UC San Diego UC San Diego Electronic Theses and Dissertations

Title

Bridging the Gap Between Set Theory and Logic: Leveraging Computing as a Mediating Tool for Learning

Permalink https://escholarship.org/uc/item/3297p2zx

Author Martinez, Antonio Estevan

Publication Date 2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

SAN DIEGO STATE UNIVERSITY

Bridging the Gap Between Set Theory and Logic: Leveraging Computing as a Mediating Tool for Learning

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy

in

Mathematics and Science Education

by

Antonio Estevan Martinez IV

Committee in charge:

University of California San Diego

Professor Paul Dawkins Professor Jeffrey Rabin Professor Beth Simon

San Diego State University

Professor Chris Rasmussen, Chair Professor Joanne Lobato Professor Daniel Reinholz

© Antonio Estevan Martinez IV, 2022 All rights reserved The dissertation of Antonio Estevan Martinez IV is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

Chair

University of California San Diego

San Diego State University

DEDICATION

To my parents, your love and encouragement keep me going.

To my sister, you were my first teacher, thank you.

To my tíos and primos, thank you for all the laughs.

To my friends, you all live life to the fullest, I am inspired every day.

To my loving wife, I couldn't have done this without you.

I dedicate this work to you.

EPIGRAPH

Yo no estudio para escribir, ni menos para enseñar, sino sólo por ver si con estudiar ignoro menos.

I do not study to know more, but to ignore less.

- Sor Juana Inés de la Cruz

TABLE OF CONTENTS

Dissertation Approval Page	iii
Dedication	iv
Epigraph	v
Table of Contents	vi
List of Abbreviations	X
List of Figures	xi
List of Tables	xiv
Acknowledgements	xvi
Vita	xix
Abstract	XX
Chapter 1: Introduction	1
Computing in Undergraduate Mathematics Education	4
Set Theory and Logic	5
Confidence, Interest, and the Importance of Student Engagement	6
Research Questions	7
Organization of Dissertation	9
Chapter 2: Theoretical Perspectives and Literature Review	
Theoretical Frameworks	
Realistic Mathematics Education	
PRIMM	
Instrumental Genesis	
Hypothetical Learning Trajectories	
Socio-Constructivism	
Literature Review	
Set Theory	
Teaching Set Theory	
Mathematical Logic	47
Computing as a Tool for Mathematics Teaching and Learning	

Students' Affective Experiences and Mathematical Identity	55
Research Questions Revisited	60
Chapter 3: Methodology	61
Conjecture-Driven Teaching Experiment Overview	63
Participants and Context	63
Participant Overview	65
Data Collection and Analysis	68
Approaches to Analysis	74
Research Question 1	76
Research Question 2	77
Research Question 3	79
Mathematical and Technological Confidence Survey	79
Three Features of Students' Affective Experiences	81
Analysis	83
TE Sessions and Example Tasks	85
TE Session 1	86
TE Session 2	91
TE Session 3	93
TE Session 4	99
TE Session 5	101
Chapter 4: Instrumented Action Schemes	103
Propositional Statements	105
Impossible-to-Answer Scheme	107
Flexible Propositional Statement Scheme	112
Set Intersection	118
Filter Every Element Scheme	119
Monitor Change in the Cardinality Scheme	128
Subsets	135
Determine Set Equality Scheme	137
Verify Each Element Scheme	146

Chapter 5: Hypothetical and Actual Learning Trajectories	
Logical Operators	
Situational Activity	
Referential Activity	
General Activity	
Set Operations and Properties of Subsets	
Situational Activity	
Referential Activity	
General Activity	
Pre and Post Mathematical Content Survey Results	
Group 1	
Group 2	
Group 3	
Group 4	
Summary	
Chapter 6: Confidence and Mathematical Identity	
Group 1	
Haven	
Judith	
Palmer	
Group 2	
Saul	
Eugene	
Leo	
Group 3	
Delia	
Juliana	
Group 4	
Julian	
Alonso	

All Students	
Chapter 7: Conclusion	
Summary of Findings	
Research Question 1	
Research Question 2	
Research Question 3	
Implications	
Limitations	
Future Research	
Final Thoughts	
Appendix A: Screening Survey	
Appendix B: List of Codes by Research Question	
Appendix C: Mathematical Content Survey	
Appendix D: Confidence and Interest Survey	
Post-Study Free-Response Questions	
References	

LIST OF ABBREVIATIONS

- ATMLQ Attitudes to Technology in Mathematics Learning Questionnaire
- DFWI D, F, Withdrawal or Incomplete
- HLT Hypothetical Learning Trajectory
- ITP Introduction to Proof
- RME Realistic Mathematics Education
- RQ Research Question
- STEM Science, Technology, Engineering and Mathematics
- TE Teaching Experiment

LIST OF FIGURES

Figure 2.1: Four Levels of Activity (Gravemeijer, 1999, p. 235)	17
Figure 2.2: Grocery List Item Task	21
Figure 2.3: Instrumental Genesis	25
Figure 2.4: Teacher and Student Diagrams	45
Figure 2.5: Shirts and Pants Combinations Task	54
Figure 3.1: Timeline of Data Collection	71
Figure 3.2: Development of Conjecture through Research and Design	73
Figure 3.3: TE Session 1 Task 1	87
Figure 3.4: TE Session 1 Task 2	
Figure 3.5: TE Session 1 Task 3	
Figure 3.6: TE Session 1 Task 4	91
Figure 3.7: TE Session 2 Task 1	
Figure 3.8: TE Session 2 Task 2	
Figure 3.9: TE Session 3 Task 1	94
Figure 3.10: TE Session 3 Task 1 – Output	94
Figure 3.11: TE Session 3 Task 2	
Figure 3.12: TE Session 3 Task 2 – Output	96
Figure 3.13: TE Session 3 Task 3	97
Figure 3.14: TE Session 3 Task 4	
Figure 3.15: TE Session 3 Task 5	97
Figure 3.16: TE Session 3 Task 6	
Figure 3.17: TE Session 4 Task 1	
Figure 3.18: TE Session 4 Definition 1	
Figure 3.19: TE Session 4 Task 2	
Figure 3.20: TE Session 5 Task 1	
Figure 3.21: TE Session 5 Task 2	
Figure 4.1: Main Study Unknown Proposition Task	
Figure 4.2: Pilot Study Unknown Proposition Task	

Figure 4.3: Kristal and Adeline's Punnett Square	114
Figure 4.4: Biological Example of a Punnett Square - Recessive Blue Eye Color	116
Figure 4.5: Truth Table Generated by Group 3	118
Figure 4.6: Set Intersection Task	119
Figure 4.7: Leo's Set Intersection Diagram	120
Figure 4.8: Eugene and Saul's Set Intersection Diagrams	121
Figure 4.9: Diagram Representation of Alonso's Comparing Elements Method	130
Figure 4.10: Alonso's Pseudocode for Set Intersection	132
Figure 4.11: Group 4 Number Theory Subset Task	136
Figure 4.12: Group 2 Number Theory Subset Task	136
Figure 5.1: Each Group's Progress through TE	151
Figure 5.2: First Task for Groups 1, 2, and 3	154
Figure 5.3: Introducing Logical Propositions and Operators	156
Figure 5.4: Juliana's Evaluation of a Compound Proposition	160
Figure 5.5: Alonso's Evaluation of a Compound Proposition	161
Figure 5.6: Group 1 General Proposition Task	164
Figure 5.7: General Proposition Task for Groups 2, 3, 4	167
Figure 5.8: Task Utilizing the Logical Operator 'or' in a For Loop	169
Figure 5.9: Venn Diagram Used with Group 1	179
Figure 5.10: Number Theory Problem	184
Figure 5.11: Proof with Group 4	192
Figure 6.1: Haven's Pre and Post ATMLQ Results	233
Figure 6.2: Judith's Pre and Post ATMLQ Results	238
Figure 6.3: Palmer's Pre and Post ATMLQ Results	244
Figure 6.4: Saul's Pre and Post ATMLQ Results	251
Figure 6.5: Eugene's Pre and Post ATMLQ Results	256
Figure 6.6: Leo's Pre and Post ATMLQ Results	262
Figure 6.7: Delia's Pre and Post ATMLQ Results	269
Figure 6.8: Juliana's Pre and Post ATMLQ Results	273
Figure 6.9: Julian's Pre and Post ATMLQ Results	

LIST OF TABLES

Table 2.1: Schemes, Techniques and Concepts	29
Table 2.2: Hypothetical Example of a Student's Subset Scheme	30
Table 3.1: Coding Framework to Document Affective Experiences	83
Table 4.1: Haven's Impossible-to-Answer Scheme	111
Table 4.2: Kristal and Adeline's Flexible Propositional Statement Scheme	117
Table 4.3: Leo's Filter Every Element Scheme	128
Table 4.4: Alonso's Monitor Change in the Cardinality Scheme	135
Table 4.5: Julian's Determine Set Equality Scheme	142
Table 4.6: Eugene's Determine Set Equality Scheme	146
Table 4.7: Leo's Verify Each Element Scheme	149
Table 5.1: Pre-Study Group 1 Set Theory Questions	196
Table 5.2: Post-Study Group 1 Set Theory Questions	197
Table 5.3: Pre-Study Group 1 Logic Questions	198
Table 5.4: Post-Study Group 1 Logic Questions	199
Table 5.5: Pre-Study Group 1 Logic and Set Theory Questions	200
Table 5.6: Post-Study Group 1 Logic and Set Theory Questions	201
Table 5.7: Pre-Study Group 1 Python Question	202
Table 5.8: Post-Study Group 1 Python Question	203
Table 5.9: Pre-Study Group 2 Set Theory Questions	204
Table 5.10: Post-Study Group 2 Set Theory Question	205
Table 5.11: Pre-Study Group 2 Logic Questions	
Table 5.12: Post-Study Group 2 Logic Questions	207
Table 5.13: Pre-Study Group 2 Logic and Set Theory Questions	
Table 5.14: Post-Study Group 2 Logic and Set Theory Questions	209
Table 5.15: Pre-Study Group 2 Python Question	210
Table 5.16: Post-Study Group 2 Python Question	211
Table 5.17: Pre-Study Group 3 Set Theory Questions	212
Table 5.18: Post-Study Group 3 Set Theory Questions	213

Table 5.19: Pre-Study Group 3 Logic Questions	214
Table 5.20: Post-Study Group 3 Logic Questions	214
Table 5.21: Pre-Study Group 3 Logic and Set Theory Questions	215
Table 5.22: Post-Study Group 3 Logic and Set Theory Questions	216
Table 5.23: Pre-Study Group 3 Python Question	217
Table 5.24: Post-Study Group 3 Python Question	218
Table 5.25: Pre-Study Group 4 Set Theory Questions	219
Table 5.26: Post-Study Group 4 Set Theory Questions	220
Table 5.27: Pre-Study Group 4 Logic Questions	221
Table 5.28: Post-Study Group 4 Logic Questions	221
Table 5.29: Pre-Study Group 4 Logic and Set Theory Questions	222
Table 5.30: Post-Study Group 4 Logic and Set Theory Questions	223
Table 5.31: Pre-Study Group 4 Python Question	224
Table 5.32: Post-Study Group 4 Python Question	224
Table 5.33: Pre-Study and Post-Study Results (number of correct responses)	225
Table 6.1: Haven's Affective Characterization	230
Table 6.2: Judith's Affective Characterization	236
Table 6.3: Palmer's Affective Characterization	242
Table 6.4: Saul's Affective Characterization	247
Table 6.5: Eugene's Affective Characterization	253
Table 6.6: Leo's Affective Characterization	259
Table 6.7: Delia's Affective Characterization	
Table 6.8: Juliana's Affective Characterization	271
Table 6.9: Julian's Affective Characterization	277
Table 6.10: Alonso's Affective Characterization	
Table 6.11: Total Scores Across all Three Categories	

ACKNOWLEDGEMENTS

First, I must thank my mentor and advisor, the chair of my dissertation committee, Chris Rasmussen. Your support and encouraging feedback at every stage of the dissertation was instrumental in the completion of this research project. Moreover, you have shown me what it means to be a thoughtful researcher, supportive colleague, and inquisitive being. I look forward to continuing our work together.

I would also like to express my gratitude to my entire dissertation committee for their support, feedback, and years of guidance throughout the doctoral program. Joanne Lobato, thank you for your years of leadership, support and helping me think through the methodological components of my study. Jeff Rabin, I appreciate all your time spent reading my work and providing me with detailed feedback that strengthened this study. Daniel Reinholz, thank you for your willingness and enthusiasm to assist in any way. You have constantly encouraged me to think critically and have also served as an academic role model throughout my time in the program. Beth Simon, thank you for your insight, guidance, and encouragement. Without your computer science education expertise, this project would not have gotten off the ground. A special thanks to Paul Dawkins, the sixth outside member on my dissertation committee. Your work inspired me to pursue a study investigating students' conceptions related to mathematical logic. Your feedback and suggestions throughout the duration of this research project were invaluable.

I am also grateful for everyone in the Mathematics and Science Education (MSED) program. I have felt constantly supported by the MSED faculty and have appreciated all the encouragement and guidance from the current and previous students in the program. A

xvi

special thanks to Matt Voigt and Naneh Apkarian for setting incredible examples for me to strive toward, as well as supporting me throughout the program when I needed guidance. I am also thankful for the friendships that I have been able to develop within the built-in community of my cohort; Ted, Makenna, Ernesto, Kevin, Nicole, and Mike, I appreciate all of you and look forward to seeing what you accomplish in the future. I know you will all do incredible things. Mike Foster, thank you for supporting me throughout the data collection process, your insight and feedback were incredibly helpful. Nicole Suarez and Kevin Pelaez, your support helped me cross the finish line. I am indebted to both of you and am so thankful that we got to graduate together. Colin McGrane, I appreciate all your hard work and support as a colleague. Your willingness to take additional responsibilities on our other projects made things easier for me as I finished my dissertation. Talia LaTona-Tequida, words cannot express how thankful I am to have you as a colleague and one of my dearest friends. What a fantastic journey it has been since we completed our bachelor's degrees together. I look forward to collaborating with you as you finish your doctoral degree and beyond. I must also thank Deb Escamilla for her determined and tireless efforts to support me and the other students and faculty in the MSED program.

To my family, your love and support have kept me strong and on an unwavering path. I would like to thank my parents, Antonio Martinez, Rosanna Jackson, and Kirby Jackson from the bottom of my heart for their unconditional love, incredibly timed words of encouragement, and setting examples for me to look up to. This achievement would not be possible without you and is as much yours as it is mine. To my sister, Lauren Martinez, you have always been a source of strength as you define what it means to work diligently.

xvii

You have been a cornerstone in my life, from lessons learned to lessons taught, thank you for always being there. To my tios, many of whom have served as role models in my life, thank you for showing me that there are no limits to what one can accomplish. To my primos, thank you for all your kind words of encouragement over the years and all the laughs that we have had together.

I must also thank my core group of friends; it is hard to explain the impact that you have had on my life. I am grateful to have such dedicated, funny, steadfast, and considerate men around me. I admire each and every one of you and cherish the friendship that we have been able to develop, cheers to you all.

Finally, to my loving and supportive wife, Haley Martinez. I am truly thankful for the sacrifices that you have made to support me in my journey in accomplishing this milestone. Your advice has spanned the wide range of available encouragements from, "just take it one step at a time," or "I know you've got what it takes to get this done," to my favorite, "let's get happy hour." You always found a way to make things better, I cannot wait to do the same for you as you embark on your next journey. I love you, thank you for everything.

VITA

2016	Bachelor of Arts, California State University, Sacramento
2017 - 2018	Teaching Assistant, San Diego State University
2018	Master of Science, San Diego State University
2018 - 2022	Research Assistant, San Diego State University
2022	Doctor of Philosophy, University of California San Diego, San Diego State University

PUBLICATIONS

- Martinez, A., Gehrtz, J., Rasmussen, C., LaTona-Tequida, T., & Vroom, K. (2022). Course coordinator orientations toward their work and opportunities for professional development. *Innovative Higher Education*, 47, 327-346. https://doi.org/10.1007/s10755-021-09579-1.
- Williams, M., Apkarian, N., Uhing, K., Martinez, A., Rasmussen, C., & Smith, W. (2022). In the driver's seat: Course coordinators as change agents for active learning in university Precalculus to Calculus 2. *International Journal of Research in Undergraduate Mathematics Education*, 8, 121-148. https://doi.org/10.1007/s40753-021-00153-w
- Martinez, A., & Pilgrim, M. E. (2021). Online grading platforms: Establishing a connection between course coordination and local data. *International Journal of Mathematical Education in Science and Technology*. https://doi.org/10.1080/0020739X.2021.1937731.
- Voigt, M., Rasmussen, C., & Martinez, A. (2021). The refiguring of students' mathematical identities: A mixed methods study of three tailored calculus courses. *International Journal of Mathematical Education in Science and Technology*. https://doi.org/10.1080/0020739X.2021.1940331

ABSTRACT OF THE DISSERTATION

Bridging the Gap Between Set Theory and Logic: Leveraging Computing as a Mediating Tool for Learning

by

Antonio Estevan Martinez IV

Doctor of Philosophy in Mathematics and Science Education

University of California San Diego, 2022 San Diego State University, 2022

Professor Chris Rasmussen, Chair

Undergraduate mathematics education research focused on Introduction to Proofs courses has gained traction as more students are experiencing challenges in their proofbased courses. While studies have analyzed the teaching and learning of proofs, there is a growing need for research in students' understanding of mathematical logic and set theory because of the foundational nature of these topics in more advanced mathematics courses. In this dissertation, I present a teaching experiment that explores how computing/programming can be leveraged to facilitate and strengthen the connection between set theory and logic. Programming is a focus of this work due to the growing need for undergraduate Science, Technology, Engineering and Mathematics majors to be prepared for their future careers, which will likely entail the utilization of computing in some form. The purpose of this study is to take a step in the direction of where undergraduate education is headed and better understand how the use of computing can potentially fit into the mathematics curriculum.

There are three main dimensions of this study. The first is a look at how programming can influence students' in-the-moment ways of reasoning about mathematical set theory and logic. The second takes a step back to consider the students' advancing mathematical activity and growth over the course of a multi-session long teaching experiment. The third is a focus on the students' affective experiences, as noncognitive factors such as confidence, interest and self-efficacy are analyzed to characterize the shifting nature of the students' mathematical identities. The results of this study indicate that the students in my study were able to leverage computing as an accessible onramp to the fundamental ideas related to set theory and logic. Moreover, my findings show that computing can have a positive effect on one's sense of confidence and interest in relation to mathematics and programming.

Chapter 1: Introduction

In this study, I focus on the development and use of concepts commonly taught in an Introduction to Proofs (ITP) course, specifically set theory and mathematical logic. In a survey of ITP courses at 179 different very high research activity and high research activity (R1 and R2) universities, David and Zazkis (2019) found that 81% of the courses covered the same main topics of mathematical logic, methods of proof, and various topics related to set theory. For many of the students, this is the first time they are introduced to this material, providing a rich opportunity for research regarding student mathematical thinking and overall experience working with mathematical material that they would encounter in an ITP course. Considering ITP research, most of the work being done has focused on students' challenges and difficulties with reading and writing proofs (e.g., Ko & Knuth, 2009; Lew & Mejía-Ramos, 2019; Moore, 1994; Weber, 2001). For this reason, I have decided to focus my efforts away from proof writing and comprehension, and instead target set theory and logic, topics that I consider essential to an undergraduate mathematician's repertoire. I consider these topics to be essential given that much of the mathematical activity in upper division mathematics courses begins with the assumption that students understand what it means for something to belong to a set. For example, in proof writing we often start our proofs with something along the lines of, "Let x be in [some set]." From there, students are expected to use deductive reasoning, and logical relations to arrive at a conclusion for whatever it is they are interested in. Without a firm grasp on what it means for an element to be a member of the set, how do we expect

students to fully understand the power of proof writing, set relations, and other advanced topics commonly taught in upper division mathematics?

Traditionally, set theory and mathematical logic have been compartmentalized in the curriculum and hence taught as individual units in a standard ITP course. In this study, the students were introduced to set theory and logic together. The conjecture that I plan to explore in connecting these two topics is that computing can be leveraged to facilitate and strengthen the connection between set theory and logic in a way that empowers learners in their learning of mathematics, without using proof writing as a tool to explore these ideas. A more developed version of the conjecture can be found in Chapter 3.

I introduce an emphasis on computing for three main purposes. The first is that data science, and computer science more broadly, are becoming increasingly valuable skills in an age where fields such as artificial intelligence and machine learning are in position to be the most influential drivers of change for our society (World Economic Forum, 2016). The second purpose draws on Lockwood et al.'s (2019) commentary on the relationship between computing and students' mathematical activity in that computing will be a focus for mathematics educators soon to come. By encouraging the use of computational software as a tool for students to learn and connect ideas in undergraduate mathematics, I am taking a step in the direction of where mathematics education will be going in the coming decade. Lastly, the content of set theory and logic lends itself to the Python environment and hence programming may serve as a useful means for learning this content.

Taking note of students' progression through undergraduate mathematics courses provides many opportunities to investigate and better understand students' mathematical growth and sophistication. Rasmussen et al. (2005) describe the growth of students' mathematical thought processes, participation, and mathematical contributions as their advancing mathematical activity. Rasmussen and colleagues proffer this terminology as an alternative to what had previously been described as the transition from elementary to advanced mathematical thinking (Tall, 1991, 1992). The benefit of using "advancing mathematical activity" is that it provides an asset-based perspective that positions all students as capable learners and doers of mathematics compared to previous approaches where students were binned as either elementary thinkers or advanced thinkers. Grouping students into bins is problematic for two main reasons. First, binning students as "elementary thinkers" carries a connotation that the students' abilities are fixed. Second, it encourages researchers to dissect the differences between the two groups instead of finding ways to understand why students are experiencing difficulties in the first place. The comparison between groups of students and measuring the "achievement gap" of student success, historically a comparison between marginalized students and middle-class white students, is known in the literature as *gap-gazing* (Gutiérrez, 2008; Gutiérrez & Dixon-Román, 2010; Young et al., 2018). By making the decision not to "gap-gaze," both in the gap between experts and novices and between students with different identities and histories, it is possible to meet the students where they are, rather than starting at an a priori expectation of where others say they should be.

Computing in Undergraduate Mathematics Education

The call for research to investigate the ways in which technology and, more specifically, computers can be used to facilitate the teaching and learning of mathematics is not new (e.g., Fey, 1989; Papert, 1980; Perlis, 1962). However, previous approaches to address this need have varied in the medium through which the technology is used such as video games (Levine et al., 2020), graphing calculators (Leng, 2011) and virtual reality (Bogusevschi et al., 2020). Also, these areas of technology integration have been a focus at multiple levels throughout the educational system (Ball et al., 2018; Oates, 2011). Lockwood and Mørken (2021) explicitly call for research in undergraduate mathematics education with a focus on machine-based computing. They define machine-based computing as "the practice of precisely articulating algorithms that may be run on a machine" (p. 2). An important aspect of machine-based computing is the development of an algorithm rather than its performance or implementation alone. This is an important component of machine-based computing that distinguishes it from other forms of computing such as the use of Desmos where one may only be required to input parameters to create a visual representation of a function. Thus, Lockwood and Mørken consider machine-based computing to be a separate construct that includes the use of programmable calculators, writing packages in Geogebra, and using text-based or block-based programming languages. For the purposes of my dissertation, I will focus on programming using the text-based language known as Python.

In their research commentary on computing in undergraduate mathematics education, Lockwood and Mørken (2021) highlight how a focus on programming can serve

as a tool for understanding student thinking and learning. They hypothesize that the use of programming can serve to strengthen student reasoning but could also pose a risk of serving as "another rote procedure that students may not understand" (p. 5). For this reason, I will integrate tasks involving writing code in Python, aimed to pique students' curiosity and support their advancing mathematical activity. I will accomplish this by providing opportunities to use Python in ways that will support the students in their reasoning about set theory and logic not always possible with a standard pencil and paper approach. From the students' perspective, my goal is to motivate an interest and appreciation of the power of computing. From a mathematics education researcher's perspective, my goal is to explore the potential and possibilities that programming can offer to enrich student learning as well as adding to the literature on the impact that programming, using a text-based language, has on students' reasoning in undergraduate mathematics.

Set Theory and Logic

Understanding students' conceptions of mathematical logic has seen some growing interest, but set theory has been particularly under-researched at the undergraduate level. Of the studies that have been conducted, which will be reviewed in Chapter 2, there has not been an analysis conducted that measures the growth over time of an undergraduate student's conceptions related to set theory. To this point, I believe it is crucial that we move beyond research that documents what students are unable to do, and instead focus on what students find to be helpful for their learning. In a recent review conducted by Dawkins et al. (2020), they explored the connections made by ITP textbook authors, and

what they considered to be helpful connections between set theory, logic and proving. Their results indicate that the strongest link exists between logic and proving, with some evidence of textbook authors using logic to explain set operations. Dawkins et al. (2020) suggest that utilizing a set-based approach to logic and proving can be a valuable approach for the teaching and learning of the standard ITP curriculum. For this reason, and those made previously, I believe it is important that additional research be conducted regarding students' conceptions of set theory and logic, as these two topics are integral in the upper division mathematics curriculum and are foundational to proofs and proving.

Confidence, Interest, and the Importance of Student Engagement

Introductory undergraduate mathematics serves as a gate-keeping experience, delaying student advancement in their STEM (science, technology, engineering, and mathematics) degree - or pushing them out of STEM altogether (Weston et al., 2019). As Koch and Drake (2018) document in their report on 36 higher education institutions across the United States, 34.3% of students received a D, F, Withdraw or Incomplete (DFWI) in calculus. The statistics for Black or African American students and Hispanic or Latinx students were even worse with DFWI rates of 47.8% and 47.9%, respectively. Given these dismal statistics, it is important that we find ways to improve the undergraduate student experience, which entails improving the curriculum to meet the needs of today's students. I believe that by infusing computing into the undergraduate curriculum, we can find ways to deliver engaging and high-quality instruction that not only prepares students for their future careers in STEM, but also fosters a sense of confidence and enjoyment in doing mathematics. This is particularly important for historically marginalized groups of students as they are often pushed out of STEM (Weston et al., 2019), thus not even having the chance to interact with programming in upper-level courses.

As a research community, it is imperative that the studies we develop are rich and lead to insightful information to improve the learning of mathematics through updating curriculum, developing effective methods of instruction, understanding student thought process, etc. However, it is also important that the studies we develop consider the research participants as human beings that bring their lived experiences to the experiment. For many students, mathematics brings about feelings of anxiety which is commonly associated with negative performance in mathematics (Namkung et al, 2019) and as a result, potentially negative mathematical identities. On the other hand, confidence, interest and other positive affective experiences have been found to be main contributors of students' positive mathematical identity development (Cribbs et al, 2015, Renninger, 2009). Therefore, I believe it is important to develop our studies with the intention of being mindful of our research participants' affective experiences, and how our intervention studies may contribute, either positively or negatively, to their sense of mathematical identity.

Research Questions

For this dissertation I explore how computing can be leveraged to motivate, facilitate, and strengthen the connections between set theory and logic. There are three main research foci for this study. The first is a fine-grained analysis of how programming may influence student reasoning and learning of mathematical set theory and logic. This was facilitated using the instrumental approach (Artigue, 2002; Guin & Trouche, 1998;

Trouche, 2004), an analytic framework that can be used to document students' mathematical activity while using a piece of technology. More details about the instrumental approach will follow in Chapter 2. The second focus takes a step back to consider students' advancing mathematical activity and growth over the course of a multi-session teaching experiment. I will consider the students' Actual versus Hypothetical Learning Trajectories (Simon, 1995) to address this focus. The third is a focus on students' affective experiences and how they may contribute to their mathematical identity as non-cognitive factors such as interest, confidence and enjoyment were analyzed in relation to the social and personal identities of the students in this study. These foci map onto the three research questions driving this study:

- 1. How does Python mediate students' learning of important concepts in set theory and logic?
- 2. What characterizes students' increasingly sophisticated ways of reasoning about set theory and logic?
- 3. How does the use of technology to learn mathematics in a small-group collaborative setting influence students' affective experiences?

These research questions will be elaborated on in more detail at the end of Chapter 2. I see the significance of this work having four major impacts. The first is that the results of this study will inform how researchers, educators, and practitioners may infuse computing into their own mathematics curriculum, particularly for those interested in ITP material. Second, this project is connecting mathematical logic and set theory in new and innovative ways to investigate how we might be able to teach these topics without the use of proof. Third, my utilization of computing to teach mathematics is fairly new territory, both in content and methodology; my goal with this work is to provide an existence proof that a study like mine can result in rich and meaningful mathematical activity for undergraduate students who may not normally experience computing. Lastly, some students in my study found the use of computing to learn mathematics to be an empowering and positive experience; understanding how and why this happened could be beneficial for other teachers, researchers, and instructional designers.

Organization of Dissertation

In Chapter 2, I present the theoretical and analytical frameworks that I am utilizing in this study. I review the instructional design perspectives that guide the development of the tasks for the teaching experiment sessions. Then I situate these frameworks within my broader theoretical approach regarding what it means to learn mathematics. These frameworks also shape how I will proceed with analysis, which will be discussed in Chapter 3. In Chapter 2 I also highlight the literature in four research domains related to my study. First, I review the research on students' conceptions of set theory and the teaching of set theory. The second area of research I present is related to students' conceptions on mathematical logic. Third, I present the research that has investigated computing as a tool for mathematics teaching and learning. Last, I review the literature that has shaped my perspective on affect and mathematical identity.

In Chapter 3, I present the methodology of my study. First, I present a top-level review of my teaching experiment which includes a description of the participants and context of my study. I also provide a description of the proposed data collection process

and approach for analysis. For each research question I will utilize different, yet compatible, analytic approaches and thus each approach is specified in Chapter 2. Lastly, I present the tasks that I will use for the teaching experiment sessions. These tasks have been informed by a three-session long pilot study that occurred in the last two weeks of January, 2021. The main purpose of the pilot study was to find an effective way to present the material to the student groups in the study which included streamlining the tasks used and understanding how to navigate between screen sharing, Jamboard, and the integrated development environment that was being used to run the Python code. Only a couple of examples of student work from the pilot study will be showcased in Chapter 4, as too many of the tasks used in the pilot study were changed between the pilot and the main study. In the presentation of the tasks, I provide a description of the purpose for each task and how it was informed by my chosen instructional design theories.

In Chapter 4 I present six schemes, or in-the-moment ways of reasoning, that highlight the effect that Python had on the students' understanding of the material. These schemes pertain to three topics: (a) logical propositions, (b) set intersection, and (c) subsets. Two schemes are presented for each mathematical topic. The purpose of highlighting these schemes is to document the influence that programming had on students' ways of reasoning and to set a foundation for future work to better understand the prevalence of these schemes across other student experiences.

In Chapter 5 I discuss the learning goals that I wanted the students to take away from participating in this study by presenting the instructional task sequence and related student work to document the developmental progression of their learning process. As an

added measure to document what, and how much, the students learned, I also present the results of a pre and post-study mathematical content survey in which I asked the students questions related to set theory, logic and programming.

In Chapter 6 I highlight the students' affective experiences and how their experiences participating in my study may have shaped their ever-changing mathematical identities. To document any shift that was due to participating in my study, I administered a pre and post questionnaire that asked about their confidence and interest related to mathematics, computers and programming, and programming to learn mathematics. Qualitatively, I also considered their responses to errors, their beliefs, attitudes, and emotions, as well as their senses of self-efficacy.

With Chapter 7 I conclude this work and provide a summary of my findings as well as a discussion on the limitations of my study. I also highlight the implications of my work and detail some potential avenues for future research.

Chapter 2: Theoretical Perspectives and Literature Review

The purpose of this dissertation is to understand how a text-based programming language can be used by students as a tool for learning mathematical set theory and logic. Specifically, I am focused on answering the following three research questions:

- How does Python mediate students' learning of important concepts in set theory and logic?
- 2. What characterizes students' increasingly sophisticated ways of reasoning about set theory and logic?
- 3. How does the use of technology to learn mathematics in a small-group collaborative setting influence students' affective experiences?

In this chapter, I first present the theoretical frameworks informing the design of the tasks used for the study as well as describe how my theory of learning is situated in relation to these design frameworks. I also present the literature on the analytical frameworks that I am using as tools to answer my research questions. In the second section I present a review of the literature on computing in mathematics education, students' conceptions of set theory, students' conceptions of mathematical logic, and conclude with a discussion on students' affective experiences and mathematical identity.

Theoretical Frameworks

Central to my approach to the teaching and learning of mathematics is that learning does not occur in a vacuum; one's previous experiences and perception of the world shape how they come to understand new material, whether that be mathematics or anything else. As described by Freudenthal (1971), mathematics is a human activity where learners organize ideas in increasingly sophisticated ways. The design of tasks to facilitate this progress is paramount. In the following subsections I provide a detailed description of the instructional theory guiding the development of tasks for my teaching experiment and highlight how the instructional theory is situated within the socio-constructivist perspective of what it means to learn mathematics.

Realistic Mathematics Education

Freudenthal's (1971) work on mathematics as a human activity eventually led to the development of the instructional theory known as Realistic Mathematics Education (RME) (Freudenthal, 1991; Gravemeijer, 1999; Gravemeijer, 2020a; Treffers, 1987). I draw on this theory in the development of the tasks that were used to guide my Teaching Experiment (TE). RME refers to the design of instruction to be realistic in the sense that the material is imaginable by the students and/or relevant to their experiences (van den Heuvel-Panhuizen, 2003). Foundational of the RME approach to the teaching and learning of mathematics are three main heuristics: (a) guided reinvention, (b) didactical phenomenology, and (c) emergent modeling (Gravemeijer, 2020b). In broad terms these three heuristics refer to the reinvention and ownership of the mathematics by the students through their own mathematical activity. My study uses these three heuristics as the guiding principles in creating the TE tasks that were used for my dissertation. I introduce guided reinvention and didactical phenomenology first to describe the role they play in the instructional design theory and provide a more detailed description of the emergent modeling heuristic as it will strongly influence the design of the TE tasks.

Guided reinvention is the process of supporting the students in a mathematical activity that leads them to a particular mathematical idea or concept that they "discover" for themselves. Of course, students are not expected to reinvent mathematics in a single class period or single TE session, but as the instructor and task designer, there are ways to guide the students through a sequence of tasks that ultimately leads them to a particular theorem or idea that is then given the formal mathematical name. Gravemeijer (2020a) explains that the creation of tasks to help students come to a point of discovery may be developed based on the history of the mathematics itself (i.e., known challenges and barriers that have been documented) and suggests that the task designer start with the informal mathematical activity that students might be bringing to the classroom or learning environment to anticipate more formal mathematics. In an instructional sequence presented by Rasmussen et al. (2019), they provide an example of how guided reinvention can be accomplished in an undergraduate mathematics differential equations class. The guidance and support to lead to the reinvention is described by Rasmussen et al. (2019) as, "the instructor nudges students toward a modified differential equation...the instructor helps students recognize contextual deficiencies...and so encourages the students to develop their own model" (p. 11). Highlighting points of interest and encouraging students to try something different are two essential instructor moves to facilitate guided reinvention.

Didactical phenomenology is the second heuristic of RME. In a sense it is the development of a need for a particular piece of mathematics that will help the learner in creating something to advance their mathematical sophistication. That is, provided with evidence of a student's particular way of reasoning, the instructor may then find ways to
motivate the need for a more robust method or process to further develop the mathematics. As the instructor, one must present the learner with situations that would call for the development of a concept to help with their own organization of concepts and relationships between those concepts. In a manuscript highlighting the important and distinguishing factors of didactical phenomenology for instructional design, Larsen (2018) summarizes didactical phenomenology as a method that "tells the designer that an instructional sequence meant to support the learning of a piece of mathematics should be situated in a context that can be productively organized by students using that piece of mathematics" (p. 25). In other words, there must be a perceived need by the students to use a particular mathematical idea or concept to make sense of the overall learning goal that is being targeted by the instructional designer.

The third heuristic, emergent modeling, is the gradual process in which learners develop more-sophisticated mathematical conceptions. Emergent modeling represents the gradual development and use of sub-models to shift from "model-of" methods for solving informal mathematics to "model-for" methods used for more sophisticated ways of reasoning. Zandieh and Rasmussen (2010) define a model as:

Student-generated ways of organizing their activity with observable and mental tools. By observable tools we mean things in their environment, such as graphs, diagrams, explicitly stated definitions, physical objects, etc. By mental tools we mean the ways in which students think and reason as they solve problems—their mental organizing activity. (p. 58)

To provide some background as to the development of the model-of, model-for, and submodel approaches, Gravemeijer (2020b) explains that meaning is not embedded within the mathematical symbols themselves, but is created by the learner. The development of sub-

models was brought about to support students in their construction of their own models of mathematics which are made up of smaller more-comprehensible sub-models. This type of construction of the sub-models is reflective of the history of mathematics in that the symbolic representation of mathematics co-evolved with the mathematics itself. This process is thus described as having a recursive relationship where a smaller model is developed to lead to more mathematics that requires another smaller model that leads to more mathematics and so on.

