
    b. Using tools they are provided by computational thinking modules?

Literature Review

The purpose of this literature review is to provide a strong backing to the theoretical lens that I would like to put forth about the way linguistics is incorporated into the academic disciplines of mathematics and CS. The three concepts I will examine are CT, mathematics as a language and translanguaging. They will link together in my theoretical lens because this research is examining how TCs interact with CT modules. I will introduce CT to show the importance of the way of thought especially in problem solving and conceptualizing scenarios, and how it is used in both CS and mathematics in sometimes overlapping ways. As we run the modules, we want to show the participants of our research that CS is accessible to them because the CT practices that they will implement are related to the work they already do in mathematics.

But how do we know what the participants in our research are understanding from these modules? That is where my theoretical lens comes into play. I believe that by observing the language the participants use as they interact with each other and the workspace, we can identify instances of where they are reasoning with the CT concepts and the ways that they are connecting what they learn to their background in mathematics. This will allow us to highlight places where opportunities exist to connect the ideas they are learning to the base knowledge they have already developed through mathematics or previous work in CS. Linguistically speaking, this mirrors the practice of translanguaging, where linguistic concepts are present across multiple named languages, and someone who is bilingual will draw from all of their linguistic background as they approach a conversation or communicative interaction (reading, writing, speech, etc.).

The following is how I plan to break down my literature review. I want to begin by defining CT, which we will find is a basis for CS as it is examined within the confines of this

research. Not only is it essential for CS, but CT also has applications in a number of other fields within and beyond STEM. I will explore theories posed by Wing (2006, 2014), Weintrop et al. (2016), and Denning (2009) with regard to these concepts. I will continue with this point by focusing on the overlaps between CT in mathematics and CS. These explanations will be broader and encompass more of the overlaps between the fields rather than between the specific linguistic elements, which will follow later in the review. The next major concepts I will consider are mathematics as a language and language in mathematics education. Through the works of Schleppegrell (2007, 2010), I will build the backing for viewing an academic discipline such as mathematics as a language. She poses the works of Pimm (1987) and Halliday (1978) among other academic papers to justify the lens. These are justifications that I plan to build upon as I develop my own lens for viewing both mathematics and CS as disciplines with linguistic components. To elaborate on the linguistic elements we see in classroom settings and beyond, I will use works of Erwig (2017), Vogel et al. (2019), and Schleppegrell (2007, 2010) to explain multi-semiotic systems and social semiotics. These are important when we examine the learning of TCs in the classroom as they collaborated with one another during their participation in CT modules. As a critical piece to my theoretical lens, I will introduce translanguaging, which is a concept that evolved from, and replaces, code-switching (posed by Schleppegrell (2010)) as a way to describe multilingual learners engaging linguistically with content. This will draw upon the works of Vogel et al. (2019) and Otheguy et al. (2015), which clarify the definition of translanguaging and how it is used in educational settings.

Defining computational thinking

First, I will define CT and explain its importance in the academic world, specifically in K-12 learning. The reason I am taking the time to discuss this broader way of thinking is because

we will find that it is useful in problem-solving situations, as well as conceptualizing various scenarios or ideas. CT is becoming central to K-12 STEM education. Wing (2008) argues that in order for CT to be fully realized at all the university and societal level, we need to develop a strong foundation at an early age (Wing, 2008). By introducing these "tools and practices into mathematics and science classrooms" we can give "learners a more realistic view of what these fields are" and better prepare "students for pursuing careers in these disciplines" (Weintrop et al., 2016). Another motivation is to use computational practices as a way to "deepen learning of mathematics and science content" (Weintrop et al., 2016). We are seeing that CT extends beyond the reaches of CS applications and there are arguments favoring the use of CT across a broader array of disciplines to meet educational goals. Weintrop et al. discuss that there are a number of definitions of CT, but none are universally agreed upon. As cited in Weintrop et al., Wing (2006) defines computational thinking as follows: "computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science." This definition closely links CT to CS, but I will also include Wing's fundamental definition in her 2014 paper applies to a wider array of academic fields. In this paper she defines CT as a "thought process involved in formulating a problem and expressing its solution(s) in such a way that a computer (human or machine) can effectively carry [them] out" (Wing, 2014, p. 1).

Within this scope of solutions that can be computed by a human or machine, we carry the skills of abstraction, algorithmic thinking, developing models, and designing efficient solutions. While these skills are traditionally associated with CS, Wing would argue that CT is one of the pillars of science (Wing, 2014). With respect to the research our team is conducting, this framework provides context to why our work is relevant beyond the modules we designed. The

TCs can bring CT techniques such as abstraction and algorithmic approaches to their classrooms without the direct link to CS. In the K-12 setting, research on CT has addressed the cross-disciplinary approach, which entails curriculum guiding students to the use of CT practices, as well as the approach to use computer simulations and tools to enhance [STEM focused] learning (Weintrop et al., 2016).

Across mathematics and science practices, there are four major categories that Weintrop et al. (2016) define, and I will highlight the one that is most relevant to this research: computational problem-solving practices. This frame centers around the use of programming, as well as understanding problems through the lens of the tools students use. This reaches beyond the use of models, to the general use of coding and computational tools for problem solving. These ideas become increasingly useful when we turn to the idea of abstractions, where ideas are generalized and reused. In the context of this research, the TCs are tasked with solving specific problems, but they engage in discussions on abstraction and automation where these tools may be put to use.

Overlaps of computer science, computational thinking, and mathematics

I want to take the time to make a distinction about the way I am summarizing the scope of CT and CS. In order to remain clear in my discourse, I will keep all discussion within reference to Figure 1, which depicts the overlaps among CT, CS, and mathematics.

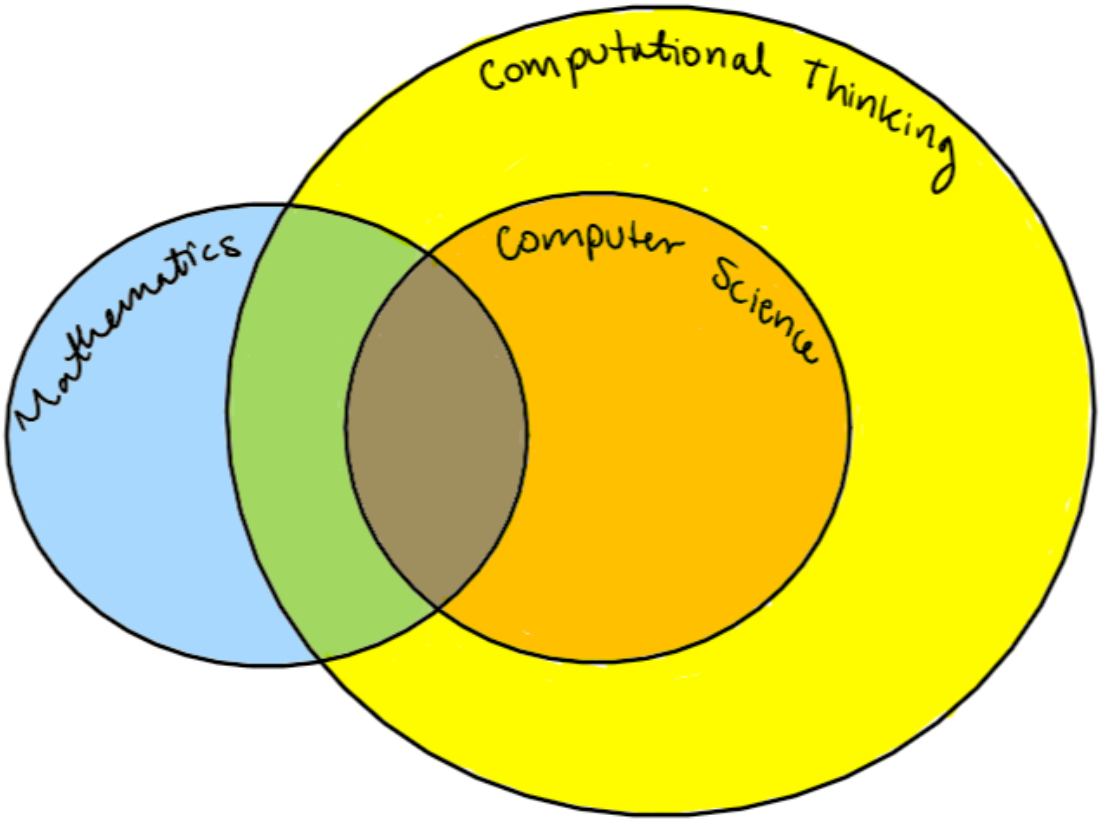

Figure 1. Venn diagram of the overlaps among mathematics, computer science, and computational thinking.

I designed the Venn diagram as a tool to describe the way one might look at each of the fields and their connection to CT. In order to express a representation of a problem or scenario in a way that a computer could process it, a computer scientist will use CT. Acknowledging that this is not all that a computer scientist might do, under broad terms, I felt comfortable fitting the entire CS category within the CT bubble. Then we have the mathematics category. In mathematics, there are a variety of ways to express a problem and its solution. While some require abstraction or an algorithmic approach, a person tasked with solving a problem could afford some level of vagueness with regard to their work. Then there are cases where one must be explicit with the work and perhaps make it generalizable to a broader scenario.

Let's take a look at a problem where two different representations deliver the same answer.

A kid sets up a lemonade stands and spend $20 to buy the supplies. They then sell lemonade at $2 a cup. How many cups do they need to sell to break even?

In our first solution, we will not pay attention to how explicit we are being. We know that there is an immediate deficit of $20, so to break even we need to earn back those $20. Since we earn $2 per cup that we sell, we can set this up as a simple algebra problem. Let *x* represent our unknown number of cups sold to break even:

$$2x = 20$$

$$x = \frac{20}{2}$$

$$x = 10$$

So, our answer is that we must sell 10 cups to earn $20 and break even. I will briefly point out that we immediately dropped units and that the solution was relatively calculation oriented.

On the other hand, one might suggest they use an equation like $y = mx + b$ to solve this problem. This solution will take more steps upfront, but I will discuss why this might prove more beneficial in the long run. To start, they would assign values to each component in the equation. They would explicitly define *b* to be their starting deficit, in this case $b = -\$20$; *m* is the amount they sell each cup for, $m = \$2$; *y* is the goal net amount of money, $y = \$0$; and *x* is the number of cups they need to sell to get to the goal, $x = \_\_\_\_\ cups$. Why use the slope-intercept form equation? Because everything is explicitly defined and can be easily altered depending on the question. Now we could change any of the values or switch the unknown and the process would

stay the same. We could also create a graph of the problem and project anticipated profits. We could even generalize further by saying that *y* represents the net earnings, *m* represents the earnings per sale, *x* represents the number of sales, and *b* represents the starting profit/cost. In this way, we do not even need to be talking about a lemonade stand or a scenario where we start off in a deficit. So here we see an example of CT because of the way we can generalize the problem to fit more than one scenario; this is called abstraction. This would therefore fall under the green overlap between mathematics and CT.