In a sense, one may think of the development of sub-models as the transition from a model-of specific or more informal problem tasks to a model-for more conceptual mathematics. An important aspect to highlight here is that each sub-model is built off the previous sub-model. The model-of/model-for transition is described in more detail by Gravemeijer, et al. (2000) as levels of activity and shown in Figure 2.1.



 Level 1: Activity in the task setting, in which interpretations and solutions depend on understanding of how to act in the setting (often out of school settings)
 Level 2: Referential activity, in which models-of refer to activity in the setting described in instructional activities (posed mostly in school)

Level 3: General activity, in which models-for make possible a focus on interpretations and solutions independently of situation-specific imagery Level 4: Reasoning with conventional symbolizations, which is no longer dependent

on the support of models-for mathematical activity

Figure 2.1: Four Levels of Activity (Gravemeijer, 1999, p. 235)

To highlight each level of activity, I will use the instructional sequence presented by Rasmussen et al. (2019) which documents the model-of/model-for transition in undergraduate differential equations (DE) students' reinvention of bifurcation diagrams. As Figure 2.1 shows, the first level is *situational activity* where the learners are using a sub-model to achieve a context-specific goal. This activity is grounded in context-specific activity to describe a particular problem situation that is experientially real for the students. At this level, students explore a realistic (from the RME perspective) method/model to describe a situation that serves as a foundation for future mathematical activity. One of the more important aspects of this level is that the situational activity should be something that the students can come back to as they progress in more sophisticated ways of reasoning. In the instructional sequence by Rasmussen et al. (2019), they begin by presenting the students with an owl population problem and help students establish the phase line as a one-dimensional representation of the entire solution space. This, in effect, serves as the context-specific situational activity where students can think of the specific solution space in terms of the phase line.

The next level is *referential activity*. At this level, students are expected to move away from the context-specific model and take a slightly more abstracted perspective of the mathematical activity. However, any abstraction must not be too far removed from the situational activity so that the students can still see the relationship with the model-of approach. The important aspect of this level of activity is that students are both able to refer to the situational activity as well as interpret what might be considered new mathematics or at least a new representation of how to describe the same material. In the DE example, the referential activity is the construction of the autonomous derivative graphs where the students are asked to graph dy/dt versus y. Note that the referential activity is not too far of a step back to the phase line as the phase line can be overlaid onto the autonomous derivative graph to represent the behavior of the rate of change function. With this information, the goal is that the students will be able to relate the autonomous derivative graph to the solution space as they did with the phase line.

Utilizing the students' understanding of how the context-situated autonomous derivative graph relates to the solution space, the next step is to present the students with other autonomous derivative graphs that are context-free. This is known as *general activity* where students use their model composed of sub-models to reason about a mathematical idea without being grounded in any particular context. So, for the DE example, Rasmussen et al. (2019) present the students with various context-free autonomous derivative graphs and ask the students to reason about what the solution space would be for each graph. Once

the students come to see how any autonomous derivative graph can be used as a tool to create solution spaces, then they have effectively used the autonomous derivative graph as a model-of solution spaces. Thus, the students have moved away from thinking about autonomous derivative graphs as a process and more so as an object that they can use for more mathematics.

The final level of activity is the *formal activity* where students are reasoning about mathematics in a much more abstract manner where they are using their previously established ideas as a model-for higher-level activity. In the context of the DE example, the students are presented with a fish hatchery problem in which the students are asked to think cohesively about the autonomous derivative graph, the phase line and slope field to make inferences about the solution space. When a student has successfully made the transition from the model-of approach to the model-for approach, they are no longer thinking about context-specific autonomous derivative graphs or a context-specific family of solutions. Instead, the students can reason about parameters on the autonomous derivative graph in a context-free environment. In the end, once a student has been able to reach the last activity level of formal reasoning, they have created a new mathematical reality for themselves where, say the effect of a changing parameter for various differential equations can then be used as part of a sub-model for situational mathematical activity in another problem situation. As seen, utilizing the levels of activity can help in the development of instructional tasks but can also be incredibly beneficial as a focus of analysis in terms of the extent to which students are able to make the transition from model-of to model-for ways of reasoning.

A more detailed description of the teaching experiment tasks that I will use can be found in Chapter 3, but here I provide an initial conjecture of the progression through the four levels of activity and the model-of / model-for transition. One of the goals of this study is to help students develop sophisticated ways of reasoning about what it means for a set to be a subset of another set. To develop this idea, we start with the situational activity of having each of the students in the teaching experiment create their own set containing grocery list items. Using For Loops in Python, we can create new sets from the original sets using set operations such as intersection and union (see lines 16-19 in Figure 2.2 below). Note that the set intersection is a result of the 'for'-'if' relationship in lines 17 and 18 and the set union operation is represented by the 'or' operator in line 18. In lines 23-28 the code is instructing the machine to check each element in the new set D (the set containing the elements in A and B or in A and C $[(A \cap B) \cup (A \cap C)])$ to determine which elements are in all three of the original sets. This process of checking each element one by one serves as the initial model-of determining whether one set is a subset of another set. As one can see, the set $D = \{8, \text{``chips''}, 6, \text{``beets''}\}$ is not a subset of all three sets as 8, 6 and "beets" are each not elements of B and C.

```
1 A = {8, "apples", "chocolate", "berries", "corn",
 2
      "juice", 13, "strawberries", 6, "avocados",
 3
      "beets", "chips")
 4 B = {1, "hot cheetos", "jalapeno", "onions",
      "cilantro", 2, "limes", "chips", 6, "cherries", 9,
 5
      "corndogs")
 6
 7 C = {8, "biscuits", "salami", "cheese", "soda",
      "water", "bananas", "beets", "watermelon",
 8
     7,"kiwis", "oranges", "coffee", 9, "cookies",
9
      "ice cream", "sugar", "honey", "butter", "milk",
10
11
     2, "pineapples", "chicken", "sausage", "beef",
12
      "ribs", "pepper", "salt", "candy", 4, "lemons",
13
     "parsley", "bread", "mayo", "mustard", "soy sauce",
14
     "ketchup", 37, "dog treats", "chips"}
15
16 D = set()
17 for x in A:
18 if ((x in B) or (x in C)):
19
      D.add(x)
20
21 print(D)
22
23 for x in D:
24 if ((x in A) and (x in B) and (x in C)):
25
    print("This item is in each of the three sets")
26 else:
27
    print("This item is not in one of the three original sets."
28
       " Thus, D is not a subset of the three other sets.")
29
```

{8, 'beets', 'chips', 6}
This item is not in one of the three original sets. Thus, D is not a subset of the three other sets.
This item is not in one of the three original sets. Thus, D is not a subset of the three other sets.
This item is in each of the three sets
This item is not in one of the three original sets. Thus, D is not a subset of the three other sets.

Figure 2.2: Grocery List Item Task

For the referential activity, the students are tasked with a number theory problem related to divisibility: "Is the set of integers from 1 to 1000, that are divisible by 21, a subset of the set of integers divisible by 3 and 7?" Python does not allow for the students to work with infinite sets, so the referential activity is in creating sets displaying these properties (of being divisible by 21, 3 and 7) for a given range and running For Loops to answer the problem task. Once the students have created the sets necessary to answer this question, the reference to the situational activity is the process of checking each element in the set of integers divisible by 21 to determine whether the set is a subset of the set of

integers divisible by 3 and 7, similar to the grocery lists where the students check each grocery list item in the new set they created (set D).

The general activity would not include the use of Python and instead ask the students if the number theory statement holds for all integers rather than for a given range. When asked how in fact they know that it is true for all integers, I anticipate that students will either give examples of specific integers that satisfy the statement or propose a hypothetical scenario where Python is able to handle infinite ranges. Ultimately, the goal is to help students develop a model-for determining that a set is indeed a subset of another set. By removing the range from 1 to 1000, I am encouraging the students to construct a sub-model of selecting an arbitrary element from the set, say set A, and examining its characteristics to show that it belongs to another set, say set B, based on how set B defines set membership (to show $A \subseteq B$). The formal activity would then be answering a problem such as "Prove $(S \cap T) \subseteq S$ where S and T are any two sets." This is considered the formal mathematical activity as there are no longer any direct references to specific characteristics that define the sets and instead requires the students to reason about any given set. Solving this task requires the students to use the model-for reasoning they developed in the general activity of checking an arbitrary element in $S \cap T$ and showing that this element is an element of S (by definition).

PRIMM

As previous research has documented, the process of learning a programming language can be difficult for students (Robins et al., 2003). This is particularly true at the K-12 level, where considerable research is being conducted regarding effective teaching

and learning of computing (Garneli et al., 2015), with evidence showing this is true at the undergraduate level as well (Bennedsen & Caspersen, 2007; Horton & Craig, 2015; Özmen & Altun, 2014; Watson & Li, 2014). With this in mind, teaching Python to students in a short TE needs to be done carefully and purposefully. Therefore, attending to the difficulties that might arise in learning a text-based programming language, I will also draw on a secondary instructional theory known as the Use-Modify-Create (UMC) model (Lee et al., 2011). Specifically, I will follow the PRIMM approach to teaching text-based programming (Sentance & Waite, 2017; Sentance et al., 2019). PRIMM stands for *Predict*, Run, Investigate, Modify, and Make. The first three map onto the "Use" component of the UMC model, *Modify* maps onto itself in the UMC model and *Make* maps onto "create" (Sentance & Waite, 2017). The creators of the PRIMM model suggest that their approach to the teaching and learning of text-based programming elaborates on the UMC model by drawing attention to the level of sophistication and understanding at each level as well as providing a more specific framework that can be implemented in the university classroom. In relation to the tasks developed for this study, the PRIMM method was used throughout the course of the teaching experiment.

I propose that PRIMM and RME are compatible in at least the following two ways. First, both frameworks approach the teaching and learning of the material through a scaffolded approach. That is, students are not expected to work with formal mathematics nor expected to write code from the onset of instruction. Instead, the goal is to provide the students with an accessible on-ramp to work with the material while maintaining a trajectory toward increasingly sophisticated material. The second aspect of compatibility is

that these two frameworks bring a systematic approach to the reinvention (RME) and creation (PRIMM) of new material. In a sense I see the Modify and Make steps of the PRIMM model as an application of guided reinvention in that the students are creating their own code to represent mathematical ideas that are new to them.

Instrumental Genesis

Programming is starting to become a fundamental aspect of learning undergraduate mathematics. For example, Buteau et al. (2020) investigates how an undergraduate student may use programming to learn both pure and applied mathematics. Rather than investigating student work with a particular mathematical concept, the authors articulate the use of a specific theoretical framework to study the use of programming as a tool for learning mathematics. The framework is known as the *instrumental approach* (Artigue, 2002; Guin & Trouche, 1998; Trouche, 2004). Importantly, Buteau et al. (2020) highlight that the instrumental approach has been used in the past for various artifacts such as graphing calculators, spreadsheets, applets, etc. but has not yet been used in the case where a programming language was considered as the artifact. They do provide illustrative examples of the relationship between mathematical inquiry and programming through the lens of an instrumental approach, but they do not provide an in-depth analysis of one specific mathematical concept.

Therefore, to investigate how Python supports students' learning and advancing mathematical activity related to set theory and logic, I utilize the instrumental approach. In Figure 2.3 I highlight the components of instrumental genesis that serve as the analytic framework to answer the first research question.

The description of the instrumental framework starts with the distinction between *artifacts* and *instruments*. An artifact may be a physical object, such as a graphing calculator, but also may be a formula, graph, or other objects that are central to a certain mathematical task (Roorda et al., 2016). Once the artifact has been determined, the integration of the artifact into a learner's mathematical activity is known as the instrument. This includes the use of the artifact to problem solve and as Trouche (2004) describes it, "an instrument can be considered as an extension of the body, a functional organ made up of an artifact component...and a psychological component" (p. 285). Figure 2.3 is a modification of a figure presented by Guin and Trouche (1998) and diagrams the relationship between the learner and the artifact.



Figure 2.3: Instrumental Genesis

In the context of this study, the artifact is the computer programming environment, Python. As discussed in this chapter, programming environments have previously been utilized with this framework in mathematics education, but never with a focus on students' advancing mathematical activity in relation to set theory and logic.

The development of the instrument is known as *instrumental genesis* (Artigue, 2002) and is tied directly to the artifact and to the mental actions of the learner in their use of the artifact to carry out a given task. This means that the learner may be afforded particular lines of reasoning but also may be constrained in their mental activity given the particular features of the artifact itself. This psychological component, the mental processes of the learner to carry out a particular task, is referred to as an *instrumented* action scheme (Trouche, 2004). The idea of a student's scheme draws on the theory of constructivism and is described by Vergnaud (2009) as "the invariant organization of activity for a certain class of situations" (p. 88). To clarify, this definition encompasses the assimilation of familiar situations which learners respond to with their learned rules or already-established ways of understanding as well as addresses the adjustments necessary to address novel situations in which a learner is required to adapt, modify or reorganize their psychological thought processes. In the literature regarding the instrumental approach, schemes are defined as consisting of four main features (Trouche, 2004; Vergnaud, 2009). These features are summarized well by Buteau et al. (2020) as:

1. the goal of the activity, with sub-goals and expectations;

2. rules of action: stable behaviors of the subject;

- operational invariants, which can be theorems-in-action (propositions considered as true) or concepts-in-action (concepts considered as relevant);
- 4. possibilities of inferences. These possibilities are essential for the adaptation of the scheme to the specific features of the situation. (p. 1026)

As Buteau et al. (2020) state, the rules of action and operational invariants may present themselves through students' mathematical activity in a situation where a student says "When I want to do this [aim of the activity] ... I always act like this [rule of action] ... because I think that [operational invariant]" (p. 1027). The rules of action are stable methods of activity that the student will rely on to accomplish a task, based on the belief or conceptual understanding (operational invariant) of how something works. The operational invariants are broken down into two categories, theorems-in-action and concepts-in-action. The theorems-in-action are constructed, from the constructivist sense, ways of understanding how something works. The concepts-in-action are the related mathematical concepts that are pertinent to the goal of the activity. For example, if a student was asked to use a graphing calculator to find the local max and min of a function given a certain domain, an example of a rule of action would be 'enter the function in the calculator'. A concept in action might be 'zero slope at local min and max,' and a theorem-in-action might be 'the window display of the graphing calculator must capture the specific domain that is asked in the problem in order to be sure of the local max and min.' Lastly, the possibilities of inference would be situations in which the student encounters a problem that might result in new rules of action and operational invariants. In accordance with constructivism, technically these schemes are constructed within the learner's mind and

thus are not directly observable, however, they can be inferred by an instructor or researcher based on the "regularities and patterns in students' activities" (Drijvers et al., 2013, p. 27). That is, for the purposes of my dissertation, a "scheme" is a model that I construct based on the actions of the student. Thus, a "student's scheme" is a construct that seemingly places possession on that of the student, but I am not claiming that the student is aware of the scheme or that the scheme is unknowable.

As Roorda et al. (2016) highlight, at the core of the instrumental approach is the relationship between one's scheme and their *technique*, or actions in response to a given situation. In contrast to a student's scheme, the technique consists of the observable actions that are carried out by the student. It is important to highlight that while the observable actions inform the student's scheme, their actions may also contradict the scheme that was originally developed based on previous actions. So, the student's technique are in-the-moment observable actions of the student that may or may not coincide with the student's scheme. In their work on a student's schemes while using a graphing calculator to reason about the derivative, Roorda et al. (2016) present the following table to highlight the various aspects of the student's scheme and technique:

Instrumentation scheme	Techniques	Conceptual elements	Technical elements
Tangent scheme	Draw a tangent on paper.	Rate of change is related to the steepness of the graph: the steepness of a tangent represents the steepness of the graph at one point.	Calculate the differences of <i>y</i> and <i>x</i> and calculate $\frac{\Delta y}{\Delta x}$.
Trace scheme	Plot the graph, move the cursor over the graph and look at the increase in <i>y</i> .	To find a minimum increase look at the slowest increase of subsequent <i>y</i> -values.	Plot the graph, press trace and scroll over the graph.The trace option gives pairs of <i>x</i>- and <i>y</i>-values. The GC makes equal steps in the <i>x</i>-values.
Trace–value scheme	Calculate the values with the GC and look at the increase over a unit interval.	The increase over a unit interval is an approximation to the instantaneous rate of change.	In the trace option, put in an <i>x</i> -value, press 'enter' and then the GC calculates the corresponding <i>y</i> -value.

Table 2.1: Schemes, Techniques and Concepts

Note. Each scheme in the first column is characterized by the techniques, conceptual elements and technical elements in its row. From "Solving rate of change tasks with a graphing calculator: A case study on Instrumental Genesis," by G. Roorda et al., 2016, *Digital Experiences in Mathematics Education*, 2(3), p. 238.

As seen in Table 2.1, the student's techniques are linked with conceptual elements that help frame an understanding for the student's scheme. In Table 2.2 I provide a hypothetical example of what a student's subset scheme might look like from an instrumental genesis perspective using the framework provided by Roorda et al. (2016).

Instrumentation Scheme	Techniques	Conceptual Elements	Technical Elements
Subset Scheme	Draw a Venn Diagram	To determine if a set is a subset, one must conclude that every element of the set is an element of the larger set	Write a For Loop to check that each element is an element of the larger set

Table 2.2: Hypothetical Example of a Student's Subset Scheme

To capture the idea of instrumental genesis in more detail, it is broken into two subcomponents, instrumentalization and instrumentation, each describing a distinct directional relationship between the artifact and the learner (Artigue, 2002; Trouche, 2004). Instrumentalization is the reshaping of the artifact in the learner's mind as to what the artifact can do to accomplish a certain task. Instrumentalization occurs through the use of the artifact and develops over time as the learner understands the functionality of the artifact and its capabilities. This includes the use of an artifact by the learner in unexpected or unintended ways. An example of this could include a learner storing mathematical results and theorems in their graphing calculator as a memory aid for future use. Instrumentation works in the opposite direction in which the artifact itself, with its built-in constraints and potential uses, is shaping the ways in which the learner conducts their mathematical activity. Roorda et al. (2016) provide examples of how an artifact may shape a learner's mathematical activity in their description of a student named Andy's schemes regarding the concept of the derivative. One specific example is Andy's use of the dy/dxoption on the graphing calculator to calculate the steepness of a graph at a particular point. Andy used this option in other examples throughout the longitudinal study when they

needed to find instantaneous rate of change and did not coordinate between the dy/dxoption on the calculator and the symbolic representations that were covered in class. The authors conclude that while Andy's graphical and numerical representations of the derivative were strong, these representations were being developed independent of Andy's symbolic representations, a negative consequence of Andy's overreliance and strong use of the graphing calculator.

Trouche (2004) comments that both instrumentalization and instrumentation work together in tandem as part of the instrumental genesis process, and thus refers to a learner's scheme as the instrumented action scheme as opposed to other terms such as *instrumentation scheme* as used by Roorda et al. (2016). For the purposes of this dissertation, I will adhere to Trouche's approach to the development of a learner's scheme (as perceived by the researcher) and either refer to it as an instrumented action scheme or just scheme for short.

Hypothetical Learning Trajectories

To improve teachers' approaches to instruction, Simon (1995) introduced the notion of a Hypothetical Learning Trajectory (HLT). As Simon presented it, a HLT consists of three main features. First, teachers must idealize a certain goal or understanding that they would like their students to have and consider the current state or level of sophistication in which their students are reasoning about that topic. Based on that reasoning, the second feature is the development of tasks that must be presented to students in a structured and intentionally organized way to facilitate the advancement of a particular kind of reasoning. The third feature is the development of the teacher's hypothesis of how

the students would be learning and interacting with the designed tasks (Simon, 1995). It is important to note that there is an interaction between the second and third features as the teacher is required to reflect on the students' perceived learning processes in response to the designed tasks. They then develop new tasks that might facilitate a new branch of discussion or potential route to proceed to the desired learning goal that was envisioned (the first feature of the HLT). Of course, how the students react to a teacher's HLT is not always going to be how the teacher intended. For this reason, what the students actually do in response to the instructional tasks and the learning that takes place is referred to as the *actual learning trajectory* (Simon, 1995). Lobato and Walters (2017) describe the relationship between the hypothetical and the actual learning trajectories well:

Prior to instruction or analysis a teacher or researcher has a planned or hypothetical learning trajectory, whereas the coproduction of mathematical knowledge during instruction or the results of retrospective analysis by researchers is often known as an actual learning trajectory. (p.84)

The HLT is only one component of what Simon (1995) refers to as the Mathematics Teaching Cycle, which involves the interrelationship between teachers' knowledge, thinking, and decision making and the mathematical activity that is occurring in the classroom.

In Lobato and Walter's (2017) literature review on learning trajectories/progressions, they describe how Simon (1995) introduced the idea of the hypothetical learning trajectory as a tool or construct for teachers. That is, a teacher develops a conjecture about their students' reasoning in the classroom and, in turn, constructs a series of tasks for the students to help support them in their mathematical sophistication to ideally reach a particular learning goal. Clements and Sarama (2004) adapted this idea for its use in research, by including the need to track students' developmental progressions:

We conceptualize learning trajectories as descriptions of children's thinking and learning in a specific mathematical domain and a related, conjectured route through a set of instructional tasks designed to engender those mental processes or actions hypothesized to move children through a developmental progression of levels of thinking, created with the intent of supporting children's achievement of specific goals in that mathematical domain. (p.83)

For the purposes of this study, when I describe the HLT, I refer to the research adaptation as described by Clements and Sarama. In order to support students through a development progression, the researcher must first have a baseline understanding of the extent to which the students are familiar with the mathematical content that is the subject of study. To accomplish this, I administered a survey before the TE and after the TE asking the students content-related questions on set theory and logic (more on the surveys can be found in Chapter 3). To further add to the capability of utilizing a HLT for research, Simon et al. (2010) highlighted the importance of understanding the learning process and thus the developmental progression by stating that researchers can use a successful task sequence to "observe the student's activity over the course of the entire task sequence. This provides data on the process by which the new learning comes about" (p. 108). In my study, I will study the research participants' learning progress and processes by documenting the actual learning trajectory of the students as they relate to the four levels of activity from the RME pedagogical framework. That is, evidence of new learning will be tracked as students move from one level of mathematical sophistication to the next (e.g., from situational to referential).

From a research standpoint, HLTs can be used to improve existing curricula as well as inform instructional practices with respect to which strategies can be most beneficial for students. Simon focused a great deal on the pedagogy of mathematics, which mostly encompassed the decision making when it came to the content and tasks that were used in class. A focus on particular teacher moves was not initially an explicit focus when the idea of a HLT was first introduced. To expand upon Simon's presentation of the role of the instructor within a HLT, Andrews-Larson et al. (2017) examined the various teacher moves that support students' reasoning as the students move across the sequence of instructional tasks. In the analysis of their results, Andrews-Larson et al. (2017) found that, "The choices instructors make as they respond to student contributions importantly shape the development of student reasoning" (p.826). Two examples of teacher moves that importantly shape what the students learn and how they learn it are knowing what (and when) to "tell" the students, and intentionally framing discussions around particular points of interest that foreground the desired learning goals. Thus, the explicit role of the instructor is considered a fourth feature of an HLT.

In undergraduate mathematics education, there has been HLT work done in the areas of linear algebra (Andrews-Larson et al., 2017), abstract algebra (Larsen, 2009) and recently in Fourier analysis (Ekici et al., 2020), among others. However, HLTs have been the focus of a recent call for future research (Laursen & Rasmussen, 2019). By asking Research Question 2, this study will be one of the first studies in mathematics education (that I am aware of) to address the hypothetical and actual learning trajectories of students

as they learn set theory and logic. In line with the four features of an HLT just described, Research Question 2 will be addressed by attending to the following four components:

- 1. The learning goals of set theory and logic that would prepare students going into advanced mathematics.
- 2. The development of tasks that would help support the advancement of the students' mathematical reasoning.
- 3. The advancing mathematical sophistication of the students as evidenced by the actual learning trajectory of the students (i.e., the progression through the four levels of activity).
- 4. The specific moves that I make as the researcher to help support the students' mathematical thought processes and reasoning.

It is important to note that the instructional design heuristics of RME, emergent modeling, didactical phenomenology, and guided reinvention are highly compatible with these four listed goals. Specifically features 2 and 3 as crafting and scaffolding effective tasks can lead to productive advancement in students' mathematical reasoning. As Simon and Tzur (2004) describe, Simon's (1995) initial presentation of HLTs "stopped short of providing a framework for thinking about the learning process and the design or selection of mathematical tasks" (p. 92). As such, I address this critique by foregrounding RME as the guiding instructional design theory for the tasks presented to the students in the TE.

At this point in the description of HLTs, it is important to note that the learning goals are set to provide a general direction for this study. Without the learning goals, each TE group has the potential to stall or go off in multiple distinct directions. However, I do want to stress that the mathematical activity of the students in this study in relation to the learning goals that are posited does not reflect any kind of failure on behalf of the students if the learning goals are not quite met, as they are all on their own paths in advancing their mathematical activity.

HLT Goals for this Study. The first goal for this HLT is that students develop operational definitions of the logical operators 'and' and 'or.' Evidence of this learning goal would be that the students can flexibly use the logical operators in the context of set theory, propositional calculus, as well as number theory questions. A subgoal would be that students will be able to reason about truth tables when determining logical equivalence of compound propositional statements. Evidence of this subgoal would be the use of truth tables with a unified approach to comparing logical statements (Hawthorne & Rasmussen, 2015). Further evidence of sophisticated ways of reasoning about logical statements and logical operators would be the proper use of logic in the construction of the For Loops when answering the tasks related to various set operations such as set intersection and union.

Utilizing Python, and their developed ideas related to set intersection and union, to determine that one set is a subset of another set is the second main goal for this HLT. Determining that one set is a subset of another set is one aspect highlighted by Dogan-Dunlap (2006) as well as Linchevski and Vinner (1988) in their noticing of student difficulty identifying situations where sets were elements of other sets, in contrast to subsets. Evidence that the students have reached this learning goal would be the proposed model-for reasoning that was previously highlighted in this chapter. This model-for

reasoning about sets as a subset of another set is the process of selecting an arbitrary element from the set and determining whether or not this element fits the characteristics of set membership of the other set in question.

Socio-Constructivism

In this section I highlight how RME is compatible with the broader perspective on learning known as the socio-constructivist theory of learning. As Gravemeijer (2020a) described, socio-constructivism "focuses attention on the crucial role of the classroom culture" (p. 219). Social interactions between the students and the instructor can not only be interpreted through the lens of a socio-constructivist theory of learning but also can be framed and interpreted as the instructor's process of didactical phenomenology and support for guided reinvention. Gravemeijer (2020a) explains the connection between RME and socio-constructivism in his paraphrasing of Cobb (1994):

(Socio-)constructivism is not a pedagogy. [Cobb] argued that if it is true that people always construct their own knowledge, then students will do so in every classroom - even with direct instruction. The issue, [Cobb] went on to say, is not whether they construct, but *how* and *what* they construct." (p. 219)

Together, RME and socio-constructivism provide a holistic approach to the teaFching and learning of mathematics. To address the relationship between theory and pedagogy, Simon (1995) posed the following question: "In what ways can constructivism contribute to the development of useful theoretical frameworks for mathematics pedagogy" (p.117). For Simon, they wanted to address how teachers can effectively support students in their development of mathematical knowledge that took mathematicians thousands of years to develop. This was the impetus in the development of the Mathematics Teaching Cycle framework, of which the HLT is a major component (Simon, 1995). For Simon, the goal was to understand how teachers can structure a sequence of tasks in response to the mathematical activity of their students. To be explicit, instructors must develop conjectures and hypotheses as to how a student might be reasoning and find an approach that prioritizes learning as something that the student is individually constructing and not something that is originating outside of the learner (Jones & Brader-Araje, 2002; Labinowicz & Frazee, 1980; Sjøberg, 2010). For instructors and researchers to approach instruction in this way, they must consider the idea that the learners are constantly organizing, interpreting, and restructuring their knowledge within the broader learning environment.

In addition to situating the instructional design theory guiding this dissertation, I highlight the relationships between RME, HLTs and socio-constructivism to ground some of the analysis plans that will be described in Chapter 3. Specifically, the main idea covered in Chapter 3 related to socio-constructivism that is important to call out here. That is the idea of developing schemes of students' learning. These schemes are constructs generated by the instructor or researcher to explain student reasoning (Norton & McCloskey, 2008). The notion of a scheme is something that was detailed by von Glasersfeld (1995) in his description of Piagetian constructivism. This idea will be strongly utilized in the analysis of instrumented action schemes. In general, this process consists of noticing the students' observable mathematical activity and generating hypotheses as to how the students might be reasoning. The instructor/researcher then tests those hypotheses

with follow up tasks and probing questions to narrow in on specific ways students might reason about a given concept.

Literature Review

I present the following sections to review the current literature in four areas: (a) the teaching and learning of set theory, (b) students' conceptions of mathematical logic, (c) computing as a tool in mathematics education and (d) students' affective experiences and mathematical identity. It is important to note that historically the literature has been fairly deficit-oriented in terms of identifying what students cannot do (particularly so for the research related to set theory). While I believe a deficit perspective of student learning is problematic, I do think it is important to highlight the literature that is framing the motivation and goals for this study. That is, by understanding where students have historically had difficulties, we can work towards finding ways to support students in the learning process.

Set Theory

In this section I review studies that have investigated the learning of set theory topics for a wide range of students. One of the first studies related to set theory and student reasoning about these topics is Linchevski and Vinner's (1988) investigation of elementary teachers' and student teachers' general understanding of sets. Their study focused on four aspects in particular that may lead to various conceptions of sets: (a) sets having a common property, (b) singleton sets, (c) a set can be an element of another set, and (d) set equality. The sample for this study consisted of 237 elementary teachers and 72 student teachers. Of the 237 teachers, 54 of them were classified as having higher levels of mathematics

training. As a product of interviews with 21 teachers, the researchers developed an openended questionnaire with five items (three containing sub-items) that is meant to elicit one's conceptions about sets. For example, item three in Linchevski and Vinner's (1988) questionnaire states,

A teacher asked her students to give an example of a set. One of the students wrote: My set has three elements: (a) 5, (b) 1.5, (c) the set of all the even integers between 2 and 100. Is this answer correct? Explain! (p. 473)

The results of the study indicate interesting findings for each of the researchers' four aspects of study. In relation to sets having a common property, 97% of the student teachers, 89% of the teachers and 60% of the teachers with a higher-level math background did not consider any arbitrary collection of elements as a set. Instead, many of the participants claimed that a set must contain elements that have some sort of commonality. Results related to the second aspect, singleton sets, revealed that the 48% of the teachers, 55% of the student teachers and 6% of the teachers with more math training believed that a single element cannot form a set. In response to item three above, 56% of the teachers, 70% of the student teachers and 76% of the teachers with greater math background claimed that the student' response was incorrect because a set cannot be an element of another set. As for set equality (the fourth aspect), 18% of the teachers, 15% of the student teachers and 56% of the teachers with greater math background correctly used the mathematical definition of set equality to determine equivalence between sets. The results of this study clearly indicate that there is room for growth in understanding topics related to set theory, but why might this be?

Fischbein and Baltsan (1998) address this question in their review of how the *collection of objects* model might lead to misinterpretations and incorrect conceptions. Generally, the collection of objects model describes the intuitive perception of a set as a group of "things" whereas a more formal approach to the mathematical concept of a set is that sets are defined by the distinct objects in the set (or lack thereof in the case of the empty set) and nothing else. The authors claim that for every finding in Linchevski and Viner's (1988) study, if one takes the perspective of a collection of objects model, "all of the misconceptions are predictable" (p.2). For example, take the misconception that a set must be composed of at least one element. The intuitive notion that a set is a collection of objects, or things, might then lead one to believe in this misconception when the truth is that a set can contain no elements, also known as the empty set. Another misconception that the collection of objects perspective may lead one to believe is that repeating elements add to the cardinality. That is, let $A = \{1, 2, 3, 3, 4\}$, someone with the collection of objects perspective might say that the cardinality is five when in fact the cardinality is four because the element 3 is repeated. One that understands sets to be defined by distinct objects would likely not make this mistake. Fischbein and Baltsan (1998) describe the intuitive model working behind the scenes as an effective contributor to how one processes mathematical problems regarding sets. The work of Fischbein and Baltsan aimed to investigate their hypothesis that the intuitive collection model pervades all apparent misconceptions regarding sets and that time and the learner's age may have a role in how one reasons about sets.

Four groups of students were the participants in Fischbein and Baltsan's (1998) study: a) 46 8th grade students, b) 51 10th grade students, c) 21 preservice elementary school teachers and d) 32 preservice junior high teachers (where mathematics was their emphasis). Data collection was conducted through the administration of a questionnaire that was given during class time. Several in-person interviews were also conducted with students who did not take the questionnaire to better understand student reasoning regarding their proposed hypotheses. The first finding from this study relates to the correctness of the participants' responses in their definition of a set. They found that 52% of the 8th graders, 71% of the 10th graders and 81% of the prospective elementary school teachers had *incorrect* conceptions of what a set is. The prospective junior high math teachers had a greater proportion of correct answers with 22% having incorrect definitions. When asked about the empty set, the junior high teachers seemed to know what it is but were unable to define it using precise language. The authors claim that this is a result of a tacit, intuitive, model taking over for their reasoning. In relation to time, the authors suggest that as students get older, unless the formal properties of sets are reinforced through study or instruction, the intuitive collection of objects model slowly becomes the primary conception that students use.

Teaching Set Theory

Given the results of Fischbein and Baltsan (1998), if students' formal conceptions of sets are not being nurtured in future mathematics courses, then research on how best to support the formal conceptions at the undergraduate level is needed. Dogan-Dunlap (2006) addresses this point of interest in their study of how a lack of mastery of set theory

concepts can lead to poor performance in linear algebra. In their review of 45 student exams across two different semesters as well as student interviews, it was clear that students were missing the necessary knowledge of set theory, particularly in the case of what it means for something to be included in a set and what it means for one set to be represented in multiple ways. Thus, the purpose of Dogan-Dunlap's (2006) study was to present a student-centered approach to teaching just-in-time prerequisite knowledge of set theory concepts to support students in linear algebra.

The basis of the pedagogical approach was to have students connect real-life experiences regarding club membership with the formal set-theory language. To help the students, a list of guiding questions was given during class that had them think about a club in terms of a gym membership or sports organization, anything that one can be considered a member of. The main purpose of this task was to encourage group discussion. Additionally, index cards were given out to the students with various representations of membership descriptions in an effort to engage students in reasoning about what it means to satisfy membership. One example the author gives is that some students are given vectors in \mathbb{R}^n that are not members of any of the clubs in the class. The purpose of this activity was to show the students that, "not every vector in \mathbb{R}^n is a member of a club even if the members of the club are vectors in \mathbb{R}^{n} " (p.405). The results of this study were limited but the author provided examples from one of the groups that participated in this activity. The group showed difficulty in being able to prove set inclusion, but they seemed better equipped with the tools necessary to relate the formal mathematical notation with their real-life experiences. In turn, the author suggests that connecting the formal mathematical

notation with their real-life experiences helped the students move away from thinking about linear algebra as pure symbol manipulation and led to a more meaningful discussion about the linear algebra concepts.

While Dogan-Dunlap (2006) considered a student-centered approach to resolving issues related to students' understanding of sets, Bagni (2006) took an in-depth look at the set theory material and highlighted the issues that may arise just in the students' representations alone. Specifically, Bagni investigated the issues related to Euler-Venn diagrams and how their usage to represent formal ideas related to sets may induce cognitive difficulties for students. To begin, Bagni (2006) describes how students may conceptualize sets using a container-metaphor (Lakoff, & Núñez, 2000) way of reasoning and how that introduces conflict with a Euler-Venn diagram in identifying a particular element as either belonging to or being a subset of a larger set. This distinction is described mathematically as $x \in I$ and $x \subseteq I$, respectively. The main point emphasized by Bagni (2006) is the following, "The key concepts of Set Theory, such as the concepts of belonging and inclusion, have an intuitive meaning that is formalized later, when they appear defined through precise verbal expressions" (p.263).

Note that the transition from an intuitive model of reasoning to a more formal one is in line with the goals of RME. However, I want the intuitive models to support student learning rather than serve as an obstacle or point of confusion. For example, Bagni explains that the precise verbal language (formal) does not always match what is drawn using diagrams in class (informal) and that drawing should not act as a replacement of a student's definition for what it means to be a set. In order to further investigate the

potential issues related to diagrams and student understanding, Bagni (2006) conducted two experiments, experiment A with 16 11-year-old students that served in a sense as a trial study to experiment B with 25 15-year-old students. Two excerpts from classroom observations in experiment B served as the main point of analysis as a student presented their work in front of the class and then revisited their work after the teacher asked them to reconsider their answer. The specific task that the student was asked to answer dealt with the set of all points on a line, R, and the set of all points on a line, S, that lie perpendicular to R. The student and teacher's diagrams are shown below in Figure 2.4.



Figure 2.4: Teacher and Student Diagrams

In Figure 2.4, the student's diagram is on top and teacher diagrams are on the bottom (Bagni, 2006, p. 271). The student drew the diagram at the top indicating a geometric relationship within the set A (the set having elements R and S). Importantly, Bagni (2006) claims that the student was reasoning about the sets R and S as subsets within A rather than *elements* in A. The process of drawing the diagram in this instance served as both an obstacle in the student's reasoning but also as a method for analysis from a researcher's

perspective. Ultimately, Bagni suggests that educators must consciously draw a coherent distinction between the verbal register when referencing sets and their visual representations. This must be done throughout the semester as well to try and encourage flexibility in transitioning between verbal descriptions and diagrams.

In this section I highlighted literature on students' conceptions of set theory to focus on the aspects that I believe most relevant to this dissertation. There are two ideas in particular that I believe students can use Python to help develop their sophisticated ways of reasoning about set theory. The first is related to the distinction between sets as elements of another set and sets as subsets. This is referenced by both Bagni (2006) and Linchevski and Vinner (1988). In Python, students must define the sets they are working with and to work with a set as an element, they must define it first as a 'frozenset()' before they can use it as an element of another set. I believe the process of defining the set first might help students conceptualize a set as an object that can be manipulated (i.e., a set can be modified or defined to whatever they want it to be). The second, is the idea of what it means to be a subset. This relates to Dogan-Dunlap's (2006) study in identifying the characteristics of a set that describe set membership. As referenced in the example of the four levels of activity, Python can potentially help support students in the model-of / model-for transition of proving that one set is a subset of another. Ultimately, I can see Python helping students reason about set cardinality, sets as elements, and sets as subsets of other sets. The power of leveraging programming, as opposed to using manipulatives or paper and pencil, is that students learn an additional skill of increasing importance in today's society. Moreover, students have opportunities to address common misconceptions

(as highlighted in the literature review in this chapter) in the programming environment as they will encounter error messages and must go through a troubleshooting process.

Mathematical Logic

Undergraduate students' conceptions of logic is a relatively new and growing area of research in mathematics education. The role of logic in the mathematics curriculum is controversial, but Durand-Guerrier (2020) argues that the development of competencies in mathematical logic can help support mathematics conceptualization and sophisticated reasoning about mathematical proofs. In this section I present a series of studies particularly focused on students' reasoning about logic with the dual goal of illuminating areas that are still under-studied and to provide the background information that motivated the conception of this present dissertation.

First, I present the results by Hawthorne and Rasmussen (2015) and their framework for characterizing students' reasoning about truth tables and implication statements. The framework they introduced consists of two main analytical dimensions: compartmentalized and unified ways of reasoning. A *compartmentalized* way of reasoning about truth tables describes students interpreting truth tables either row by row, column by column or even by each individual symbol. In contrast, someone with a *unified* view of truth tables is able to reason about the whole table which describes multiple cases as one idea and can be used to show logical equivalence between various logical statements. Through a series of interviews with students enrolled in a discrete mathematics class, their findings suggest that students generally struggle with viewing truth tables using a unified way of reasoning. Four of the six students demonstrated compartmentalized ways of

reasoning and the other two students showed partially compartmentalized ways of reasoning about the truth table on their way toward a unified view. Only the professor of the course presented fully unified ways of reasoning about the truth table to show equivalence between two logical statements. One of the primary takeaways from their study is that it is important to consider how students new to logic may find it helpful to work in a semantically meaningful context rather than pure symbol manipulation. This meaningful context could contribute to a reified view of the truth table not only as a process but also as an object that can be analyzed.

Along these lines of providing meaningful contexts for mathematical logic, Dawkins and Cook (2017) present results from a series of teaching experiment sessions with undergraduate students focused on students' reasoning about disjunctions. The design of the teaching experiment was guided by the RME heuristic of guided reinvention with the purpose of understanding how untrained mathematics students, who have not taken a proof-based mathematics course, reason about problems related to mathematical logic. Specifically, the researchers conducted teaching experiments with 13 students, six pairs and one individual, enrolled in Calculus 3. The teaching experiments began with asking the students to reason about a set of disjunctive statements, and the researchers' goal was to focus on the spontaneous behavior and interpretations that the students utilized to reason about the statements. During the first session with the students, the researchers provided a set of relatively basic disjunctive statements such as, "A1. Given an integer number *x*, *x* is even or *x* is odd" and asked the students to determine whether the statement was true or false (p.246). The authors presented results from one pair of students, Eric and Ovid, as

their teaching experiment was representative of the others. Results from Eric and Ovid's session indicated that after an initial confusion, the students developed what Dawkins and Cook (2017) refer to as a *Part True-All True Heuristic* to correctly reason about the disjunctive statements.

On the second day however, the researchers provided the students with more difficult disjunctive statements such as, "B7. Given any triangle, it is acute or it is not equilateral" which resulted in students having to use more advanced ways of reasoning (p.246). Importantly (and relating back to the set theory literature), students were able to leverage what the authors refer to as *set-based reasoning* to help them answer the logical statements. For example, with the statement B7 provided above, one student, Erin, was able to use set complement relations to help justify their answer (p.254):

It is true because an equilateral triangle is an acute triangle. So if it's not equilateral, then there's a chance that it's, it could be anything else [...] Like, every type of triangle. So if it says, "it is acute," so an equilateral is an acute triangle, so it would fit into that category. 'Or it's not equilateral.' That would include like right triangles, and obtuse, and all those other types.

This student's reasoning relied on being able to partition the set of all triangles into two complementary subsets, equilateral triangles, and non-equilateral triangles. The authors suggest that the set-based strategy, "may be productive for reinventing the normative logic of quantified disjunctions" (p.254). While not every student in the study was able to reason about mathematical logic using this type of set-based reasoning, all the students eventually finished the teaching experiment sessions demonstrating reasoning with the *Part True-All True Heuristic* where only one part of a disjunctive statement needs to be true for the entire statement to be true.

In a follow-up article explaining in more depth the importance of set-based reasoning, Dawkins (2017) provides more examples of student work from the teaching experiments used in the Dawkins and Cook (2017) study. In this article Dawkins characterizes several unique ways of student reasoning about mathematical logic. In this process, Dawkins also characterizes different ways students may reason about categories and properties (each of these being important ideas related to set theory). Specifically, Dawkins highlights the difficulties that some students encountered as in the tendency for Eric and Ovid to over-rely on familiar categories which resulted in an avoidance of negative ones. A clear example of this reasoning was Ovid's substitution of "not acute" with "obtuse." As one might suspect, this and other similar ways of reasoning led to complications in the ways they reasoned about disjunctive statements. Dawkins (2017) refers to this approach as *reasoning about properties*. In contrast, Erin's approach (as described in the previous paragraph) was able to draw on set-based reasoning to determine the truth value of the logical statement. This is what Dawkins (2017) refers to as *reasoning* about predicates. I present the findings of this study for two purposes: a) to highlight the difficulty that students may encounter in the development of normative ways of reasoning about logical operators, and b) to bring awareness to the distinction of what *reasoning* about properties and reasoning about predicates may look like. Given that a focus of this dissertation is to understand how students reason about sets, I anticipate these ways of reasoning will be helpful in describing students' mathematical activity. The main idea related to mathematical logic that I will explore further is the reinvention of normative ways of reasoning about the logical operators 'and' and 'or.'
Computing as a Tool for Mathematics Teaching and Learning

The use of computing as a tool for mathematics learning is not quite a new phenomenon as efforts in the past have been made to learn various topics such as elementary algebra using Logo and Excel (Sutherland, 1994) and BASIC (Tall, 1989), set theory (Dubinsky, 1995) and abstract algebra (Dubinsky, & Leron, 1994). The work done by Dubinsky on set theory and abstract algebra is more in line with the focus of my dissertation as the goal is to use programming to develop conceptions of pure mathematics topics. To accomplish this goal, Dubinsky and colleagues developed a specific programming language, called ISETL, that was designed specifically for the purpose of learning these topics. Findings from the work with ISETL did in fact reveal connections between programming and the development of students' mathematical conceptions, although the use of ISETL as a programming language never gained any sustained traction.

I would argue that it wasn't until Wing's (2006) influential (more than 6600 citations on Google Scholar) description of computational thinking and its influence on society that computing made significant headway in mathematics education. Wing (2006) describes computational thinking as the process of "parallel processing. It is interpreting code as data and data as code...It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable" (p. 33). Wing updated this definition in a more recent blog post by defining computational thinking as the following, "Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer - human or machine can effectively carry out." Wing's insight has led to many changes in mathematics education as new approaches have been developed to incorporate computing into the K-12 curriculum (Grover & Pea, 2013). Drawing on some of the lessons learned from the K-12 success, researchers have been making an effort to incorporate computational thinking into the undergraduate curriculum in other disciplines as well (Settle et al., 2013).

Mathematics is one such field where computing is a natural fit into the curriculum and will continue to become important with an ever-growing emphasis on data science (National Academies of Sciences, Engineering, and Medicine, 2020). Additionally, computing has recently been a focus in undergraduate mathematics education as there has been a push for computing to be considered a mathematical disciplinary practice (Lockwood et al., 2019). In the context of this study, drawing on the definition used by Lockwood et al. (2019), computing is defined as "the practice of using tools to perform mathematical calculations or to develop or implement algorithms in order to accomplish a mathematical goal" (p. 3). The following literature highlights the research that has been conducted with respect to computing as a tool for mathematics learning at the undergraduate level.

To gauge the prevalence of computing as a focus for undergraduate mathematics students in the United Kingdom (UK), Sangwin and O'Toole (2017) sent a survey to all departments in the UK that offer a Bachelor of Science degree in mathematics. 46 (63% of the total number of surveys sent) departments replied to the survey. Of the 46 respondents, 78% of the departments indicated that their undergraduate mathematics students are exposed to programming, primarily using a mathematical programming language such as MATLAB, Maple, and R rather than general purpose programming languages Python,

C++ or Java. While these results reflect the prevalence of programming in the UK, there are also calls for computer science and computational thinking to be integrated into the undergraduate mathematics curriculum in the United States as well (Li et al., 2020; Lockwood & Mørken, 2021). This has led to a surge (and need) in research on how undergraduate students can leverage programming as a tool for learning mathematics.

Avigad (2019) takes one step closer to this need with a description of the theorem prover known as Lean to teach an ITP course. They describe that they use Lean to teach their students three languages, formal symbolic logic, informal mathematical language, and computational proof language. The theorem prover is a programming language that allows for the user to carry out a proof using deductive reasoning as one might do in a standard mathematical proof. Using logical statements, the program can verify that the proof is valid. Avigad (2019) explains that "Lean's logic is very expressive, so that, in principle, *any* ordinary mathematical theorem can be formalized and proved in the system" (p. 280). Avigad did not carry out a formal investigation of student work in relation to the use of Lean, but they did provide anecdotal data from their student evaluations. Their results suggest that students do in fact find the use of a programming language to be helpful in the learning of formal mathematics such as logic as well as developing an understanding of the structure of proof writing.

Lockwood and De Chenne (2020) explicitly address the need for research in how computing can impact undergraduate students' mathematical activity in their study exploring student reasoning about combinatorics problems. The data from this study comes from two paired teaching experiments in which the participants were recruited from a

Vector Calculus course. In the teaching experiment sessions, the students were given a series of counting tasks in which they were asked to use Python code to help determine their answers to permutations and combinations problems. An example of one of the tasks they used can be seen in Figure 2.5.

1a) Given a set of shirts and a set of pants we would like to know the total type and number of outfit combinations possible. Look at the code below. What do you think this code does? What will the output of this code be?

```
Shirts = ['tee','polo','sweater']
Pants = ['jeans','khaki']
outfits = 0
for i in Shirts:
    for j in Pants:
        print(i,j)
        outfits = outfits+1
print(outfits)
```

1b) Why is one of our print statements located on the inside of the loops and the other on the outside?1c) What would happen if we put the **print(outfits)** statement inside the for loops? First make a guess and then try changing the code. Discuss the results.

1d) How would your program change if you wanted to print the outfits so the pants are before the shirts?

Figure 2.5: Shirts and Pants Combinations Task

The task presented in Figure 2.5 was the first task presented to the students in the teaching experiment by Lockwood and De Chenne, (2020, p. 317). While Lockwood and De Chenne do not explicitly mention it, they are incorporating the first four steps of the PRIMM model as the students are asked to Predict in (1a/b/c), Run in (1a/b/c), Investigate in (1b/c) and Modify in (1c/d). The goal of their study was to understand how Python can help students develop a robust understanding of the four different problem types (repetition allowed/order matters, repetition not allowed/order matters, repetition allowed/order does not matter, repetition not allowed/order matters) that involve selecting *r* objects from *n* items.

While the students' counting strategies are not particularly germane to the focus of my dissertation, how students used Python as a tool for learning is. For example, one pair of students were able to leverage nested For Loops to reason about a problem where order matters and repetition is allowed. The nested structure allowed for the students to impose an ordered structure while allowing for repetition in that the For Loops were repeatedly drawing from the set of objects that were being counted. Lockwood and De Chenne (2020) conclude that the use of Python seemed to have "enrich[ed] the students' combinatorial reasoning by affording opportunities for the students to strengthen connections to the kinds of outcomes they were counting" (p. 338). In Chapters 4 and 5 I present my findings on students' reasoning about newly constructed sets using For Loops which are in line with the conclusion presented by Lockwood and De Chenne. This is a new opportunity and capability for the students as they will be able to see visually what happens when the code runs through every element in a set and creates new sets through set operations.

Students' Affective Experiences and Mathematical Identity

While computing is a major focus of this dissertation, with that comes the need to examine the current state of computing and mathematics education from a social-emotional perspective to understand how my study will have an impact on students. Impact in mathematics education can be measured in many ways, from an external qualitative ethnographic evaluation of one's educational identity, to quantitative survey analysis on how students report their experiences in the classroom. For example, Rubin (2007) utilized ethnographic methods to document the oppressive environment at a low-income and urban high school in which Rubin argued, "few individuals would be able to persist and fewer still would be able to gain the skills necessary to succeed in higher education" (p. 244). In Voigt et al. (2021) we used both qualitative and quantitative methods to document the shift, or refiguring, of a group of students' mathematical identities. There are many ways to do this work, however, the goal of understanding how a teaching intervention or learning experience can impact one's developing mathematical identity is the same, and is the purpose of asking Research Question 3. The design of this dissertation predominantly utilizes qualitative methodology (more in Chapter 3) to investigate the development of the students' mathematical identities as the students progress throughout the study. In this section I highlight a brief review of previous work that shapes my perspective, and my intent, when I aim to study impact on students' mathematical identity.

Some studies on students' non-cognitive behavior and identity can loosely be described as attempts to label or define. For example, Gee (2000) defined and illustrated *natural, institutional, discursive,* and *affinity* identities to describe the "kind of person" someone is, as they are perceived in the educational context. Other work that falls under this category of identity-defining can be seen with Cobb and Hodge's (2011) Normative, Core, and Personal Identities, as well as Nasir and Saxe's (2003) discussion on the tension between one's Ethnic and Academic Identities (Nasir & Saxe, 2003). We can use this work to better understand how certain students may behave in a particular educational context, but there is more to be desired in terms of understanding the affective factors that contribute to one's identity development. As Darragh (2016) describes, these identity-labeling studies focus on identity as an acquisition, or something that an individual possesses inside of themselves. Instead, Darragh suggests that research should focus on

identity as a process, or something that we do. The advantage of discussing identity as a process, or something that changes over time, is that it is possible to take a sociological perspective on student behavior in educational contexts. This allows the researcher to analyze the interpersonal exchanges between students and the power dynamics that exist in STEM education (Gutierrez, 2013).

When I consider the concept of one's mathematical identity, I often first start with Martin's (2006) definition of identity, "the dispositions and deeply held beliefs that individuals develop, within their overall self-concept, about their ability to participate and perform effectively in mathematical contexts and to use math to change the conditions of their lives" (p. 206). While this definition does a particularly good job in framing the importance of one's mathematical competence, research suggests that mathematical identity is composed of more than just one's ability to do mathematics. In fact, Cribbs et al. (2015) find that mathematical competence has an indirect effect on one's mathematical identity, while interest and external recognition by their peers and others has a direct effect on their identity. For this reason, when I discuss impact on one's mathematical identity, I am considering competence, evidence of interest and enjoyment, and if possible, how the students are being perceived by their working group partners. The perception component is the most difficult to analyze given that I am not asking the participants to discuss their perceptions of one another. However, if there is a situation in which Student A makes a comment about how smart Student B is, which happened in the pilot study, then I will consider that comment as a component of the development of Students B's mathematical identity development.

As for studies on students' affective experiences involving computing, Psycharis and Kallia, (2017) found that computing may have a positive impact on high school students' self-efficacy in mathematics. Similar results were found by Weese et al. (2016) in their description of two interventions aimed at improving self-efficacy in computer science for grade 5-9 students. Relatedly, at the undergraduate level, Satyam (2020) explored the types of positive emotions and satisfying moments that students experience in a standard ITP course. These satisfying moments fell into four categories: accomplishment, sensemaking, properties of mathematics, and interactions with people. Analyzing the data for evidence of students' positive affective experiences will help shape how we as educators and researchers understand how students experience teaching interventions in the context of advanced mathematical content. It is also important to consider the potential negative experiences that students may encounter. In a study investigating how gender and race influence group dynamics in physics classes utilizing computing, Shah et al. (2020) found that computing and physics were maintained as spaces for patriarchy and white supremacy. Specifically, white students dominated participation in small group settings compared to their black and latinx peers, and in groups with a greater proportion of girls than boys, the girls experienced greater participatory equity. As group work is often a major component of active learning classrooms, it can also potentially be an aspect that leads to inequitable experiences for women and students of color. Johnson et al. (2020) highlight this idea in their research on inquiry-oriented instruction classrooms by reporting that the inquiryoriented classrooms seemed to have a negative effect on women compared to their peers in non-inquiry-oriented classrooms. Johnson et al. (2020) stress that additional research on

small-group collaborative interactions is necessary in the post-Freeman et al. (2014) era of active learning in STEM.

In view of this, it is important to consider the social and personal identities of students as we explore how computing and STEM education more broadly may have an effect on students' non-cognitive experiences. Therefore, documenting how the student participants in my study experience the TE in terms of affective measures such as confidence, enjoyment, and interest is an explicit focus of my work. Additionally, attention to how certain students in each group are recognized by their working group partners will be a component in the analysis of the participants' shifting mathematical identities. Given the focus on intra-group relationships and the interactions between students, the social identity markers for each student are described in the next chapter. The students were asked to provide these markers in the initial screening survey which I then used to compose the groups. More on group composition can be found in Chapter 3.

As Adiredja and Andrews-Larson (2017) describe, postsecondary mathematics education research is taking a sociopolitical turn in that there has been a necessary push to better understand social discourses in the educational context and how these discourses impact equity, power and students' mathematical identity development. For social discourses, I draw on Gutierrez's (2013) definition of "taken-for-granted ways of interacting and operating" (p. 43). These discourses emerge in all aspects of our lives as we move through the world and interact with others. Given that I am asking my research participants to work in a small-group collaborative setting, it is important that I consider the taken-for-granted ways of interacting within each group in my study. That is, I am taking up the sociopolitical push in mathematics education research to attend to the factors that may contribute to one's identity development and better understand how these factors are manifested through the unique interpersonal dynamics that arise in each group of my study.

Research Questions Revisited

Having reviewed some of the literature on instrumental genesis, HLTs, RME, PRIMM, and mathematical identity, I revisit my research questions here.

- 1. What kinds of instrumented action schemes develop while using Python in learning about set theory and logic?
- 2. Over the course of an actual learning trajectory, what characterizes students' increasingly sophisticated ways of reasoning about set theory and logic?
- 3. How does the use of Python to learn mathematics, in a small-group collaborative setting, influence students' affective experiences and the development of their mathematical identity?

Each research question has been elaborated to target the specific focus of analysis. The methodology and details of the approach for analysis for each research question are detailed in Chapter 3.

Chapter 3: Methodology

In this study I aim to characterize a way in which computing can be used as a tool for the teaching and learning of mathematical set theory and logic. To accomplish this goal, I leveraged what is known as Design-Based Research (DBR). DBR consists of the following five characteristics: (1) the design of learning environments and learning theories go hand-in-hand, (2) development and research undergo continuous cycles of design, enactment, analysis and redesign, (3) research must lead to relevant and applicable theories for other design educators and practitioners, (4) research must consider the implementation of design and learning theories in a real learning context, and (5) findings from the research must be documented using methods that can connect students' mathematical activity to the outcomes which lead to learning theories (Design-Based Research Collective, 2003).

One form of DBR is that of a Conjecture-Driven TE (Confrey & Lachance, 2000). Unique to a Conjecture-Driven TE is that the conjecture is a means to "reconceptualize the ways in which to approach both the content and the pedagogy of a set of mathematical topics" (Confrey & Lachance, 2000, p. 235). The conjecture driving this work is the following: *programming can not only be leveraged as a processing tool, but also serve as an experientially real context in which students will be able to connect mathematical logic and set theory that also positively influences their identities as mathematicians*. Central to this conjecture is the instructional design and delivery of how the computer programming language, Python, can be used to teach students about set theory and logic. In this case, the content is set theory and mathematical logic and the pedagogy was informed by the instructional design theories of RME and PRIMM. There are four reasons in particular for selecting Python: (a) the nomenclature of Python is quite similar to that of standard mathematical writing; (b) Python has built-in operators and functions that can process mathematical logic and set-related commands; (c) Python is a widely used programming language in mathematics, data science, and web development; and (d) Python is free and available through the use of Google Colab¹.

This chapter is divided into three main sections. First, I provide an overview of the TE which will cover the context, research participants, and general approach to data collection and analysis. The second main section is divided into three parts to provide a review of the analysis for each research question. As a reminder, the three research questions are:

- What kinds of instrumented action schemes develop while using Python in learning about set theory and logic?
- 2) Over the course of an actual learning trajectory, what characterizes students' increasingly sophisticated ways of reasoning about set theory and logic?
- 3) How does the use of Python to learn mathematics, in a small-group collaborative setting, influence students' affective experiences and the development of their mathematical identity?

I use the instrumental approach to answer Research Question 1 (RQ1), with the goal to analyze in-the-moment student reasoning. Analysis of the HLT and the corresponding actual learning trajectories as they relate to the RME levels of mathematical activity is used

¹ https://colab.research.google.com/notebooks/intro.ipynb

to answer Research Question 2 (RQ2). In the following section, I provide a description of the survey instrument that was used to inform analysis and consider additional aspects of students' affective experiences that will be used to answer Research Question 3 (RQ3). In the final main section of this chapter, I provide the tasks that were used in all five TE sessions across all four groups with some extra tasks that I used in my final session with Group 4. Along with the tasks I include sample code and a brief description of various design elements.

Conjecture-Driven Teaching Experiment Overview

In this section I highlight the specific elements involved with the data collection process for this study. This includes a description of who was recruited to participate, the context in which this study occurred, how the data was collected, when the study took place, and a broad overview of my approach to analysis for each research question.

Participants and Context

This study incorporates data collection from a pilot study and a main study. The pilot study consisted of one group of two students and the main study consisted of four groups, two groups of two and two groups of three. Participants were recruited from a four-year Hispanic-Serving Institution and were purposefully selected using criterion sampling (Patton, 1990) in that they have already taken, or are currently enrolled in, differential calculus or integral calculus and not enrolled in an ITP course. At the institution in which the participants were recruited, differential calculus is the prerequisite for the ITP course offered. Not being enrolled in or having already taken an ITP course was the primary selection criterion.

The secondary purposeful selection criterion was to meet set thresholds based on students' demographic information (e.g., race, gender, first-generation status, parent, or guardian, etc.). What this entailed was the selection of students with the primary intention of maintaining balanced gender identities (i.e., roughly half women and half men - with the awareness and acknowledgement that some of the research participants either may not want to disclose their gender identity or may identify elsewhere on the gender spectrum), as well as selecting a minimum of 30% underrepresented minority (URM) students. I gathered the 30% figure based on the undergraduate student population at the institution from which participants were recruited, as more than 30% identify as URM students. I stress the need for this secondary selection criterion to ensure that the research participants represent a wide range of social and personal identities. As the research shows, active learning in STEM classrooms is beneficial both for student performance (Freeman et al., 2014) and narrowing the achievement gap (Theobald et al., 2020), but there is still concern about the opportunity for negative and potentially harmful experiences in an active learning setting (i.e., Cooper & Brownell, 2016). Therefore, it is important for this study to include as wide a range of students as possible to learn about their affective experiences in a collaborative setting to combat the traditionally white, male, middle-class narrative that is often told in STEM education research.

To recruit students for the pilot study I asked one of the instructors of integral calculus (who taught eight virtual sections in the fall semester of 2020) to email their students with information about my dissertation and to ask them to consider participating. Included in the email sent to the students was a link to a screening survey, should they be

interested in participating, asking the students for their academic information in terms of their major, class standing, and whether they have taken the ITP course or plan to take it. The screening survey was administered through the online survey software, Qualtrics. This method of recruitment is known as an opt-in recruiting strategy, a method preferred by research participants (Willison et al., 2003). I used this same strategy of recruitment when I recruited students for the main study in the spring semester of 2021. For the main study I emailed the differential calculus instructors as well as the integral calculus instructors. In addition to general academic background information such as their major and whether they plan to enroll in an ITP course, the survey also asked for demographic information from the students to aid my secondary selection criterion. The full screening survey can be found in Appendix A.

Participant Overview

All the names used to reference the participants in my study are pseudonyms. The pseudonyms were selected using a random name generator that pulled names from a large database of names constructed from many different cultures and time periods.² The following subsections profile each of the ten students in my study. The numbers next to each participant's name represent their responses to the survey questions asking them to rate their knowledge and experience with set theory, mathematical logic, and computer programming, respectively. The wording of the question on the survey was the following: "On a scale from 1-10, where 1 represents little to no knowledge or experience and 10 represents extremely knowledgeable and experienced, how would you rate your knowledge

² https://www.behindthename.com/random/

and experience with the following." For example, Kristal (1/1/5), this means that Kristal reported a 1 for set theory, which represents little to no knowledge or experience, a 1 for mathematical logic, which represents little to no knowledge or experience, and a 5 for computer programming which represents somewhere in between little knowledge and extremely knowledgeable. The students also had the opportunity to put 0 or not answer the question at all, which I am interpreting as zero knowledge or experience with that topic.

Kristal (1/1/5). Kristal identified as a Black or African American and Native Hawaiian or Pacific Islander woman. She was a first-year Computer Engineering major and was also a student athlete. Kristal participated in all three sessions of my pilot study. Kristal was enrolled in integral calculus at the time of the pilot study.

Adeline (1/0/5). Adeline identified as a white woman. She was a second year Applied Mathematics major and participated in all three sessions of my pilot study. Adeline was enrolled in integral calculus at the time of the pilot study.

Judith (0/1/0). Judith identified as a first-generation college student and as a white woman. She was a second-year Physics major and a commuter student at the time of the dissertation study. Judith participated in all five sessions in Group 1 and was enrolled in integral calculus.

Haven (0/0/0). Haven identified as a first-generation college student and as a white woman. She was a second-year Environmental Engineering major at the time of the dissertation study. Haven participated in all five sessions in Group 1 and was enrolled in integral calculus.

Palmer (1/2/2). Palmer identified as a first-generation college student and as a white man. He was a first-year Civil Engineering major and a commuter student at the time of the dissertation study. Palmer participated in all five sessions in Group 1 and was enrolled in differential calculus.

Leo (1/1/2). Leo identified as a first-generation college student and as a Hispanic or Latinx man. He was a first-year Computer Engineering major and a commuter student at the time of the dissertation study. Leo participated in all five sessions in Group 2 and was enrolled in integral calculus.

Eugene (1/2/3). Eugene identified as a first-generation college student and as a Hispanic or Latinx man. He was a first-year Computer Science major at the time of the dissertation study. Eugene participated in all five sessions in Group 2 and was enrolled in differential calculus.

Saul (0/4/3). Saul identified as an East Asian and white man. He was a first-year Mechanical Engineering major at the time of the dissertation study. Eugene participated in all five sessions in Group 2 and was enrolled in integral calculus.

Juliana (0/0/0). Juliana identified as a first-generation college student and as a Hispanic or Latinx woman. Juliana also identified as a current or former English language learner. She was a first-year Environmental Engineering major at the time of the dissertation study. Juliana participated in all five sessions in Group 3 and was enrolled in differential calculus.

Delia (0/0/1). Delia identified as a Middle Eastern or North African woman. She was a first-year Computer Science major at the time of the dissertation study. Delia participated in all five sessions in Group 3 and was enrolled in differential calculus. **Alonso (1/1/2).** Alonso identified as a Hispanic or Latinx and white man. He was a first-year Mechanical Engineering major at the time of the dissertation study. Alonso participated in all five sessions in Group 4 and was enrolled in integral calculus.

Julian (0/0/2). Julian identified as a Southeast Asian man. He was a first-year Mechanical Engineering major at the time of the dissertation study. Julian participated in all five sessions in Group 4 and was enrolled in integral calculus.

As reported by the students, there were six women and four men in my study. Of the ten participants, seven reported a race other than white (including those that listed white and another race). By drawing on Shah et al.'s (2020) findings, I composed groups by avoiding unbalanced groups by gender and race/ethnicity (e.g., more men than women or only one person of color in a group of three). Intentionally attending to the group composition is not enough on its own to eliminate the possibility of harmful speech, or create an entirely safe space for the students, but it is one way to mitigate potentially problematic scenarios and create a space for the students in my study to feel welcomed and safe.

Data Collection and Analysis

The main TE consisted of five one-hour sessions and was conducted with four groups of two to three students in each group. I met with each group separately, which resulted in 20 hours of contact hours with the students across the four groups. Five sessions

were set as a target for two reasons. The first is that having conducted the pilot study, five sessions seemed to be enough time with the students to accomplish the goals of this study. The second was a desire to equitably compensate each participant for their time. With a max of five sessions per group, I was able to pay each student \$20 per session. Two surveys were also administered to the student participants, once before the TE and once after the TE concluded. The first survey is focused on the students' sense of confidence and enjoyment related to programming and mathematics. The second survey is content based. More details on the two surveys can be found in the following sections on Research Questions 2 and 3.

To inform the design of the first few sequences of TE tasks that I used in the TE, I conducted a pilot TE with one group of two students. The pace of the first two sessions of the pilot study ended up being much slower than I anticipated, so the tasks of the first two sessions were more streamlined for the main study. One example of this streamlining process was the removal of an index counter which was used to support the idea of iterations through a For Loop. As was the case with the pilot study, the main TE was conducted virtually using Zoom (due to the ongoing COVID-19 pandemic) and Jamboard for the participants to collaborate. Zoom is a video and web conferencing platform that all students at the institution have access to use. Jamboard is a Google app that serves as a collaborative and interactive canvas. For the pilot study, the students were asked to run their own code in Google Colab, a free Integrated Development Environment (IDE) accessible from any computer connected to the internet. After the pilot study, I realized that having the students copy and paste from Jamboard to the IDE was taking too much

time, and I could not see the code that the students were running on their own screens without screen sharing. To make the process more efficient, I decided that I would run all the code for the main study using an IDE (IntelliJ IDEA Version 2020.2.2) on my own machine and screen sharing the code with the participants. So, half of the screen that the students saw contained the task that they were asked to complete, and the other half of the screen was the IDE with the code. All but one of the TE sessions were recorded to the cloud via Zoom which automatically transcribed the sessions with fairly reliable accuracy. However, in order to fully analyze the data from the TE sessions I cleaned the transcriptions for formatting, grammar, and any errors that the automatic transcript produced. Also, filler words such as "um," and "like" were removed from student quotations for easier reading. A timeline of the data collection process for my study is provided in Figure 3.1.



Figure 3.1: Timeline of Data Collection

As Figure 3.1 shows, I collected data for the pilot TE in late January 2021. The rest of the data collection occurred in the spring semester and concluded just before the summer of 2021. Data collection for Groups 1-4 occurred using a staggered approach with two groups of students completing the first two sessions of the TE, then the next two groups started the TE. A staggered approach is used to incorporate a micro-cycle of DBR iterations (Prediger et al., 2015). By incorporating a cyclical approach to data collection, slight adjustments and modifications were made to home in on discussions and ideas that informed the conjecture driving this research. Confrey and Lachance (2000) describe two types of analyses that are often used for a conjecture-driven TE: (1) ongoing and preliminary, and (2) retrospective analysis of the data corpus. Ongoing and preliminary analysis consists of frequent (after each TE session) and critical reflections of emerging issues throughout the intervention. The focus of this aspect of analysis is to understand the implications of the students' mathematical activity related to the future plans of the TE. This step of analysis is a crucial aspect of being able to reflect on the conjecture (to elaborate rather than change) and be responsive to the students' needs from one TE session to the next. To conduct this step in analysis, I created contact summary forms (Miles & Huberman, 1994) to highlight the main concepts, themes, and salient points related to my research questions from the TE sessions with the students.

At a global scale, I conducted ongoing and preliminary analysis not only from one TE session to the next within each group, but also across each group of students. I developed cross-cutting themes and relevant hypotheses that influenced the design of the tasks and tested my conjecture throughout the data collection process. Figure 3.2 shows how the staggered approach to data collection for my study is leveraged in conjunction with the cyclical nature of retrospective analysis and ongoing development. This diagram was inspired by the developmental research cycle presented by Cobb and Yackel (1996) and modified to reflect the development of the conjecture for this conjecture-driven TE. Note that in Chapter 2 I highlight RME and PRIMM as guiding instructional theories. In a subsequent section in this chapter, I introduce the analytic frameworks that were used for the research phase of the TE.



Figure 3.2: Development of Conjecture through Research and Design

The second method of analysis that Confrey and Lachance (2000) describe occurs once the entire TE has concluded. At this point, a more thorough retrospective analysis of the entire data corpus using a theory-based approach is necessary. I approached the qualitative analysis for each of RQ1, RQ2 and RQ3 differently, with all three guided by existing analytical frameworks, but modifying the existing frameworks or using them as templates for my own purposes. This approach falls in the middle of the qualitative coding continuum as described by Miles & Huberman (1994). In Miles and Huberman's (1994) description of qualitative data analysis, they explain that their preferred method of analysis is to create a "provisional 'start list' of codes prior to fieldwork" (p. 58). These codes are used to assign meaning to a particular piece of information in the data. Unlike a pure grounded theory (Glaser & Strauss, 1967) method where one must create codes as they emerge, I approached the data with an existing framework relevant to addressing each research question (Appendix B). In the next sections I provide details of how the frameworks were used, but a description of the general coding process, based on Miles and Huberman (1994) is given here first.

First, one considers the analytical frameworks and previous empirical findings to develop a list of codes based on the problem of investigation. The next step is to apply this list of codes to the first set of data. The codes are then examined for fit and power (i.e., how well do the codes describe the data and to what extent are the codes leaving out information). At this step the original list of codes are revised or removed and new codes may be added. The third step is to ensure that the revised list of codes maintain a general structure and relate to each other in a coherent way. Once the set of codes seem to be descriptive and relevant for answering the research question of interest, one may continue to code future data while remaining open to the idea that codes may change, and new codes might still be added.

The entire data corpus that was analyzed using qualitative methods consists of the discourse between research participants, my interactions with the students, and the four surveys that were administered (two before the TE and two after). To carry out the analysis, I used the qualitative analysis software, MAXQDA to tag excerpts from the transcripts, and group the codes by research question. In the next section I describe my approach to qualitative analysis for each research question and I provide a full list of the types of codes that I used in the MAXQDA software in Appendix B.

Approaches to Analysis

I incorporate three analytic frameworks into my dissertation to understand the potential relationship and harmony between set theory, mathematical logic, and computing.

The first is instrumental genesis, as described in Chapter 2, which was used to characterize snapshots of students' reasoning in terms of how Python can connect ideas in set theory and logic. The purpose of introducing two mathematical topics (which have historically been taught independently) was to provide rich opportunities for students to see the connections between set theory and logic, as well as develop hypotheses of undergraduate student understanding as the students think about these topics jointly. The second is the use of a HLT, a framework that was used to conceptualize the growth of students' reasoning and in this case, to document what the students can learn over the course of five TE sessions. Results from the TE illuminated potential areas in which computing can be infused into the standard ITP curriculum. The third framework was built from various affective components based on my perspective on students' mathematical identity development which incorporates students' sense of self with respect to their confidence and interest in mathematics, their beliefs, attitudes, emotions, response to errors and selfefficacy. To guide the analysis of the third research question, pre- and post-study surveys were the most significant components guiding the analysis for the third research question. Specifically, these survey results guided the qualitative analysis of students' reflections and interpretations of participating in this TE. Results from the analysis provided insight into the students' sense of confidence and interest related to mathematics, computers and programming and programming as a mediating tool for learning mathematics.

Research Question 1

RQ1: Instrumental genesis

• What kinds of instrumented action schemes develop while using Python in learning about set theory and logic?