But let's take this one step further. Now say we want to look at the same scenario, but with a CS lens. We could write a program to automate our calculations. This could look like using our $y = mx + b$ function in code, where *y*, *m*, *x*, and, *b* are variables. Finally, we have crossed into the brown section of the diagram where all three categories overlap. We have taken a simple mathematics problem, drawn out the important details, abstracted so the details can be altered within the structure of our representation, and used a computer program to automate any calculations. This also enlightens us to the ways in which a mathematician might be daunted by CS, when in reality they have been using CT practices in their daily work. The only difference is that they have been able to move in and out of explicitness without major consequences, whereas in a coding program they might encounter an error or break their code with the same lack of discreteness.

The distinctions that I have laid out in this example are relevant to this research because there is not necessarily one correct way to hold each of the fields or ways of thinking. The works of Denning (2009), Wing (2014), and Weintrop et al. (2016) agree that CT is critical beyond the field of CS. That being said, there are distinctions even within the sources I have raised. In the work of Denning (2009), he discusses how the movement to conjoin CS and CT aims to attract

others into the CS field by demonstrating the fundamental need for CT (Denning, 2009). However, he also argues that it may serve as a limitation to the field in the same way that equating CS with programming might do a disservice to the extent of the field. His work explains that CT is a practice that is used by computer scientists but that does not define the field. This would stand in contrast to the depiction in Figure 1, where CT is a blanket that falls over all of CS. For the work within this research, the participants are working in a CS setting, learning to use CT to address specific tasks. Therefore, all of the CS they are working with would fall under CT. This is not to discredit the line of thought in which CS does not completely lie within CT, or the line of thought in which CT practices are a tool that computer scientists use, among others.

There are other limitations to the diagram as well. We have two disciplines: CS and mathematics; yet, we only have one practice or way of thought depicted in the figure. One might argue that we would need a mathematical thinking category which would encompass all of mathematics and overlap with CT and CS to some extent. The reason I did not include this in the Venn diagram or concern myself with the development of a theory that incorporates this is because it is beyond the scope of our focus. The modules for the TCs were not designed to introduce them to mathematical thinking because they should already be familiar with this. Instead, the focus is to remove them from their comfort zone while drawing connections to the material they understand.

Representations and semiotics

I would like to take a detour now in order to clarify some of what I described in my lemonade stand example. Earlier, I made mention of the idea of a representation. There are numerous discussions that I could raise with regard to representations and their importance, but

that would cross into an area of theory that perhaps extends beyond where we need to be for purposes of this theoretical lens. That being said, I would like to define representations and hint at their usefulness in discussing how students approach problem solving in the classroom. This is especially helpful to raise because the examples I just discussed were a great instance of comparable representations of the same problem. One representation was devised only within the scope of the problem, specifically to derive an answer to the question at hand. The second representation built a fuller perspective around the elements of the question, thus allowing for it to be applicable to more than one scenario. Neither representation would be "wrong" within a solution format because both can be used to find an answer to the question. The better question is which one is more useful? Before I go further on this discussion, I will allow for Erwig's book on computing to give us a nice insight to representations and their use.

In order to compute an algorithm or an answer to a problem like we saw before, we need to devise a representation of the situation. Let's start by discussing what constitutes a representation. According to Erwig, a representation needs two components: a signifier and a signified. That is, "something that represents and something that is represented" (Erwig, 2017, p. 49). The signifier and the signified make up a *sign*, which was an idea developed by linguist Ferdinand de Saussure (p. 51). In essence, signs are representations. The signifier is some object or word that represents – or stands for, as Erwig puts it – the signified. For instance, the English word "dog" stands for the concept of a dog. When I say that word to someone who speaks English, they will most likely picture a dog, or perhaps the concepts commonly associated with a dog like "fluffy", "cute", "friendly", "loyal." However, in some ways the word "dog" is a vague signifier. Perhaps the person interpreting my reference to a "dog" is afraid of dogs. Then simply saying the word "dog" might not properly represent the signified meaning for which I was

aiming. In this case, if I wanted to bring up the common adjectives that accompany the concept of a dog, perhaps I would want to include a drawing of a happy dog or include the word "friendly" in my description of the dog. Either way, the real purpose of this conversation about signs and representations is that *their use is to convey meaning*. To solve a problem, one must represent that problem somehow; but in a broader sense, the representation that is choosen shows some level of how the person understood the scenario and it carries meaning to whomever sees their work. In reality, one could use representations to devise a solution to a problem, or they could just set up the scenario around the problem and convey some meaning about it. Problem solving is just one way that representations are useful for us.

But let's keep our conversation on problem solving in order to focus on the example I levied earlier. By the time we end up with a situation where we are naming variables and using an algorithm, like in the lemonade stand scenario, we probably are not just using one or two signs. Representations can be built out of a series of signs. We can start by looking at the parts of our lemonade stand for which we could create representations. We have the cost of supplies, the earnings per cup of lemonade, the number of cups sold, and the net earnings. In the first representation, I did not explicitly represent each component. I skipped the step of showing the goal of a net earnings of $0, and instead decided to solve for the amount needed to break even: $20, the cost of supplies. Here I let "$20" represent the cost of supplies. This was not a very explicit step in my work, but hopefully for outside readers, they would see the context of the problem and the dollar sign in front of the 20, and this would help them see the meaning behind my representation. I did make an explicit sign when I stated "let x represent our unknown number of cups sold to break even." I won't continue to go into details here because the part I want to talk about is the representation as a whole.

Despite the lack of clear signs throughout the development of the representation, it is possible to see the interpretation of the problem. In fact, if someone only saw the representation and was asked to identify the scenario from a series of simple mathematics problems, they might be able to do it. This representation did not *need* to be explicit the whole time, and there is not anything inherently wrong with the ways each component fit into the picture. In the second representation, each signified concept is given an explicit signifier. We essentially have a "let ___ represent ___" for each component. As I have discussed, this requires more work on the front end, but this representation is more generalizable in future instances. This is key in the CT practice of abstraction, where a representation is designed so that it can easily be used repeatedly and in a variety of scenarios without requiring a major change to its structure. In both representations we have expressed each component as a variable or mathematics term, which makes it possible to perform a computation.

One piece of representations that I briefly described in this last section is the language aspect. Language functions as a representation because we use words to convey meaning and stand in the place of concepts or ideas. When we look at how we construct broader representations, it is important both to understand what language is used to convey meaning, as well as the strategies deployed in building the representation. In academic disciplines there are common ways that students are expected to build representations and communicate about what they are learning. To better understand what this looks like in this research, I will describe how mathematics is constructed as a language and the ways that this is relevant to mathematics education.

Mathematics as a language and language in mathematics education

I want to start by referencing the work of Schleppegrell (2007, 2010) because of the variety of frames she provides in the linguistic scope of mathematics and mathematics education. The first step in her work is to explain what language is and what is meant when we say that mathematics is a language. Schleppegrell (2010) references the work of Pimm (1987), who explained that mathematics is like a language because of its use of a specific vocabulary and ways to construct words and phrases. More so, Pimm's work noted that "mathematics is like language in that it is a system for making meaning" (Schleppegrell, 2010, p. 75). Just as representations are used to create meaning and understanding, language is a method of communication and meaning-making. That is why language itself can be a form of a representation. Schleppegrell articulates that from a semiotic perspective, language goes beyond the uses described above and extends as a "tool for thinking and constructing knowledge" (2010, p. 76). One might argue that representations are also tools for thinking and constructing knowledge, so perhaps it is redundant to make this distinction. The semiotic perspective is important to note, however, because our earlier discussion on the signifier and signified is rooted in semiotics. Ferdinand de Saussure, the linguist referenced in Erwig's work, was one of the founders of semiotics, which is the study of signs and symbols and how they are used to create meaning (Erwig, 2017).

In the case of the mathematics classroom, meaning is constructed in what Schleppegrell describes as multiple semiotic systems (Schleppegrell, 2007, p. 141). These systems are "symbols, oral language, written language, and visual representations such as graphs and diagrams" (Schleppegrell, 2007, p. 141). These systems are used together to fully convey a message or describe a scenario. Oftentimes there are places where a single system might fall

short, and the use of another adds context or understanding. In the case of the study with mathematics TCs, they are able to create meaning in a number of ways; they speak with each other, navigate with two different representations of code (block and script as described in the *Methods* section), and write their thoughts in response to reflection questions. They can flush out their understanding by creating a representation in their code and then discussing it with each other and elaborating in writing. Each semiotic system is a pathway to engaging with the material and with the other TCs.

The other way to view the meaning-making systems in mathematics is the mathematics register, coined by Halliday (Schleppegrell, 2010). A register is defined by Halliday (1978) as "a set of meanings that is appropriate to a particular function of language, together with words and structures which express these meanings" (Schleppegrell, 2010, p. 79). The mathematics register is the set of words and symbols required to communicate mathematical concepts and the ways that they are used to convey meaning. One important feature of the mathematics register is the combined use of technical language and what is referred to as "everyday" or "ordinary" language. The mathematics register requires the communicator to use everyday language in ways that may be unfamiliar and to combine this with highly technical vocabulary. It may be important to address the question of whether it truly is necessary to communicate using the full extent of the mathematics register in order to articulate a mathematical idea. Who determines which language is considered "technical" and what the proper way is to communicate an idea? The language that a student utilizes when articulating an idea should not be dismissed as inherently incorrect, but instead evaluated for demonstration of understanding and analyzed to identify knowledge gaps. Schleppegrell discusses some of this dilemma. Students must practice the use of technical language as they work in classroom settings, but they also need opportunities to

engage in discourse with others using their own words (Schleppegrell, 2010, p. 86). Those who are more fluent in the language of mathematics might move between their daily language and the mathematics register with a sense of fluidity. In many ways, we see this demonstrated by teachers as they speak at both a student's level and at a technical level, but this practice can extend to anyone in any interaction. For example, as I will elaborate in the *Results and Discussion* section of this paper, a TC with prior experience with both CS and mathematics could decide when and how to use technical CS terminology when working with a fellow mathematics TC with no prior CS experience. They might choose to rely more heavily on everyday language or mathematics terminology, while combining only parts of their technical CS vocabulary into the conversation to remain at the appropriate level. This is a point articulated by Otheguy et al. when discussing how monolingual and bilingual people both translanguage in their daily lives (Otheguy et al., 2015, p. 292). For instance, a teenager might use slang terms and internet references with friends but keep vocabulary more standardized when interacting with parents or other adults. Given this brief introduction to the topic, I will now take the opportunity to segue into a discussion on code-switching and translanguaging.

Translanguaging and disciplinary repertoires

First, I want to be clear that the discussion of code-switching is to give context to the field of thought surrounding language and multilingual learners. However, I will pair this with the introduction of translanguaging, which is a theory that challenges the notion of code-switching and replaces it as a way to describe multilingual communication. The reason this is relevant to the study is because I will posit that translanguaging can be generalized from languages to disciplines, specifically mathematics and CS in the context of this research. The TCs in the study come from a mathematics background, and their movement into a CS setting

via the modules prompted a form of translanguaging between disciplinary repertoires, which I will define in this section. Note that while translanguaging is used in relation to multilingual meaning-making, we can extend its definition to fit monolingual communication because of the ways that language is sociolinguistically defined (Otheguy et al., 2015, p. 293). This is why I was able to discuss this briefly in the previous section, and hopefully this will make more sense after translanguaging is properly defined. Nonetheless, before we go further on the discussion of translanguaging, I will provide the context of code-switching.

Code-switching is defined as "the practice of moving between languages in seamless ways, a practice common in bilingual and multilingual settings" (Schleppegrell, 2010, p. 88). Code-switching was an idea that bilingual people move between named languages (i.e. Spanish and English) in their head with a level of fluidity. As acknowledged by Otheguy et al., code-switching was a theory developed in an attempt to shed a positive light on multilingualism (Otheguy et al., 2015, p. 282). Bilingual and multilingual learners have historically been undervalued or misunderstood in classrooms despite the extent of their linguistic repertoire – their meaning-making background (Vogel et al., 2019). Notwithstanding its similarities to translanguaging, which we will define shortly, code-switching falls short because it views the languages within the communicator's mind as separate systems that they move between as they convey meaning.

Translanguaging, on the other hand, is "the deployment of a speaker's full linguistic repertoire without regard for watchful adherence to the socially and politically defined boundaries of named (and usually national and state) languages" (Otheguy et al., 2015, p. 283). In other words, named languages are not defined by linguistic distinctions, but by social, cultural, and political reasons. They have their importance in the world in which we live, but within a

learner's mind they would not be so characteristically separated without external influence. What an outside influence might identify as the use of multiple separate languages should actually be considered a use of one's full idiolect, which is "a person's own unique, personal language, the person's *mental grammar* that emerges in interaction with other speakers and enables the person's use of language" (Otheguy et al., 2015, p. 289). In that sense, a person would not speak a named language but instead their individual idiolect. This is where code-switching falls short; it views multilingual communication as the movement between languages, when in reality it is the use of a collection of linguistic knowledge within the mind of the communicator. As I previously discussed, all of this is important because we judge learners for their ability to communicate within a specific named language, regardless of whether they have a far larger repertoire given their multilingual background.

All of this falls under the broad discussion of translanguaging and multilingual communicators, but now I want to turn back to the classroom. Schleppegrell discusses English Language Learners (ELLs) in mathematics classrooms. They encounter extra challenges when navigating the mathematics register especially if they have not been introduced to a mathematics register in their native language (Schleppegrell, 2010, p. 91). Of equal importance, there are opportunities that arise when we acknowledge the linguistic repertoire of multilingual learners. Vogel et al. explored the learning of K-12 students in a bilingual learning environment as they engaged in CS and CT practices. The students who attended a bilingual Spanish-English middle school program were assigned partners and worked through a CS integrated unit plan (Vogel et al., 2019). Students were allowed to work in both Spanish and English. In one case, their teacher paired a student who was more comfortable working in English with a student who was more

comfortable working in Spanish so that they would encourage each other to move out of their comfort zones.

The students used elements of Spanish and English as they worked, as well as their own personal experiences. As an example, one of the activities was to design the code for an interview with the author of a novel. Students drew from their own knowledge about the authors they chose and mixed Spanish and English components as they created their work. By completing tasks together, students expanded each other's linguistic repertoires (Vogel et al., 2019, p. 4). I raise this research because in many ways it mirrors interactions in the TC research setting and the learning opportunities of those moments. All of the TCs entered the modules with a background in mathematics, and a number of them also had a background in CS. Together, they were asked to discuss both mathematics and CS concepts.

Here is a good place to raise an important note about the translanguaging lens, which I will discuss further when I expand on my theoretical lens. Translanguaging pertains to the use of one's full linguistic repertoire. In the case of the TCs, there are instances where they are directly translanguaging in the linguistic sense; they pull from their linguistic background in technical mathematics discourse as well as from their everyday natural language and the words used in CS. However, I intend to borrow the concept of translanguaging to discuss their full *disciplinary repertoire*. To elaborate, the mathematics register that Halliday (1978) raises would fall under the disciplinary repertoire of mathematics. In the same vein, I expect that one could devise a CS register to parallel Halliday's work in mathematics, and this would fall under the CS disciplinary repertoire. That being said, TCs do not necessarily consider their own sensemaking—naming one practice as their use of the mathematics register and another as the use of their CS register.

Instead, they "translanguage" across their full disciplinary repertoire when they put problem-solving strategies or reasoning to use.

Another important component to note is the discussion by Vogel et al. as to whether programming languages can be equated with named languages such as Spanish and English. One issue with conflating the two is that it might mask the ways that named languages function in learning CS. I acknowledge this challenge, and there are two reasons why I adapt translanguaging in the ways I have discussed. The first reason is that I am using translanguaging to refer to the discipline of CS as it arises in this study, which includes not only specific coding languages such as Python and BlockPy, but also CT practices such as algorithmic thinking and abstraction. When I discuss how TCs call upon their full disciplinary repertoire, I am referring to the linguistic aspects (symbols, terminology, syntactics, and grammar) of the varied disciplines they draw upon. I am also referring to the types of representations the TCs know from their background (e.g. linear equations or For loops) and the ways that they use these representations. While this can involve programming languages for someone with a CS background, their full disciplinary repertoire would not be confined to just the languages they use to code.

The second reason that I believe translanguaging can be a useful lens to adapt is because I believe that there are similar mistakes in how we analyze learning in the minds of multilingual students and students with expanded disciplinary repertoires. For instance, someone who can speak both Spanish and English would have aspects of both languages in their linguistic repertoire. They could draw upon elements of either as they speak, which might make it seem like they are switching fluently between two separate languages to an outside observer. However, by the definition of translanguaging, the speaker is accessing their repertoire without making such concrete distinctions between languages in their own mind. We can adapt

translanguaging to fit a scenario with disciplinary repertoires. For example, someone with a background in CS and mathematics would have aspects of both fields in their disciplinary repertoire. When they draw upon them, an outsider might identify certain aspects as a use of their CS knowledge and other aspects as a use of their mathematics knowledge. If we extend the definition of translanguaging to fit this scenario, then the person communicating would not be making such exact distinctions in their own mind as they draw upon the CS and mathematics tools in their disciplinary repertoire.

Beyond the mistake that an outside observer might make when viewing either a multilingual person or someone employing their full disciplinary repertoire, there is another possible implication. If we make conclusions on how well a TC is learning or how much they understand concepts in a certain discipline based solely on if they are using appropriate technical terms or ideas, we might be neglecting an entire aspect of their repertoire. As an example, a TC could be well-versed in technical mathematical terminology and concepts, and therefore access CS material through elements of this mathematical aspect of their repertoire in a non-standard way. They might use imprecise terminology, borrow mathematics terminology to approximate the ideas in CS, or use multiple terms to describe a concept, whereas a TC that is experienced in CS could use fewer terms and be more precise. Rather than assessing their mathematical knowledge and new CS understanding separately, it could be beneficial to view this learning through the nuanced lens of translanguaging. This would allow more insight to how much the TC is understanding CS and building their overall knowledge.

Additionally, this implication extends to multilingual students who could, for example, be fluent in one or more languages but be new to English. Rather than focusing on the deficits of their abilities in this new language (e.g. inaccurate pronunciation, word choice, sentence

structure, etc.), it may be more beneficial to pay attention to how they draw upon their developed linguistic repertoire as they access new vocabulary and patterns in English. This could reveal more about how they are making sense of the new language, as well as shed a more positive light on the process of language learning and multilingual communication.

Before I wrap up this section, I would like to turn back to the work of Schleppegrell one more time in a discussion of social semiotics. To do so, I will connect Schleppegrell's insights to those of Vogel et al. (2019). Through the earlier sections, I mentioned the importance of multiple semiotic systems and representations to create meaning. Now that I have returned our conversation to the classroom, I can focus on the ways that interactions with peers and educators aid in the construction of meaning. When Schleppegrell (2010) discusses the relevance of social semiotics, she does so with a systemic functional linguistics perspective, which is a more linguistically specific lens than I will use. Nonetheless, she brings essential elements to the conversation. Students are developing meaning in their own minds, but when they interact with others in a learning environment, they expand the ways that they conceptualize ideas. This is what Schleppegrell (2010) refers to as the synthesis of cognitive and social activity in learning (p. 84). Social interaction also layers context onto meaning. I have talked about the multiple semiotic systems, including symbolic, written, visual, and spoken meaning-making (Schleppegrell, 2007). These are developed further when I add the lens of social semiotics. Rather than individual learners seeking meaning as an isolated action, there is a collaborative element to the process. Each learner carries their internal understanding, that we could refer to as the idiolect of their meaning-making repertoire, and when they communicate with others they bridge their understandings and expand each other's repertoire.

This is where we can now connect to the work of Vogel et al. (2019). This research concluded that while they originally believed that translanguaging was the mode by which students would learn CT, they found that translanguaging was only a component of the broader set of interactions in the computational setting (Vogel et al., 2019, p. 6). The students did not separate their past experiences, the context of their tasks, and the community they were a part of as they worked, in the same way that they did not separate the named languages of Spanish and English. To construct their code, the students connected with each other and found ways to communicate across their differing repertoires and understandings. These students engaged in social semiotics by constructing meaning with one another and using multiple representations to express their ideas.

Conclusion

Up to this point, I have taken you on the following path: I started with the introduction of CT in the K-12 setting and incorporated language and representations as a way to analyze how participants interact with CT concepts as well as each other. I will go into greater detail here to ensure that the intent of this literature review has been clear to follow. It was necessary to start with the discussion of CT in the section *Defining computational thinking* because it is the primary motivation of this research. There is a drive to integrate CT into the K-12 classroom setting and therefore a drive for pre-service K-12 educators to learn how to incorporate it into their curriculum (Yadav et al., 2017). As follows, this research focuses on mathematics TCs jointly participating on CT modules.

Because we are examining mathematics TCs in particular and working with basic CS concepts, it was necessary to clarify the overlaps between the academic disciplines of mathematics and CS in relation to CT. This was what followed in the section *Overlaps of*

computer science, computational thinking, and mathematics. The next two sections, *Representations and Semiotics* and *Mathematics as a language and language in mathematics education* were necessary to transition to the discussion of language. Language is a critical component of my theoretical lens, so I started with the broader scope of representations, which can include language. Then I moved specifically to mathematics as a language because I could use the works of Schleppegrell (2007, 2010) to define it as such and bring its relevance in analyzing learning in mathematics education. This is where I transitioned to the discussion of translanguaging and social semiotics. These topics allowed me to talk about disciplinary repertoires to describe the full set of meaning-making devices and representations that students carry as they work in multiple academic fields. In the end, this all centers back on CT when we examine the language and representations used by the TCs as they formulate ideas in the CT setting. At this point, I can now introduce my theoretical lens.