For analysis, I utilized Roorda et al.'s (2016) framework for identifying the techniques, conceptual elements, and technical elements as the basis for developing the students' instrumented action schemes. Recording the conceptual elements is done through the identification of the mathematical concepts that the students are learning. In analyzing the techniques used by the students (e.g., the output students produced, diagrams they drew, logical statements they considered), I watched the Zoom video recordings to document what exactly the students were producing in relation to conceptual elements. To document the technical elements (e.g., the code the students typed, the mathematics they wrote in jamboard), I analyzed the Zoom video recordings to document what the students typed and when they typed it. This analysis was done by tagging the audio transcriptions using MAXQDA with Concepts, Techniques, Tech-Elements as the main codes for analysis. If a student typed something without saying out loud what they had typed, I used the 'memo' feature within MAXQDA to document what the student typed. This way I was able to analyze all three aspects of the students' instrumented action schemes within MAXQDA without having to analyze data in separate documents.

By coding the techniques, conceptual elements and technical elements used, I was then able to construct each students' instrumented action schemes. Importantly, the students' schemes were constructed with the four features as described by Trouche (2004) and Vergnaud (2009) which included the goals, rules, operational invariants and the possibilities for inference. These four features also served as codes within MAXQDA to document excerpts from the TE that reveal information about their schemes, but were not a major component of the coding process. For example, coding the "goal" of a given instructional task was not particularly helpful as they were self-evident in many cases. In Appendix B I provide the list of codes that I utilized in my analysis with sub-codes that I used to maintain a coherent structure.

Research Question 2

RQ2: Learning trajectories in the context of set theory and logic

• Over the course of an actual learning trajectory, what characterizes students' increasingly sophisticated ways of reasoning about set theory and logic?

In analyzing RQ2, I focused on the four features of an HLT to measure the overall growth of what the students were able to learn throughout the TE. These four features served as codes in MAXQDA and are listed as Goals, Tasks, Mathematical Activity, and Instructor Moves. Again, the Goals code was not utilized as much as originally thought because the design of the instructional sequence was partitioned in a way that addressed the first goal of the HLT at the beginning of the TE and the second goal near the end of the TE. The Task code was used as a general code to document the task as helpful, unhelpful, confusing, etc. This code was used to document the engagement of the students for each task either by the questions they asked or the discussions they had with their peers. The Mathematical Activity code was not used in its general form, but subcodes were used to characterize the students' increasingly sophisticated ways of reasoning. The subcodes used

were the RME levels of activity as detailed in Chapter 2. The four levels are *situational activity*, *referential activity*, *general activity*, and *formal activity*. One might pause at the use of the levels of activity as constructs for analysis given that I introduced them as features for the instructional design heuristic of emergent modeling. However, the four levels of activity have proven to be useful as an analytic tool to frame the model-of/model-for transition (Rasmussen & Blumenfeld, 2007). The RME subcodes were used sparingly, mostly as documentation between tasks that I thought served as important transitional moments for the students. The Instructor Moves code started as a list of codes drawn from the findings presented by Andrews-Larson et al. (2017) and I added new codes that were relevant to my own instructional methods. The full list of codes are listed in Appendix B. This approach to analysis is in line with Miles and Huberman's (1994) preferred method of analysis given that I was open to the codes changing and often added new emergent codes.

Additionally, as stated in Chapter 2, I utilized a HLT and I was interested in monitoring the students' growth over time. To measure any growth or evidence of learning, the pre-study content survey served as a baseline for what the students may or may not have known coming into the study. The content survey is composed of 14 questions, eight of the questions are related to set theory, five of the questions are related to logic and the last question asks students to evaluate a piece of Python code. The only difference between the pre-study content survey and the post-study content survey is the last question about the Python code. The code was modified to be slightly more challenging and involve a For Loop, something that we spent a lot of time working with during the TE sessions. The full survey can be found in Appendix C.

Research Question 3

RQ3: Affective student experiences

• How does the use of Python to learn mathematics, in a small-group collaborative setting, influence students' affective experiences and the development of their mathematical identity?

Mathematical and Technological Confidence Survey

In their work to understand how MATLAB can be used to aid university students learning topics in algebra and calculus, Cretchley et al. (1999) asserted that students' attitudes must be considered when investigating the use of technology to support the teaching and learning of mathematics. As a result of being on this team of researchers, Fogarty and colleagues set out to validate an instrument that would measure the psychological constructs of attitudes towards mathematics, attitudes towards technology, and importantly, attitudes towards the use of technology to learn mathematics. This instrument was named Attitudes to Technology in Mathematics Learning Questionnaire (ATMLQ) (Fogarty et al., 2001). 289 university students completed the survey before an Algebra and Calculus 1 course and 184 students completed the survey after the course was completed.

Fogarty et al. (2001) found that their 34-item questionnaire loaded on three factors (as intended): Math Confidence, Computer Confidence, and Math-Tech. These factors accounted for 48% of the variance. Importantly, the findings from their efforts to test the validity of the instrument resulted in sound internal consistency, reliability, as well as high test-retest reliability. I administered this survey before the students participated in the TE

and again after they had completed their last TE session. While this dissertation is not designed to be a large quantitative study, data from the pre-and post-study surveys revealed changes in individual students' attitudes towards each of these three factors, illuminating areas for more inquiry. Analysis of the questionnaire is composed of documenting a positive or negative shift in the students' responses to the questionnaire items. The questions on the questionnaire are asked in the negative as well as the affirmative, which means that analysis consists of what I consider to be a positive or negative shift in the students' responses. For most of the questions, it is clear in which direction is the positive, for example, a positive shift in response to the question "I find mathematics frightening" would be in the direction of an "Agree" to a "Disagree." However, for the questions related to using programming to learn mathematics, I am operating under the assumption that programming is a useful and powerful tool that can be used to learn mathematics, a perspective that some students may not necessarily agree with. For these questions, I considered positive responses as those that agree with my perspective and negative responses as those that disagree with my perspective. The full survey is only 34 questions long and is presented in Appendix D.

I have made slight modifications to this survey in the last set of questions where there was an extra emphasis on the use of graphing calculators being considered a part of technology. My reasoning is that I do not want the use of calculators to dominate the students' perspectives on technology due to the high likelihood that they have had varied experiences working with graphing calculators in their previous mathematics courses. Further, I made modifications to the wording of the survey to include

"computing/programming" instead of general "technology." As mentioned, results from this survey helped inform and support my analysis of the students' experiences in the TE. In the following section I provide a more specific framework for what I focused on when discussing students' affective experiences.

Three Features of Students' Affective Experiences

Examining students' affective experiences throughout a teaching experiment can mean many different things. For the purposes of this study, I considered the following three components of students' affective experiences. First is their self-efficacy (Bandura, 1997) as some participants were unsure about their ability to program using Python and others found it fun and engaging. Before each TE session began, I conducted a quick check in with the students and asked how they were doing and asked how they were feeling about the days' activities. I also left time at the end of each session to ask the students to informally reflect on the tasks that they were asked to complete. However, some sessions ran all the way up to the hour and I was not able to ask the students to reflect on the day's activities. Mid-session reflections and comments by the students were also considered as points for analysis as the students showed evidence of self-doubt or strong interest in a certain task.

Given that the tasks presented to the students were challenging, and that this was the first time the participants were asked to write and analyze code, I anticipated that students would have to cope with not succeeding on their first try. Tulis et al. (2016) provide a model describing the processes that students might undergo as they experience errors. These processes describe both the motivational and emotional states that a student

might experience in relation to learning and making errors. Part of the process model provided by Tulis et al. (2016) and important for my analysis was the students' direct reactions towards errors (e.g., shutting down, frustration, curiosity), their indirect/secondary reactions (e.g., revise their answer to show that they now understand) and their emotional and motivational regulation strategies to "activate and sustain their cognitive, metacognitive and affective functioning" (p. 18). For some students, their motivational regulation strategies were to acknowledge that the problem was really difficult, and they had no idea what was going on, others increased in their determination to understand the material. It is important to note that my instructional approach entailed not evaluating students' wrong answers. That is, if a student said something incorrect, my first reaction was not to tell them that they were wrong, but to ask their partners what their thoughts were and try to start a discussion. For this reason, it was not entirely evident what the students' reactions were to their wrong answers at the moment, but the students did comment on the difficulty of the tasks, which I included as a secondary component of my analysis regarding their disposition towards errors.

The third feature of analysis draws on McLeod's (1992) general focus on students' beliefs, attitudes and emotions throughout the learning of mathematics. While beliefs, attitudes and emotions are quite general terms, McLeod emphasized the variety of components and factors that each contains. For example, beliefs can encompass one's beliefs about a given piece of mathematics, belief about oneself, beliefs about mathematics teaching and beliefs about the social context. Attitudes consist mostly of one's moderately stable positive or negative reaction to a given piece of mathematics. Emotions are less-

stable forms of affective measurement but are important to paint a fuller picture in terms of one's experience. For example, aspects of boredom, panic, fear, embarrassment, anxiety, happiness, and excitement are important emotions to consider. Although my analysis focused mostly on the students' sense of self efficacy and response to errors, analyzing the data with respect to general beliefs, attitudes and emotions will serve as a check to ensure that thorough analysis was conducted. As emphasized in Chapter 2, one's interest and recognition by their peers also plays a great deal into the development of their mathematical identity. I consider aspects of interest and perceptions of others to fall under beliefs, included in the fifth column of the table as presented in Table 3.1.

Analysis

Table 3.1 highlights how these features were considered in response to the three general categories from Fogarty et al.'s (2001) survey.

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics				
Computers and Programming				
Programming to Learn Math				

Table 3.1: Coding Framework to Document Affective Experiences

I used the framework as shown in Table 3.1 to analyze each student's general affective experiences throughout my study. Using the qualitative analysis software, MAXQDA, I

used the three features of affective experiences highlighted in this section as the main codes for analysis. Based on these codes, construction of the table consisted of copying and pasting short excerpts (or timestamps for longer excerpts) from each TE session into the table. As mentioned, participant comments made before, during and after the TE sessions were analyzed, as well as the free-response questions which were included in the post-study survey asking about their general experience of participating in the study. Once the table was constructed for each student, a smaller table was generated as a summary or general characterization of each student's affective experiences. Some of the excerpts that were posted in the original larger table were then used as data points to provide the reader with a perspective of each student's affective experience. Given that there were only 10 students in my study, all their answers to the free-response questionnaire items were provided for each student as well.

With respect to the second column labeled Pre/Post Survey Change, I measured shifts in either direction to capture changes in the students' beliefs and perceptions towards mathematics and technology from pre-study to post-study. Given that the sample size is quite small, general conclusions cannot be made about all students' experience in a TE utilizing Python to understand set theory and logic. However, with a small sample size I was given the opportunity to deeply examine the data on a question-by-question and individual student level. With this information I was able to provide a more robust interpretation of each students' overall experience in the TE.

TE Sessions and Example Tasks

The following tasks represent the HLT for my study. These tasks were informed by the pilot study and the structure of which tasks were used on each day were not consistent from group to group. That is, the tasks are presented here as the original hypothetical organization as opposed to how they actually occurred in the study for each group, as some groups moved through the material quicker/slower than others. The main lesson learned from the pilot study was that asking the students to manage their own IDE, work in Jamboard and being asked to reason about topics in set theory and logic was asking too much of the students for the first two sessions. The pace at which we were working through the activities was far slower than anticipated. In response to this, my approach to the main study was to share my screen with the Jamboard and my own IDE document up, side by side. Following the first three steps of PRIMM, the students were still able to Predict, Run (which I did on my end), and Investigate. For the Modify and Make steps of the PRIMM model, the students still had access to the Jamboard where they were encouraged to type code that I then copied and pasted into the IDE. As mentioned, the following TE sessions are presented in a general anticipated format, something one might be able to use in their own teaching experiment. That is, the days and specific tasks are not exactly as they occurred as I conducted the TE sessions and moved from one group to the next. I modified some tasks to streamline the activities, and the original versions of the tasks will be given in Chapter 4 as I discuss the students' progression in their advancing mathematical activity.

TE Session 1

The main goal of the first TE session is to introduce features of Python that will be used in the future TE sessions. For example, later in the TE the plan is to teach the students how to use a For Loop, which will be leveraged as a tool to work with sets. For Loop construction is not always simple and might be a substantial roadblock for students. Thus, establishing a strong foundation in the building blocks of Python is a major goal and necessity to get to writing For Loops in future tasks. This entails covering some of the unique syntax requirements for the code to run properly, but also addressing the established norms of writing code such as proper line length and code block separation. A secondary goal for the first session is to introduce some of the basic ideas of set theory and logic.

Given that my TE is only five sessions long, the introduction to Python starts immediately in the context of set theory. The following figures are what the students see in the Jamboard slides and I provide a description of each figure to explain how the example activities and tasks tie back to RME and PRIMM. In some cases, I provide example code to document ideal student responses.
In mathematics, a set is a collection of objects defined explicitly by the objects in the set. In Python, we use curly braces to indicate that we are working with sets. For example, take a look at the following code in Python: setA = {"dog", bird, "lion", "cat", "fox"} setB = frozenset(["dog", "lion", 4, "lion", "red", 4.37]) setA.add(9) len(setA) setB.add("whale") len(setB) What do you notice? What do you wonder? What do you predict will happen when you run this code? What does the 'len()' function do?

Figure 3.3: TE Session 1 Task 1

There are multiple aspects of this code that are important to highlight. First, the students are asked to point out features that they notice or have more questions about. Some students may notice or wonder that setA is defined using curly braces and setB is defined using 'frozenset(),' "lion" is repeated twice in the set B, the 'len()' function is doing something to setA and setB and the last two lines of code are in green text with the '#' symbol in front.

The purpose of this first activity is to start a conversation about the different presentations of sets, the types of classes that Python identifies and whether these classes are acceptable as part of a set. For example, "dog" is a *string*, 4 is an *integer*, and 4.37 is a *float* - all of these class types are acceptable as elements of a set. However, bird is an

undefined variable name and is not acceptable. When the participants run this code, they will get an error message.

 NameError
 Traceback (most recent call last)

 <ipython-input-13-1a72de625718> in <module>()

 ----> 1 setA = {"dog", bird, "lion", "cat", "fox"}

 2 setB = {"dog", "lion", "lion", 4, "red", 4.37}

NameError: name 'bird' is not defined

With this error message our conversation will focus on the Investigate step in the PRIMM method and we will continue to discuss the different types of elements allowed in a set. Without explicitly calling out this aspect of mathematical set theory, the idea behind highlighting the different types of elements in a set addresses Linchevski and Vinner's (1988) study in which most of the participants believed that sets can only contain the same type of elements. We will also consider the number of elements in each set. At first glance, it might seem that there are five elements in setA and six elements in setB, but in fact there are five elements in each set as sets do not count duplicate items.

Once the students have a firm grasp of the types of elements allowed in a set and know how to find the number of elements in the set, we will transition to logical statements and propositions in Python. cities = {"Sacramento", "San Diego", "New York", "Miami", "Portland"} a = 21 b = 5 print("Hello") print(a + b) print(a + b) print(a - b) print(a % b) print(a % b) print(a < b) print("Miami" in cities)

Figure 3.4: TE Session 1 Task 2

This task is designed to introduce the participants to the 'print()' function and to document the different types of outputs that the print function can produce. The statements, 'a < b' and "Miami" in cities' are the first propositions that the students will encounter, and the print statement will return the Boolean value (False and True, respectively in this case) when the argument is a proposition with a known truth value. To explore Boolean expressions in more detail, I would like the students to Predict, Run, Investigate, Modify and Make with the following activity.

Now consider the following: setA = {"dog", "bird", "lion", "cat", "fox"} setB = {"dog", "lion", "lion", 4, "red", 4.37} p = "dog" in setA q = (len(setB) == 5)r = "San Francisco" in setB print(p) print(q) print(r) print() print(not p) print() print(p or r) print(p and r) What do you predict the output will be? What is the 'not' command doing? How would you describe what the 'and' and 'or' operators are doing?

Figure 3.5: TE Session 1 Task 3

At this point I would say that the students are using Python and the 'print()' function as a model-of reasoning about propositional logic to determine the truth value of a propositional statement. Once the students have had the chance to work with their own propositional statements using the 'and' and 'or' logical operators, the next task will ask the students to combine the operators into one print statement.

Write one print function that involves multiple 'and' and 'or' operators. Before you run your code, predict what each other's output will be.

Figure 3.6: TE Session 1 Task 4

A student might come up with something similar to the following, 'print(p and q or r or p and r).'

I would consider the students to be engaging in referential activity with this task as they are taking a slightly more abstracted approach to the 'and' and 'or' operators in that there are multiple operators within the same print statement. The reference to the situational activity is that they can still go back to their previous print statements to reason about the new compound propositions they are being asked to create.

TE Session 2

The main goal of this session is to encourage further discussion about the relationships between propositional statements. The first task is a continuation of the last session, but we will engage in more general activity as the students will be asked to work with propositions with unknown truth values.

Let's assume that *s* is a proposition with the Boolean value of False, and *t* is a proposition with an unknown Boolean value.

Interpret the following propositional statements: (s and t) and (s or t)

Figure 3.7: TE Session 2 Task 1

With this task in TE Session 2, I keep the wording intentionally vague with the use of "interpret" to make room for the students to reason about these propositional statements in various ways. In the pilot study I asked the students what print(s and t)would produce and that seemed to have caused more confusion since that specific code cannot be run without defining s and t first. I do anticipate that the students will use what they learned from the first session, specifically from TE Session 1 Task 3, to reason about the propositional statements. If the students in the TE do decide that they want to run 'print(s and t)' to see what happens, then I will encourage that and support them. If not, then we will have other interesting discussions about the propositional statements.

Ultimately, the goal is to guide the students to develop a model-of reasoning that entails developing operational definitions of the 'and' and 'or' operators which they can then use as models-for reasoning about more sophisticated mathematical problems. The students will be working with different combinations and scenarios, for example, where s is true and t is false and thus will be guided to write their work in a more structurally organized way. Once we have the structure of a truth table, we can then reason about additional logical statements like (s and not t) and (s or t and u) (note that the second compound propositional statement depends on the placement of internal parentheses). Once the students are at the point of reasoning about compound propositions and solving

92

more advanced mathematical problems (model-for reasoning about propositional statements), then the students may be presented with more abstract logical problems which would then be considered the formal mathematical activity. I anticipated that Task 1 during Session 2 would occupy most of the time for TE Session 2 and thus leave this as the only main task for Session 2. I also anticipated that some groups will not finish the tasks planned from TE Session 1 and thus will need time at the beginning of TE Session 2 to finish. To conclude Session 2, or to begin Session 3, I planned to use the following task as shown in Figure 3.8.

How many elements are in the set $A = \{s, e, t, t, h, e, o, r, y\}$? How can you verify your answer using Python?

Figure 3.8: TE Session 2 Task 2

An ideal student response would be to write the following code:

$$A = \{"s", "e", "t", "t", "h", "e", "o", "r", "y"\}$$

len(A)

I anticipated that the students will utilize the 'len()' function due to our focus on its use in TE Session 1. This is then a good opportunity to present the students with an alternative way of finding the number of elements of A using a For Loop.

TE Session 3

The main goal of this session is for students to develop an understanding of how to write a For Loop. A secondary goal for this session (and more broadly across the entire TE) is for me, as the primary investigator, to gauge the extent to which students are able to reason about sets as objects that can be manipulated.

How many elements are in set A? How do you know? A = {"s", "e", "t", "t", "h", "e", "o", "r", "y"} for element in A: print("-----") What do you notice? What do you wonder?

Figure 3.9: TE Session 3 Task 1

C→	-	-	-	-	-
	-	-	-	-	-
	-	_	-	-	-
	-	_	-	-	-
	-	_	-	-	-
	-	_	-		-
	_	_	_	_	_

Figure 3.10: TE Session 3 Task 1 – Output

Here, the purpose of this code is to start a discussion with the students about what might be occurring in the For Loop. Some questions that might be asked at this stage are "What is 'element' doing in the For Loop?" and "How might the number of lines in the output relate to the set A?" We take this one step further with the next task as seen in Figure 3.11.

Now let's consider the following code: city = "San Diego" A = {"s", "e", "t", "t", "h", "e", "o", "r", "y"} for character in city: print(character) print("-----") print("*******") for element in A: print(element) print("-----") Run the first For Loop. What do you notice? What do you wonder? What do you predict the output for the second For Loop will be?

Figure 3.11: TE Session 3 Task 2

Again, the purpose here is to encourage the students to ask more questions and notice that their predictions about why the output for the second For Loop might not be what they anticipated. In the original design of the tasks focused on the For Loop, I introduced the idea of a running index counter that would count how many times the For Loop iterated through each iterable object. I did this with Group 1, and unfortunately it introduced unnecessary complications and confusion for the students. For Groups 2, 3, and 4 I instead introduced the tasks as presented in Figures 3.9 and 3.11. Figure 3.12 shows what the output would be for this code.

s C≯ ____ а ____ n ____ ____ D ____ i ____ e ____ g ____ 0 ____ ****** 0 ____ r ____ е ____ s ____ У ____ h ____ t ____

Figure 3.12: TE Session 3 Task 2 – Output

Notice that the string type, "San Diego" is ordered, and the elements of set A are not. This will be a topic of discussion to highlight that sets are not ordered and to serve as a reminder that sets do not count repeated elements. The next task will introduce the students to the programming procedure of first defining an empty set to create new sets.

What do you predict will be the output of the following code? A = {"s", "e", "t", "t", "h", "e", "o", "r", "y"} B = set() for x in A: B.add(x) print(B) print(B)

Figure 3.13: TE Session 3 Task 3

What do you predict will be the output of the following code? A = {"s", "e", "t", "t", "h", "e", "o", "r", "y"} D = set() for x in A: if ((x == "e") or (x == "o")): D.add(x) print() print(D)

Figure 3.14: TE Session 3 Task 4

In this case D is a set of the vowels from set A. Also note that the students are required to reason about the logical operator 'or' that is situated within the For Loop. Now that we have an introduction to the use of For Loops, I present the following task that will be the basis for the following TE sessions.

Create your own sets of items that you might get at a grocery store. Make sure to include elements other than strings. For example, {20, "bananas"} could represent the set containing \$20 cash back from your purchase and bananas.

Figure 3.15: TE Session 3 Task 5

The sets used in the pilot study are presented in Figure 3.16.

A = {8, "apples", "chocolate", "berries", "corn", "juice", 13, "strawberries", 6, "avocados", "beets", "chips"} B = {1, "hot cheetos", "jalapeno", "onions", "cilantro", 2, "limes", "chips", 6, "cherries", 9, "corndogs" } C = {8, "biscuits", "salami", "cheese", "soda", "water", "bananas", "beets", "watermelon", 7,"kiwis", "oranges", "coffee", 9, "cookies", "ice cream", "sugar", "honey", "butter", "milk", 2, "pineapples", "chicken", "sausage", "beef", "ribs", "pepper", "salt", "candy", 4, "lemons", "parsley", "bread", "mayo", "mustard", "soy sauce", "ketchup", 37, "dog treats", "chips"} How many elements do we have in each set? How many unique elements do we have overall? How do you know?

Figure 3.16: TE Session 3 Task 6

A plausible student response would include the use of For Loops and the '.add()' function

(which the students saw in TE Session 3 - Task 4 to create a union of the three sets,

followed by a 'len()' function of the newly created set:

len(A) len(B) len(C) U = set() for i in A: U.add(i)

```
for i in B:
U.add(i)
for i in C:
U.add(i)
len(U)
```

Once we have found the union of the three sets, the next TE session will start by asking about the set intersection.

TE Session 4

The goal of this session is to start the movement up through the levels of activity as we transition from the situational activity of working with grocery item sets to sets of integers. First we will begin by finding the intersection of the sets from TE Session 3. Then we will transition to a number theory problem.

Using Python, how would you find the common elements across all three sets?

Draw a diagram of what this might look like first before you write any code.

Figure 3.17: TE Session 4 Task 1

The goal of this activity is to help students see that logic may be used in the context of set theory-based questions. Specifically, 'in', 'and' and 'or' operators can be used within For Loops to control the creation of their new sets. By having the students construct their own sets, I am engaging the students in a situational activity that they will use to develop the ideas of set operations such as intersection, set difference, and symmetric difference. For example, the intersection of sets produces a new set containing the elements present in all sets. If there are no common elements between the sets, the intersection will produce the empty set. Ideal student code for finding the intersection between all three of the sets would look like the following:

```
I = set()
for x in A:
if ((x in B) and (x in C)):
I.add(x)
print(I)
```

In the sets I provided in Figure 3.17, the intersection of the three sets is {"chips"}. Here I will make the decision to "tell" the students that {"chips"} is a subset of the three sets. I make the decision to tell the students that {"chips"} is a subset of the three sets to bring up the concept of subsets and start a discussion about the difference between a set as a subset and a set as an element.

{"chips"} is what we refer to as a subset. By definition, if every element in set A is an element of another set, set B, then A is a subset of B and this fact is denoted by $A \subseteq B$

Figure 3.18: TE Session 4 Definition 1

The next task will introduce the number theory problem that will be the context of the referential activity.

Let's consider the integers from 1 to 1000. In Python we can call out these integers by using the 'range()' function.

range(1, 1001)

Is the set of integers divisible by 21, a subset of the set of integers divisible by 3? Is the set of integers divisible by 21, a subset of the set of integers divisible by 7? How do you know? How can you use Python to help answer this question?

Figure 3.19: TE Session 4 Task 2

The description of the task in Figure 3.19 and transition from model-of to model-for

reasoning about determining whether one set is a subset of another was described in detail

in Chapter 2 in the RME section. There are two other tasks that I used for the groups that

moved more quickly through the TE. These tasks are represented in Figures 3.20 and 3.21.

TE Session 5

The purpose of this session is to either finish the tasks that were not completed in the previous sessions or to engage in more mathematics with the following tasks. The first task can be used as a warmup for the fifth session, as seen in Figure 3.20.

How many elements belong to the set $A = \{1, 3, 7, \{1, 3\}, \{\}\}$. How might you use Python to help you answer this question?

Figure 3.20: TE Session 5 Task 1

Set A has 5 elements. Drawing a Venn diagram might actually cause more confusion as described by (Bagni, 2006) as the set containing elements 1, 3 might make it seem that the set A has duplicate elements, namely 1 and 3. My hypothesis is that by writing code in Python to describe set A, the students will be able to develop a more sophisticated understanding of what it means to be an element of a set. Driving this hypothesis is that

Python requires sets as elements to be defined as a 'frozenset()' first. As initially presented in the first TE session, one cannot modify frozen sets once they have been defined. Example code to answer the activity above would look something like the following:

B = frozenset([1, 3])C = frozenset() $A = \{1, 3, 7, B, C\}$ len(A)

The following task is the first step in approaching more formal mathematics.

Let D be the set of integers divisible by 21. Let E be the set of integers divisible by 3. Let F be the set of integers divisible by 7. How would you show that D is equal to the intersection of E and F?

Figure 3.21: TE Session 5 Task 2

We got to this task with Groups 3 and 4. Given the time constraints with five sessions, the goal is not necessarily to write out a formal proof, but talk about what it means to select an arbitrary element from a set and use that element to prove that each set is a subset of the other.

Chapter 4: Instrumented Action Schemes

This chapter is dedicated to answering RQ1: What kinds of instrumented action schemes develop while using Python in learning about set theory and logic? To answer this question, I draw on the instrumental approach (Artigue, 2002; Guin & Trouche, 1998; Trouche, 2004). As Buteau et al. (2020) describe, with the instrumental approach we can better understand how text-based programming can be integrated into students' mathematical activity, and as a result, understand how the use of the artifact can help or constrain one's mathematical understanding. In this chapter I present six instrumented action schemes of students' mathematical activity, each pertaining to one of three main mathematical ideas. Each of the six schemes are presented by primarily highlighting examples of one student's work. That is, each of the schemes are not analyzed for every student. However, where appropriate, I bring in examples of how the schemes may relate to examples of other students' work. The goal of this chapter is to serve as a foundation for future studies, each focused on one particular scheme or mathematical idea. These future studies will analyze the emergence of the instrumented action schemes for every student in my study, but that is not the focus of this chapter. The purpose of this chapter is to highlight the schemes that pertain to the following three main conceptual ideas: propositional statements, set intersection, and subsets.

As a reminder, there are four main features of a scheme as described by Buteau et al. (2020):

- 1. the goal of the activity, with sub-goals and expectations;
- 2. rules of action: stable behaviors of the subject;

103

- operational invariants, which can be theorems-in-action (propositions considered as true) or concepts-in-action (concepts considered as relevant);
- possibilities of inferences. These possibilities are essential for the adaptation of the scheme to the specific features of the situation. (p. 1026)

A full description of the instrumental approach, including rules of action, operational invariants and possibilities of inferences can be found in Chapter 2. For each scheme, I present the emergence of the instrument (the coordination of the artifact and the user to solve a mathematical task) through examples of student work and qualitative excerpts. In this chapter, I only focus on theorems-in-action for the students' operational invariants instead of the concepts-in-action. I find that the concepts-in-action do not convey as much information as the theorems-in-action relevant to my perspective of the students' construction of their schemes. The way I interpret concepts-in-action is that they mostly serve as contextual or descriptive components within the larger instrument development. I provide a table with an overview of the scheme for each student that is selected for analysis.

As stated, six schemes are presented in this chapter relating to three main mathematical ideas. The first two schemes, Impossible-to-Answer scheme and Flexible Propositional Statement scheme relate to conceptions of propositional statements, specifically, what it means for a proposition to be "unknown." The Filter Every Element scheme and Monitor Change in the Cardinality scheme relate to finding set intersections. The Determine Set Equality scheme and Verify Each element scheme address students' conceptions of what it means for one set to be a subset of another set.

Propositional Statements

A proposition is a statement that carries a Boolean (true or false) value. For example, the statement, 'The Earth orbits the Sun' is a proposition that carries a True Boolean value and the statement, 'The Sun orbits the Earth' is a proposition that carries a False Boolean value. If I say that *s* is a proposition, without any additional information, then all we can determine is that *s* carries either a True or False Boolean value and it cannot be both at the same time. It is not that *s* represents some statement that is unknown, say, 'Life is a simulation,' which no one knows the answer to. In Python, it is possible to work with propositional statements by assigning a particular statement to a variable. Consider the following code, which is pulled from Figure 3.5:

The single equals sign acts as an assignment, where the expression on the left-hand side of the equals sign is assigned to represent the expression (propositions in these cases) on the right hand side of the equals sign. The double equals sign is used to determine whether the values of the two expressions on either side of the double equals sign are equivalent. In this example, p is a proposition that has a True Boolean value because "dog" is an element of setA, q is a proposition with a True Boolean Value because there are five elements in setB, and r is a proposition with a False Boolean Value because "San Francisco" is not an element of setB. The students were asked to verify the Boolean value of each of these

propositions by investigating the output of the following code, which utilized the 'print()' function:

print(p)
print(q)
print(r)

This code would produce the following output:

True True False

The students were also asked to evaluate print statements such as the following:

print(p and q)
print(r and q)
print(q or p)
print(r or q)
print(p and q or r or p and r)

Working with these print statements helped the students build an understanding of how the logical operators 'and' and 'or' functioned when they evaluated propositional statements composed of propositions with known Boolean values. It is important to note that in Python, if one tries to use the print function with a variable that is unknown, or has not been assigned, say 'print(t),' one will receive the following error message:

NameError Traceback (most recent call last) <ipython-input-2-8061e4e0faac> in <module>()

---> print(t)

NameError: name 't' is not defined

The following subsections highlight two students' schemes of how they conceptualized what we referred to during the study as an unknown proposition, that is, a proposition with a Boolean value that was not made explicit to the students. In the first example, I provide a brief example of how one student initially conceptualized the concept of an unknown proposition and how this conception developed through an instrumental process. The second scheme is more representative of how the other students in my study interpreted the unknown propositions.

Impossible-to-Answer Scheme

The following task, represented in Figure 4.1 was given to the students in Group 1.

Provide an example of a proposition with an unknown truth value.						
Let's assume that <i>s</i> , and <i>t</i> are two propositions with unknown truth values.						
Interpret the following propositional statements:						
(s and t) and (s or t)						

Figure 4.1: Main Study Unknown Proposition Task

At this point in the TE, the students had already worked with the logical operators 'and' and 'or.' The students were familiar with the idea that the 'and' logical operator requires two true values on either side of the operator to make the entire statement true and the 'or' operator only needs one true value to make the entire statement true.

When I introduced this task to Group 1, Haven's initial reaction was to consider what it means to divide something by zero, "At first, I was thinking of zero and how, you know, you can't divide by zero and stuff. I don't know if that would be unknown, that's just not possible." It wasn't clear to me at the time why Haven would use dividing by zero as an example of a proposition, so after a few seconds I asked Haven if she could clarify her answer. It was evident that Haven was still thinking, but she changed her response while keeping the theme of working with zero by offering the following proposition, "I don't know, zero isn't an even number, technically that's unknown." It seemed to me that Haven was wrestling with two ideas, the first was defining a proposition, which is a statement that carries either a true or false value, and the second is determining what it means for a proposition to be "unknown." My interpretation of her perspective is that she was understanding an unknown proposition as a statement that, from her perspective, no one knows the answer to. However, from a mathematical standpoint, we know that zero is an even number, and we also know that division by zero is undefined. Perhaps the undefined nature of dividing by zero, that is, an expression not having any meaning at all, is what Haven had in mind for an unknown proposition. However, it is more likely that Haven was attempting to find an expression that no one would know the answer to, which in my mind is different from an expression being undefined. This falls under the same category as my 'Life is a simulation' example, and thus I am characterizing Haven's initial perspective on an unknown proposition as an Impossible-to-Answer scheme.

Palmer, one of Haven's partners offered the following definition of an unknown proposition, "If we use a new variable that isn't defined, it would be unknown because there'd be no way to tell if it was true or false." To elaborate, Palmer was referring to an undefined variable in the computational programming sense. For example, in the previous scenarios, we had defined p, q, and r, but if we had another variable, say s, and not assign sto any propositional statement using the single equals sign, then that would be an undefined variable, which is what Palmer was referring to. This definition that Palmer offered was closer to what I meant by "unknown proposition," so I asked Haven to interpret the following piece of code in Python:

print(s and t)

This print statement was the only line in the code block, and thus, *s* and *t* were not defined. As an attempt to help Haven answer this question, I assigned hypothetical examples of what *s* and *t* could be. I suggested that perhaps one could consider that *s* represents 'Haven will ace her next exam' and *t* represents 'Judith will ace her next exam.' Each of these propositions are unknown given that these statements refer to a situation that will take place in the future, but they are not impossible to answer. They are propositions that one will be able to answer in the future.

As for Haven's scheme, the goal of the activity was to determine the output of the statement 's and t.' With the hypothetical propositions in mind, Haven stated that "either [proposition] can be true or false, I was just thinking that it would be funny if it just popped out the answer, 'maybe.' Like, maybe? I don't know." Haven's primary rule-of-action here is to 'run the print statement to determine the output.' After running the print statement, 'print(s and t),' we saw that the output produced an error message similar to the one above, that stated 's is not defined.' For the operational invariants in Haven's scheme, my interpretation is that there are two primary theorems-in-action at play. The first is that 'A single output, or correct answer, will be produced when asked to evaluate a print statement.' The second is that 'Propositional statements can only be evaluated if the propositions are designated with one truth value.' These theorems-in-action are supported

by a secondary instance in which prior to seeing the error message, Haven mentioned that she was curious about how the 'and' and the 'or' operators "would change the output because we don't know what either are." The first theorem-in-action, 'A single output, or correct answer, will be produced when asked to evaluate a print statement' is supported in her use of "the answer" in her statement about Python producing the output of "maybe." Additionally, Haven references "the output" in her statement about not knowing what the print statement would produce because "we don't know what either are." In both cases, Haven is referring to a singular output, or answer. Haven had seen error messages in previous tasks, and thus was aware of the possibility of an error message but did not mention the possibility of one occurring in this instance.

In Haven's statement about not knowing what the output will be because, "we don't know what either are," Haven was referring to the truth value of the propositions. While Haven did state earlier that either proposition could be true or false, it is likely that she was referring to the possibility of them acing their exams in the future, but that it was impossible to know at the time whether these propositions were actually true or false. Therefore, the inference (the fourth feature of her scheme), is that the output would be indeterminate in a scenario in which one is asked to evaluate an unknown proposition. This contrasts with the possibility of the output either being True or False depending on the potential Boolean values of the propositions. This would lead to multiple outputs, in the case of 's and t', we would have four outputs depending on the truth values of the propositions (i.e., False, False, False, True). While it is true that we do not know whether the propositions would carry either a True or False value, Haven's response that the output

could be "maybe" suggests that neither of the propositions could be True or False at the time of evaluation, it is impossible to determine. Therefore, the output would reflect the unknowable nature of the propositions. If Haven believed that the propositions could take on either a True or False value, then the output would then be True or False, depending on the logical operator that was used. Table 4.1 represents a summary of Haven's scheme.

Instrumentation Scheme	Techniques	Conceptual Elements	Technical Elements
Unknown	Run the	Propositional	The statement 's and t' will produce one output if and only if s and t are propositions that are answerable.
Propositions lead to	print	statements can only	
Impossible to	statement in	be evaluated if the	
Answer	the IDE to	propositions are	
Propositional	determine	designated with one	
Statements	the answer	truth value	

 Table 4.1: Haven's Impossible-to-Answer Scheme

In Haven's case, the instrument she developed was a direct product of the information that was presented to her. Up until this point, we had only worked with propositions in Python that we explicitly defined (e.g., 'p = "dog" in setA'). Furthermore, each time we ran a 'print()' statement, it would produce only one output. Evaluating propositional statements in multiple different ways was not yet part of the instructional sequence, and perhaps not yet part of the students' conceptual understanding of propositions. Other students in my study also showed glimpses of this impossible-to-answer scheme, Haven just served as a good representative example. I would like to note that Haven did eventually experience a shift in her understanding of unknown propositions, and her shift falls under what I describe in the next section as a flexible perspective on propositional statements.

Flexible Propositional Statement Scheme

In this section I highlight a perspective on unknown propositions that I am calling a *Flexible Propositional Statement scheme*. For this scheme, I highlight Kristal and Adeline's work, the two students in the pilot study, as they reasoned about unknown propositions and the logical operator 'and.' Kristal and Adeline were presented with a similar task as Group 1, but with less structure, see Figure 4.2.

Let's consider two propositions with unknown truth values, *s* and *t*. What would the following print statement produce? print(s and t) Please explain your reasoning.

Figure 4.2: Pilot Study Unknown Proposition Task

Notice in this task the students were not asked to first provide an example of an unknown proposition. Also, in this task the students were asked to evaluate the statement 's and t' as an argument of a print statement.

When the students were asked about their initial interpretations of the problem,

Kristal stated, "My very first thoughts were that it would throw an error, because I didn't really understand what unknown truth values are." My response to Kristal's comment was to inform them that *s* and *t* are propositions that can hold either a True or False value. With this information, Kristal's thought process moved to think about the print statement from a probabilistic perspective, which Adeline took up in her own way of reasoning:

Kristal: So, there's a 75% chance of getting it False, and only a 25% chance- Oh wait no, that's not right...Interviewer: Can you say a little bit more about that reasoning?Kristal: They both have to be True in order for the whole thing to be True, and that's only going to happen one out of the four times. There's four outcomes that could [happen] and that's that they both could be True, they

both could be False, or one could be True, and one could be False and then switch that, the other is True, and the other is False. That was a really bad way of explaining, I'm sorry.

Interviewer: No, no, it's okay. Adeline, why don't you put into your own words what Kristal is saying, or reiterate what Kristal was saying.

Adeline: I think Kristal is- I didn't notice it before, but when she said it, I was like, 'Yeah that sounds like it's on the right track, it sounds like it's right.' Because if you do the little... I'm thinking of the little Punnett Squares from like, I don't know, it was biology.

Kristal: Oh yeah.

Adeline: When you do them and you pull them down, if it's the 'and,' since both of them have to be True for the 'and,' that's only going to happen 25%of the time. So, if we say False, it's 75% likely to be False- more likely to be False but it still has the option to be True? But then I think if it was 'print(*s* or *t*)' that would change the percentages, but since it's 'print(*s* and *t*),' what Kristal said was right with the two Truths, you know. Kristal: It's like the truth is the recessive gene. Adeline: Exactly.

The Punnett square the students are referring to was drawn immediately after this exchange

and shown in Figure 4.3. This excerpt is a good example of the instrumentation process

(the one-way influence of the artifact on the learner), in which Python's built-in

functionality of producing one output for a valid print statement is strongly influencing the

thought processes of Kristal and Adeline. That is, Kristal and Adeline are discussing the

likelihood that the print statement will produce either a True or False value, because it

seems they believe that the propositional statement must produce only one output. If we

take the assumption that Kristal and Adeline believe the propositional statement must

produce only one output because the statement is presented as an argument of a print

function, their determination that Python will produce either a True or False output given

the probability of each occurrence with the logical operator 'and' makes sense.

As for their schemes, there are the obvious similarities between Kristal and Adeline's ways of reasoning with that of Haven's (presented in the previous section). Specifically, all three students are operating with the same goal, to determine the output of the statement 's and t,' and the same rule-of-action of running the code in Python to determine the output. Kristal's initial reaction was that the output would produce an error (which requires first running the print statement), which supports this rule-of-action to be true. What is interesting is where Kristal and Adeline's schemes diverge from Haven's. First, the main operational invariant that they are utilizing is the theorem-in-action of 'All propositions carry either a True or False Boolean value, which determines the output of a propositional statement when evaluated. 'This idea of a proposition carrying either Boolean value at any given time is what I consider a flexible perspective on the nature of unknown propositions. By flexible, I mean that Kristal and Adeline both understand that the propositions can take on either Boolean value at the time of evaluation, and as a result, they are able to flexibly reason about the potential outcomes. This is demonstrated well in their use of the Punnett Square analogy, with their work represented in Figure 4.3.



Figure 4.3: Kristal and Adeline's Punnett Square

In Figure 4.3, the red "T:" on the left-hand side of the figure represents the proposition, t, and the red "S:" at the top of the figure represents the proposition, s. The red "T" above the

top left square represents the scenario in which s is True, and the red "F" above the top right square represents the scenario in which s is False. The red "T" to the left of the top left square represents the scenario in which t is True and the red F to the left of the bottom left square represents the scenario in which t is False. Lastly, the values inside of the squares represent the outcomes of the statement 's and t' corresponding to the Truth values of the propositions. For example, the T inside the top left square represents the scenario in which both s and t carry True Boolean values. For the statement 's and t, ' this is the only scenario in which one would have a True output.

I see a strong connection between Kristal and Adeline's use of the Punnett square with a traditional truth table. Drawing on Hawthorne and Rasmussen (2015), Kristal and Adeline demonstrate a form of a unified view of a truth table. Even though Adeline was the one to bring up the idea of a Punnett square, Kristal's last comment, "It's like the truth is the recessive gene," proves to me that Kristal understood Adeline's idea clearly. See Figure 4.4 (Genome Research Limited, 2021) for the similarity between Kristal and Adeline's Punnett square with a Punnett square that one may encounter in a biology classroom to demonstrate how recessive genes operate in nature.



B - dominant brown eye alleleb - recessive blue eye allele

Figure 4.4: Biological Example of a Punnett Square - Recessive Blue Eye Color In only one scenario, *bb*, do we get blue eye color. This scenario relates directly back to the one scenario in which we get True for the logical statement '*s* and *t*. 'To complete the description of Kristal and Adeline's scheme, one possibility of inference could be the scenario in which the students are asked to evaluate '*s* or *t*, ' and as Adeline said from the excerpt above, this would change the percentages. By "change the percentages" I am interpreting this to mean a new assignment of the values that would go inside of the Punnett square, in which we would have True, True, True, and False. We did not get to the creation of a new Punnett square for the logical operator 'or,' but Kristal and Adeline did demonstrate clear understanding of the 'or' operator in other future tasks. Table 4.2 summarizes Kristal and Adeline's flexible propositional statement scheme as described in this section.

Instrumentation Scheme	Techniques	Conceptual Elements	Technical Elements
Flexible Outcomes of Propositional Statements	Draw a Punnett square to determine the possible outcomes	Propositions take on the Boolean value of True or take on the value of False	The value of the statement 's and t' depends on the Boolean values of the propositions s and t. When evaluated, there are four outcomes, three being False and one being True

Table 4.2: Kristal and Adeline's Flexible Propositional Statement Scheme

For Kristal and Adeline, the instrument they developed was a direct result of describing how they believed Python may determine the output of the propositional statement '*s* and *t*' when *s* and *t* are unknown, or undefined propositions. To be clear, my understanding is they believed the code in Python would produce only one output. That is, there was a 75% chance of Python producing the output False and a 25% chance of Python producing the output True. While Kristal and Adeline were not technically correct from a computer programming perspective (the code would produce an error message), their understanding of the 'and' operator and how it functioned in the context of two unknown propositions was mathematically correct, and was a result of their desire to understand how Python might determine the output of the propositional statement.

This scheme of being able to flexibly reason about the output of a propositional statement was reflected by other groups in the study as well. Without documenting each step of very similar schemes built by students in Group 3, I present Figure 4.5 to document another case of the students coming up with a version of a truth table for the statement 's or t.'



Figure 4.5: Truth Table Generated by Group 3

This screenshot was taken from Group 3's Jamboard slide that had the same task as presented in Figure 4.1. With some guidance and annotation, the students reasoned about the different scenarios and possible outputs of the statement '*s* or *t*.' That is, the students in Group 3 showed evidence of flexibly reasoning about the various propositional statements and their outputs given unknown propositions. The primary piece of evidence for students operating with this scheme is the ability to reason about all the potential outputs at once, either through a truth table or otherwise, and be able to describe why each scenario is different as a result of the changing Boolean values of the propositions.

Set Intersection

11} and $B = \{1, 2, 3, 4, 5\}$, then the intersection of A and B is a new set containing three elements, $\{2, 3, 5\}$ because 2, 3, and 5 are elements of both sets A and B. In the following sections, I describe how two students used specific programming concepts, For Loops and If Statements, to find the intersection of multiple sets. As additional information, every group in the study was presented with the same three sets:

A = {8, "apples", "chocolate", "berries", "corn", "juice", 13, "strawberries",
6, "avocados", "beets", "chips"}
B = {1, "cheetos", "jalapeno", "onions", "cilantro", 2, "limes", "chips", 6, "cherries", 9, "corndogs"}
C = {8, "biscuits", 6, "cheese", "soda", "water", "bananas", "beets", "watermelon", 7, "kiwis", "chips"}

The intersection of the three sets, A, B, and C is a set containing two elements, {6,

"chips"}. Every group was presented with the following task, seen in Figure 4.6, asking the

students to reason about the intersection of three sets (we did not use the word

"intersection" until after the students created an algorithm to find the intersection).

Using Python, how would you find the common elements across all three sets?

Draw a diagram of what this set relationship might look like first before you write any code.

Figure 4.6: Set Intersection Task

Filter Every Element Scheme

In this section I highlight the work of Leo from Group 2 during the fourth session. While I focus mainly on the work of Leo, similar to the last section examining Kristal and Adeline's work, I found that this scheme applies to all of the students in Group 2. The main reason why I highlight Leo specifically is because up until this point he had not been as vocal about his thoughts as compared to his peers and during Session 4 Leo led the group in solving the task presented in Figure 4.6. Before drawing the diagram, I asked each student in Group 2 to share their thoughts on what the question was asking. Leo's response was that the task was to find the "elements that are shared between all three sets." Leo's other two partners agreed with this statement. The diagram that Leo drew is shown in Figure 4.7.



Figure 4.7: Leo's Set Intersection Diagram

Leo's diagram showcases three sets, A, B, and C, with the element of 2 in all three sets. This element is then added to a new set, which Leo called D. There is also the matter of what appears to be the elements A, B and C in the sets A, B and C, respectively. Without ignoring this fact, I am hesitant to speculate on exactly what Leo was intending when he included those elements in his diagram. It could be that the first elements that came to mind in Leo's mind were elements that were called "A," "B," "C." It could also be that perhaps Leo was using those letters as stand-ins for the other elements that belonged to the sets A, B, and C. The more important fact is that Leo's diagram represents that his conception of "common across all three sets" is in line with the set operation of intersection, which was the goal of this task. I mention this point because other students in the TE study interpreted "common across all three sets" as the set union instead of the intersection. Leo's other partners, Eugene and Saul, drew Venn diagrams for their representation of what the goal of the task was. Their Venn diagrams are shown in Figure 4.8.



Figure 4.8: Eugene and Saul's Set Intersection Diagrams

The distinction between Leo's diagram, versus Eugene and Saul's diagrams, is an important one. I argue that Leo approached the design of his diagram from a computational thinking perspective, an argument that cannot necessarily be made about the diagrams of his partners. This is supported by Leo's description of his diagram in that when asked about it, Leo said that he created a new set D and he was thinking that there might be "some code that would look for repeated values and add it to the new set D. Number two was repeated in all of them, so I brought it down to the new set." With this statement, it is possible to infer that Leo drew his diagram as pseudocode, where he did not know the exact code that needed to be written to achieve the goal, but represented a computational

process in the form of a diagram that searched every element in each set and selected the element that existed in all three sets. Pseudocode can also be a rough set of instructions, steps or processes that outline the computational process involved in constructing an algorithm to solve a task, without using all of the involved syntax that is required in text-based programming languages. This is often the first step in writing a computational process or algorithm to solve a problem. In contrast to Leo's approach, when Eugene and Saul were asked about their diagrams, they each said that they drew their diagrams to represent an overlapping area that would contain elements that all three sets shared. Eugene and Saul were right in their reasoning, and I do not want to downplay this fact, but what I want to highlight is that there is a closer connection between the computational process involved in finding the intersection between sets and Leo's diagram compared to his partners'.

As for Leo's scheme, the goal of the activity was to find the intersection of all three sets. Leo's rule of action was to '*Implement a code that would check each element in all sets and try to find a match for that element in each of the other sets*.' This is supported by Leo when he was asked to explain his diagram in more detail, "So originally what I had thought of was, you'd have three sets, or as many sets as you want, and I'm not sure if there's code that can search through each set and then just look for repeated ones. But then just add those values, the repeated ones, into a new set." Leo's explanation is in line with his diagram and also in line with the idea of set intersection.

Given that Leo was unsure about the specific code that we would write to make this computational process happen, I suggested that we go back to an example of a For Loop

122
that was used in the previous session, Session 3. The For Loop presented below was the last For Loop that Group 2 examined in a series of examples that I used to present the functionality of a For Loop to the students. Thus, by this time in the study, the students in Group 2 had a good understanding of the process of a For Loop and understood how the For Loop iterates through the designated object. In the following example, the For Loop iterates through a set with elements consisting of the letters that spell out "set theory."

One potential output of this code is the following:

h o {'o'} t r e {'e', 'o'} y s {'e', 'o'}

Looking at the output, we can see that the first step in the For Loop is to print each element in A. The next step is to verify whether this element is either "e" or "o." If the element is either "e" or "o," then this element is added to the set D and the set D is printed. This is why we see the set {'o'} right after the output of 'o' and the set {'e', 'o'} right after the output of 'e.' Reflecting on this For Loop, Leo said that one would "need code that would relate the sets. But you don't know which values repeat." Leo points to the line in the above code that states 'if ((i == "e") or (i == "o"))' and says that we can only use this "if you know which [elements] repeat.' As stated earlier, we know that '6' and 'chips' are the two elements common across all three sets. If the students knew which elements were repeated, they could theoretically just replace 'e' and 'o' from the if statement with '6' and 'chips.' This is what Leo was referring to by saying that they don't know which elements are repeated. The main goal is to find this information out by constructing a computational process that would work with any number of given sets. At this point I gave Leo's partners an opportunity to step in and offer their thoughts as to how we might be able to construct this process of verifying that a single element existed in all three sets. Eugene confirmed that we could use a For Loop, but then also stated that we can use an If Statement that would establish a condition that would need to be met:

So, yeah you would use a For Loop for this. And I guess you would use an If Loop as well. So, you could say that if x is in A and B and C, then you put the function to add.

Before writing any code, I asked Saul if he would like to offer his thoughts as well. Saul took up Eugene's thoughts and stated the following:

We're going to use a For Loop for each [element in] A and then we're going to compare it to B and C by saying if x is in B, I don't know how you write the code, but- And then you do 'and' because both values have to be True, instead of 'or,' I think. So you do 'and' x in C or something like that. And then you do the last line, which is add, or D dot add or something like that.

As we can see from Saul's statement, he was not confident in how exactly one would write this code. So, as Saul was verbalizing his thoughts, I was writing code in Python that represented his ideas. After the code was written, presented below, I asked Leo to reflect on the code and share whether the code resonated with his original thought of a computational process that checked every element across all of the sets.

D = set() for x in A: if ((x in B) and (x in C)): D.add(x) print(D)

Leo stated that the If Statement was "acting as restraints" to make sure that the element coming from set A exists in both B and C. When asked why we are using the 'and' operator and not the 'or' operator, Leo said that "we're using 'and' and not 'or' because if just one was True then it would add [the element], but you want both to be True. So, x in both A and B. C and B I mean." By "you want both to be True," I am interpreting this to mean both propositions, 'x in B' and 'x in C.' Both expressions are propositions because they can carry either a True or False Boolean value depending on the element that is selected in A. As the For Loop is iterating through the elements of A, the If Statement is being used to verify that both propositions are being met, that the element exists in B and the element also exists in C. For example, in the case of the element, "beets," this element exists in C but is not an element of B. Thus, the proposition 'x in C' is True, but the proposition 'x in B' is False. Since the conditional statement, 'x in B 'and' x in C' is not met, the program exits out of the If Statement and moves onto the next random element in A. In the case of the elements that exist in all three elements, like "chips," both of the propositions are True, satisfying the conditional of the If Statement, thus prompting the next line of code, which is to add "chips" to the set D. The output of the code written above produces the following output:

```
\{6, 'chips'\}
```

This is indeed the intersection of all three sets.

Given the data, I infer that Leo is operating with two main theorems-in-action. The first is 'Using a For Loop, one can iterate through every element of a defined set in *Python.*' The second theorem-in-action, and the one that I see as the foundation for his construction of the meaning of set intersection is 'Using an If Statement in a For Loop, with a specific conditional statement, one can filter for particular elements.' From the very beginning, with Leo's diagram, we saw the presence of a desire to search through every element of the given sets to find the repeated elements. This approach was then taken on by Leo's partners by utilizing the For Loop and If statements. Leo was then able to verify the computational process and determine that the written code did indeed reflect his original idea of looking for repeated elements. The fourth feature of Leo's scheme is the possibilities of inference. This emerged quite clearly in the fifth session with Group 2 as they were tasked to determine whether or not the set of integers divisible by 21 is a subset of the set of integers divisible by 7 and 3. This task will be the focus of the next section, but for the purposes of examining Leo's use of inference, I only highlight the first step of the code that the students in Group 2 came up with. The code is below.

126

A = set() for x in range(1, 1001): if (x % 21 == 0): A.add(x)

Here we can see that the students constructed a For Loop that iterates through a sequence of integers ranging from 1 to 1000 (the range function stops one value short of the second specified number). One can also see that the students in Group 2 utilized an If Statement and a specific conditional statement to filter for the values that are divisible by 21. The '%' sign in Python functions as the modulus. So, in this case we have x mod 21. Leo explains the code by stating the following:

Leo: The second line is looking through a range from one to one thousand and one. And then the third line it's just saying for any constant from one to 1000, where you do the mod 21, if that equals zero, then you add that in to set A.
Interviewer: That's right. And my question to you, Leo, why is it that we're setting x mod 21 double equal sign to zero?
Leo: You don't want a remainder. Yeah, you don't want a remainder, because if you have a remainder then it's not fully divisible I guess.

In Leo's first statement he says that the second line is looking through a range from 1 to 1001, but this was just a reference to the range function which stops one value short of the second specified number. The important part here is that both of Leo's theorems-in-action are present here in this possibility of inference, with a slight adjustment in that the code is not iterating through a set. First, Leo states that one can iterate through every value in a specified sequence of numbers. Second, the conditional statement is used to filter for specific values in that range. As with all possibilities of inference, one's operational invariants are subject to change. In Leo's case, the major change here is that one can

extend this filtering process beyond sets, in this case we ended up filtering for specific values in a range of integers.

As for the utilization of the For Loop and If Statement to filter for the intersection of the three sets, these computational tools were not designed to find the intersection of sets or any other specific mathematical task. They were designed to make computational processes more efficient. Leo, and his partners in Group 2, took these computational tools and used them to solve a mathematical task. This is the definition of instrumental genesis, the process of integrating an artifact, or artifacts, into one's thought processes as a means to solve a mathematical task. In this case, both the For Loop and the If Statement served as artifacts in the development of the instrument of finding the intersection of given sets. A summary of Leo's scheme is below in Table 4.3.

Instrumentation Scheme	Techniques	Conceptual Elements	Technical Elements
Find the Set Intersection by Verifying Every Element	Write a For Loop using an If Statement to filter for specific elements	The intersection produces a new set containing all elements that are shared in all of the original sets	By writing a For Loop, one can pass through every element in a set. These elements are then passed through an If Statement containing the logical operator 'and' to filter for the elements that are shared across sets.

 Table 4.3: Leo's Filter Every Element Scheme

Monitor Change in the Cardinality Scheme

In this section I present the work of Alonso, a participant in Group 4 as they worked on the same task as presented in Figure 4.6. The goal of the activity was to find the intersection of the three given sets. I highlight Alonso's work due to the unique nature of his approach to solving this task. No other student in my study attempted to solve the task in the same way that Alonso did. First, I will briefly present some information that will be pertinent to understanding Alonso's solution method. First, repeated elements in a set, in Python and in mathematics in general, are not counted multiple times for the total number of elements that belong to the set. For example, if we define $A = \{1, 2, 3, 4, 5, 5\}$, the number of elements that belong to A, otherwise known as the *cardinality* of A, is five. Second, the cardinality of a set can be determined in Python using the 'len()' function. This function finds the length, or the number of items in the iterable data object.

As with all of the other groups, I asked Julian and Alonso to describe, in their own words, the goal of the task. Alonso didn't quite describe the goal of the task and instead verbalized his initial thoughts in how one would solve this problem:

What you could basically do I guess is compare two sets and then, if they're the same- oh sorry, you could compare elements from both sets and if they are the same then you could print it or add it to another set. But, yeah it'd be a little bit more tricky because you have to check all the possible combinations. You can't just add it one at a time or check one at a time like we did with the last code.

The "last code" that Alonso is referring to is seen below, which was used to find the union of three sets, A, B and C. Set union results in a new set that contains all of the elements from all of the original sets. For example, if $A = \{1, 2, 3, 4\}$, $B = \{4, 5, 6, 7\}$ and $C = \{7, 8, 9, 10\}$, then the union of the three sets would be a set containing the elements $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

The code above uses a For Loop to iterate through all of the elements in sets B and C and add all of those elements to set A. This would result in the union of the three sets as the new version of set A contains all of the elements that were in the original sets A, B, and C. As Alonso said, one cannot just add all of the sets to a new set without first verifying that the element is repeated across the other sets. With respect to Alonso's description of the goal of the set intersection task, Alonso is considering the scenario where one has two sets and the intersection would involve pairing each element in one set to all of the other elements in the other set. Figure 4.9 is a diagram representation of Alonso's initial idea.



Figure 4.9: Diagram Representation of Alonso's Comparing Elements Method The dots in each set represent unique elements. In the figure I only draw arrows for the first and fourth elements in Set A, but the idea is that every element in Set A is being compared to every element in Set B. As an example, the first dot in Set A is being

compared to every other element in Set B and the second dot in Set B is a match to the first dot in Set A. This element would then be added to a new set, which is the intersection of Set A and Set B. All the different possible arrows that could be drawn in the diagram represent all of the "possible combinations" that Alonso was referring to. In his description, Alonso mentions comparing "two sets," which I asked for clarification on and he said, "All three, well all sets," indicating that this method could be done with more than three sets in the event that there were more sets. The reader may observe that this approach is quite similar to the approach taken by Leo and Group 2 in which every element in A was passed through an If Statement to determine whether or not the element had a match in both sets, B and C. What is interesting is that Alonso took a different direction when he started to develop his diagram documenting the computational process of finding the intersection of the three sets.

Once the goal of the task was agreed upon by both Julian and Alonso, I gave them about six minutes of silent work time to draw a diagram or representation of the computational process to find the intersection of all three sets. Also, it is important to note that before drawing the diagram, Alonso wanted to clarify what I meant by "diagram." I told him that I would "leave it up for interpretation." Figure 4.10 is a screenshot of Alonso's diagram, which he used as an opportunity to write pseudocode. Step through values in set A add value to set B and C Check to see if length has changed if it has not changed add value to set D Step through values in set B add value to set A and C Check to see if length has changed if it has not changed add value to set D Step through values in set C add value to set B and A Check to see if length has changed if it has not changed add value to set D

Figure 4.10: Alonso's Pseudocode for Set Intersection

Alonso and his partner (diagram not shown) were the only two in my study to write pseudocode. The other students either drew Venn diagrams, circled certain elements in the sets, or did some type of representation of what the intersection process might look like. Julian's approach was more in line with how Leo and Group 2 solved the problem, which is why I am not presenting Julian's work and instead mainly focusing on Alonso's work.

Here I will approach Alonso's pseudocode step by step. The first process in Alonso's pseudocode is to step through (another computer science term for 'iterate through') all the values, or elements, in A and add those elements to the set B and the set C. This step is fairly straightforward but requires a little interpretation as to what is meant by 'set B and C' in the second line. One could interpret this to mean the logical operator 'and,' because he refers to it as a 'set' instead of 'sets,' and would thus possibly imply the intersection of the two sets B and C. However, in his description of the pseudocode, Alonso clarified this line when he said, "So, if you had set A, you would add it to both set B, and C. So that all the values from set A get cycled through the two sets." This confirms to me that Alonso was referring to two separate sets, set B and set C. The next step in Alonso's pseudo code is to check the length, or in mathematical terms, check the cardinality of the two sets B and C. As Alonso describes it, "With each iteration you check to see if the length has increased. If the length has not increased, then you know that there has been a repeat." I am interpreting this statement as a process of adding each element from A to both sets B and C. One then finds the cardinality of the sets B and C. If there is a change in the cardinality from before the element in A is added to the sets B and C, then one can determine that this element is not shared with the set. For example, let $A = \{1, 2, ..., ...\}$ 3} and B = $\{4, 5, 6\}$. If we add the element 2 to B, then the cardinality of the set B would change from three to four since 2 is not an element of B. The next step in Alonso's pseudocode is to add the element to a new set, D, if the cardinality does not change. An overview of the whole computational process is described by Alonso in the following way:

This would require three For Loops and at the end you would just print the length of D and you would get the number of common elements. Or you could just print D to get which elements are in common.

The three For Loops that Alonso describes represent three iterative processes to add all the elements from each of the three sets to the other two remaining sets. Of course, three For

Loops are not required, which is an interesting distinction from Leo and Group 2's approach in which they realized that only one For Loop was required. We did not write out Alono's code as we were running close to the end of time during Session 4 and we still needed to hear from Julian about his diagram, but given that Alonso's method was so detailed, my hypothesis is that having the actual code would not change anything about his scheme or his conception of finding the intersection of multiple sets.

As for Alonso's scheme, we know that the goal is to find the intersection of the three sets. I consider the rule-of-action to be similar to Leo's, 'Implement a code that would check each element in all sets and verify whether or not that element existed in each of the other sets.' The way in which the elements are verified to exist in the other sets is the point of divergence between Alonso's approach and Leo's. For Alonso, the first theorem-in-action is the same, 'Using a For Loop, one can iterate through every element of a defined set in Python.' The second theorem-of-action is where the two approaches differ in that for Alonso, 'One can determine the existence of repeated elements by monitoring any change in the cardinality of a set once an element has been added to the set.' These theorems in action led Alonso to the solution method that has been presented in this section. Without Python, it is unlikely that Alonso would have come to that solution method, which is evidence that Alonso constructed an instrument, using the For Loop and the 'len()' function as artifacts, to solve the mathematical task of finding the intersection of multiple sets. For the possibilities of inference, I present Julian and Alonso's first step in solving the number theory task of determining whether the set of integers divisible by 21 is a subset of the set of integers divisible by 3 and 7. What is interesting is that Alonso and

Julian came up with the same first approach as Group 2 in writing a For Loop to construct the set of integers divisible by 21, but solved the task using different methods, reminiscent of their approaches to the set intersection task. A summary of Alonso's scheme can be found in Table 4.4.

Instrumentation Scheme	Techniques	Conceptual Elements	Technical Elements
Determine Set Intersection by Monitoring Change in the Cardinality	Add an element to a set and calculate the cardinality of the set	The cardinality of a set changes when a new, non- repeated element is added	By writing a For Loop, one can pass through every element in a set. These elements are added to the other sets and the cardinalities of the other sets are calculated. If there is no change in the cardinality from before the element was added to after for all sets, then the element is a repeated element can can be added to a new set.

Table 4.4: Alonso's Monitor Change in the Cardinality Scheme

Subsets

The third and final mathematical concept that is covered in this chapter is that of subsets. Consider any two sets A and B. A is a subset of B if and only if every element in A is an element in B. So, for example, let $A = \{1, 2, 3, 4\}$ and $B = \{1, 2, 3, 4, 5, 6, 7\}$. A is a subset of B because all four elements are elements in B. The goal of the task for all three groups (Group 1 did not get to this question) was the same, but the wording of the problem was slightly different between Group 2 and Groups 3 and 4. Figure 4.11 is the task that was presented to Group 4 and Figure 4.12 is the task that was presented to Group 2.

Let's consider the integers from 1 to 1000. In Python we can call out these integers by using the 'range()' function. range(1,1001)

Is the set of integers divisible by 21, a subset of the set of integers divisible by 3? Is the set of integers divisible by 21, a subset of the set of integers divisible by 7? How do you know? How can you use Python to help answer this question?

Figure 4.11: Group 4 Number Theory Subset Task

Let's consider the integers from 1 to 1000. In Python we can call out these integers by using the 'range()' function. range(1,1001)

Is the set of integers divisible by 21, a subset of the set of integers divisible by 3 and 7? How do you know? How can you use Python to help answer this question?

Figure 4.12: Group 2 Number Theory Subset Task

The only difference between the two tasks is breaking the first question into two separate questions. I highlight the difference because this ultimately resulted in two different approaches, one by Leo in Group 2, and the other which was represented by Leo's partners as well as the participants in Group 4. Group 2 and Group 4 are highlighted again in this section by presenting two different schemes. This will serve as a follow up on the previous section as the schemes presented for Leo and Alonso documenting their conceptions of set intersection very much coincide with the two schemes developed to solve the subset tasks.

Determine Set Equality Scheme

As an extension of what was presented at the end of the Filter Every Element Scheme section, and as was the case with Group 2, the code below is what the students in Group 4 settled on to find the sets of integers divisible by 21, 3 and 7.

```
D = set()

E = set()

F = set()

for x in range(1, 1001):

if (x % 21 == 0):

D.add(x)

for x in range(1, 1001):

if (x % 3 == 0):

E.add(x)

for x in range(1, 1001):

if (x % 7 == 0):

F.add(x)
```

This code represents three computational processes in which the product is three sets, D which is the set of all integers from 1 to 1000 which are divisible by 21, E the set of all integers from 1 to 1000 which are divisible by 3, and F the set of all integers from 1 to 1000 which are divisible by 7. The next step was to determine whether A was a subset of B and C.

Given that we focused on Alonso in the previous section to represent Group 4, I would like to highlight Julian's work and ideas in the development of a scheme that I am calling the *Determine Set Equality* scheme. Julian's initial idea was to conduct a check of each element in D to see if all those elements exist in E: To find if the elements divisible by 21 are an actual subset of the integers divisible by three, you just would have to find if the elements of 21 are equal to, like all the elements that are divisible by 21, are equal to the elements in three, but not all of [the elements in E], obviously, because the [elements] divided by three would have much more, but just to make sure that they are an actual subset.

My interpretation of Julian's statement that "All of the elements that are divisible by 21,

are equal to elements in three" is that one needs to determine that every element in D has a

match with an element in E. Further, by stating that "the [elements] divided by three would

have much more," I am interpreting this as the cardinality of E is much greater than D,

which it is. When asked how one would go about computing this check process in Python,

Julian offered the following idea:

Julian: Because you're trying to find if the elements in D equal certain elements in E, then I was thinking about doing a 'for x in' statement and then doing a print. So I was thinking of doing 'for x in D, if x equals E', then you can [add] that into another set. So if the elements in E are the same as D, then you [add] that into another set and then you just find that set equal to D. If it's True, then obviously all of the elements in D are in E. So like 'for x in D, if x equals'... could you do 'if x is in set E?' or 'x in E' then just do 'G,' or 'A add x' Interviewer: And you want to add these elements to an empty set, is that correct? Julian: Yeah. And then after that do 'print A equals D' Interviewer: Okay Julian: And if that works, if that is True, then D would be a subset of E.

As Julian was talking I was writing code in Python. The resulting code is below.

A = set()for x in D: if (x in E): A.add(x)

print(A == D)

Again, Julian refers to an element being "equal" to E, but I interpreted that in the moment, and now, as an element having a match in the other set. As such, the third line in the code uses an If Statement to verify whether the element from D is also an element in E. If this is True, then the element from D is added to a new set, A. Running the code, we found that the output was indeed True. Alonso was not completely sure about the print statement in the final line of the code that Julian came up with, so I asked Julian to explain why he wanted to compare the sets A and D using the double equals sign (the double equals sign is used to verify that two expressions produce the same value whereas a single equals sign is used for assignment). In the following exchange Julian describes the purpose of the print statement and Alonso was able to jump into the conversation and provide evidence that he too was on board with why the print statement is a significant step to showing that D is a subset of E:

Julian: So set A is basically made up of elements that are the same in D and E. Basically we are making sure that the elements in A are the quoteunquote 'subset of E,' that they are actually the subset of E because then they would equal D. Which is what we wanted to know, if it was a subset or not.

Interviewer: Okay. What would happen, let's just say a hypothetical situation where we got an output of False? What would that mean about the sets D and E?

Alonso: So I think that would mean that the set of integers that are divisible by 21 is not a subset of the integers divisible by seven. So I think what that means is that... yeah I don't how else to say it than that. and ensures visible by.

Interviewer: And did you mean the set of integers divisible by three? **Alonso:** Yes, what did I say?

Interviewer: You said seven.

Alonso: Ah okay, I meant three.

Interviewer: Okay, and Julian, what are your thoughts?

Julian: Yeah I was thinking the same thing because, obviously, since we are checking if D is a subset then somewhere in there, the total amount of integers - or at least one of the integers in set D are not equal to set E so it

would not add it to set A and obviously the statements would not be the same.

I was initially confused with what Julian meant by his first statement, saying that "the elements in A are the quote-unquote 'subset of E.' However, after proposing the hypothetical scenario in which the output is False, it seems to me that Julian has a firm grasp on the purpose of the print statement. The purpose being a verification that the two sets, D, the set of integers divisible by 21 and A, the set of integers divisible by 21 and also divisible by 3, are the same. That is, all of the elements in D are divisible by three, which means that all of the elements in D are also elements in E, which by definition implies that D is a subset of E. This same process was done with set F, to find that D is a subset of F, as seen with the code below.

$$B = set()$$

for x in D:
if (x in F):
B.add(x)

$$print(B == D)$$

With this code the output is once again True, which means that all of the elements in D are divisible by 21 and also divisible by 7. Thus, all of the elements in D are elements in F, which implies that D is a subset of F.

Julian's scheme is composed of the goal, which is to determine whether D is a subset of E as well as a subset of F. The primary rule-of-action is to '*Construct a For Loop that will find the intersection of the two sets in question*,' (D and E, or D and F in this case). While Julian did not verbalize that the goal was to find the intersection, his

construction of the For Loop to find the elements divisible by both sets is the process of finding the intersection. As for the main theorem-in-action, it is my interpretation that Julian believes that 'If the set intersection of D and E(A) is equal to D, then D is a subset of E.' Again, the purpose of determining the truth value of the print statement 'print(A == D)' is to conclude whether or not D is a subset of E. If the output of this print statement was False, then this means the two sets are not equal and thus implies that there exists an element in D that is not divisible by 3. Therefore, every element in D would not be an element in E which would mean that D is not a subset of E. Lastly, stating the possibilities of inference is somewhat difficult, as this was the last task that I worked on with the students. However, Julian and Alonso did not skip a beat when they applied their Determine Set Equality scheme to determine that D is a subset of F. For this reason, they showed that their conceptual understanding was strong in determining whether a set was a subset of another set. As seen in their Monitor Change in the Cardinality scheme, the instrument that Julian and Alonso constructed to determine whether one set was a subset of another set utilized a For Loop and an If Statement as artifacts. I would argue that the typical approach (for someone not experienced or exposed to proof writing) to solving this task would be to check every element and verify that each element belongs to the other set in question. Julian and Alonso did not take this approach. Instead, their instrument was constructed utilizing the capability and efficiency of set relations in Python. A summary of Julian's instrument is below in Table 4.5.

Instrumentation Scheme	Techniques	Conceptual Elements	Technical Elements
Determine one set is a subset of another by utilizing set equality	Write a For Loop to find the intersection of the two sets in question	D is a subset of E if every element in D is an element in E	Using a For Loop, one can construct a new set A to be the intersection of two sets, D and E. A True value of the equality 'A == D' means that D is equal to A and thus a subset of E as every element in D is an element in E.

Table 4.5: Julian's Determine Set Equality Scheme

Group 2. Two members of Group 2, Eugene and Saul, also showed evidence of a Determine Set Equality scheme, and I highlight their code below to showcase a different example of how this scheme can be represented. It is important to note that the labeling of the sets used in Group 2 were different from the sets used in Group 4. The code that Group 2 wrote to determine the sets divisible by 21, 3 and 7 is below.