Theoretical Framing

My theoretical lens can be divided into two sections and further parsed from there. These sections are both useful as we return to my research questions. We are looking to identify affordances and challenges for the mathematics TCs as well as the ways they navigate them with each other and with the tools provided by the modules. Here is where I will borrow from the theory of translanguaging in my theoretical lens. I posit that the key to addressing the research questions will be to use the idea of disciplinary repertoires and analyze instances where TCs "translanguage," drawing from their full meaning-making background.

The two sections are as follows: linguistic foundations and use of representation. The linguistic foundations are the common and technical language in each discipline. This section also addresses varying uses of terminology by two TCs with different disciplinary repertoires, which might result in misinterpretations of each other. The use of representations brings these elements into play as participants solve problems, interpret scenarios, and build meaning with each other. By design, the sections of this theoretical lens will overlap in use and relevance per each situation. Another note to make is that based on the ways I defined representations in the section *Representations and Semiotics* of the literature review, the linguistic elements are categorized as representations. When I discuss the "use of representations" as a separate category it is because I want to go further than looking at the specific word choice of the TCs. Both sections are centered around tools of meaning-making.

Linguistic foundations

In this piece of my theoretical lens, I want to expand upon elements discussed by Schleppegrell (2007) and Halliday (1978) on the mathematics register. Recall that a register is a

set of words and structures and the way that they are used to communicate a group of ideas. The mathematics register, defined by Halliday (1978), contains everyday language and technical language and each are used in unique ways to communicate mathematical ideas. For an academic register such as the mathematics register to exist, it makes sense that there should be a CS register as well. I believe that in terms of this research, we will be able to identify instances where there are crossovers of the mathematics and CS registers, and this could allow for opportunities and challenges for the TCs.

For instance, they might see common terms between the two such as variables, functions, equal signs and equality, and range. This could raise questions like "Does this mean the same thing in CS as it does in mathematics and in everyday language?" and "Do the same rules still apply to how I use these terms?" which are excellent places to explore. We may also see terms that apply more strictly in the CS register than in the mathematics register, but that might still use familiar words. Several examples could be weakly typed vs. strongly typed languages, For loops, While loops, and strings. That being said, the distinction between "technical language" and "common language" is not necessary for the purposes of this research, nor are the lines between these categories solid enough to easily identify.

Either way, through the use of these terms, participants or researchers more fluent in the CS register versus the mathematics register or vice versa might find themselves misinterpreting each other as they communicate. As I discussed in the *Translanguaging and disciplinary repertoires* section of the literature review, social semiotics is an important part of constructing meaning. Therefore, there is value in evaluating these aspects of communication and repertoire building as the TCs use the academic registers of CS and math.

Use of representations

I want to examine the language that the TCs use because of the overt connections between the mathematics and CS registers. That being said, there is a step that learners take that goes further than how they word their ideas. Students build broader representations using their meaning-making repertoires, which involves key vocabulary, but extends to the ways they know how to address tasks or problems. The TCs have experience with a variety of concepts – functions, parameters, variables, etc. – through their mathematical background. Their experiences give them guidance on how to use these concepts, where they might show up, and how they relate to one another. Some of the TCs also have experience with these concepts or others like them through a CS background. Whether they've taken courses or had experience coding, they now bring with them the perspective and understanding of how these pieces fit together in the CS context.

So how does this build on what I talked about in the previous section? What I just described is the foundation for what I discussed in the *Translanguaging and disciplinary repertoires* section of the literature review as "translanguaging" across one's disciplinary repertoire. This lens gives us a way to talk about how when the TCs are asked to address a specific task or problem, they will construct meaning using common concepts as they understand them. We will see moments where they communicate with each other and clarify meaning through the use of vocabulary of which they have a common understanding. They also might discuss concepts that they are familiar with but with a new context that broadens how they think about it.

In this research, the TCs are tasked with designing code for a variety of simple scenarios. We can look at the vocabulary they use and which pieces of their disciplinary repertoire that they

draw upon as they construct representations in the BlockPy code. Each TC has experience with setting up representations in a mathematics context and a number of them have experience with doing so in CS. Part of this section of the theoretical lens is to look at what kind of knowledge they reference as they work together and build representations in this CS setting.

Methods

In this section, I will lay out the context of my own analysis as well as the study that provided the data. I will explain the format of the data I am using as well as the ways it was collected. I will also provide information about the parts of the data I am specifically analyzing. Then I will go into the details of my analysis, such as how I am qualitatively coding and which themes I have highlighted.

Researcher's positionality

I am an undergraduate student studying physics with a minor in mathematics; I had no experience with CS or programming prior to this study. I am interested in learning Python and am taking computational courses as a part of my undergraduate education. Another one of my interests is the learning of languages, which I have pursued by studying Spanish as a second language at the university level.

I come to this research as a student, and the history I bring influences how I lead myself through this work. Throughout my own education I have seen instances of where ideas and disciplines connect together, and this influences the lens with which I view this research. Beyond my own language education, I am interested in bilingualism and multiculturalism, and it is my personal opinion that these aspects are undervalued in our education system. I have a curiosity to how we can expand multilingualism, multiculturalism, and interdisciplinary education. This gives me a motivation to find the ways that these ideas can be connected together.

As an aside, in this research, I was not a part of the module design or data collection process, but I will be analyzing data that was collected in this study.

Research context

I looked at data from a study that the research team conducted on how mathematics TCs engage with CT modules through joint participation. These modules were run during the first three days of the study, whereas the fourth and fifth days involved student teaching and are not the focus of this paper. The participants of this study were enrolled in a methods course for secondary mathematics education in their university's education master's program.

The research team created a module run on an interface called BlockPy, which follows the structure of Python but removes aspects of the syntax by formatting components as blocks and puzzle pieces. Through this interface, TCs were able to use either the block space or a regular Python code space, and their work would automatically appear in the adjoining space. Beyond the simplification of syntax, another purpose of the dual representations was to offer a variety of ways to reason within the coding setting and draw connections to mathematics concepts.

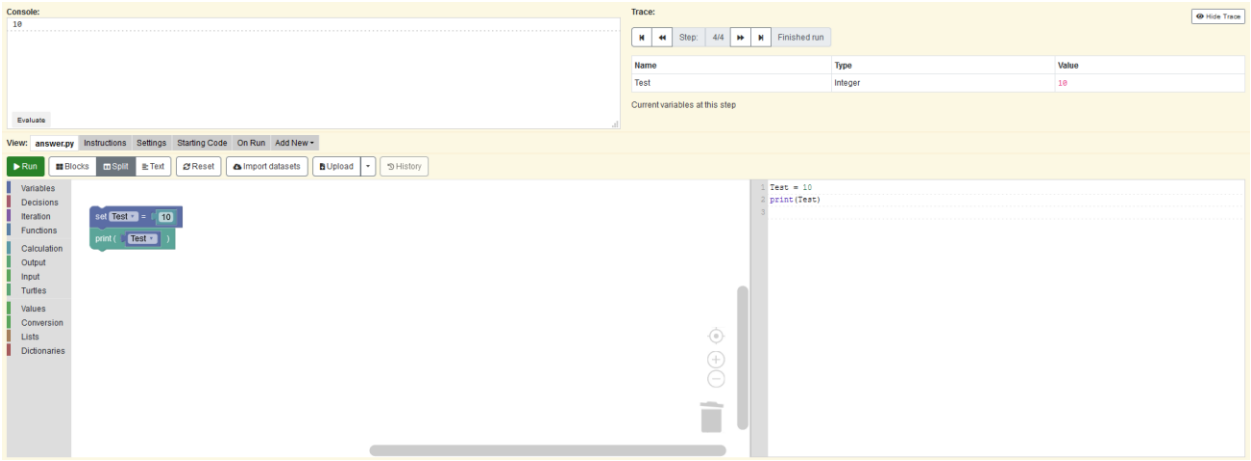

Figure 2. Screenshot depicts the BlockPy interface that the TCs used in the modules. The left side shows the block-based code, and the right side shows the script-based code, both of which the TCs could use.

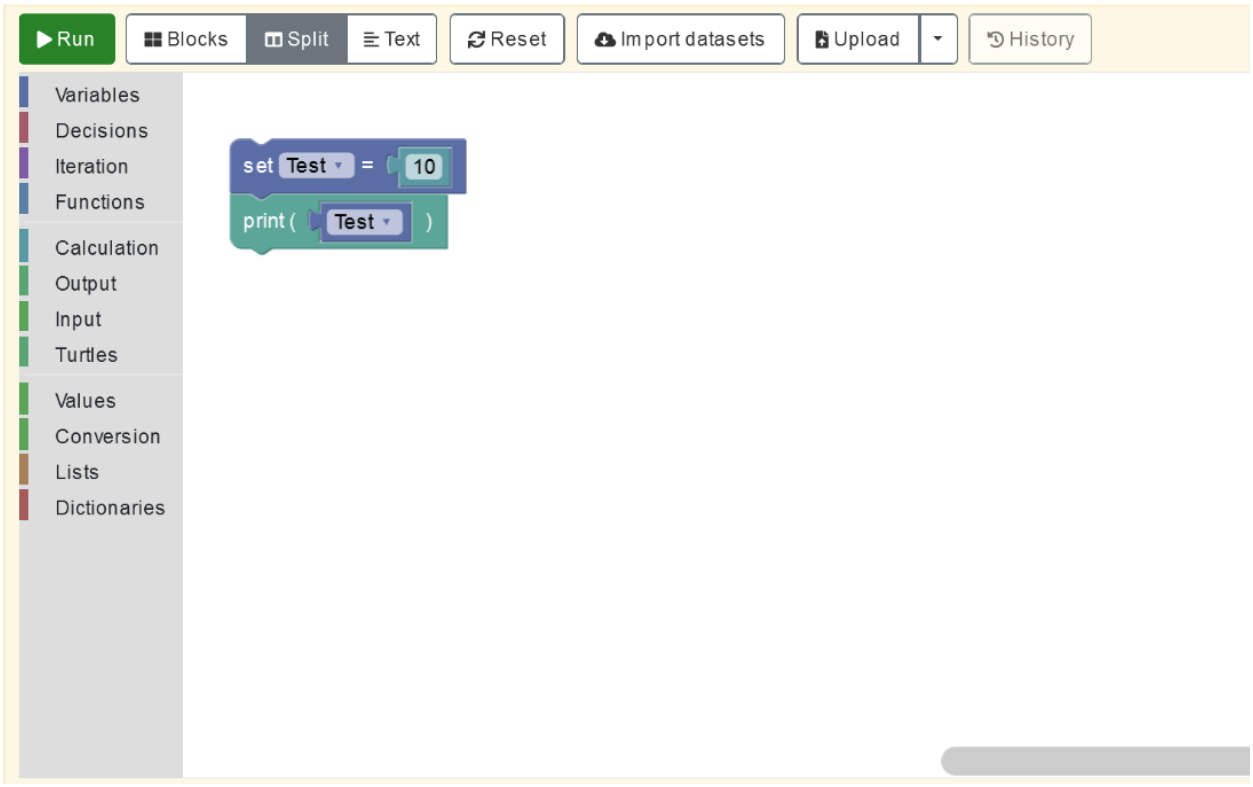

Figure 3. Close-up screenshot shows solely the block-based side of the BlockPy view. In the sidebar, TCs could choose from a wide variety of blocks with which to create their code.