As one can see, Group 2 used the labels A, B and C instead of D, E and F. With that, Eugene described his initial thought process in determining whether A was a subset of B and C:

I initially thought of using a proposition which would be like the 'and' proposition since both would be True. But I guess I'm blanking on how to write that using the If Loop. Obviously, you start off with 'for x in A' to get the values inside the set of A. And then we have to check if those values, 'if x is in both B and C.' Therefore, if they are, we already know that it's a subset from just that- if all the values are True. So that's where I'm stumped, on writing the If.

If we take a look at Figure 4.11, we see that the question asks if the set of integers divisible by 21 is a subset of the set of integers divisible by "3 and 7." Given Eugene's immediate use of the 'and' operator inside of an If Statement, I am interpreting this as Eugene interpreting the question already asking about the intersection of the set of integers divisible by 3 and the set of integers divisible by 7. This was not the intended interpretation of this question, but Eugene handles this well with his use of the 'and' operator in the If Statement. He says that this computational process would immediately tell you whether A is a subset of B 'and' C (the intersection) given that the If Statement is True for every element in A. Even though Eugene essentially said exactly what should go in the If Statement, he says that he is stumped. At this point Saul jumps in and says that the If Statement should read 'if x in B and x in C then add to D.' Eugene followed up Saul's statement and said that "we already know that any value we pick is already inside the set A and we're trying to make sure it's a subset of both B and C. So, yeah I think that would work." When asked about how one would then check, or make the final determination that A is a subset of B 'and' C, Eugene said that we could "get the length of the new set and the

original set A and compare those two to see if it's true." By "true" I am interpreting this to mean the same as in previous tasks we used the double equals signs to compare two expressions and determine the truth value. As I had done previously, I helped the students by writing the code as they were talking, without going ahead of them and made sure only to write what they had said out loud. This resulted in the following code:

D = set() for x in A: if ((x in B) and (x in C)): D.add(x) print(len(D)) print(len(A))

To test Eugene's instrument that he constructed to determine whether A is a subset of B

'and' C, I asked him to consider a hypothetical scenario in which an element in A is not an

element of B or C, and consider what that might mean about the length of the sets:

Eugene: Okay you made me think of something else. Using the 'and' proposition both have to be True, right? Both the left prop- the left thing and the right thing have to be True for it to run.
Interviewer: [nods]
Eugene: Okay, I was confused with the 'or,' but it still stands. Yeah, one [of the lengths] would be less than the other, because they wouldn't contain an element that was in the original, which would mean that one of those elements is not in B or C, which would make it not a subset.
Interviewer: Okay, and I'm sorry I got cut up with something you said about the 'or,' but your answer to the question is that [the length of D] would be less than [the length of A]?
Eugene: Yeah the new [set] should be less than the old, so yeah you're right. D would be less than A.
Interviewer: If- if there was what?
Eugene: If one of them was not included.

Running this code produced the following output.

47

Thus, the length, or cardinality of D is equal to the cardinality of A. This means that every element in A exists in the intersection of B and C. Therefore, the conclusion is that A is a subset of B 'and' C.

Eugene's scheme is composed of the goal to determine whether A is a subset of B and C. The primary rule-of-action to 'Construct a For Loop with the condition 'x in A." The main theorem-in-action is that 'If the cardinality of the set containing elements in A and in the intersection of B and C is equal to the cardinality of A, then A is a subset of the set intersection of B and C.' This is supported by his determination that this is true by verifying the lengths of the two sets. Confirming that the two sets have the same cardinality was proof that A is a subset of the intersection of B and C, which was the goal for Eugene and Group 2. As the case with Julian's scheme, determining the possibilities of inference is difficult given that this was the last task with the students, and they did not have an opportunity to adjust their rules-of-action or operational invariants. A summary of Eugene's scheme is shown in Table 4.6.

47

Instrumentatio n Scheme	Techniques	Conceptual Elements	Technical Elements
Determine one set is a subset of another by utilizing set equality	Write a For Loop to pass the elements of A through an If Statement filtering for elements that belong to B and C	A is a subset of B 'and' C if every element in A is an element in B 'and' C	Using a For Loop, one can construct a new set by adding all of the elements in A that satisfy the conditional statement 'x in B 'and' x in C' to a new set, D. If the cardinality of D is equal to the cardinality of A, then A is a subset of the intersection B 'and' C

Table 4.6: Eugene's Determine Set Equality Scheme

While Julian's scheme and Eugene's scheme are different in the computational processes, I argue that the scheme name of 'Determine Set Equality' is fitting for both. The name encapsulates a broader conception dealing with the properties and relations of sets. For Julian, this entailed determining the set equality between D with the intersection of D and E. For Eugene, this entailed determining that the cardinality of the two sets, A and the intersection of B and C were equal. This contrasts with the next and last scheme presented in this chapter which focuses more on the individual elements that belong to the set of integers divisible by 21.

Verify Each Element Scheme

In the previous section we saw how Leo's partners solved this task by determining that the length, or cardinality of A is the same as a new set containing values that belong to A and are also elements of the intersection of the two sets B and C. In this section, I briefly highlight Leo's alternative method because his interpretation of the problem led to another unique solution. The examination of Leo's scheme is not as in-depth as the other schemes, but I thought it interesting as it coincides with his earlier Filter Every Element scheme. Immediately after Eugene presented his thoughts on utilizing the cardinality, Leo offered the following idea of finding the union of B and C, which is written mathematically as B \cup C:

We can add the integers of three and seven together to make a new set and then compare the integers of 21 to the new set to see if all of the integers in the set divisible by 21 - if those are in the new [set] of 3 and 7, then 21 has to be a subset of that one.

My interpretation of Leo's statement is that he wanted to combine all of the elements in B and C into a new set. This would effectively find the union of the two sets. Leo then wanted to compare A to the new set containing all of the elements from 1 to 1000 that are either divisible by 3 or divisible by 7. Leo offered his solution to this task immediately after Eugene presented his solution:

So I was thinking of- it's a little bit different than what [Eugene] has here. I'm not sure if the way I'm thinking of it is correct too, but I was originally thinking after where we said 'create a new set D,' like how [Eugene] did. Then you put D add B and D add C and then you just do the If Statement too. And if A in D then you'd put... I guess a proposition? The ending was kind of confusing. But I was just thinking kind of like that. You add both of them into a new set and then Just compare the A with the new set. If A is in D, then you do a proposition or I'm not sure exactly how you would do that.

As Leo was speaking, I was writing code that he could see and he verified that the code I wrote is how he was imagining it. The end is where Leo got stuck, he wasn't sure what could go after the If Statement which was checking to see that each element in A is an

element in D. I offered a suggestion because we had not seen something quite like this and we were running out of time during our last session. The code is below.

```
D = set()
for x in B:
    D.add(x)
for x in C:
    D.add(x)
for x in A:
    if (x in D):
        print("A is a subset")
    else:
        print("A is not a subset")
```

For the If Statement, I wrote an If Else statement that would produce the output 'A is a subset' if the element selected from A is in the set D and would produce the output 'A is not a subset' if the element selected from A is not an element of D. The output produced 47 lines of 'A is a subset.' Leo's Verify Each Element scheme is summarized in Table 4.7.

Instrumentation Scheme	Techniques	Conceptual Elements	Technical Elements
Determine one set is a subset of another by verifying each element	Write a For Loop to pass the elements of A through an If Statement as a check to determine if every element in A is an element of D	A is a subset of B U C if every element in A is an element in B U C	Using two For Loops, one can construct a new set by adding all of the elements in B and C to the new set, D. One then uses another For Loop and If Statement to check all of the elements in A and determine whether they also exist in $B \cup C$

Table 4.7: Leo's Verify Each Element Scheme

As for Leo's scheme, the goal in this case was to determine whether A is a subset of the union of the sets B and C. The rules-of-action were to '*Add all the elements in B and C to a new set, D*' and '*Use an If Statement to check every element in A to see if it exists in D.*' The primary theorem-in-action is that '*If every element in A satisfies the If Statement, then A exists in D and thus is a subset of D.*' This theorem in action was supported by my own writing of 'A is a subset' and then seeing the 47 lines of code with this output. While Leo did not solve exactly the same task as his partners, he did determine that A was a subset of the intended set in question, which was the union between B and C. The whole goal for this task was to work with the mathematical idea of a subset, and Leo achieved this goal. As with the last two cases, the possibilities of inference are nearly impossible to determine.

Chapter 5: Hypothetical and Actual Learning Trajectories

In this chapter I revisit the goals of the HLT as outlined in Chapter 2 and address the degree to which these goals were met by each group in this study. The research question that I answer in this chapter is the following: *Over the course of an actual learning trajectory, what characterizes students' increasingly sophisticated ways of reasoning about set theory and logic?* Given the nature of a conjecture-based teaching experiment with multiple groups, the tasks used in the instructional sequence were modified from one group to the next in an effort to narrow in on the most productive tasks and ideas for the students. Thus, the actual learning trajectories were slightly different for each group in my study. As a result, the extent to which the goals of the HLT were met also varies by group. As a reminder, the goals of the HLT are the following:

- Students develop operational definitions of the logical operators 'and' and 'or' and are able to flexibly reason about these logical operators to solve problem tasks.
- 2. Students are able to utilize Python and their conceptions of union and intersection to determine that one set is a subset of another set.

Note that in this chapter some details are lost in the comparison between the actual learning trajectories of each group in the service of a more detailed look into the instructional sequence. That is, this chapter highlights and how the instructional sequence supported the developmental progression, or movement between situational, referential, and general mathematical activities across all the groups. In this chapter I do not highlight

150

the step-by-step instructional task sequence for each group, which would result in the presentation of the same/similar tasks four times (one for each group).



Figure 5.1: Each Group's Progress through TE

This chapter is divided into three main sections, with the first two focusing on the two respective HLT goals. Within the first two sections, I highlight the instructional tasks relevant to the HLT goal, with examples of student work presented to represent the progression of the students' work. I present multiple student solutions in cases where students took multiple unique approaches to solve a task. However, some tasks were solved using the same or similar approaches. In these cases, I do not present all of the students' work. Additionally, some of the figures presented in this chapter are repeated figures from Chapter 3. Presenting the tasks here in this chapter is not only easier for the reader, but also highlights the evolution of the tasks used for the students across all four groups. As previously mentioned, the tasks presented in Chapter 3 are to be used as a reference or foundation for future work. The tasks presented in this chapter are what was

actually presented to the students. Additionally, the students' work is analyzed using the four levels of mathematical activity to track the developmental progression of the students in each group. However, it is important to note that the instructional tasks did not entirely support a clear progression from one level of mathematical activity to the next for both goals. With respect to the first goal, all of the students in the study showed evidence of constructing a model related to logical operators, and using that model to solve more sophisticated mathematical tasks. However, the instructional task sequence designed to support the second goal of the HLT did not result in student work that was evident of a model-of/model-for transition. This is largely due to an oversight in the design of the tasks related to set intersection and union, and how those tasks related to the number theory task meant to engage the students in reasoning about subsets. Specifically, the oversight was in not providing the students a situational context in which they were given the opportunity to develop a model-of reasoning about sets as subsets (compared to sets as elements) and set equivalence. The students' mathematical activity for those tasks are still presented to document their mathematical progression, but the focus on the model-of/model-for transition is not a focus in addressing the second goal of the HLT.

For the tasks that were modified from one group to the next, I highlight the differences between tasks and the reasons why I changed the task. For the tasks that I did not modify, I present examples of student work and document the extent to which this work was representative of all the groups, or highlight the differences between students' work from one group to the next. In the third section, I present the results from the mathematical content surveys that were administered before and after the study. The

surveys were administered as a way to determine what the students learned over the course of the teaching experiment study, and the survey questions consist of both multiple-choice questions as well as open-ended response questions. The survey results are presented by groups to document the differences in learning outcomes between the four groups.

Logical Operators

The purpose of this section is to highlight the developmental progress of the students' mathematical activity with respect to the first goal regarding the logical operators 'and' and 'or.' Specifically, I highlight the model-of/model-for transition that resulted in what I am characterizing as the 'proposition/operator/proposition' model. Moreover, the ideas that the students are reinventing with this model are the set operations of set union and set intersection.

Situational Activity

The first session with the students was mostly designed to help them become familiar with the Python environment and learn the basic principles of set theory and logic such as set membership, set cardinality, logical propositions as well as first exposure to the logical operators 'and' and 'or.' All of the groups were exposed to working with Python and exploring various outputs from the very beginning. For example, the first task presented to Groups 1, 2, and 3 is shown in Figure 5.2. In mathematics, a set is a collection of objects defined explicitly by the objects in the set. In Python, we use curly braces to indicate that we are working with sets. For example, take a look at the following code in Python: setA = {"dog", bird, "lion", "cat", "fox"} setB = frozenset(["dog", "lion", 4, 4.0, "red", 4.37]) setA.add(9) len(setA) What do you notice? What do you wonder? What do you predict will happen when you run this code? What does the 'len()' function do?

Figure 5.2: First Task for Groups 1, 2, and 3

With this task the students started to become familiar with the syntax of Python in working with sets as well as working with the idea of cardinality, or the size of the sets. Given that this task is not directly associated with the first HLT learning goal, I will skip providing excerpts on what the students had to say about this task. The purpose of providing this task is to document the beginning of the HLT as well as to use this task as a reference for the following section. The only difference between the first task presented to Groups 1, 2, and 3 and the first task presented to Group 4 is that I removed '4.0' from setB and replaced it with another 'lion.' Python interprets 4 and 4.0 as the same element within a set, and I wanted to stress the idea that only unique elements contribute to the cardinality of a set. By the time I got to Group 4, I realized that Groups 1, 2, and 3 spent more time discussing why 4 and 4.0 are considered as the same element in Python (because one is an integer and

the other is a float) as opposed to the general idea of repeated elements. By replacing '4.0' with 'lion,' I streamlined the discussion to the important conceptual topic of cardinality. The second important idea with this task is to understand the functionality of the 'frozenset(),' which is an immutable version of a regular set (typically denoted using the curly braces, {}). Immutability is the idea that one cannot change the basic properties of the object using a function in Python, such as removing or adding an element. In order to remove or add an element in setB, one must change the line that assigns the frozenset to the variable 'setB.'

The first relevant activity introduced to the students on the idea of logical propositions and operators is shown in Figure 5.3.

```
Now consider the following:
setA = {"dog", "bird", "lion", "cat", "fox"}
setB = \{"dog", "lion", "lion", 4, "red", 4.37\}
p = "dog" in setA
q = (len(setB) == 5)
r = "San Francisco" in setB
print(p)
print(q)
print(r)
print()
print(not p)
print()
print(p or r)
print(p and r)
What do you predict the output will be? What is the 'not' command
doing?
How would you describe what the 'and' and 'or' operators are
doing?
```

Figure 5.3: Introducing Logical Propositions and Operators

All of the groups in my study were presented with this task during the first or second session (Group 2 saw this problem in Session 2). In terms of the levels of mathematical activity, this task serves as the situational activity for the students. Often with a situational activity, the students are provided with an opportunity to start constructing their own models of certain mathematical ideas. In this case, the students are constructing models on how the 'and' and 'or' operators function when utilized in a propositional statement. Below are some excerpts of the students' reasoning on their predictions as well as their initial interpretations of what the 'and' and 'or' operators are doing.

In Group 1, Haven and Palmer discussed their initial thoughts on the output of the print statement 'print(p),' which produces a value of True because the element 'dog' is in setA:

Haven: I want to guess that it's going to print dog. Or maybe say True? I can't tell the difference between the propositions and then when you just do it, you know?
Interviewer: Yeah, yeah. Anybody want to tag in on that one?
Palmer: I think it'll say true.
Interviewer: Okay, and why do you think that?
Palmer: Because it's printing p, and p is a proposition. Kind of like declaring that this element is in the set.

Haven's initial thought is that the print statement would print 'dog,' but as Palmer described, the proposition is declaring that 'dog' is in setA and the print statement is

determining the validity of that declaration, which is True in this case. As a result of our

discussion, Haven determined that when we print a proposition, we determine the validity

of the proposition which means that the output will be either True or False. This reasoning

was picked up by the other groups in the study as well. The next step in this task was to

determine the output of the propositional statements 'p or r' and 'p and r' as well as

formulate an idea on how the logical operators 'and' and 'or' operate. I present an

exchange that occurred in Group 1 first, followed by excerpts from Groups 3 and 4:

Palmer: I'm thinking for [p 'and' r] it'll print the results of both propositions p and r.
Interviewer: And we saw that p was True and r was False. Am I understanding correctly that maybe the output will be True, False?
Palmer: Yes. That's what I think.
Interviewer: And Judith?
Judith: I have an idea. For the 'and' one, I think it'll say False because True and False it's 'and' so, True, is not equal to False so it's not both it can't be both True and False, but the one for 'or,' it would be True *or* False you know what I'm saying?

Interviewer: So what do you think the output, when you have the 'or' would be?
Judith: I'm not sure actually. I don't know.
Interviewer: Okay, Haven?
Haven: Ooh yeah, this is a tough one, because I was at first, I was thinking kind of like how Palmer was where if it's 'p 'and' r,' they'll show both outputs for p and r, but yeah the 'or' is throwing me off.

There are two instructor moves that I find important to highlight here that helped provide insight into the students' reasoning. First, I did not evaluate the students' answers. Evaluating Palmer's answer would likely have led to a different response by Haven, who revealed that she also thought the output would be the two values, True and False. Second, I asked each student to voice their thoughts on the problem task, and for two of the students I probed with follow-up questions to get a better understanding of their reasoning. Note, as the fourth component of the HLT, I will continue to highlight additional different instructor moves that helped further the instructional agenda throughout the rest of this chapter. As for Group 1's reasoning, Palmer predicted that the output would produce two values, True and False, because those were the individual outputs when we evaluated p and r on their own, respectively. Judith had a different idea and said that the output would be False because True and False (the values of the propositions on their own) are not equal, or not the same. I couldn't determine what exactly Judith meant with her last comment regarding the 'or' operator, but given her tone, my interpretation is that she reasoned that the print statement will produce either True or False, as if at random.

Group 2 took a different approach in that they wanted to see more examples of the print statements using different propositions to get a better idea of what the 'and' and 'or' operators were doing. To assist them, I asked Group 2 to come up with another proposition
that would produce a False output, and Saul came up with 'm = "LA" in setA.' With this new proposition, I wrote out four lines for Group 2 as presented below:

print(p and q)
print(p and r)
print(r and q)
print(r and m)

All of the potential cases are covered here (True 'and' True, True 'and' False, False 'and' True, False 'and' False). This was another intentional instructor move in that I provided a scenario with complete information for the students to easily investigate the outputs and compare print statements. After running this code, Saul said "the 'and' is checking that both sides are True, and if not, then it is False." Leo asked about the situation in which we would have 'or' and I wrote the same bit of code, replacing the 'and' operator with the 'or' operator:

print(p or q)
print(p or r)
print(r or q)
print(r or m)

Eugene described the propositional statements with the 'or' operator as "wanting [the proposition] to be True in order to get True." All three of the students in Group 2 quickly came to the same conclusion that the 'and' operator requires two True premises in order for the output to be True and the 'or' operator only requires one True premise. Seeing all of the different possible combinations of the propositional statements significantly helped Group 2 reason about the logical operators 'and' and 'or' much more quickly than Group 1. In fact, Group 2's success and confidence in their reasoning about the logical operators

'and' and 'or' motivated me to change the problem task for Groups 3 and 4 by presenting all of the possible combinations to the students and asking them to reflect on the outputs, as compared to predicting the output for just the two statements 'p or r' and 'p and r,' which I asked for with Group 1. Being able to determine the output for the simple propositional statements of 'p or r' and 'p and r' was the first step in the students constructing their 'proposition/operator/proposition' model.

Once the groups showed evidence of understanding how the 'and' and 'or' operators functioned, I asked the students to create their own print statements that contained multiple logical operators and multiple propositions. For example, students in Groups 3 and 4 created and evaluated the following two print statements, respectively: (a) print(p and q and r), and (b) print(((r and m) or p) and (q and r)). Figure 5.4 showcases the work that Juliana in Group 3 came up with.

```
8 print(p and q and r)
9 print((p and q) and r)
10 print((True and True) and False)
11 print(True and False)
12 print(False)
False
False
False
False
False
False
```

Figure 5.4: Juliana's Evaluation of a Compound Proposition

Juliana's description of her thought process highlights how she reduced the compound proposition from left to right:

So for 'p and q,' p is True and for q it is also True. So, for 'True and True' it would be- So I have 'True and True and False' and for the first statement 'True and True' it would give me a True statement and I have the False statement left so it would be 'True and False' and since when we have the 'True and False' statement it will give us a False statement since it favors that.

The code shown in Figure 5.4 was written by me after Juliana's description to verify that I understood Juliana's reasoning. Juliana and her partner Delia often used the language of a logical operator "favoring" a certain Boolean value. In the case of the 'and' operator, Juliana and Delia said that the 'and' operator favors False since only one False value will result in a False output. For the 'or' operator, they said that it favors True since only one True value is required to produce a True output. Alonso typed out the step-by-step method for the problem analyzed in Group 4, shown in Figure 5.5, after they had discussed why the final output would be False.

print(((r and m) or p) and (q and r)) (F or T) and F T and F

Figure 5.5: Alonso's Evaluation of a Compound Proposition

For Group 4, 'm' was assigned to be the proposition "LA" in setB' which is False. I asked Julian and Alonso to reflect on the compound proposition shown in Figure 5.5 and they described an approach similar to that of Juliana from the last example, but their focus was on which operator would be the last to be evaluated:

Interviewer: So, since Alonso wrote this one, Julian, can I get your thoughts on the output?

Julian: Yeah so I looked at it as I kind of want to go the simplest route first. So, I'd say the final [operator] would be 'and' because 'r and m' and then 'p,' those are together. And 'q and r' is [together]. So I want to lookbecause it's an 'and' statement, if those two don't match up, then it would be False. And 'q and r,' I looked at it and because it's an 'and' statement, r is False so that statement will be False and because that statement is False, the whole entire thing will be False. Interviewer: Okay, and Alonso, can you rephrase in your own words what Julian was saying? He was saying something with the 'and' and then he referred to this relationship right there [mouse pointing to 'q and r']. Alonso: Yeah so the last operator that happens is the 'and' in the middle between the two parentheses. And since 'q and r' is False, it doesn't matter what's on the other side because of the operator 'and' so it's all False.

The one instructor move to highlight from this exchange was asking Alonso to rephrase in his own words what Julian said. Julian was thinking out loud when he was describing why he thought the answer would be False, so he was jumping to several ideas in his mind at the same time. Rather than asking Julian to explain his reasoning again and risk making Julian feel uncomfortable that I was putting him on the spot, I asked Alonso to rephrase in his own words Julian's reasoning, specifically calling out the 'and' operator and the propositional statement 'q and r.' After Alonso's rephrasing, it is clear that both Julian and Alonso agreed that the focus should be on the second 'and' operator in the compound proposition and determined that the simplest approach would be to show that the propositional statement 'q and r' is False. This then implies that the entire compound proposition would then be False as the 'and' operator requires two True premises. In both cases, the students in Groups 3 and 4 showed a multi-step process of simplification down to a single statement of the form 'proposition/operator/ proposition.' Up to this point, the 'proposition/operator/proposition' model has only been used as a model-of solving for one output, which has been the Boolean value of a given propositional statement. In the general activity, this model will become an important component for reasoning about a more complicated mathematical task. The students in all of the groups eventually came to the same understanding and method of reducing down to simpler propositional statements and I consider the students' work on compound propositional statements to still be situational activity, as they were working with the same defined propositions, 'p,' 'q,' and 'r' with the two sets, 'setA,' and 'setB.'

Before moving on to the referential activity, I want to spend more time discussing the actual learning trajectory of Group 1, as this task was a point of divergence for Group 1 compared to the other three groups. We spent all our time in the following session (Session 2) to discuss how the logical operators were functioning, as it was still unclear for the students why the propositional statements were producing various outputs. Specifically, Palmer said "If I remember correctly, I think the 'and' is making sure both of [the propositions] have the same result, whereas the 'or,' well, it just kind of chooses like one of them randomly." Palmer's reasoning about the 'and' operator is in line with how Judith was initially reasoning in that as long as the two propositional values are the same, then the 'and' operator would produce a True output. This means that for Palmer, 'False and False' would produce a True output. Palmer's reasoning is also in line with my interpretation of how Judith initially described the 'or' operator as well in that Python was producing a True or False output at random when evaluating a propositional statement with the 'or' operator. At the time, during Session 2, I thought that we could resolve this reasoning by looking at compound propositional statements. After analyzing multiple compound propositions, it seemed that Group 1 had a good understanding of the logical operators as they were able to solve the following compound propositional statements correctly: 1) print(p and q or r), 2) print((r and q) or p), and 3) print(r and (q or p)). Unfortunately, it wasn't until the end of Session 2 that we cleared up that the 'and' operator requires two True values, not just two of the same values. This clarification occurred as we moved onto the next task which was designed to support students' referential mathematical activity.

Referential Activity

With the next task for all the groups, the goal was to support students in their reasoning about any proposition, not just defined ones like 'p,' 'q,' and 'r' in the previous example. The task presented to Group 1 is shown in Figure 5.6.

Provide an example of a proposition with an unknown truth value. Let's assume that *s* and *t* are two propositions with unknown truth values. Interpret the following propositional statements: (s and t) and (s or t)

Figure 5.6: Group 1 General Proposition Task

Even after we discussed some potential examples of what a proposition with an unknown truth value could be, Palmer said that, "For me when I see that I wonder what *s* and *t* mean, or what they are? They're just kind of random variables." Judith and Haven also seemed to have some reservations about what exactly a general proposition is, so I decided to try and provide a different example using two sides of a coin that could relate back to the propositions:

Interviewer: How about we do something else, like a coin flip. If I say I have two coins, *s* and *t*. And I flip them, it's either heads or tails right?

Associating to True or False. So if I flip *s* and - let's do this. Let's set heads equal to True and Tails is equal to False...What are the different possible combinations and different outcomes for these two propositions *s* and *t*? Now we're thinking about it in terms of coins, but we can still kind of relate it to the same idea. **Judith**: I'm not sure about the possible combinations but does it have something to do with probability? **Interviewer**: Can you say a little bit more about that? **Judith**: If there were one coin, the possibility of getting heads, being True, would be 50%? If it was 'print *s*' it'd be like 50, but when you introduce a second variable *t*, the probability goes down to I don't know 25% or something? **Interviewer**: Okay. **Judith**: I don't remember statistics, but yeah something like that.

I realized that I could leverage Judith's response by relating the only scenario (out of four)

of 'True and True' to the 25% that she mentioned for the propositional statement 's and t,'

but I needed to first make sure that everyone in Group 1 was on the same page about the

possible outcomes of this propositional statement. Palmer reveals that he still was unclear

on the functionality of the 'and' operator:

Interviewer: What situations would you get True and in what situations, would you get false?

Palmer: True, would be if both coin flips are the same and then False would be if one is heads and one is tails.

Interviewer: Okay. Let me go back really quickly. Because I think I want a little bit of clarification here. Let's do 'm' is equal to the proposition 'whale in setA.' Okay, so we know this proposition is False, because 'whale' is not in setA right? Now, one thing that you said was that it would produce a True value for the same, but I want to highlight that if we have 'r and m' that you actually get a False output. So, even though *r* and *m* are both False, you still get a False output.

Palmer: Oh, okay.

Interviewer: Haven or Judith, can you explain why you're getting False here?

Haven: Yeah I kind of noticed it right after he said it too. Because even if they're both the same, if they both were tails, they're both going to be False and the whole thing would be False. Even though they're the same, that wouldn't make it True.

Interviewer: So let's think about the situation where we would get a True value for these two coins.

Palmer: Oh, so both land heads.

Interviewer: Okay. And so Judith, going back to what you said about the 25%, I think you're right there, in terms of probability. Can you kind of relate the 25% back to this situation here in which you would get a True output?

Judith: Yeah because to get both heads is one of the four options of getting heads and tails, tails and heads, tails and tails, and heads and heads, with *s* and *t* respectively.

I typed out the possible combinations of coin flips that Judith described and asked Palmer

and Haven to reflect on the outcomes for the four statements:

Palmer: Yeah okay so now I see that they would all- or the first three would all be False and only the last one would be heads, or only the last one would be True.

Interviewer: Haven, can you rephrase in your own words, what Palmer was saying there?

Haven: Yeah I was thinking the same thing that only heads and heads would be True because heads is True and then even though heads and tails has that True value, it's still False because tails is False. And all of them have tails, the rest of them do, so they would be False.

This discussion on relating the propositions to coin flips occurred at the end of the second

session for Group 1. Given that Group 1 had some difficulty with the task as it was

presented, I decided to change the task slightly for the other groups in my study. I modified

the task by assuming that s is a proposition with a known Boolean value of False. The

modified task is presented in Figure 5.7.

Provide an example of a proposition with an unknown truth value. Let's assume that *s* is a proposition with the Boolean value of False, and *t* is a proposition with an unknown Boolean value. Interpret the following propositional statements:

(s and t) and (s or t)

Figure 5.7: General Proposition Task for Groups 2, 3, 4

and you have to take 'False or False.'

My reason for assigning a False Boolean value to s was to first ask about the propositional

statement 's and t' with the hope that the students would realize that it does not matter

what the value of t is. From there, we could reason about the other propositional statement.

There were some minor clarifications that needed to be made across Groups 2, 3, and 4

with what was meant by "unknown proposition," but for the most part, Delia's

contributions from Group 3 serve as good representations of the mathematical activity

across the three groups that solved this task. I present Delia's contributions below followed

by Julian and Alonso's contributions from Group 4 when asked about 's and t' and 's or t':

Interviewer: What are the possible outputs for 's and t'? Delia: Yeah so I guess it would just both be False so it would just come out as False. Interviewer: Both be False. So do you mean- is that what you mean when you say it'll both be False? [typed out 'False 'and' True,' and 'False 'and' False'] Delia: Yeah. Interviewer: What would we have for 's or t'? ... Delia: Well we would just be taking both options that could be presented when you run a proposition statement that has an unknown value. You have to take all of the options that could possibly happen. So, *s* is a definite, so we always know it will be False. And while it was true- [*t*] can be True or False, so you have to take both options. You have to take 'False or True' The ellipses between Delia's contributions represents a brief discussion with Juliana on her reasoning of the propositional statement 's or t.' Juliana's reasoning was not clear in terms of what the two cases would be, which is why I did not include her contribution in the above excerpt. As mentioned, students in Groups 2 and 4 provided similar arguments in their reasoning about the propositional statements, as shown in Julian's reasoning about 's and t':

So 's and t.' Since one is False then that means the statement will probably be False already, or- yeah because s has a value of being False and t is unknown, but because it's an 'and' statement it would be False because you would need both of them to be True to be True.

Alonso agreed with this reasoning and followed Julian with his own reasoning about the statement 's or t,' "That one can be either True or False because s is False, but if t is True then the entire thing will be True, but if t is False then the entire thing will be False." Similar to Julian's reasoning, Alonso determined that the value of the statement 's or t' depends on the value of t because we already knew that s was False.

At this point in the learning trajectory, the students have elaborated on their modelof understanding mathematical logic (or constructed a sub-model). That is, they have determined that the value of a propositional statement is dependent on the values of the propositions at the time of evaluation, which can change as the values of the propositions can shift between True and False. This model is a broader, more holistic perspective on evaluating propositional statements compared to the previous situational activity in which the propositional statements were fixed given the defined propositions in the form 'proposition/operator/proposition.' In the next activity, the students use their developed model to solve a different mathematical task in the context of set theory. Solving a task in

168

a different context, or a context-free environment represents the next step in the evolution of mathematical activity, general activity.

General Activity

The purpose of this section is to present two tasks that are relevant to the students' mathematical activity involving the logical operators 'and' and 'or.' Both tasks involve the use of a For Loop, which was introduced to the students before the task presented in Figure 5.8. More information on the tasks used to introduce the For Loop to students can be found in Chapter 3.

Each group in the study was presented with the following task, but I will highlight the work of Groups 1 and 4 as each group had different initial experiences with this task, with Group1 having slightly more difficulty than Group 4. Groups 2 and 3 had similar initial experiences as 1 and 4, respectively.

> What do you predict will be the output of the following code? A = {"s", "e", "t", "t", "h", "e", "o", "r", "y"} D = set() for x in A: if ((x == "e") or (x == "o")): D.add(x) print() print(D)

Figure 5.8: Task Utilizing the Logical Operator 'or' in a For Loop

Following the instructional theory of PRIMM, I first asked students to predict what they thought the output would be, or to provide any thoughts that they had on certain lines in

the code. Palmer in Group 1 said the following about the if statement in the fourth line of

the code:

Well the code is probably going to recognize those as important results or something. Those are the elements where it's going to notice that- we're highlighting those [elements] and we're going to want it to do something when it reaches an 'e' or an 'o.'

Judith and Haven did not have anything to add, so I ran the code and asked the students to

reflect on the output of '{e, o}.' Again, Palmer was the first to respond, but I used Palmer's

ideas to inquire into Judith's reasoning:

Palmer: When I initially saw this line of code, I was looking at every single individual line trying to figure out what everything does. Then I saw between three and five [the if statement], it's kind of like giving it steps. So, the first one, we already know it's going to like list out all of the [elements] in the set and then the second one is kind of like highlighting specifically the 'e' and 'o' and then when you told me to like to analyze it, you kind of just wanted me to look at that one specifically, but I also noticed that [line] five is saying 'D add x.' I was thinking if it sees an 'e' or an 'o' in set A, it's going to add the 'e' and the 'o' to the empty set. That's kind of what I was thinking.

Interviewer: Great. Judith, what do you think about what Palmer just said? **Judith**: Yeah I think that makes a lot of sense.

Interviewer: So, Judith, what happens when let's say x randomly grabs 'r.' Let's say that's the first one it does. Okay, so, 'for r in A.' Why isn't 'r' in this final set D?

Judith: Because 'r' is not what 'x' is defined as, 'x' is defined as either 'e' or 'o.' Since 'r' is not either 'e' or 'o,' it isn't printed.

Given that Palmer and Judith are both seeing this material for the first time, I did not want

to harp on the terminology that they used too much. For example, Palmer says that the For

Loop is going to "list out all of the [elements]," and it is unclear whether Palmer thinks

that every element in set A will be printed, or he is using "list" as another term for iterating

through all of the elements in A. Similarly, Judith responds to my question saying that

"Since 'r' is not either 'e' or 'o,' it isn't printed." I am interpreting Judith's use of the word

"print" to mean printed in the set D, or not added to the set D, and thus does not appear when set D is printed. In either case, the important aspect of both Palmer and Judith's responses is the idea that only certain elements from set A are being added, specifically the elements 'e' and 'o.'

What was not clear to me at the time of this activity was if the students in Group 1 could describe line by line how the For Loop was iterating and evaluating each element in A. That is, what happens next after the code has identified either the element 'e' or the element 'o.' As an important component of any HLT, I wanted to test Group 1's reasoning, so I came up with another task by asking the students to consider the following code:

```
comp = "computer"
D = set()
```

```
for letters in comp:

print(letters)

if ((letters == "e") or (letters == "o")):

D.add(letters)

print(D)
```

print(D)

Again I ran the code for the students to see the output, which is shown below:

c o {'o'} m p u t e {'o', 'e'} r {'o', 'e'}

Haven recognized that all of the letters in computer are printed, but was not sure why the

sets were printed after 'o' and 'e':

Haven: So we still have computer spelled out from the For Loop, but now in between...I don't know how [Python] chose it because after 'o' you have the 'o' in brackets because that's specified in the 'if letters 'o.' So, yeah I'm not sure why they're in those spots, because then as it goes down, you have the [set] 'o' and 'e.'

Interviewer: Mm hmm. Palmer or Judith? What are you thinking? **Judith**: Well what I'm thinking is since it's spelling out computer, maybe it's going through 'c,' 'o,' and then it recognizes that there's an 'o' and then it keeps going, and then when the 'e' comes up it said oh, there was an 'e' and then now there's an- or there was an 'o' and now there's an 'e.' It's like going through the letters of 'computer' to find 'o' and 'e' and showing it when they find the letters.

Neither student referred to line 7, 'D.add(letters).' I wanted the students to see that when

the For Loop is iterating through each element in the string 'comp,' that when the elements

'e' or 'o' are passed through, they satisfy the If Statement and are then added to D, which

is then printed. After asking again about the If Statement, Haven refers to the logical

operator 'or':

I know with the 'or' and 'and' operators, it's like 'or' is the one that if it's True then it's all True or something, like if one of them is. So, that is true, there is an 'o' in computer, so it kind of picks that up and then prints it

Again, the terminology used was not precise, but the important aspect of Haven's thought process is that the If Statement is satisfied because one of the propositions is true when the element 'o' is passed through. Haven utilized her constructed 'proposition/operator/ proposition' model to help her come to the correct conclusion. We spent a considerable amount of time (about 20 minutes) on this task to understand how the For Loop was functioning, but I still was not entirely convinced that all the participants in Group 1 felt

comfortable explaining the processes of a For Loop. In an effort to keep momentum going, I transitioned to the next task knowing that I would need to provide additional support in solving the next task asking about set intersection.

Before getting to the next task, I present some of the work done by the participants from Group 4 on this task to highlight the difference in the language used by Alonso and Julian and their detailed descriptions of how the For Loop functioned. I highlight this difference as an important result from the perspective of computer science education in that with Groups 3 and 4, I switched to an IDE on my own computer rather than using Google Colab. I made this switch because Google Colab does not have a Debug feature that enables the user to step through each line of the code (typically as a method of troubleshooting, or "debugging" code that is producing an error message). The Debug feature was useful for Groups 3 and 4 in showcasing how a For Loop steps through each element in the iterable object. With Google Colab, the students in Groups 1 and 2 were not able to see how the For Loop would randomly select an element from the set (since sets are not ordered) and step through the code with that element.

In the following exchange between Julian, Alonso and I, I ask both participants to share their reasoning on what they think the output will be for the code presented in Figure 5.8. Prior to this exchange however, Julian was thinking out loud and offered a couple of different ideas but ultimately reasoned that the final output would be a set containing three elements, three 'e's and one 'o.' In an effort to understand how Julian was thinking, I asked him to repeat his thought process in how he arrived at his final answer:

173

Interviewer: And can you walk me through just one more time how you got to that output?

Julian: Yeah. So, I was thinking before that it would just add any of the set letters, but because the action of adding the element into 'D' only works when 'x' equals 'e'- or when the element equals 'e' or when it equals 'o.' So, every single time it hits 'e' then it adds the element and when it hits 'o' it adds the element into 'D.' And because there are only two 'e's and one 'o' then- I would assume that it would keep doing that over and over again until it hits 'e' and 'o' and then it would just add those. And then going to the next 'e' and then add that again. So I assume it would create the equal amount of 'e's and 'o's in 'D' as in 'A.'

Interviewer: Okay, great. Thank you very much. Alonso, what are your thoughts?

Alonso: I think something slightly different. So I think the output would be an empty line, and then it would be the brackets with either 'e' or 'o' inside of it. Then another empty line, and then it would be after that empty line would be the brackets with both 'e' and 'o' as part of that set. But it would only be the two elements, I think. And the reason for that is in previous examples, when we did 'x in A,' it stepped through all the elements, but it didn't do any repeats. So, what this code does is for the For Loop, it steps through all the element is 'e' or if the element is 'o,' then add that element to set 'D.' So only 'e's and 'o' ance.

Interviewer: Okay great. Thank you very much for that explanation as well. So, let's run this and see what happens.

[{'e', 'o'}]

So both of you were very, very close and I think both of you had similar reasoning-

Alonso: Oh that's why. That's what's wrong

Interviewer: Julian, can you reflect on this output here?

Julian: Yeah so it seems Alonso was kind of right with it only printing 'o' and 'e' and no repeats. I kind of just didn't realize that. Because it's the elements, it would not do the repeats. So, it would only be the 'o' and 'e' the one time so it would print those.

Both Alonso and Julian were able to step through the entire process, with the important

aspect of their contributions being that the if statement is satisfied with either the elements

'e' or 'o' and then moves on to the next step which is to add these elements to set 'D.' This

understanding is supported by the model that they constructed in their situational and

referential mathematical activity in interpreting propositional statements of the form 'proposition/operator/proposition.' Neither student was entirely correct, but after seeing the output they both understood why the output was just the one set containing the two elements 'e' and 'o.'

With the next task in the sequence, my intention was that students would use what they have learned of utilizing a For Loop to iterate through elements in a set, and use an If Statement to filter for certain elements that they want. Specifically, the goal of this task was to find the intersection of the following three sets, A, B and C:

A = {8, "apples", "chocolate", "berries", "corn", "juice", 13, "strawberries",
6, "avocados", "beets", "chips"}
B = {1, "cheetos", "jalapeno", "onions", "cilantro", 2, "limes", "chips", 6,
"cherries", 9, "corndogs"}
C = {8, "biscuits", 6, "cheese", "soda", "water", "bananas", "beets",
"watermelon", 7, "kiwis", "chips"}

Each group in the study were given the same three sets and eventually came to some version of the following code:

D = set() for x in A: if ((x in B) and (x in C)): D.add(x)

print(D)

In Chapter 4 I highlight how students from Group 2 came to this code which resulted in the output of '{chips, 6}.' At this point in the study for all of the groups I first mentioned the notion of a subset. I used the example of '{chips, 6}' to show that every element in

'{chips, 6}' exists in A, B, and C which makes '{chips, 6}' a subset of A, a subset of B, and a subset of C. There will be more on the idea of subsets in the next section.

Like Group 2, Groups 3 and 4 also arrived at some version of the above code on their own given their understanding of how to filter through a set for select elements. For Group 1, I provided additional support by writing the basic structure of the code (starting with a For Loop and If Statement) and asked them to fill in the blanks given the goal of finding the set intersection. With Group 1 we spent most of our time investigating the general functionality of a For Loop in sessions 3 and 4, which is why I wanted to provide them with extra support in our last session and finish the TE by solving the task of finding the set intersection. As mentioned, Group 1's TE stopped short of the designed instructional sequence. Their actual learning trajectory was composed mostly of understanding how a For Loop works in session 3 and 4. Unfortunately for Group 1, they helped me figure out what didn't work so well with the tasks as they were initially designed. They also helped me realize that Google Colab was not the best choice. However, as far as understanding For Loops, I made sure that we spent the time understanding how they worked and how they could be used to make longer code more efficient. To this degree the students in Group 1 learned a great deal, even if what they learned was not exactly what was initially envisioned with the design of this study.

Going back to the above code, I want to highlight that the students in all of the Groups were able to utilize the 'and' logical operator to solve a mathematical task in the context of set theory (line three). That is, the structure of 'proposition/operator/proposition' appears as a way to filter for certain elements across three different sets. In the beginning of the designed instructional sequence, the students were asked to evaluate simple compound propositions such as 'True or False' as well as compound propositions such as '(True and False) or True.' In reasoning about the compound propositions, the students reduced the longer compound proposition down to a shorter form of 'proposition/operator/proposition.' This constituted the students' Situational activity. The students were then asked to reason about propositional statements in which they were not directly given the Boolean value of the propositions, such as 's and t.' I consider their work on this task to be referential as the form was short, but the content itself was slightly more abstract and removed from context. Lastly, using their initially constructed model-of ways of reasoning about logical operators in the form of 'proposition/operator/proposition,' the students were able to solve a more sophisticated mathematical problem asking them to find the intersection of three sets. Thus, the students were able to utilize their model-of ways of reasoning about logical operators as a model-for reasoning about sets and set relationships. In the next section I start with the set intersection task as the situational activity for reasoning about set relationships and how I attempted to support the students develop models-of reasoning about set intersection as models-for reasoning about subsets.

Set Operations and Properties of Subsets

In this section I present the sequence of tasks that were designed to support students in their reasoning about set operations, and ultimately reason about what it means for one set to be a subset of another set. This section will be briefer than the previous for three main reasons. The first is that most of the student work related to the core tasks for this instructional sequence has already been presented in Chapter 4. The second is that work related to these tasks involved introducing the idea of the For Loop to the students and these tasks are presented in Chapter 3. Third, the instructional sequence presented to the students, which was meant to foster the developmental progression from situational activity to formal activity, did not support the model development as intended. That is, the situational activity meant to support students in their reasoning about set intersection did not lead to mathematical activity that was entirely coherent with the referential and general activity related to subsets. Specifically, the goal with the situational activity was to generate a discussion about how the intersection of sets is a subset of the sets that were used to find the intersection. In the original conception of this task I thought that this would be enough to serve as the situational activity. However, in retrospect, a task focused on understanding the difference between sets as subsets and sets as elements would have likely been much more beneficial for the students to have an opportunity to build a modelof way of reasoning about sets. As such, a model-of reasoning did not necessarily emerge from their mathematical activity, but I highlight their work on these tasks as a foundation for future work that may utilize these tasks and build more coherent instructional tasks. The first task in the sequence was the set intersection task, which was intended to provide an opportunity for students' situational mathematical activity.

Situational Activity

With this task, the students were given the three sets:

A = {8, "apples", "chocolate", "berries", "corn", "juice", 13, "strawberries",
6, "avocados", "beets", "chips"}
B = {1, "cheetos", "jalapeno", "onions", "cilantro", 2, "limes", "chips", 6, "cherries", 9, "corndogs"}

C = {8, "biscuits", 6, "cheese", "soda", "water", "bananas", "beets", "watermelon", 7, "kiwis", "chips"}

The students were asked to find the elements that exist in all three sets. The students were also asked to find the union and other set relationships (e.g., A union B not C). The problem context for this task was that these sets represent items that one may find at a grocery store. The integers in the sets represented cash back that they might receive when checking out. Some of the student work for this problem has already been highlighted in Chapter 4, but the focus of this section is specifically on how the students were reasoning about set operations. This was the last task presented to Group 1, so I present some of their work below. With Group 1, I drew a Venn diagram to represent the three sets A, B, and C which is shown in Figure 5.9.



Figure 5.9: Venn Diagram Used with Group 1

Once Group 1 constructed the code that would find the intersection of the three sets, I asked the students what we could do to find the elements in the green shaded area labeled 'G.' Judith first idea was to find the intersection of A and B, "I guess, could you do 'for x in A, if x in B...and then G dot add x?" As Judith was talking, I was writing the code which is below:

G = set()for x in A: if x in B: G.add(x)

The output produced the same set as the intersection of all three sets, '{chips, 6}.' I asked

the students if they could identify what was wrong with this output, which led to the

following exchange:

Judith: Okay I think I have an idea. Okay, because it doesn't include the blue area- so we're saying it includes those in A and B, but not those in A, B, and 'C.' So, wouldn't it be like... nothing? You know? Interviewer: Palmer, what are your thoughts on that? Can you rephrase what Judith was saying? Palmer: Umm... Interviewer: Or do you have any questions about-Palmer: Yeah I think we're just kind of continuing on what we did last week. Yeah I guess the Green is just what only A and B share and not what all three of them share like we were trying to find last week. **Interviewer**: Right, and so this first output for 'print F' [{chips, 6}]. That set contains all of the elements that exist in all three sets. Palmer: Yeah. **Interviewer**: 'Print G' is referring to- let's do this. Okay, so that area is G, and this area is F [labeling the green and blue shaded regions, respectively]. Haven, what are your thoughts on the two outputs and maybe if you want to refer back to what Judith was saying. Haven: Yeah when she mentioned how it should be nothing I was like 'Oh wait?' Because I don't know I'm kind of still figuring it out and thinking about it. So, yeah I don't know really.

Interviewer: Okay. What is it that you're curious about in terms of when she said it could be nothing? Haven: I don't know because when we were looking at 'print G,' it is supposed to draw from A and B and then 'F' is supposed to draw from A, B, and C. So I'm kind of curious when Python is running through it, I'm kind of curious of where it's going wrong, like when it's running that. Interviewer: Mm hmm. Well let's take a look at the code. What can I add in here [pointing to the If Statement]? Judith, do you have what we might be able to add here to tell Python that we only want the elements in A and B and not to include any elements in 'C?' Judith: We could do 'x not in C?' Maybe? **Interviewer**: Yeah so that's gonna go right here... Judith: Oh and 'and!' 'and' will go there Interviewer: Let's try that. So, let's run this. We end up with the empty set here. So, Palmer, can I get your reflections on why we got the empty set? Palmer: Yeah because A and B, aside from the 'chips' and 'six,' they don't have anything else in common with each other. So, there's nothing in the "the G zone."

From the excerpt, both Judith and Palmer noticed that the original code produced the

intersection of A and B, but they didn't just want the intersection, they wanted the

intersection not including any of the elements that also exist in C. Since the intersection of

A and B was equal to the intersection of all three sets, they figured that must mean that

there are no elements that exist in A and B and not in 'C.' With this understanding, Group

1 also found the elements that exist in A and C and not B by modifying their code slightly:

H = set()

for x in A: if ((x in C) and (x not in B): H.add(x)

The output produced a set containing two elements, '{8, beets}.' Judith commented that this code is working the way it is supposed to because "8' and 'beets' are repeated in A and 'C.' But 'chips' and '6' are not mentioned because 'chips' and '6' are repeated in all three. We only want the ones that are repeated in those two [A and C]." The only other group in the study that constructed different For Loops to find various intersections of the three sets was Group 4, who also found the set of elements shared between A and C and not in B. All of the groups did find the union of the three sets, with most of the groups opting to either add all of the elements from B, and C to A or creating a new set and adding the elements from all three sets to the new set. Group 3 took the latter approach, and their code is below:

D = set()

for dog in A: D.add(dog)

for dog in B: D.add(dog)

for dog in C: D.add(dog)

The reader may notice that the variable used in the For Loop in the above code is 'dog' which came out of a discussion about what 'x' represented in another For Loop that we were looking at. In Group 3 we discussed how the variable in the For Loop can be any letter or string that we want. Juliana asked if this means it can be something like 'dog' and I ran the code using 'dog' as the variable to show that the code still works. This was another instructional move that worked well in this study in terms of using the words or vocabulary that the students were using to relate to their ideas. While their terminology was not always technically correct, I decided not to correct them in the moment and return

to that idea if it needed to be addressed, usually after the students had a better idea of the mathematical concept.

As mentioned in the previous section on logical operators, once the students found the intersection of all three sets, I used the intersection to talk about what it means for one set to be a subset of another set. We discussed how the two elements, 'chips,' and '6' are elements in all three sets, A, B, and C. In the initial task design, the work that the students put towards finding this result was supposed to be the situational mathematical activity. However, the work going into finding the intersection had nothing to do with subsets themselves. A better situational activity would have been providing the students with sets and subsets and asking the students to find different relationships between the sets using Python. At the time of conducting the fourth session with Group 2, I realized that the set intersection activity did not adequately generate the need or motivation to investigate the idea of subsets further, so I provided the students with an additional block of code to show the difference between a set as a subset and a set as an element of another set:

A = frozenset([1, 2, 3])
B =
$$\{1, 2, 3, A\}$$

print(len(B))

The output for the above code is '4' as there are four elements in B: 1, 2, 3, and A. What is interesting in this case is that A is not only a subset, but also an element of B. Investigating this code was beneficial in discussing the idea of sets as subsets and sets as elements, but it did not challenge the students in the same way a traditional situational activity from the RME perspective would have, it was mostly just direct instruction in telling them what the

definition of a subset was. With the next task, the students in Groups 2, 3 and 4 were given a task in the context of a number theory problem, which was intended to serve as an opportunity for the students' referential mathematical activity.

Referential Activity

I am presenting this task and the following task as potential referential and general mathematical activity with the intention of developing tasks in the future that support situational mathematical activity. The goal would be to create tasks that provide more of an opportunity for a coherent progression and opportunity for model development. For this task, the students were no longer in the same problem context of the grocery store and working with grocery store items. Instead, the students were asked to reason about a number theory problem which is presented in Figure 5.10 below.

Let's consider the integers from 1 to 1000. In Python we can call out these integers by using the 'range()' function.

range(1, 1001)

Is the set of integers divisible by 21, a subset of the set of integers divisible by 3? Is the set of integers divisible by 21, a subset of the set of integers divisible by 7? How do you know? How can you use Python to help answer this question?

Figure 5.10: Number Theory Problem

For more details on how Groups 2 and 4 arrived at their final code, refer to Chapter 4.

Group 3's work was not highlighted in Chapter 4 because their approach was essentially

the same as Leo's from Group 2. Below are the codes written by each group to solve the

number theory task presented in Figure 5.10. Group 2's code was the following:

D = set()

for x in A: if ((x in B) and (x in C)): D.add(x) print(len(D)) print(len(A))

A represented the set containing the integers between one and 1000 divisible by 21, B represented the set containing the integers between one and 1000 divisible by 3, and C represented the set containing the integers between one and 1000 divisible by 7. This approach used a For Loop to find the set intersection between B and C, and finds the cardinality of the two sets A and D. Group 2 determined that since the two sets have the same number of elements, A is a subset of D (and is in fact equal to D). It is also important to remind the reader that the original language used in this task for Group 2 was "Is the set of integers divisible by 21, a subset of the set of integers divisible by 3 and 7? Having the "and" in between 3 and 7 likely contributed to the difference between Group 2's code and Groups 3 and 4. Group 3's code is below:

```
for x in A:
    if (x in B):
        print("A is a subset of B")
    else:
        print("A is not a subset of B")
```

for x in A:
 if (x in C):
 print("A is a subset of C")
 else:
 print("A is not a subset of C")

Group 3 utilized a For Loop to iterate through the elements in A, and used an If-Else structure to verify that each element in A was an element in the sets B and C. Group 4's approach is similar to Group 2's as they relied on a set relationship (equality) to solve the task:

$$A = set()$$

for x in D:
if (x in E):
A.add(x)
print(A == D)
$$B = set()$$

for x in D:
if (x in F):
B.add(x)

print(B == D)

For Group 4, D represented the set containing the integers between one and 1000 divisible by 21, E represented the set containing the integers between one and 1000 divisible by 3, and F represented the set containing the integers between one and 1000 divisible by 7. The output for this code produced a value of True. Their method included adding all of the elements that existed in D and E to a new set A. They determined that since A and D are equal, then all of the elements in A are also in E, which means that A is indeed a subset of E. The same case was true for the relationship between sets B and D.

With each group in the study, the students used a For Loop which enabled them to filter for certain elements. Group 2's actual mathematical activity was most closely aligned

with the original design of my study in that they used the idea of set intersection inside of the For Loop to determine that the set of integers divisible by 21 is a subset of the set of integers divisible by 3 and 7. As alluded to, the connection between the situational and referential tasks is lacking coherence in terms of set intersection supporting the students' development of a model that can serve as a model-of way of reasoning that could be used as a model-for way of reasoning in a later task about subsets. Future work on these tasks would entail a different direction for the situational activity to think about what it means to be a subset. In fact, the code on the two sets, A and B where $A=\{1, 2, 3\}$ and $B=\{1, 2, 3, A\}$ provided an opportunity for the students to engage in discussions about the difference between a set as a subset and a set as an element. This scenario will likely be developed as a starting point for future work on this task. For the general activity, I wanted the students to take a step back from Python and consider the same question about whether the set of integers divisible by 21 is a subset of the sets of integers divisible by three and seven, but for all integers, not just those between one and 1000.

General Activity

Removing the constraints of Python is what I considered to be the next step in the students' mathematical activity. This section is divided into two smaller subsections with each subsection focusing on the groups, 2 and 4 to showcase the students' thinking without requiring Python to be used as a tool to solve the problem. Only Groups 2 and 4 will be the focus of this section as Group 3 only had a couple of minutes to think about this task. Both students in Group 3 said that they think it is still true for all integers, that the set of integers

divisible by 21 is a subset of the set of integers divisible by three and a subset of the set of

integers divisible by seven.

Group 2. Group 2 was only given about 10 minutes to think about this question as

it was the last task with about 15 minutes left in the final session. Below is an exchange

presenting each of the three students' ideas:

Interviewer: Now we only have about 15 minutes left, but I'd like to think about what you all think if we strip away Python. Okay, let's consider all of the integers, not just the integers from one to 1000. Is the set of integers divisible by 21 a subset of the sets of integers divisible by three and seven, how do you know?

Leo: I don't know if it's true, but I see three and seven have a similarity with 21. So I would say that it's true. So it would be like- but you said we couldn't use what we figured out on Python... So it would just be, I guess mathematically what we thought?

Interviewer: Yeah, I would like you to expand a little bit more. It might just be your initial thoughts, so I want to give you some time if you need a little bit more time. But can you say a little bit more about what you're thinking about three and seven?

Leo: Well the numbers divisible by 21- Since 21 is divisible by three and seven, then any number that's divisible by 21, you can [divide it by] three and seven. You can put 21 is equal to three times seven. And if you do 21 squared, that would be seven times three squared. I think. No, that wouldn't work. 21 times seven...

Interviewer: 21 squared is 21 times 21 right.

Leo: Yeah.

Interviewer: So we can sort of write this as three times seven [(21)(21) = (21)(3)(7)]. We know that 21 is equal to three times seven. 21 squared is equal to 21 times three times seven.

Leo: So there would be a pattern, with after each- With one power higher then you'd add an extra set of that.

Interviewer: Mm hmm... Okay, so we have this idea that 21 can be written as three times seven. How does that relate to this idea of a subset? Eugene and Saul, what are your thoughts? Leo offered some insight, but I want to hear how you two are thinking about it.

Eugene: Yeah I mean I'd say- I mean, I wouldn't have known if we didn't go through this process. Just my math skills aren't on that level. But if we're assuming that from one to 1000 you know it's true, I guess we're using all integers now so including negative values, it should still be true. Because if it's a negative value, it's just negative for both- all three subsets you know.

Using the same integer. So, that's what I'm thinking. It should be the same. It should be the same, yeah. **Interviewer**: Saul, what are you thinking? **Saul**: Yeah kind of just the same thing. Since the factors of 21 are three and seven, then you're going to have similar factors for 42 and everything. For set A. There's going to be a three and seven there, so that means as you go up there can be a six and nine. I guess.

Looking back at this excerpt, Leo, Eugene and Saul all had great ideas about the properties of the elements in set A, the set of integers divisible by 21. Leo brought up that there is a relationship between 21 and the numbers three and seven, Eugene figured that if it's true for one to 1000 then including the negative numbers shouldn't be an issue, and Saul even mentioned the factors of the elements in A. Instead of following up on their ideas in a more informal way by discussing the factors further, I decided to try and do a rough sketch of the proof to show that any arbitrary element in A will also exist in B and C. I asked the students to consider any element in A and I showed them that this element could be written at 21 times some integer 'm.' We can then write this as 3 times 7 times m which ultimately shows that this element exists in both B and C. We only had about five minutes left in the session, so we didn't get a chance to spend too much time on this rough sketch of the proof. Looking back, Leo and Saul were close enough to determining the answer on their own and the last five minutes of the session would have been more productive if I asked more questions and inquired further into Saul's reasoning about the factors instead of telling them the answer. Even though we only had a few minutes to spare, Group 2 made it all the way through to the end of the designed teaching experiment.

As the reader may have already noticed, the goal of this task was to identify that the set of integers divisible by 21 is equal to the intersection of the set of integers divisible by

three and the set of integers divisible by seven. My thinking in the design of this task was that the students might first find the intersection of the two sets of integers divisible by three and seven, then we could explore the relationship between the intersection and the set of integers divisible by 21.

Group 4. Group 4 made it the furthest in the TE compared to the other three groups. In fact, the students in Group 4 were one whole session ahead of the other groups by the time we got to this task. Below I highlight their initial thoughts on this task, which occurred during the last five minutes of the fourth session.

In response to the task asking about all of the integers, the following is what Julian and Alonso had to say:

Julian: I'd say yes, I mean obviously the integers follow certain sets of rules because it's, you know, going up by ones. So I'd necessarily say yes. If it's you know infinitely going, then obviously- I think... Yeah I'd say it's true because since we've already done it inside the limited type of range, we can obviously say it is true for a higher type or just infinite in general. Interviewer: Okay, and Alonso, what are your thoughts? Alonso: Yeah I think this holds true as well. So, three and seven go into 21 evenly and as long as those numbers go into the other numbers I think it's going to be true for all integers.

Both Alonso and Julian say this statement is true, and the above excerpt is where we finished Session 4. Julian uses the argument that since it is true for the range from one to 1000, then it must be true for all integers and Alonso uses the factors argument as justification that this statement is true. In the fifth session I did not come back to this exact problem and instead asked the students what their thoughts were on how to prove that the set of integers divisible by 21 (D) is equal to the intersection of the set of integers divisible by 7 (G).

We started the session with a couple of warm-up tasks on set intersection and the definition of a subset. One of the questions asked if the set of integers divisible by 3 (E) is a subset of the set of integers divisible by 7 (F) and both students reasoned that E is not a subset of F. As Julian explains:

Okay okay. Yeah so basically we're just disproving the fact that all integersor all elements in E are included in the integers in F- or the elements in F. If we disprove that three itself is not divisible by the [number] that is getting divided in F- So, we just find that one integer, you know, three is already not included, so that means E itself cannot be a subset because not all of its elements are included in F.

Julian uses a counterexample argument to reason that E is not a subset of F. That is, since three is not divisible by seven, then three cannot be an element of F, which automatically means that E is not a subset of F as not all of its elements are elements in F. The purpose of this warmup task was to verify with both students the meaning of a subset.

The warmup tasks only took about 10 minutes, and we spent the rest of the session discussing the proof showing that D is equal to G. The goal of the fifth session with Group 4 was to show that two sets are equal by proving that each set is a subset of the other. I wasn't entirely sure what to do with Group 4 in the last session since we had completed the sequence of instructional tasks earlier than expected. I figured a discussion on proving set equality would provide an opportunity to discuss selecting an arbitrary element as well as provide closure to the number theory problem that the students solved in the fourth session. Figure 5.11 shows a screenshot of the proof that was constructed with help from Julian and Alonso in Group 4.

(Step 1: show D is a subset of G)	D = {set of integers divisible by 21}
consider an arbitrary element x in D	E = {set of integers divisible by 3}
So we know that x is divisible by 21.	F = {set of integers divisible by 7}
x = 21*n , where n is any integer.	
x = 21*n	Prove that $D = (E \cap F)$ (E $\cap F$) = G
= 3*7*n = 3*m , where m is 7*n	(Step 2: Show that G is a subset of D)
Since we can write x as 3 times any	Considerany element, vin G. Then v is divisible by 3 and 7.
Integer (In this case, m). We can say	So, we can write y as $y = 3^{+}m$ and $y = 7^{+}n$, where m and n are any integers
that X is divisible by 3. Therefore, X is	
an element of E.	[since we can write y as $3*m$ we can say that y is divisible by 3 and y is an
x = 21*n	element of E. since we can write y as 7*m we can say that y is divisible by
= 7*3*n	/ and y is an element of F.
= 7*p, where p is 3*n	We want to show that y is divisible by 21]
since we can write x as 7 times any	we want to show that y is divisible by 21.]
integer (in this case, p). We can say	Since y is divisible by 3 and 7, we know that we can write y as
that X is divisible by 7. Therefore, X is	y = 21*t, where t is some integer.
an element of set r.	
Thus x is an element of F and F which	Thus, y is an element of D. Therefore G is a subset of D.
means that x is an element of G. This	D - C
proves that D is a subset of G.	<i>D</i> = 0.

Figure 5.11: Proof with Group 4

Note that in the above proof there are some mistakes. First, in "since we can write x as 3 times any integer," the "any" should say "some." This mistake was made in other similar lines in the proof. In a situation where the students were able to construct a model-of reasoning about subsets and then use this model as a model-for reasoning about one set being a subset of another, and proving this fact, then that would be considered formal mathematical activity. However, I provided a lot of support for Julian and Alonso as we were working through the proof. I had a major role in the construction of the proof by frequently reminding them of the goal of the proof and structuring the conversation that would get us to showing that D is a subset of G and G is a subset of D. I did challenge

Julian and Alonso by asking questions throughout the construction of the proof, but my questions were very specific. For example, in showing that D is a subset of G, I started us off by asking Julian and Alonso to consider any arbitrary element, 'x,' in D:

Interviewer: What are the properties of 'x,' or what can we say about 'x'? **Julian**: That they're divisible by 21

I then wrote 'x = 21*n' and showed that 'x' can be written as '3*m' and can also be written as '7*p.' Alonso was initially concerned about the process of selecting an arbitrary element in D, as he explained,

I'm just thinking of the example where if you have two sets which are randomly generated, then if you just randomly pick an [element], then it is likely that [the randomly selected element] will be in the other set, but they are probably not subsets.

Alonso's comment was a nice contribution that helped guide our conversation about what it means to select an arbitrary element. I told the students that the goal in selecting an arbitrary element is to not to look "at just one element, you're looking at all the elements in a set by describing it using some property. And the property that we have [in this case] is that [the elements] are generated by 21." From there, showing that 'x' is an element in G was straightforward for Julian and Alonso. We took a similar approach to show that G was a subset of D, and I offered them a hint by asking them what they recalled about the least common multiple of two factors. In the last five minutes of this session, I asked Julian and Alonso to reflect on the proof and describe their thoughts about what we proved and how we proved. Julian doubted his ability to write the proof on his own, but also explained what we proved: I don't think I could do this by myself, I'm going to be honest. You kind of walked us through it, so I kind of mostly get it. That we're trying to findobviously we're finding that D is a subset of 'E and F' and then we're finding that G is a subset of D by using E and F for just the set D. We're just using regular division and stuff, and multiples, to then find how 'x' or those arbitrary elements are pieces of the sets. That's how we basically just connect each part to then make D and G both subsets of each other.

Alonso had been contributing the most as we worked through the construction of the proof, but he wasn't very sure about the method of showing that the two sets were subsets of each

other:

It seemed a little circular to me. I guess what we did is we used the fact that anything divisible by 21 is divisible by seven and three. Then we used the fact that if something is divisible by seven and three, then it is divisible by 21. So, I don't know, I guess the proofs are important, it just seems a little circular.

We did not have time to talk more about the proof, but I was impressed with their thoughts and contributions throughout the whole session. In the next section I present the results of the pre- and post-study mathematical content surveys as one way to document what the students learned in this study.

Pre and Post Mathematical Content Survey Results

Pre/post mathematical content surveys are one way to measure the extent to which a student learned the mathematical content that is the focus of study. In my study, the foci are mathematical logic, set theory, and computer programming. There were 12 questions on the survey: six questions were asked about set theory topics, two were about mathematical logic, three included ideas of both set theory and logic, and there was one question asking the students to interpret Python code. This section is divided by each Group in the study and their survey results are presented by each content topic. In the
tables, the cells highlighted with a green color are correct answers, and the cells highlighted with the salmon color are either wrong answers or unsure as indicated by the participants. Big-picture ideas from the survey results will be presented after the survey results, with some commentary in between on interesting findings.

Group 1

Haven, Palmer, and Judith composed the first group in my study. As mentioned previously, I learned a lot from Group 1 as their reception to the tasks and instructional methods informed the direction of the rest of the study. However, this meant that we spent more time than intended on the functionality of a For Loop and less time on some of the set theory ideas such as set intersection and subsets. On the pre-study survey, Haven, Judith and Paul scored 3, 4, and 4, respectively. On the post-study survey they scored 4, 10, and 9 respectively. Their survey results are presented below with the pre-study and post-study results presented separately.

Name	$\begin{array}{c} Q1\\ A = \{ \{1, 5, 7\}, \\ \{3, 4, 8\} \}. Are\\ any odd\\ numbers\\ between 1 and\\ 10 elements of\\ set A? Please\\ provide details\\ with your\\ answer. \end{array}$	Q2 S = {1, 3, {3, 4} }. Is 3 an element of S?	Q3 S = {1, 3, {3, 4} }. Is 3 a subset of S?	Q4 S = {1, 3, {3, 4} }. Is {3} a subset of S?	Q5 S = {1, 3, {3, 4} }. Is {3, 4} an element of S?	Q6 S = {1, 3, {3, 4} }. Is {3, 4} a subset of S?
Haven	Yes. 1,5,7 are odd numbers and are in set A	Yes	Yes	I'm not sure	Yes	Yes
Judith	Yes. 5,7,3	Yes	Yes	Yes	No	Yes
Palmer	Yes. 1,3,5,7	Yes	Yes	No	Yes	Yes

Table 5.1: Pre-Study Group 1 Set Theory Questions

Name	Q1 A = { {1, 5, 7}, {3, 4, 8} }. Are any odd numbers between 1 and 10 elements of set A? Please provide details with your answer.	Q2 S = {1, 3, {3, 4} }. Is 3 an element of S?	Q3 S = {1, 3, {3, 4} }. Is 3 a subset of S?	Q4 S = {1, 3, {3, 4} }. Is {3} a subset of S?	Q5 S = {1, 3, {3, 4} }. Is {3, 4} an element of S?	Q6 S = {1, 3, {3, 4} }. Is {3, 4} a subset of S?
Haven	Yes. 1,5,7 are odd numbers and are in set A	Yes	Yes	No	Yes	Yes
Judith	No. Set A contains only 2 subsets, {1,5,7} and {3,4,8}, the contents of which are not pertinent to answering this particular question.	Yes	No	No	Yes	Yes
Palmer	Yes. 1,3,5,7	Yes	No	No	Yes	Yes

Table 5.2: Post-Study Group 1 Set Theory Questions

In response to Q1, Judith mentions that '{1, 5, 7}' and '{3, 4, 8}' are "subsets" but my interpretation is that she is conflating the meaning between a set as an element and a set as a subset. I still considered her answer to be correct given that she answered the question "No" and she says that the "contents," or elements, in the sets are not relevant to the question that is being asked. This implies that she saw set A as having two objects, neither of them being integers and thus not relevant to the question. Another interesting aspect of

the answers to these questions is that none of the students in Group 1 answered Q4 or Q6 correctly. However, as mentioned, we did not spend much time on the topic of subsets compared to the other groups, so it is not entirely surprising that the students answered these questions incorrectly.

Name	Q7 Is the following statement true or false? "Given an integer number x, x is even or x is odd"	Q8 Is the following statement true or false? "The integer 15 is even or 15 is odd"
Haven	True	False
Judith	True	False
Palmer	True	True

Table 5.3: Pre-Study Group 1 Logic Questions

Table 5.4: Post-Study Group 1 Logic Questions

Name	Q7 Is the following statement true or false? "Given an integer number x, x is even or x is odd"	Q8 Is the following statement true or false? "The integer 15 is even or 15 is odd"
Haven	True	False
Judith	True	True
Palmer	True	True

Name	Q9 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A and B, what elements are in C?	Q10 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A or B, what elements are in C?	Q11 Consider any two sets, A and B. What does it mean for an element to not be an element of A and B?
Haven	1,3,4,5,7,8	1,3,4,5,7,8	Not sure
Judith	1,5,7,3,4,8	i don't know	the element is not in both A and B, but is in A or B or neither.
Palmer	{1,5,7,3,4,8}	{1,5,7,3,4,8}	To be a new element not found in A or B

Table 5.5: Pre-Study Group 1 Logic and Set Theory Questions

Name	Q9 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A and B, what elements are in C?	Q10 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A or B, what elements are in C?	Q11 Consider any two sets, A and B. What does it mean for an element to not be an element of A and B?
Haven	1,5,7,3,4,8	1,5,7,3,4,8	it is not in the set{} of both A or B
Judith	C={{3,4,8}}	C={{1,5,7},{3,4,8},1,5,7}	If an element is not an element of both A and B this means the element can either exist in set A or in set B or in neither set A nor set B. The element cannot exist in both set A and set B.
Palmer	{3,4,8}	{1,5,7,{1,5,7},{3,4,8}}	for it to not be in both A and B

Table 5.6: Post-Study Group 1 Logic and Set Theory Questions

Haven's post-study answer to question Q11 is not entirely clear, and I initially considered her answer to be wrong, but the word "both" made me think that her conception of the situation is correct, or else she would likely have used the word "either."

Name	Q12 What would be the output of the following code in Python? city = "San Diego" for x in city: print(x)
Haven	Not sure
Judith	San Diego
Palmer	print("San Diego")

Table 5.7: Pre-Study Group 1 Python Question

Name	Q12 What would be the output of the following code in Python? prop = "proposition" for x in prop: print(x) if ((x == "o") or (x == "p")): print(x)
Haven	prop="The integer 15 is even or 15 is odd" for x in prop: print(x) if ((15 is even) or (15 is odd))
Judith & Palmer	p p r o o p p o o s i t i o o n

 Table 5.8: Post-Study Group 1 Python Question

Given that we spent more time on For Loops than the other groups, I am not surprised that Judith and Palmer answered Q12 correctly on the post-study survey. However, it seems that the task sequence was not as beneficial for Haven as compared to her peers.

Group 2

On the pre-study survey, Saul, Eugene and Leo scored 3, 4, and 4, respectively. On the post-study survey they scored 8, 11, and 5 respectively. Eugene and Saul were usually the two to step in and provide their thoughts first throughout the study, but I was surprised at Leo's post-study survey results. However, I am considering the possibility that perhaps the reason Leo did not improve as much as his peers is because of his poor internet connection. In almost every session I asked Leo to turn his camera off to potentially help with connection, which could have been the reason why Leo was not participating as much and missing out on some of what was said by me or his group partners, ultimately missing out on the learning process.