Within the modules, the mathematics TCs were tasked with completing basic coding tasks and answering reflection questions. The questions required them to consider links between their mathematics background and the principles they worked with in the code. The questions were also designed to encourage discussion around what was happening in the code.

The tasks were built around secondary mathematics curriculum. The TCs spent three days working through these tasks. The first two days were prefaced with an investment and interest rate scenario of which they would be able to build a representation in code by the end of the modules. The instructions explicitly conveyed that the TCs would draw connections between computation and mathematics as they worked. On Day 1, they worked with basic variable

concepts such as data types, creating and assigning information to variables, and the order of the input and variable about the equal sign. On Day 2, they were introduced to For loops and were given the tools to set up a representation of the investment scenario. In contrast, Day 3 worked with a feature called “turtle” that allows the user to draw shapes through their code. For purposes of this analysis, I will be focusing on a specific group’s path through Day 1 of the module, the reasoning of which I will explain later in my methods section.

Data collected

To complete the modules, the TCs were divided into five groups of 2-3 per computer. In each group, the participants rotated who was the driver operating the computer and who was on the instructions. The study was conducted in the TCs’ normal classroom to provide a natural setting for discourse to occur. In order to form groups, the research team considered two main pairing criteria: who would share airtime and work well together and who had prior coding experience versus none. Members of the research team were familiar with the participants and used this knowledge to determine balanced pairings. Additionally, the participants filled out a pre-study survey in which they disclosed any prior experience with CS and programming. From these results and research team knowledge, the pairs were made to spread out those who had prior experience.

The research team collected data by setting up cameras and audio recorders, as well as a screen recording technology on the computers. From this data, the team constructed enhanced transcripts coordinating the screen cast and audio recording and verifying with the video as needed.

I have examined the enhanced transcripts of Group 2, which consisted of two TCs that I refer to by the pseudonyms Ginny and Rhoda. Rhoda disclosed that she has had previous experience with CS courses, while Ginny has little to no previous experience. As mathematics TCs, both participants have extensive previous experience with mathematics.

Methods of qualitative analysis

The thematic analysis coding method was a multi-step process (Fereday & Muir-Cochrane, 2006; Swain, 2018). First, I read through the data with a second researcher to get a broad sense of the transcripts. Next, I developed a set of a priori codes for the transcripts based on themes in my theoretical lens (Fereday & Muir-Cochrane, 2006; Swain, 2018). I used these to make a color-coded key and highlight instances throughout the transcripts. Then I went through a process of adding and adapting my codes through subsequent read-throughs. I did this both on my own and with an additional researcher to develop inter-rater reliability of the codes.

In the initial coding, I generally highlighted the key words that I saw that fit the code, but in subsequent reads I looked at the context of what they were saying and added highlights to additional places. This also involved adjusting the highlights in cases where the original code did not fit the context. One case of this is adding code where the participants' dialogue is built from the specific vocabulary for which I originally coded. For example, "representations" is a word but also an idea of how one thing stands for another. In the first read I highlighted any variance of the word "representation," and in the subsequent reads I added highlights for instances of representation regardless of whether they contained the word itself. Part of the reason why it was still important to have codes for the exact words even after I went back and reread the context is because we can use the linguistics lens to analyze the specific vocabulary and phrasing of the TCs. The last adjustment I made as I read through the transcripts was adding additional codes to

clarify the a priori codes that I initially developed. In one instance, I added a code on Day 2, and reread Day 1 to highlight this code as well. Throughout this process, I also wrote short notes and memos about the themes and to help me decide which instances to examine further with my lens.

The a priori key that I used on Day 1 and Day 2 of the data highlighted the following codes: variables, input related vocabulary (Boolean, integer, values, strings), CS terms, mathematics terms, and representations. The variables highlight was used for any instance of the use of the word "variable" as well as discussion around setting a variable and rules of variable use. This code was developed from the *Linguistic foundations* section of my theoretic lens because "variable" is a term used in both CS and mathematics. The input related vocabulary was also developed form the *Linguistic foundations* section of my lens due to its use of a number of words that have meaning in both academic registers.

The codes for CS terms and mathematics terms were difficult to create because they were somewhat neglectful of the inevitable overlaps of the two. I also felt this might contradict the translanguaging lens, where in the mind of the communicator the distinction between languages is not clearly defined. That being said, these codes were helpful for identifying moments of translanguaging because I could see the TCs utilizing the full extent of their disciplinary repertoires. I did not create either of the codes until after I started the analysis because of my reluctance to strictly define the two, but I added them for the above reasons.

Lastly, the representations code was developed out of the *Use of representations* section of the literature review. It was used to highlight any instance where the words "representation" or "represent" were used, as well as moments where the participants talked about how they constructed representations.

I also added two codes: “instructor” and “Ginny introduces mathematics concept.” I wanted to color the instructor dialogue to distinguish it from the text of the TCs. I added the a posteriori code about Ginny’s introduction of mathematics concepts because as I read through the transcripts with an additional researcher we noticed that there might be a pattern in terms of these contributions (Swain, 2018). The way she interrupted the dialogue around CS concepts felt like a marker, as these moments were commonly followed by colorful groupings of dialogue.

Oftentimes, I made comments in sections where more than one color of highlight was present. These multi-coded sections flagged as rich instances for discussion. This was not entirely intentional, but it seemed to be prevalent enough for me to note. As I determined which sections to analyze deeper, I returned to these sections.

I selected several sections of development over Day 1 because of the clear progression of the discussion around variables in CS and their connection to mathematics. I have focused on Group 2 over the course of Day 1 because the two members, Rhoda and Ginny, have different undergraduate coursework backgrounds: CS and mathematics, and just mathematics, respectively. Throughout the day, they build meaning across their distinguished disciplinary repertoires to understand the concept of a variable and how it differs between CS and mathematics. The sections I chose to analyze were those where I noticed the use of translanguaging and social semiotics to construct meaning about a concept in the activity. I will go on to analyze and elaborate these sections.

The last step of my analysis was to run through the coded transcripts with an Excel spreadsheet and analyze with several questions in mind:

1. How are the teacher candidates using the idea of variable?
2. What purpose do the participants ascribe to variable?

3. How are the participants using computer science or mathematics terminology?
4. How are the teacher candidates making connections to their teaching experience?
5. How are the teacher candidates making connections to their mathematics knowledge?

These questions stem both from my research question, as well as some of the codes that I developed over the two days. This Excel spreadsheet was created by a second researcher. From the spreadsheet I identified two main themes with the guidance of this second researcher. I will elaborate on these in the following section.

Results and Discussion

I will return now to my research questions:

1. What are the linguistic affordances and challenges that K-12 mathematics teacher candidates face as they engage in a computational thinking setting?
2. How do the teacher candidates navigate those affordances and challenges:
    a. With each other?
    b. Using tools they are provided by computational thinking modules?

To address these questions, I will now introduce two main themes that arose from the data.

Theme 1: The disciplinary repertoires of the teacher candidates granted them both affordances and challenges.

Theme 2: The teacher candidates created meaning together by translanguaging and coordinating their disciplinary repertoires.

These themes overlap with each other and each helps to give clarity to the questions posed at the beginning of this paper.

Theme 1: The disciplinary repertoires of the teacher candidates granted them both affordances and challenges.

Each TC had their own disciplinary repertoire from which they drew knowledge and vocabulary. I will start by examining Ginny's and Rhoda's disciplinary repertoires separately because they offer individual insight to the concept of disciplinary repertoires and how they can either present affordances or challenges depending on how they are accessed within discussions.

Rhoda drew upon a CS-math disciplinary repertoire. Rhoda was an undergraduate CS major prior to changing her major to mathematics with a focus on education. As a result, she brought both a mathematics and CS background to the module. Across the data, Rhoda made explicit reference to understanding how things worked in the code space but also experienced confusion as she tried to navigate the BlockPy interface.

Affordances. Rhoda's understanding of CS granted both affordances and challenges. On one hand, knowing what technical terms meant in CS – such as variable, data type, and For loop – allowed her to understand more about what was happening behind the code than someone who was new to the content. Across the first day, Rhoda offered insights verbally to Ginny on these different constructs as she explored the BlockPy interface in the process of completing the module coding activities. An example of this was when Rhoda was prompted to explain the concept of variable to Ginny. In this case, they were comparing a variable in CS to a variable in mathematics, as guided by a reflection question in Reflection #1.1 (see *Appendix*). Rhoda elaborated that in CS, a variable works as a placeholder; it is useful for when you have something in your code that you want to call on later. Rhoda engaged in conversations about CS constructs – variable and datatypes – and explained how these constructs were used in CS. In this respect, Rhoda's prior experience with CS granted her the capacity to navigate across different uses of technical CS vocabulary and explicate the different elements to Ginny. Rhoda's capacity to draw upon CS constructs is something that I will return to in the second theme as I highlight the ways the two participants navigate the concept of variable over the course of Day 1.

Challenges. Rhoda's programming experience also presented her with challenges in this setting as she learned to code using BlockPy. To provide context to this challenge, I would like to return to an idea from the literature review about representations, as well as to the Venn

diagram that depicts the overlaps between CT, CS, and mathematics. In terms of the setting of this research, Rhoda falls in the centermost overlap of the Venn diagram due to her prior experience with CS and mathematics. As I explained in the *Overlaps of computer science, computational thinking, and mathematics* section of the literature review, I have chosen to place all of CS under the umbrella of CT for purposes of description in this research. Rhoda's descriptions of a variable as a placeholder are just one example of how she demonstrated CT in her use of CS. Despite her position in the centermost overlap, she still encountered difficulties in the CS setting because she was working with a different representation of code than she was used to. Recall from the *Representations and semiotics* section of the literature review that according to Erwig, a representation needs two components: a signified and a signifier. The signified is the thing that is represented, and the signifier is the thing that represents (Erwig, 2017, p. 49). For Rhoda, she was comfortable enough with CS content that she often knew what she wanted to signify, but in the new BlockPy space, she didn't necessarily know which signifiers to use. Her experience was with a script-based representation of Python. In order to learn the new block-based representation of Python, she needed to build an understanding across the representations.

The coding interface included both script and block-based code. As Rhoda interacted with the interface, the screen cast data showed her attempts to perform certain actions with BlockPy that she expected to be able to do from her previous experience. Often Rhoda interjected in the discussion with Ginny narrating her exploration of the block-based code. There are a number of instances in which Rhoda and Ginny were navigating the BlockPy interface, engaged in a discussion of the module directions, and Rhoda would diverge to make sense out of BlockPy. She would mention a type of representation that she was familiar with, such as setting a variable or assigning a data type, and would get caught up in how to create it in the block-based

code space. In these moments, Rhoda would express her confusion. Although often a problem was not insurmountable, it constrained what she wanted to accomplish at that time.

Example of movement across representations. An example of Rhoda's expressed confusion occurred when they were first introduced to the block-based code space. She wanted to allow the user to input a quantity of money to find out how much they would earn over a certain number of years, a key idea in the scenario posed at the beginning of Day 1. In the BlockPy code window Rhoda had defined some variable "x" and was exploring different block options to try and find an input where the user could choose the value. She tried to ask a professor for help with this because she was unable to find the block that would signify what she wanted in the block-based space. The instructor realized that she was inquiring about something that was beyond what the module was asking for at this step. During this process Rhoda also introduced an operator on the variable that subtracted 2 from the input value when it printed, which is an idea she returned to later to aid her description of variable as a placeholder.

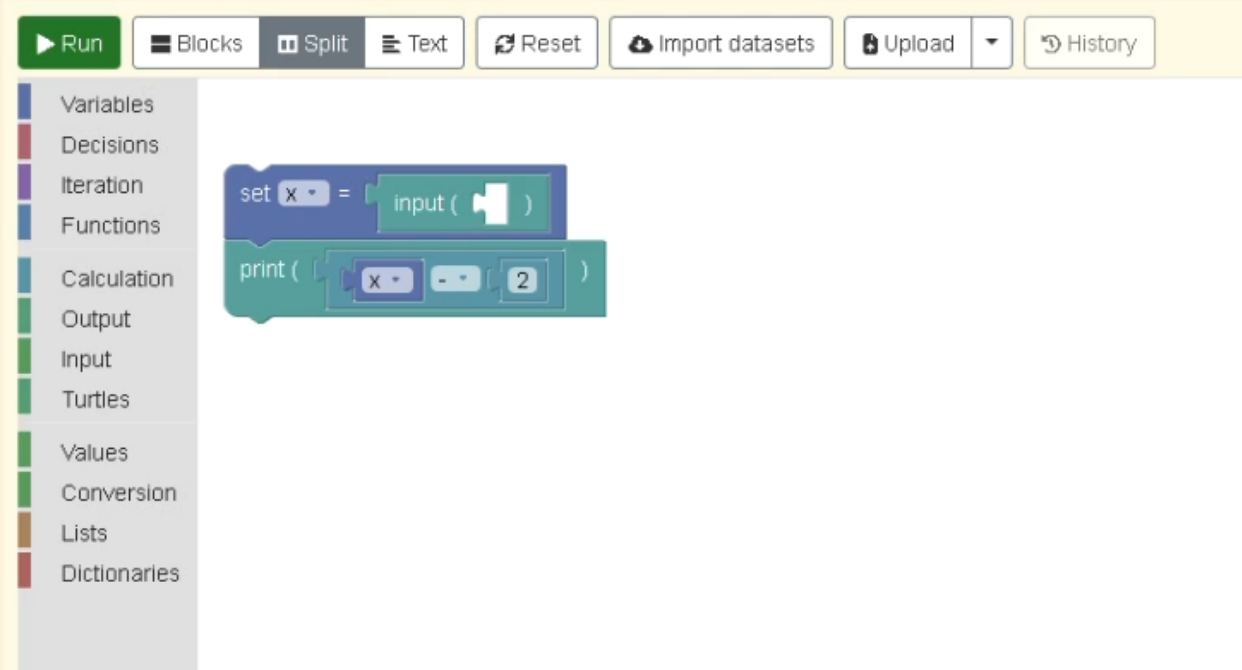

Figure 4. Block-based code in BlockPy shows Rhoda's attempt to create a variable that could receive a user input. She left an empty input puzzle piece because she could not find a block to signify what she wanted. This also demonstrates her use of the -/= block, which she later used to describe how a variable can act as a placeholder.

In this instance, Rhoda was afforded the insight about user inputs and ways to manipulate variables based on her previous CS knowledge, and as a result, she felt compelled to understand how this would correspond to the BlockPy representation. This inquiry was not resolved because her exploration was beyond what the module was asking for in that moment. Once she and Ginny realized this, they went back to the instructions for guidance. Before moving on, Rhoda expressed that she "was trying to figure out how the block is related to the Python" (line 259), meaning how the block-based code space related to the script representation that she was more familiar with from her prior CS experience.

Without resolution, Rhoda still showed signs of wanting answers to her earlier question later in the module. One instance was while Rhoda and Ginny were responding to the reflection question on variables that I previously referenced. During the discussion, Ginny and Rhoda were

comparing ideas and Rhoda contrasted that in "Python a variable is typically a set thing" (line 542). Rather than clarifying this further, Rhoda moved into narrating her earlier confusion with user input and setting a variable with the blocks to Ginny.

Interestingly, as Rhoda spoke, Ginny also spoke, but Ginny was expressing confusion about how the variable would work in Python. I saw this as an instance where Rhoda and Ginny were in a dialogue to create a common understanding, but then they diverged in their conversation because they had confusion in different places. When Rhoda was not caught up in the coding discomfort, she was able to navigate Ginny's confusion around the connections between CS and mathematics. Part of this navigation was listening to Ginny's questions which connected to her background in mathematics, and interpreting how they related to similar ideas in CS. This is something I will discuss further in Theme 2 when I elaborate on their use of translanguaging. In this case, Rhoda was aware that there was something she could do in the code – create a user input variable – but because she did not know how to do it in BlockPy, she was side-tracked from the conversation with Ginny.

Regardless of whether she had prior experience, Rhoda would most likely have encountered challenges when faced with a new setting like BlockPy. That being said, Rhoda specifically demonstrated the need to negotiate with her prior knowledge of CS before making sense of what was happening with the block-based coding. Rhoda's CS-math disciplinary repertoire afforded her access to CS constructs as evidenced in her elaboration on ways that a variable was similar and different in CS and mathematics (this will be expanded on in the second theme). Her previous coding experience positioned Rhoda to know what was possible in programming (an affordance) and uncovered how the block-based coding system constrained her seamless access to creating that code in the module. To summarize, the main affordance that

Rhoda encountered with her CS knowledge was that she could dive deep into the technical content, and the main challenge she faced was that she got caught up when she wanted to coordinate her disciplinary repertoire with the new representation in the BlockPy interface.

Ginny drew upon a general mathematics and mathematics teaching repertoire. In comparison, Ginny's disciplinary repertoire in mathematics was an affordance in the module when she was asked to draw upon it to contrast concepts in mathematics and CS, and it posed challenges when she did not see resolution in the differences. The TCs were called to consider similarities and differences between mathematics and CS by the reflection questions in the module, and by instructors when small group or whole class discussions were prompted.

While Rhoda fit in the centermost overlap of the Venn diagram from the literature review, Ginny fit more readily in the mathematics or math-CT overlap of the Venn diagram. Therefore, when asked to work on CS material, Ginny needed to move from the math-focused areas of the Venn diagram into the overlap where Rhoda resided. This presented challenges not only because she was accessing new material, but also because she had to negotiate with how each component overlapped with what she already knew.

The interesting part about Ginny's repertoire is that it was not just built out of general mathematics knowledge, but also from her student teaching background. In her references to mathematics concepts that I will continue to discuss in Theme 2, several related directly back to her recent TC experiences. Ginny also explicitly discussed how she'd been going over certain concepts in her classroom and wanted to understand how they connected to the modules. Rhoda shared this piece of Ginny's repertoire as well, but Ginny specifically drew upon it because it was an area of confidence for her in a setting where she was otherwise uncertain.

Affordances. One instance where she demonstrated this confidence was when Rhoda and Ginny were responding to the third question in Reflection #1.1. I discussed the latter half of this conversation in reference to Rhoda's repertoire, but now I'd like to look closer at the earlier part of their discussion. The module prompted them to discuss the similarities and differences in the ways Python and mathematics use the idea of variable. After reading the reflection question, Rhoda was the first to respond by saying that in math, a variable "is normally just the number" (line 529). Ginny quickly interjected that a variable is like a varying quantity (line 530). Her interruption to Rhoda's contribution was a sign of how ready she was to pull from her teaching background. As she elaborated on her varying quantity idea, she referred to the topics she'd been going over in her class that day, including the mathematics concept of parameters. While the conversation then diverged, the two TCs returned to the ideas of varying quantities and parameters as guiding points in their discussion over variables in CS. For Ginny, the connections to her mathematics teaching experience were a jumping off point for deeper dialogue on newer CS information.

There was also another instance earlier in the day where Ginny compared using set blocks to writing a 'let statement' in response to the second question in Reflection #1.1. A let statement was the name she used for defining a variable for a function in mathematics, (i.e. let "x" represent the number of apples in a bag), which came from her student teaching. The reference to something familiar in Ginny's disciplinary repertoire was a way for her to develop an understanding of how a variable works in CS. In both instances, rather than building all of her new CS knowledge on a clean slate in her mind, she was creating connections to her previous mathematics knowledge to make meaning of the CS terms. I will return to the definition of translanguaging to explain this behavior. Just as translanguaging is formed from the idea that

languages are not separate in the mind of the communicator, the disciplinary repertoire is not divided into separate fields in the mind of the TC. Ginny's disciplinary repertoire afforded her a base to build CS knowledge upon, thus intertwining her previous mathematics and teaching knowledge with the new CS concepts. Ginny used mathematics concepts and terminology in discussion with Rhoda and learned how they connected to the CS information, which helped her move into the centermost overlap of the Venn diagram. This is something I will discuss more in Theme 2.

Challenges. At the same time, Ginny's disciplinary repertoire presented challenges as she approached the module. While she was able to pose mathematics concepts as a starting and/or clarifying point for discussion around variables in CS, she also struggled when her understanding of the mathematics concepts didn't align with the CS version. For example, during Rhoda and Ginny's discussion about how a variable is like a varying quantity in mathematics, Ginny expressed confusion about the way this translated to how a variable is described in CS. This confusion led to an in-depth discussion with Rhoda about the uses of a variable in CS, and Ginny continued to pull various concepts from her repertoire into the conversation to negotiate which pieces of her knowledge about variables fit with this new CS description. It was advantageous for Ginny to have so many parts of her mathematics and teaching background to draw upon as she eased her confusion, but it also meant that she could not move on from her discomfort until she had navigated each of the different similarities and differences of the mathematics and CS use of variable. By the end of the Day 1 module, Ginny still had not fully negotiated the differences of the two main ways variables were described, which were as a varying quantity (Ginny's description) and as a placeholder (Rhoda's description). She expressed this in a piece that I will share as I elaborate on this in Theme 2.

Theme 2: The teacher candidates created meaning together by translanguaging and coordinating their disciplinary repertoires.

The second theme that arose pertained to how the TCs navigated meaning-making between their disciplinary repertoires. As I have explained, Rhoda could readily access a CS repertoire, but what I would now like to address in more detail is how Rhoda's disciplinary repertoire included mathematics based in her undergraduate degree and her experience student teaching. This is important because it was a knowledge base that Rhoda and Ginny shared, and it allowed them to have a common basis for communication. Throughout Day 1, Ginny foregrounded her mathematical knowledge as she grappled with the new CS concepts, and Rhoda acted as a bridge between their common repertoire and the CS material. Another way to think about this is to return to the Venn diagram, where Ginny started in the mathematics and math-CT overlap, while Rhoda was in the math-CT-CS overlap. Rhoda and Ginny could communicate using their shared mathematics overlap, and Rhoda's experience with CS allowed her to guide Ginny towards the math-CT-CS overlap.

Ginny draws on mathematics to link to CS. As I began to discuss in Theme 1, there were several instances where Ginny engaged with Rhoda about a CS concept and used mathematical language to do so. This made sense because Ginny was familiar with the mathematical concepts that she used as a mathematics TC in her courses and student teaching. For example, early in Day 1, Rhoda was explaining what a variable was in brief terms because they had just created their first variable in code with Rhoda writing the block-based code on the computer. Over writing the code, Ginny expressed how she did not understand what the variable was doing. Until this point, they did not have a context for the use of a variable and Rhoda wanted to elaborate on what she did in the code. She said, "So variables are like if you're gonna

have to use it in multiple lines and you don't wanna keep typing" (pg. 11, lines 167-8). Rhoda's explanation hinted at her use of CT, which I will return to later in this section in more detail. In response, Ginny said "Oh yeah … because in Python that's how you'd let something go to x or something" (pg. 11, lines 169-170). Rhoda affirmed this before continuing on to show Ginny more about the variable. This was a brief interaction, but it was the first place in Day 1 where we saw Ginny articulate a CS idea comparing it to a mathematical idea and using mathematical terminology. Although it was not entirely clear what Ginny meant and there was not much discussion about this exact contribution, Ginny's subsequent contributions about variables had us inferring that she meant assigning a value for x, such as when you evaluate a function for a value at x. Because the variable concept was unfamiliar in a CS context, she used concepts that she was more familiar with to draw connections. Unlike for Rhoda, the CS terminology was very new to Ginny, so she used mathematical ideas and vocabulary to communicate. This helped her move into the centermost overlap of the Venn diagram, as I discussed before.

Rhoda and Ginny did not just create meaning around the concept of variable in one instance—it was a process that developed throughout the course of the day. In each instance, Ginny navigated the terrain of the math-CT-CS overlap of the Venn diagram, and Rhoda was encouraged to find new or elaborated ways to describe her understanding of variables. Moreover, they developed meaning about variables in mathematics and CS by coordinating their two disciplinary repertoires. Rhoda brought her CS understanding of the variable to the module, and because she shared a common set of vocabulary and knowledge with Ginny, they could have conversations using that shared base. Ginny formed connections in a way that flushed out Rhoda's definition of a variable in CS and in a way that built Ginny's CS knowledge upon the solid mathematics base. This mode of creating meaning and then clarifying and building it

throughout the day was aided both by the curiosity of Ginny to negotiate her understanding in mathematics to CS and by the module's persistent questioning of the connection between the two disciplines.

In the literature review, I briefly touched on multiple semiotics systems and how in this study, the TCs had access to two coding representations (block-based and script), their oral communication, and written responses as a way to form meaning in the modules. Now is a good time to elaborate on how this relates to the ideas of social semiotics and translanguaging. I will start by returning to the work of Vogel et al. (2019). This paper explained how in a study of how translanguaging worked in a K-12 multilingual CT environment, participants used translanguaging and interactions with each other to form meaning. They accessed their backgrounds and experiences to make meaning as they constructed code and brought their understandings together in the process. While the researchers were clear that translanguaging had an important presence in the study – in this case the participants were in a bilingual Spanish-English middle school program – they found particular interest in the interactions between students and the ways their backgrounds helped them build computational representations. This overlaps with Schleppegrell (2010), which introduced social semiotics as a way to create meaning. Both posit that when students come together, they expand the ways that they make sense of a concept through their communication.

Module design supported collaboration. Individually, Rhoda and Ginny could have accessed the multiple semiotic systems I discussed before to form meaning about the concept of variable, but the module required that they communicate with each other and discuss their understanding of what they were doing. Their experience mirrored the ways that translanguaging and social semiotics played out in the Vogel et. al (2019) study. The two TCs translanguaged

with the mathematics and CS disciplines as they created meaning, but they did so jointly and in such a way that uniquely collaborated their two disciplinary repertoires. To explain this further, I will now share the path that Rhoda and Ginny took through the course of Day 1 as they developed their discussion on variables.

Their exposure to variables was guided by the module; first they were tasked with creating variables, and then they answered their first set of reflection questions about how CS and mathematics relate (Reflection #1.1). During their first instance of creating variables, Ginny created a link to her mathematics knowledge, which I discussed earlier in this section. She continued to create these connections while trying to understand how variables were used in CS. An example of this is the connection she made to 'let statements' that I elaborated on in Theme 1.

Through Ginny's prompting and questions, Rhoda began to deepen her explanations of how a variable works in CS as compared to math. Rhoda had an idea in her mind about what variables were and how they could be used, which was formed from the CS background in her disciplinary repertoire. Her explanations of variables also connect to the definition of CT posed in the *Defining computational thinking* section of the literature review. Recall that Wing defines CT as a "thought process involved in formulating a problem and expressing its solution(s) in such a way that a computer (human or machine) can effectively carry [them] out" (Wing, 2014, p. 1). This includes the skills of abstraction and algorithmic thinking. The main idea Rhoda wanted to divulge to Ginny was that variables are representations, and as such they can be used as placeholders in code. The use of variables as placeholders such that something can be represented in multiple lines of code without writing them out each time is a basic example of

algorithmic thinking. This supports the placement of Rhoda in the CT-CS-math overlap of the Venn diagram because the CS concepts she explained were rooted in CT.

The objective of part of the discourse between Ginny and Rhoda was finding a way for Rhoda to communicate the signified CS-CT concepts to Ginny. This involved constructing different signifiers or ways to explain what she was thinking. When they started, Rhoda mentioned that variables allow the user to reference something in multiple lines of code, which is an example I used earlier in this section. Shortly after, she affirmed an idea from Ginny that a set block in the block-based code lets someone create a variable and say what it represents. These were brief moments where Rhoda touched on how variables work as representations. While these explanations drew from her background from the start, the social semiotic setting allowed for her to flush out her meaning and construct it with Ginny's mathematics repertoire. Rhoda pulled from different resources in her repertoire such as examples in code or different wordings in order to respond to Ginny's prodding of her explanations.

Ginny's use of similes. One of the ways that Ginny prompted this coordination of repertoires was through her use of comparative reasoning using similes. For instance, a common phrase in these moments was, "is this like…?" where Ginny tried to make connections between an idea drawing upon her mathematical background and a new idea in the modules. Rhoda responded to Ginny's questions or claims that used the mathematical examples to explore a CS concept by either affirming them or correcting the ideas that Ginny raised. Again, these moments were opportunities for Ginny to step from her mathematics repertoire into an overlap of mathematics and CS, as depicted in the Venn diagram from the *Overlaps of computer science, computational thinking, and mathematics* section of the literature review. Together, Ginny and Rhoda used these moments to navigate Ginny's discomfort as she moved into math-CT-CS

region of the Venn diagram. There were four instances throughout the construction of the meaning of variable in which Ginny used comparative reasoning with Rhoda to explore the CS concept (lines 169-170, line 448, lines 555-556, and line 603). These examples included connections to parameters, 'let statements' (as discussed in Theme 1), and linear equations in the form of $y = mx + b$. In these moments, Rhoda demonstrated that she understood Ginny's references to their shared mathematics background.

By translanguaging between her own CS background and her understanding of the mathematics concept that Ginny raised, Rhoda was able to bridge the gaps between their two repertoires and provide an explanation with which Ginny could engage. When something did not make sense or reminded her of a different segment of her mathematics understanding, Ginny would form another connection. This prompted Rhoda to provide further explanations to the uses of variables in CS. One instance of this occurred further in the sequence of Day 1 after Ginny made a comparative statement to linear functions, which Rhoda did not affirm. Rhoda wanted to clarify how the variable worked in CS as opposed to math, so she revisited her earlier touches on the idea of variables as a representation by explaining that variables in CS are like a placeholder. This time, instead of briefly explaining her understanding of variables, Rhoda provided examples and elaboration.

The two TCs continued to draw connections to mathematics and to engage with each other at the prompting of instructors and the module. Notably, near the end of the day Ginny was prompted by an instructor to elaborate on how she felt that the concept of a variable related across CS and mathematics. In her response, she employed terminology that both she and Rhoda had used throughout the day (pg. 46, lines 846-852):

> It's sort of... I mean, I feel like, I feel like there's a way to relate it 'cause of... I... But, I just get really confused 'cause when we're saying like set Name equal to, you know, like somebody's name and they use the term variable there… To me, I think of a variable as something that, like, we were talking about is like a placeholder in some instances and a varying quantity in others and, uh, so in this situation like I think I'm starting to see how we can use this later on. Like instead of using like a person's name, we're using Name to give it a code.

The key terms that stood out in this quote were "placeholder" and "varying quantity." Ginny first posed the varying quantity idea when she talked about a variable in mathematics. As I mentioned in Theme 1, Ginny felt that she understood how a variable could be a varying quantity in math, but she did not know how this carried over to a variable in CS. By the end of the day, she had heard Rhoda's explanations of how a variable is like a placeholder that lets the user call upon something later in the code. In this prompted moment, Ginny tried to make sense of the differences between the two uses of the concept of variable. While she did not strictly use mathematics vocabulary to describe a CS term or vice versa, she utilized vocabulary from BlockPy such as "set ____ equal to" for creating a variable, which demonstrated her tendency to "try on" certain words as she communicated. Ginny's borrowing of terminology as she played with her sensemaking of variables in CS versus mathematics is an example of how she translanguaged. She constructed her new knowledge about variables in CS on top of the mathematics version that she had a strong basis for in her disciplinary repertoire, which was demonstrated in this vocalized negotiation of the two descriptions of variable. It is interesting to see instances where Ginny translanguaged by trying on the new terminology as opposed to the

instances where she translanguaged by using her mathematics terminology to discuss the CS concepts.

To summarize, over the course of Day 1, the two TCs worked together in a social semiotic setting to construct meaning around variables in CS and mathematics using their two disciplinary repertoires. Rhoda's participation in the module drew upon her mathematical and CS background, which centered her in the math-CT-CS overlap of the Venn diagram from the literature review. In this respect, she was able to pull from either her mathematics or CS background depending on the context of the situation. Ginny, on the other hand, was new to the CS material, positioning her in the mathematics and math-CT overlaps of the Venn diagram. In her case, she translanguaged and moved into the innermost overlap by trying on the CS technical language or describing her understanding of CS concepts using familiar mathematical vocabulary. Due to their shared background in mathematics and mathematics teaching, Rhoda was able to coordinate concepts between the technical mathematics language that Ginny used and her knowledge of CS.

Conclusion, Limitations, and Implications

This research serves as a foundation for further examination of the ways that K-12 mathematics TCs approach CS material in a CT setting. I presented a case study of two specific K-12 mathematics TCs, one with a math-CS background, and one with just a mathematics background. The results showed that in this setting, the two TCs translanguaged between their disciplinary repertoires and the content of the modules. They also negotiated the overlaps of their disciplinary repertoires to create meaning together.

Limitations

One limitation of this research is that I only examined two TCs so the conclusions of this study are not necessarily comprehensive of all mathematics TCs. I also only chose to analyze the first day of the five-day module because of the development of Rhoda and Ginny's discussion of variables in CS and mathematics. This means that there was an excluded set of information that could have revealed more about the progress of meaning-making between the two TCs over the course of the module. As a result of these limitations, a future study might examine whether similar results would arise for a broader set of K-12 mathematics TCs, perhaps with varying backgrounds in the mathematics and CS subject matter. For instance, would the same interactions arise between two TCs with no prior CS experience? Or two TCs that both had prior CS experience?

An advantage of the study design is that there was some knowledge of the participants' prior experience with CS, and pairs were formed between TCs of varying experience. This could have been one of the contributors to the productive series of conversations between Rhoda and Ginny about how variables work in CS versus math. If Ginny had not been so curious about how

the CS material related back to her mathematics and mathematics teaching experience, and if Rhoda had not had the prior experience with CS, they might not have flushed out the meaning of variables in CS so thoroughly. By examining other pairings in a similar setting, it might be possible to discern whether or not this factor of the module design contributed to the depth of their social semiotic construction of meaning around variables.

Implications

Implications of the theoretical lens. As discussed in the literature review, there are theoretical lenses that consider mathematics and CS as languages. Building from this, I have connected to theories of translanguaging to think about these disciplines. It would be an interesting question to explore whether the use of translanguaging in a disciplinary context could hold if we were working with the overlaps of another set of disciplines. In order to fully understand the ways that disciplinary languages are leveraged by those who have access to one or more of the disciplines in their repertoire, we need to use this theoretical construct in other transdisciplinary contexts. One possibility, for example, could be a study with CS and physics. In fact, an interesting place to examine this research further could be in a course such as the Computational Physics Lab (PH 365-7) at Oregon State University.

In this course, physics students with varying CS experience are paired each class period in a similar fashion to that of the TCs in this module, with a driver and a navigator that switch off throughout the class period. The students are given physics situations closely tied with a concurrent upper-division physics curriculum and asked to code them in Python. As a student currently enrolled in this course, I am curious to explore how my peers and I situate our physics knowledge in a CS setting. I am also interested in my own overlaps of mathematics, physics, and CS, which other students in the course carry to varying degrees in their disciplinary repertoire.

The course explicitly requires the overlap of physics and CS by the nature of the tasks. I am wondering if more discussion could be fostered around certain concepts? For instance, students use For loops as equivalent representations of integrals, which they have used to sum infinitesimal surface charges to find electric potential. This requires an overlap of CS, physics, and mathematics. Another example is when students use arrays as equivalent representations of matrices, which hold information about quantum states. There may be ways to anticipate these overlaps and the resultant affordances and challenges that students in the course face as they create meaning together in pairs. This would be an interesting parallel to the results of this study.

Implications of the module. The case study of Rhoda and Ginny revealed that there were affordances and challenges that arose when they entered a setting that overlapped with knowledge in their respective disciplinary repertoires. For Rhoda, she needed to navigate a new representation of a coding language with which she was familiar. For Ginny, she was introduced to an entirely new discipline, but in the context of her familiar mathematics and mathematics teaching background. The two TCs found ways to leverage the knowledge from their repertoires to explore new territory, but they also demonstrated a pressing need to negotiate the distinctions between their prior knowledge and the new information. One implication of this result is that there may be ways to support the navigation from a TC's disciplinary repertoire to the new content.

In this study, the module provided reflection questions that asked the TCs to consider similarities and differences between mathematics and CS, which gave Ginny explicit space to reference her mathematics knowledge. These moments seemed to support meaning-making and allow the two TCs to unpack some of their own understanding about the concepts. This leads to a follow-up question of how often the TCs would have explicitly drawn upon their previous

mathematics or mathematics teaching knowledge without the prompting of the reflection questions. From the number of instances where Ginny drew specifically on her mathematics teaching knowledge without prompting, it seems that she might have been inclined to do so regardless of the design of the module. However, this prefaces another question as to how one could design a module that anticipated or addressed possible overlaps of disciplinary knowledge.

We saw Ginny branch out from the mathematics knowledge in her repertoire to form connections to CS information, and we saw Rhoda create meaning from CS knowledge in her disciplinary repertoire. It could be advantageous to explore ways for module design to provide opportunities for TCs like Rhoda and Ginny to foreground certain aspects of their repertoires as they enter a new disciplinary setting. In this way, the affordances in meaning-making granted by one's disciplinary repertoire might outweigh the challenges that arise from the negotiation with the overlapping set of information.

Moreover, if potential challenges from the overlapping repertoires can be properly anticipated and prepared for in the module design, there could be further advantages. While the results of this module are not necessarily conclusive enough to make a statement about navigating the challenges in a productive way, I would like to pose an idea that could be further explored in future research. I wonder if Rhoda and Ginny would have faced the same challenges that we were able to see in the results regardless of whether they discussed them explicitly through the prompting of reflection questions or other aspects of the module. I imagine that without guidance from the module, there would have been more internal struggle without the same level of vocalization. If this is the case, then it may be beneficial for the module to intentionally bring these challenges to the open, where they can be handled with the guidance of

prepared instructors and material. With proper guidance, perhaps the challenges in meaning-making and repertoire navigation would create more opportunities for growth and understanding.

Another implication to consider is the overall organization and design of the module to support TCs building from their disciplinary repertoires. Across the first day of the module, Ginny expressed confusion at the nature of their task. This is most likely due to instances where the module, which was new and designed specifically for this research, lacked clarity. Ginny expressed several times that she wanted to have a set task to complete or that she needed more context. Part of this may be because they were introduced to the interest rate scenario at the very beginning that they were meant to work towards over the course of the module, but then they were disconnected from it for the majority of Day 1. Ginny, who was brand new to the coding, may have needed more concrete guidance and context to feel grounded in the work she was doing. Rhoda, on the other hand, was happy to explore the new interface, but she did not realize there were instructions in a provided handout to create and print variables until an instructor informed her of this. Her pathway through the beginning of Day 1 implies that she saw the investment scenario and was ready to begin planning for it from the start. In particular, this may have been what led her to investigate how to create a user input in the block-based script (as described in Theme 1 of the *Results and Discussion* section). That being said, both Rhoda and Ginny were able to collaborate within this space, and the instructors were available to keep them generally on track through the module.

Conclusion

In this paper, I explored how K-12 mathematics TCs interact with each other as they explore mathematics and CS concepts in a CT setting. I introduced a new construct called a disciplinary repertoire, which I defined as the full set of meaning-making and communicative

devices that a person carries, specifically geared towards learning and problem solving within the various academic disciplines. I adapted the theory of translanguaging to describe how the TCs navigated overlaps of their disciplinary repertoires with elements of the modules.

The data I presented supports the conclusion that K-12 mathematics TCs face affordances and challenges based on the knowledge that they carry in their disciplinary repertoire. They hold their disciplinary repertoire as a foundation for new information, but it also creates roadblocks when they cannot rationalize discrepancies between familiar and new information. When working in pairs, the TCs leverage their disciplinary repertoires together. Through discourse they can begin to trace overlaps between their bases of knowledge and the new content. The CT modules provided a setting where both TCs had overlaps with the tasks and their disciplinary repertoires, so they were able to translanguage or compare representations as they worked. Of the implications I discussed in the previous section, one conclusion to highlight is the importance of explicitly acknowledging the disciplinary repertoires of TCs when they enter a transdisciplinary context. It is advantageous to provide opportunities for TCs to foreground their knowledge as well as address the potential challenges that they will face as they navigate the new setting.

References

Denning, P. J. (2009). The profession of IT: Beyond computational thinking. *Communications of the ACM*, *52*(6), 28–30. https://doi.org/10.1145/1516046.1516054

Erwig, M. (2017). *Once upon an algorithm: How stories explain computing*. The MIT Press.

Fereday, J., & Muir-Cochrane, E. (2006). Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *International Journal of Qualitative Methods*, *5*(1), 80–92. https://doi.org/10.1177/160940690600500107

Otheguy, R., García, O., & Reid, W. (2015). Clarifying translanguaging and deconstructing named languages: A perspective from linguistics. *Applied Linguistics Review*, *6*(3), 281–307. https://doi.org/10.1515/applirev-2015-0014

Schleppegrell, M. (2010). Language in mathematics teaching and learning: A research review. *Language and Mathematics Education: Multiple Perspectives and Directions for Research*, 73–112.

Schleppegrell, M. J. (2007). The Linguistic Challenges of Mathematics Teaching and Learning: A Research Review. *Reading & Writing Quarterly*, *23*(2), 139–159. https://doi.org/10.1080/10573560601158461

Swain, J. (2018). *A Hybrid Approach to Thematic Analysis in Qualitative Research: Using a Practical Example*. SAGE Publications Ltd. https://doi.org/10.4135/9781526435477

Vogel, S., Hoadley, C., Ascenzi-Moreno, L., & Menken, K. (2019). The Role of Translanguaging in Computational Literacies: Documenting Middle School Bilinguals' Practices in Computer Science Integrated Units. *Proceedings of the 50th ACM Technical*

Symposium on Computer Science Education, 1164–1170. https://doi.org/10.1145/3287324.3287368

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wing, J. M. (2014). Computational thinking benefits society. *40th Anniversary Blog of Social Issues in Computing, 2014*.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, *60*(4), 55–62. https://doi.org/10.1145/2994591

Appendix

Scenario for Day 1 and Day 2:

You have money to invest that will earn simple interest each year based on the interest rate and principal investment, and you will implement a program to compute how much money you would earn over a specific number of years.

Questions from Reflection #1.1:

1. What do you notice about how the variables and values for the variable are used with the "set" block?
2. How are Python and mathematics similar and different?
3. What are the similarities and differences in the ways Python and mathematics use the ideas of variable and the equal sign?
4. Given the three data types a variable may hold in Python, in what ways, if any ,do these data types relate to mathematics?

Transcript codes:

Instructor = blue text
Ginny introduces math concept = orange text
Linguistic elements:
Variable = pink highlight
Input related: input, Boolean, integer, values = turquoise highlight
Math terms = yellow highlight
CS terms = green highlight
Representations:
Representation = red highlight