Name	Q1 A = { {1, 5, 7}, {3, 4, 8} }. Are any odd numbers between 1 and 10 elements of set A? Please provide details with your answer.	Q2 S = {1, 3, {3, 4} }. Is 3 an element of S?	Q3 S = {1, 3, {3, 4} }. Is 3 a subset of S?	Q4 S = {1, 3, {3, 4} }. Is {3} a subset of S?	Q5 S = {1, 3, {3, 4} }. Is {3, 4} an element of S?	Q6 S = {1, 3, {3, 4} }. Is {3, 4} a subset of S?
Saul	Yes. 1,5,7,3	Yes	Yes	No	Yes	Yes
Eugene	Yes. 1,5,7,3 are odd numbers in the set A.	Yes	I'm not sure	I'm not sure	Yes	I'm not sure
Leo	Yes. 1,5,7,3 are odd numbers that are part of set A	Yes	Yes	I'm not sure	Yes	I'm not sure

Table 5.9: Pre-Study Group 2 Set Theory Questions

Name	Q1 $A = \{ \{1, 5, 7\}, \{3, 4, 8\} \}$. Are any odd numbers between 1 and 10 elements of set A? Please provide details with your answer.	Q2 S = {1, 3, {3, 4} }. Is 3 an element of S?	Q3 S = {1, 3, {3, 4} }. Is 3 a subset of S?	Q4 S = {1, 3, {3, 4} }. Is {3} a subset of S?	Q5 S = {1, 3, {3, 4} }. Is {3, 4} an element of S?	Q6 S = {1, 3, {3, 4} }. Is {3, 4} a subset of S?
Saul	No. I think A only consists of two elements, and each is a subset.	Yes	Yes	No	Yes	Yes
Eugene	No. The only elements of set A are two other sets not the elements within those sets.	Yes	No	Yes	Yes	No
Leo	Yes. 1 as there is only a value of 2 in this set as the two sets in set A ony count as 1 element	Yes	Yes	No	Yes	Yes

Table 5.10: Post-Study Group 2 Set Theory Question

Similar to Group 1, Saul and Leo in Group 2 did not answer the subset questions correctly. We spent more time on the idea of subsets than Group 1, but this goes to show that even after an instructional sequence on set theory topics, the students still had a difficult time between the idea of a set as an element and a set as a subset. As for Leo's answer to Q1 in the post-study survey, it seems that Leo recognizes that there are only two objects in set A, but it is not clear to me what he meant with his explanation.

Name	Q7 Is the following statement true or false? "Given an integer number x, x is even or x is odd"	Q8 Is the following statement true or false? "The integer 15 is even or 15 is odd"
Saul	True	False
Eugene	True	True
Leo	True	True

Table 5.11: Pre-Study Group 2 Logic Questions

Name	Q7 Is the following statement true or false? "Given an integer number x, x is even or x is odd"	Q8 Is the following statement true or false? "The integer 15 is even or 15 is odd"
Saul	True	True
Eugene	True	True
Leo	True	True

Table 5.12: Post-Study Group 2 Logic Questions

Name	Q9 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A and B, what elements are in C?	Q10 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A or B, what elements are in C?	Q11 Consider any two sets, A and B. What does it mean for an element to not be an element of A and B?
Saul	1,5,7,3,4,8	1,5,7,3,4,8	It won't be part of C either
Eugene	Both elements in A and B will be within C.	Both elements in A and B will be within C.	The element would not be defined in either set A or B.
Leo	The elements in C are (1,5,7)(3,4,8). Which are the elements in A and B	The elements in C are (1,5,7)(3,4,8).	For an element to not be an element for A or B that element can't be those numbers in the list.

Table 5.13: Pre-Study Group 2 Logic and Set Theory Questions

Name	Q9 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A and B, what elements are in C?	Q10 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A or B, what elements are in C?	Q11 Consider any two sets, A and B. What does it mean for an element to not be an element of A and B?
Saul	<i>{3, 4, 8}</i>	{3, 4, 8} , 1, 5, 7, {1, 5, 7}	Any element besides {3, 4, 8}. It's either in neither set or only one set.
Eugene	C= {{3,4,8}}	C= {{1,5,7},{3,4,8},1,5,7}	The element is not inside both sets A and B.
Leo	{{1,5,7},{3,4,8},1,5,7}	{{3,4,8},{1,5,7},1,5,7}	For an element to not be an element of A or B it has to differ in value. For example set B has elements of 1,5,7. If an element was not those values then it wouldn't be an element of B.

Table 5.14: Post-Study Group 2 Logic and Set Theory Questions

Regarding Saul's answer to Q11 in the post-study survey, Saul was answering the question with set B from Q9 and Q10 and follows up by saying that an element not in 'A and B' is either in A or in B or neither set.

Name	Q12 What would be the output of the following code in Python? city = "San Diego" for x in city: print(x)
Saul	X
Eugene	San Diego
Leo	I'm not sure.

Table 5.15: Pre-Study Group 2 Python Question

100000000000000000000000000000000000000	
Name	Q12 What would be the output of the following code in Python? prop = "proposition" for x in prop: print(x) if $((x == "o") \text{ or } (x == "p"))$: print(x)
Saul	o p
Eugene	proposition
Leo	o,p

Table 5.16: Post-Study Group 2 Python Question

None of the students in Group 3 got the Python question correct, which was a bit surprising considering that I thought the students had a good grasp on the functionality of the For Loop. Obviously, Leo and Saul knew that 'o,' and 'p' were important and perhaps they missed the first print statement in the For Loop which would have caused each element in 'prop' to be printed with the 'o' and the 'p' to be printed a second time each time it was run through the loop.

Group 3

On the pre-study survey, Delia and Juliana scored 5 and 2, respectively. On the post-study survey, they scored 6, and 4 respectively. I did have to support Delia and Juliana more than some of the other groups in the study, and I found it more difficult to

engage the students in discussions about these topics as they were not as likely to ask

follow-up questions or challenge one another's ideas.

Name	Q1 A = { {1, 5, 7}, {3, 4, 8} }. Are any odd numbers between 1 and 10 elements of set A? Please provide details with your answer.	Q2 S = {1, 3, {3, 4} }. Is 3 an element of S?	Q3 S = {1, 3, {3, 4} }. Is 3 a subset of S?	Q4 S = {1, 3, {3, 4} }. Is {3} a subset of S?	Q5 S = {1, 3, {3, 4} }. Is {3, 4} an element of S?	Q6 S = {1, 3, {3, 4} }. Is {3, 4} a subset of S?
Delia	Yes. 1,5,7,3	Yes	Yes	Yes	Yes	No
Juliana	Yes. 5,7,3	Yes	Yes	I'm not sure	No	Yes

Table 5.17: Pre-Study Group 3 Set Theory Questions

Name	Q1 A = { {1, 5, 7}, {3, 4, 8} }. Are any odd numbers between 1 and 10 elements of set A? Please provide details with your answer.	Q2 S = {1, 3, {3, 4} }. Is 3 an element of S?	Q3 S = {1, 3, {3, 4} }. Is 3 a subset of S?	Q4 S = {1, 3, {3, 4} }. Is {3} a subset of S?	Q5 S = {1, 3, {3, 4} }. Is {3, 4} an element of S?	Q6 S = {1, 3, {3, 4} }. Is {3, 4} a subset of S?
Delia	Yes. 1,5,7,3	Yes	Yes	Yes	Yes	Yes
Juliana	Yes. 1,5,7,3	Yes	Yes	No	Yes	Yes

 Table 5.18: Post-Study Group 3 Set Theory Questions

As with the other groups, the subset questions (Q3, Q4, and Q6) seem to give the students the most trouble and the element questions (Q2 and Q5) seem to be easier to answer.

Name	Q7 Is the following statement true or false?	Q8 Is the following statement true or false?
	Given an integer number x, x is even or x is odd"	The integer 15 is even or 15 is odd"
Delia	True	False
Juliana	True	False

Table 5.19: Pre-Study Group 3 Logic Questions

Table 5.20: Post-Study Group 3 Logic Questions

Name	Q7 Is the following statement true or false? "Given an integer number x, x is even or x is odd"	Q8 Is the following statement true or false? "The integer 15 is even or 15 is odd"
Delia	True	False
Juliana	True	True

Name	Q9 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A and B, what elements are in C?	Q10 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A or B, what elements are in C?	Q11 Consider any two sets, A and B. What does it mean for an element to not be an element of A and B?
Delia	im not sure	im not sure	im not sure
Juliana	C={{1,5,7},{3,4,8}}, {1,5,7,{3,4,8}}	No idea	No idea

Table 5.21: Pre-Study Group 3 Logic and Set Theory Questions

Name	Q9 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A and B, what elements are in C?	Q10 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A or B, what elements are in C?	Q11 Consider any two sets, A and B. What does it mean for an element to not be an element of A and B?
Delia	1,5,7,3,4,8	1,5,7,3,4,8	That means the element will not be found in A and B
Juliana	{1,5,7,3,4,8}	{1,5,7,3,4,8,1,5,7,3,4, 8}	it means that the element is not defined in either set A or B

Table 5.22: Post-Study Group 3 Logic and Set Theory Questions

Delia's answer to Q11 in the post-study survey was borderline for me in terms of whether she really understood the answer, especially after not answering Q9 correctly, which also uses the logical operator 'and.' However, she wasn't wrong, so I marked her answer as correct.

Name	Q12 What would be the output of the following code in Python? city = "San Diego" for x in city: print(x)
Delia	'San Diego'
Juliana	print(San Diego)

Table 5.23: Pre-Study Group 3 Python Question

Name	Q12 What would be the output of the following code in Python? prop = "proposition" for x in prop: print(x) if ((x == "o") or (x == "p")): print(x)	
Delia	p p r o o p p p o o o s I t o o n	
Juliana	{ <i>p</i> , <i>o</i> }	

Table 5.24: Post-Study Group 3 Python Question

Group 4

On the pre-study survey, Julian and Alonso both answered 3 questions correctly. On the post-study survey they scored 7, and 12 respectively. Alonso was the only student to answer all 12 questions correctly on the post-study survey. It was clear throughout the study that Alonso was highly interested in the material and was often quicker to pick up on some of the concepts than Julian. However, Alonso allowed space for Julian to voice his thoughts which ultimately led to differing ideas and an open dialogue between the two students. As a result of their effective communication, we were able to have in-depth discussions about programming and the mathematical content at a much quicker pace than the other groups.

Name	Q1 A = { {1, 5, 7}, {3, 4, 8} }. Are any odd numbers between 1 and 10 elements of set A? Please provide details with your answer.	Q2 S = {1, 3, {3, 4} }. Is 3 an element of S?	Q3 S = {1, 3, {3, 4} }. Is 3 a subset of S?	Q4 S = {1, 3, {3, 4} }. Is {3} a subset of S?	Q5 S = {1, 3, {3, 4} }. Is {3, 4} an element of S?	Q6 S = {1, 3, {3, 4} }. Is {3, 4} a subset of S?
Julian	I'm not sure	Yes	Yes	Yes	No	Yes
Alonso	Yes. I'm guessing but there are odd numbers that are between 1 and ten such as 1,5,7,3	Yes	Yes	I'm not sure	Yes	Yes

 Table 5.25: Pre-Study Group 4 Set Theory Questions

Name	Q1 A = { {1, 5, 7}, {3, 4, 8} }. Are any odd numbers between 1 and 10 elements of set A? Please provide details with your answer.	Q2 S = {1, 3, {3, 4} }. Is 3 an element of S?	Q3 S = {1, 3, {3, 4} }. Is 3 a subset of S?	Q4 S = {1, 3, {3, 4} }. Is {3} a subset of S?	Q5 S = {1, 3, {3, 4} }. Is {3, 4} an element of S?	Q6 S = {1, 3, {3, 4} }. Is {3, 4} a subset of S?
Julian	No. There are no number elements in set A rather there are group elements {1, 5, 7} which are made as 1 element and not as individual number elements.	Yes	Yes	No	Yes	Yes
Alonso	No. The elements of set A are sets not integers. The sets that are elements have the odd numbers 1,5,7 and 3.	Yes	No	Yes	Yes	No

Table 5.26: Post-Study Group 4 Set Theory Questions

Again, we see that the subset questions proved to be a challenge, with Julian not answering Q3, Q4, or Q6 correctly.

Name	Q7 Is the following statement true or false? "Given an integer number x, x is even or x is odd"	Q8 Is the following statement true or false? "The integer 15 is even or 15 is odd"
Julian	True	False
Alonso	True	I'm not sure

Table 5.27: Pre-Study Group 4 Logic Questions

Table 5.28: Post-Study Group 4 Logic Questions

20. 1 Ost Study Gloup + Logic Questions									
Name	Q7 Is the following statement true or false? "Given an integer number x, x is even or x is odd"	Q8 Is the following statement true or false? "The integer 15 is even or 15 is odd"							
Julian	True	False							
Alonso	True	True							

	2 1 0			
Name	<i>Q9</i> <i>A</i> = { {1, 5, 7}, {3, 4, 8} <i>J</i> , <i>B</i> = {1, 5, 7, {3, 4, 8} <i>J</i> . If we define C to be the set of elements that exist in A and B, what elements are in C?	Q10 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A or B, what elements are in C?	Q11 Consider any two sets, A and B. What does it mean for an element to not be an element of A and B?	
Julian	3,4,8	1,5,7,3,4,8	I am not sure	
Alons o	not sure	not sure	I'm not sure	

Table 5.29: Pre-Study Group 4 Logic and Set Theory Questions

Name	Q9 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A and B, what elements are in C?	Q10 A = { {1, 5, 7}, {3, 4, 8} }, B = {1, 5, 7, {3, 4, 8} }. If we define C to be the set of elements that exist in A or B, what elements are in C?	Q11 Consider any two sets, A and B. What does it mean for an element to not be an element of A and B?
Julia n	The elements in C would just be {3, 4, 8} because is is the only similar element as 1, 5, 7 is not the same as {1, 5,7}	The elements in C would be 1, 5, 7, {1, 5, 7}, {3, 4, 8}	That means the element is not present in both A and B rather it is either present in A or B but not both.
Alons o	{3,4,8}	{ {1, 5, 7}, {3, 4, 8}, 1, 5, 7, {3, 4, 8} }	"and" is being used as a logical operator which means that the element is not in both the sets. That means the element could either be in one set or no sets.

Table 5.30: Post-Study Group 4 Logic and Set Theory Questions

Both Julian and Alonso answered all of the combination logic and set theory questions correctly, which was an interesting result considering that Julian did not answer the subset questions earlier correctly. However, the combination logic and set theory questions did not explicitly ask about subsets, which is an oversight on my part in the design of these questions.

Table	5.31:	Pre-Stu	dv Grou	in 4 Pvt	hon Oues	stion
1 uoic	5.51.		ay Oroc	ap i i yu		Juon

Name	Q12 What would be the output of the following code in Python? city = "San Diego" for x in city: print(x)
Julian	print(San Diego)
Alonso	San Diego

Table 5.32: Post-Study Group 4 Python Question

Name	Q12 What would be the output of the following code in Python? prop = "proposition" for x in prop: print(x) if $((x == "o") \text{ or } (x == "p"))$: print(x)
Julian	(o, p, p, o, o,)
Alonso	not enough lines to show realistic output in console. Would print pprooppoositioon but it would be arranged vertically

While there should have been enough space in the console for Alonso to write out his answer, his description of the output is entirely correct. As for Julian's answer, it is interesting that there are multiple 'p's and 'o's in his answer which indicates some idea of an iteration through 'prop,' I am just not entirely sure how he came to that answer.

Summary

Below is a table of the results of the pre-study survey and the post-study survey.

Table 5.33: Pre-Study and Post-Study Results (number of correct responses)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
Pre - Study	0	10	0	3	7	1	10	3	0	0	1	0
Post - Study	5	10	4	3	10	2	10	7	6	7	8	4

All of the students on the pre-study survey answered Q2 correctly, indicating that '3' is an element of 'S = {1, 3, {3, 4}}.' My hypothesis as to why all of the students answered Q2 correctly is that the students relied on their colloquial understanding of what an "element" is, which is synonymous with "component," "part," or "constituent." Similarly, we see that seven out of the 10 students answered Q5 correct on the pre-study survey, which asked whether '{3, 4}' is an element of S. With the idea that "element" is synonymous with "component," "part," or "constituent," it is sensible that the students were able to identify an object in a set (whether that be an integer or a set) and were successful on those two questions before the study took place. All of the students also answered Q7 correctly, claiming that the statement "*Given an integer number x, x is even or x is odd*" is True. Additionally, only three students answered Q8 correctly on the pre-study survey which asked whether or not the statement "*The integer 15 is even or 15 is odd*" is true or false.

These results are further evidence of Dawkins and Cook's (2017) Part False-All False Heuristic in which the presence of one false statement in a disjunction is reason for students to declare the whole disjunction false.

The three questions that students struggled with most were Q3, Q4, and Q6 asking whether '3,' '{3},' and '{3,4}' are subsets of S. Given the lack of coherence in the instructional task sequence on subsets, it was not entirely surprising to see these results. The results from Q3, Q4, and Q6 support a need to revise the task sequence on subsets and find a better way to introduce the idea of subsets instead of using the set intersection task to promote students' situational mathematical activity. Compared to the concept of elements of a set, the students were provided with minimal learning opportunities to construct their own understanding of what it means for a set to be a subset of another set. Most of what we discussed during the TE sessions focused on the elements of a set, either finding the cardinality, adding elements to a set or finding the intersection and union of sets and discussing the elements that belonged in the new set. All of these ideas focus on the elements as the primary property of the set. It wasn't until the last session or the penultimate session for some groups did we discuss what it means for a set to be a subset. With that said, it is peculiar that only half of the students answered Q1 correctly on the post-study survey asking about the elements of the defined set, A. It could just be that asking the students to extrapolate about the properties of a set and provide a justification for their reasoning is a more difficult question to answer than answering a true or false question.

226

The greatest improvements were on items Q9, Q10, and Q11, with six, seven and seven additional correct answers, respectively. The structure of Q9, Q10, and Q11 were similar to the structure of the questions that the students were asked to solve during the TE. That is, we spent most of our time working with set theory and logic in the same context, which could be the reason for the students' success on these questions. In terms of learning opportunities, the idea of the For Loop with an If Statement to filter for certain elements was a major focus of discussion and opportunity for students to build an understanding of set relationships such as union and intersection. The average score on the pre-study survey for the students was 3.5 and the average score on the post-study survey was 7.6, more than double that of the pre-study survey.

Chapter 6: Confidence and Mathematical Identity

The purpose of this chapter is to address RQ3: How does the use of Python to learn mathematics, in a small-group collaborative setting, influence students' affective experiences and the development of their mathematical identity? This chapter is divided into four sections, one for each group, with subsections for each research participant. As outlined in Chapter 3, there are five units of analysis with respect to the students' affective experiences. The first is the main component of analysis, which is the documentation of a quantitative change in their pre-study and post-study ATMLQ results. Changes in the students' responses from the pre-study questionnaire to the post-study questionnaire, which are presented as a difference, reveal a shift in their confidence and interest as learners and doers with respect to three categories: mathematics, computers and programming, and programming to learn mathematics. The second, third and fourth units of analyses are smaller components which are used to provide a more robust qualitative description of their dynamic mathematical identities. Lastly, their open-ended responses to the questionnaire are presented to document their final thoughts on their experiences in the study. For each student, a summary table is presented to provide an overview of their experience and characterization of their shifting mathematical identity in this study, as well as individual histogram charts to highlight the three different blocks of the ATMLQ.

Given that there are only ten research participants in this study, generalizable conclusions from the ATMLQ cannot be made. However, analysis of each student's responses can serve as a component of the bigger-picture description of their affective experiences. As a reminder, analysis of the questionnaire items was conducted to document a positive or negative shift in their attitude towards mathematics, computers and programming. For example, consider the following item on the questionnaire, "I find mathematics frightening." If the student's response to this question on the pre-study questionnaire was "Strongly Agree" (-2) and their response to this question on the poststudy questionnaire was "Strongly Disagree" (2), then they had a positive shift of four units (2- (-2)) for that question in the Mathematics block of questions. This analysis was conducted for each question on the questionnaire for each student. For the students in my study, the average score on the mathematics block of questions on the pre-study questionnaire was 6.9, post-study was 9.3. The average score on the computers and programming block of questions on the pre-study questionnaire was 10.2, post-study was 10.5. Lastly, the average score on the programming to learn mathematics block of questions on the pre-study questionnaire was 11.8.

Group 1

Group 1 was composed of Haven, Judith and Palmer. There were two main reasons why I grouped these students together, the first is that all of the students identified as firstgeneration college students, and the second is that I wanted to make sure that there were at least two women in a group of three. Additionally, all these students identified as white, which made me more inclined to group the only white man in my study with the two white women. Both Judith and Palmer kept their cameras turned off throughout the duration of the study and Haven kept her camera on. In the first session I told the students that if they feel comfortable, they can turn their cameras on, but Judith and Palmer never did. Given that Judith and Palmer's cameras were turned off, it was much harder for me to get a read on how they were receiving the material, and it felt difficult to build community among the students.

Haven

Haven's mathematical identity development can be described as a significant shift, from coming into the study and seeing herself as a low performer in mathematics to seeing herself as someone that can do math and programming. Based on the questionnaire results, Haven had the greatest positive shift in her mathematical confidence and interest from prestudy to post-study with a difference of 32 points. This result was initially unexpected, as Haven was seemingly the least confident and not as strong mathematically compared to her partners throughout the duration of the study. However, after analyzing Haven's reflections after each TE session, Haven always had a positive attitude and seemed to really enjoy learning new material. A summary of Haven's affective experiences is presented in Table 6.1.

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics	+8			Positive
Computers and Programmin g	+12	Not confident in her answers, but not afraid to guess to get	Allowed space for others Made it clear she	the tasks Excited to participate
Programmin g to Learn Math	+12	a conversation started	found the tasks to difficult	Interested in new mathematics

Table 6.1: Haven's Affective Characterization
As a reminder, the summary tables presented in this chapter represent a synthesis of the data collected and analyzed related to each student's affective experiences. For example, in analyzing the data, I constructed a larger table that contained quotes by Haven that related to each of the above columns or notes that I took as part of my descriptive accounts of each session. These notes and quotes were then pared down to reflect the most salient ideas related to Haven's affective experiences. Haven was always upbeat and excited to participate throughout the study. Usually in the first few minutes of informal conversation time she would talk about her classes and discuss any upcoming tests that she was either looking forward to or nervous about. During the TE, if there was a long pause and none of the students quite knew what the correct approach was, Haven typically jumped in and offered an idea even though she wasn't quite confident in her answer. For example, we had a proposition, 'p = "dog" in setA' which was a True proposition, and I asked the students what they thought 'print(p)' would produce (this was on the first day of the TE). There was a long pause before the following exchange occurred:

Haven: I want to guess that it's going to print 'dog.' Or maybe say 'True.' I can't tell the difference between the propositions and then when it just [prints it], you know?
Interviewer: Yeah totally, does anybody want to tag in on that one?
Palmer: I think it will say 'True.'
Interviewer: Okay, and why do you think that?
Palmer: Because it's printing 'p,' and 'p' is a proposition. Kind of like declaring that this element is in the set.

Before the 'print(p)' command, we had seen other print statements such as 'print(hello)' which would print out the string 'hello.' Haven wasn't sure whether the print statement would print out the string 'dog,' or evaluate the proposition, but she offered her thoughts

on the question anyway. Haven's contribution ultimately prompted Palmer to provide his thoughts, which when we ran the print statement, proved correct. After the first session of the TE, I asked all of the students to reflect on the day's activities and comment on their thoughts about how the first day went, Haven's response is a great example of the excitement that she brought each day to our sessions:

I mean this stuff is really cool, I've never seen this stuff before. Like that percentage sign [modulus] thing, I was like 'oh my mind is blown.' And then even at the end, I was like 'whoa,' relating [the logical operators] to math and stuff. So, yeah I'm excited to learn more!

As we started to get more and more into programming, introducing For Loops and If Statements, Haven seemed to need slightly more help than her peers, but always kept a positive attitude and felt comfortable asking questions when she needed to or letting everyone know that she wasn't sure what was going on. In one instance, I asked the group how we would find the intersection of three sets using Python, and Haven responded immediately by saying "I'm not quite sure...I'm kind of getting 'lost in the sauce' now. There's a lot [to think about]." Haven made the comment with a smile on her face and her honesty lightened the mood as Judith joined in with Haven in her laughter. Haven never seemed embarrassed when she didn't quite understand something, and at the end of Session 3, Haven commented on her interest with the programming ideas:

This stuff keeps getting cooler. It's crazy how- you mentioned before, with the 'range()' and how it will just do all this math and spit out all these numbers super fast. And you can have one line do all that instead of working more for it. So, that's kind of cool. And even...yeah, I don't know. I'm still wrapping my head around the whole For Looping stuff.

Haven acknowledged that she was still trying to understand how the For Loop was working, but she still commented on how the material "keeps getting cooler." Haven's interest in the mathematics we were exploring, as well as her desire to understand the programming material, are likely the reasons why there was such a large difference in her ATMLQ results. Haven's ATMLQ results are presented below in Figure 6.1.



Figure 6.1: Haven's Pre and Post ATMLQ Results

Haven came into the study with a low sense of confidence and interest in mathematics, reporting a score of -10 on the ATMLQ, but by the end of the study, Haven experienced a positive growth to -2. The greatest shifts occurred with respect to Haven's perspectives on computers and programming, and programming to learn math, each with a positive shift of 12 points. Given her participation throughout the study, her results are not surprising; even though Haven was not always correct or confident in her answers, she gave her best effort and found a genuine interest in the material. Haven's responses to the free-response questions at the end of the post-study ATMLQ are presented below, offering a more detailed look at how she perceived her experience as a research participant.

In response to the question, "What was your favorite part about participating in this study?" Haven said, "I got to learn some basics of set theory in python. The people in my group were great and always kept me on my toes. This was something new and exciting for me to learn." While there wasn't a specific topic that Haven called out as being her favorite, again she reflects a general interest and excitement in her experience. In response to "What was your least favorite part about participating in this study?" Haven had the following to say:

Trying to work around everyone's schedule can definitely be difficult but for me. This was the 6th zoom call of the day for me on Wednesdays and sometimes I dreaded going to it but it always turned out to be worth it and time always flew by in the sessions it seemed. Usually I hate hate hate cold calling in my classes but I know you needed our input and actually it wasn't bad. I didn't mind it for this setting. You were very patient with our answers.

Haven mentions that she hates cold-calling (calling on someone for an answer without them volunteering), but she also said that for this study, "it wasn't bad." I did my best to let the students know that it was okay to not know the answer and that when I called them, I was really just interested in how they were thinking about the material. She also mentions that she had six Zoom meetings on the days that Group 1 met. At the time of data collection, everything was still virtual in terms of meetings and classes, so, "Zoom fatigue" was certainly a very real factor that was likely affecting many of the students in the study. Relatedly, the third open-ended question addressed the virtual nature of data collection, "What are your thoughts on working collaboratively with others in the Zoom virtual setting?" Haven's response primarily addressed her experience in terms of logistics: I thought the zoom setting was great. It made this really accessible for me and easy to be a part of. It's just the click of a link. Sometimes I wish the other people would have had their cameras on but it really wasn't a huge deal.

As mentioned, Judith and Palmer had their cameras turned off throughout the duration of the study. I also thought that the group would have benefited if everyone's cameras were on, but I did not want to press the issue. The last question asked about the student's identity, "What about you or your identity contributed to your success in this study?" Haven reflected on her identity as someone that gives 100%:

I'm a really passionate person like when I'm doing something I enjoy, I go all in. No distractions. I felt like this was just new enough and exciting enough for me that I was able to really dive in and focus a lot on it.

Haven's response reflects the main takeaway, that her positivity, excitement and interest really helped her succeed and ultimately led to large shifts in her perspectives on mathematics, computers and programming to learn mathematics.

Judith

Judith's mathematical identity development can be described as one of maintaining a strong sense as a mathematician and a growing interest in how programming can relate to mathematics. Across the three blocks of her ATMLQ results, Judith scored the highest in mathematics for both the pre-study questionnaire and the post-study questionnaire. Judith typically wasn't the first to offer a solution to a task, but when she did, she was usually right. Also, when Judith did provide an answer, most of the time her answer was phrased as a question, indicating a lack of confidence. For example, in the first session I asked if anyone had any thoughts on what the command, 'B.add('whale')' would do, Judith responded by saying, "Yeah it's going to add 'whale' to set B maybe? Or not inside the-

235

wait, I don't know. And then there would be seven elements, maybe?" Judith's reasoning was correct, the '.add()' function adds the argument to the specified set, but like this one, many of her contributions were qualified with a "maybe" or "I don't know," even though she was right. A summary of Judith's affective experiences is presented in Table 6.2.

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematic s	+5	Semi- confident in	Karan ahar	Positive attitude
Computers and Programmi ng	+3	her answers, typically not the first to volunteer	Knew when she was wrong Tended to think out loud	Strong interest in mathematics and found
Programmi ng to Learn Math	+3	an answer, and her answers were most often correct	to start a conversation about a difficult task	connections between programming and mathematics

|--|

Judith often found ways to relate what we were doing to something that she had seen before in her mathematics classes. For example, when the students were first learning about the 'and' and 'or' logical operators, Judith commented that the operators were functioning in a similar way to the union (U) and intersection (\cap), where the union is commonly used in Precalculus and Calculus classes to discuss the domain and range of functions:

It reminds me of the thing in math where you have the thing in parentheses, and then you have the 'U looking thing.' That means like 'or' and then you have the two parentheses, and you have the 'and looking thing' in between the two sets of parentheses that mean this and that. So, I was thinking, like

the 'p or r' since both 'p' and 'r' exist, it'll either be True or False and since True falls in True [or] False, True is- it's True you know? And then the 'p and r,' it can't be both True and False. If that makes any sense.

It was interesting that Judith brought up this idea in the first session, before we had even gotten to union and intersection of sets. Relating the 'or' and 'and' operators to the union and intersection was the goal of the TE and Judith saw the connection way ahead of what I had planned. Given that Judith had seen the union and intersection in a previous mathematics class, it is worth considering the influence this may have had in terms of her participation and ideas throughout the study. If Judith had a strong understanding of union and intersection before the study, then her contributions should be reanalyzed. However, since Judith refers to the union as the "U looking thing," this makes me question the significance or meaning that the union actually had for Judith as well as the other students in the study. At the end of the first session, Judith's reflection highlighted her interest in mathematics, "I think it's super cool. I like math a lot, so it's cool to do this kind of stuff." She also called out her interest in mathematics and enjoyment in finding connections to mathematics at the end of the third session, "I think it's super cool. I like when you introduce a new topic. I'm trying to figure out like, 'Oh, how does this connect to math' or something. It's really cool." As mentioned, Judith came into the study with a high sense of confidence and interest in mathematics, and she reported even greater confidence and interest after the study. Her results on the ATMLQ are presented in Figure 6.2.



Figure 6.2: Judith's Pre and Post ATMLQ Results

Judith showed gains in every block on the ATMLQ from the pre-study questionnaire to the post-study questionnaire, with a five-point gain in mathematics, and three point gains in computers and programming, and programming to learn math. Given that there were not any major shifts in her scores, my hypothesis is that this experience solidified (if not bolstered) her belief and confidence in herself as a mathematician and supported her interest in programming. The only time Judith mentioned anything about her participation and contributions in the study was her reflection about the entire experience after Session

5:

I liked it a lot. It's definitely something that I feel super engaged in, which is super cool. Because whenever I have this meeting thing I'm always paying 100% attention to what's going on. And I liked that you would like- you're active about participation, you're like 'Oh Haven what do you think? Or Judith, Palmer? What do you think?' I liked that. It was cool. Yeah, it's been a good experience. Judith addressed my approach to engaging the students, in that I would never immediately evaluate a student's response by saying "that's not right," or "good job" and move on. Instead, I would ask each student what they were thinking, which was helpful to hear from all of the students, but I have also realized that this approach did not give the students a chance to reflect on their incorrect answers. The times that Judith did provide a wrong answer (which was very rare) she would typically give her answer as more of a think-out-loud explanation of the process. I found two examples that represent this phenomenon. In the first example, I asked the students how we might go about finding all of the unique elements between three different sets. Judith responded by saying the following:

Well, I was thinking you could just copy and paste. Like make another set D, and then copy and paste everything from A, B, and C in it. But that might be hard in practice if there were like a lot of really big sets or something.

Judith's reasoning isn't wrong, but the goal was to use Python in a way that would simplify this process, and as Judith mentioned, copy and pasting would be difficult if you were working with large sets. In another example, I asked the students how we would find the intersection of three sets and the students got to a point where they knew that they wanted to use a For Loop and an If Statement to find the common elements. However, they weren't sure how to filter for the ones they wanted, so I wrote the 'and' operator in the second line of the following code:

for x in A: if and :

In the following exchange, Judith doesn't provide the right answer and immediately follows up her answer with an out-loud explanation as to why she is wrong:

Judith: What you wrote with the 'and' makes a lot of sense because instead of the 'or' we're using 'and' because we're finding what is in sets A, B and C.

Interviewer: Okay, and so, what do you think would go in these missing spaces here.

Judith: 'Chips' and '6,' but I mean... I guess we would have to know that 'chips' and '6' are repeated, but we might not know that for larger sets.

Judith's first response was correct, we need the 'and' operator to filter across all three sets, but what should go in the missing spaces is 'x in B' and 'x in C.' Judith knew that 'chips' and '6' couldn't go in the missing spaces because we would have to know beforehand that 'chips' and '6' were the only two elements common across all three sets. Judith's thinkout-loud process was very helpful to engage the other students in the conversation as I would follow up Judith's response by asking one of the other students to reflect on Judith's thought processes.

As for Judith's open-ended responses, Judith's perspective on her favorite part about the study was that "This study was super interesting and engaging. Looking for the connections to mathematics and learning a new skill were probably my favorite things about this study." Judith didn't have anything that she wanted to share about her least favorite part of participating as she put "N/A." As for her perspective on working on Zoom, she said that she, "would prefer in person, but working collaboratively on zoom is definitely a great substitute, given the current state of things." I thought her preference to be in person was a little surprising as Judith had her camera off throughout the entire study, but perhaps Judith was working in a physical space with others (like a dorm room or common living area with family) which could have been distracting for the other participants. Lastly, in response to the question "What about you or your identity contributed to your success in this study?" Judith responded, "My interest in the topic. I love Math and Linguistics, and computer programming languages give me a bit of both." For Judith, this experience solidified her positive sense of mathematical identity as a capable learner and doer of mathematics. Her ideas furthered the mathematical agenda, and my impression was that she was well-respected in the group as we all quickly learned that Judith was usually right when she answered a question.

Palmer

Palmer's mathematical identity development can be described as a process of maintaining a positive sense of mathematical identity and finding an appreciation in how complex programming can be. Across all of the students in this study, Palmer scored the highest in the computers and programming block for both the pre-study questionnaire and the post-study questionnaire with scores of 26 and 16, respectively. 26 was the highest possible score on the computers and programming block of the questionnaire, which showed a great deal of confidence in his ability to understand computers and interest in programming. His confidence came through in his participation as Palmer was generally sure of himself in his answers and not afraid to voice his opinion, but not in a rude or overly confident way, it was just clear that he enjoyed participating. For example, in the first session I asked if anyone had any ideas what 'print(21 % 5)' would produce, and Palmer was the first to offer his thoughts, "I have two ideas. The first one, maybe it will divide [21] by [5]. My other idea was maybe it would find 21% of five." Neither of Palmer's guesses were correct, but his thoughts spurred the others to throw out their own ideas too. Another instance of Palmer's confidence came at the beginning of the third

241

session when I asked the students to reflect on their experiences up to that point in which he said, "I feel like it's kind of easy to pick up. As we've been going through, nothing seems too complicated so far." A summary of Palmer's affective characterization is presented in Table 6.3 below.

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics	-2	Confident in his	Appreciated	Belief in himself to
Computers and Programming	-10	answers Confident in his ability to understand the material	difficult tasks and was most engaged when the group was struggling	material
Programming to Learn Math	+4			Overwhelmed at the end

 Table 6.3: Palmer's Affective Characterization

Throughout the study Palmer was very even-keel and nothing seemed to faze him. He did have his camera turned off and he was muted until he wanted to say something, but that was just representative of his approach to learning. That is, he didn't have much to say, but I could tell that he was actively listening to me and his partners because he always made productive contributions, either in the form of a question or an answer. In his reflection on the first TE session, Palmer said the following:

I'm in the same boat as Haven and Judith. I think when we were doing this, I was seeing a whole bunch of concepts that I learned all throughout high school and college here. They're kind of like...I don't know, I noticed them being used in a different way. It was like a new perspective. So, like Judith, Palmer also was able to see connections between what he was learning in the TE and what he had seen in his previous mathematics classes. After the third session, Palmer still had a positive perspective on the material:

Palmer: Everything is kind of like building upon each other, like every little facet of the coding. So as we go on it never gets that hard because we're just introduced to new topics one at a time.
Interviewer: And do you think that's manageable for you, or is it too fast? Are things moving too fast?
Palmer: No I think it's really manageable because it makes it really easy to kind of backtrack a little. If a new topic does seem a little confusing at first, I just have to look at what we did before, and then kind of see what's changing- like when you introduced the For [Loop]

At this point in the TE, Palmer is still feeling good about programming and feels that

everything is building nicely, which from a design perspective, is good feedback to hear.

However, once we reached the fourth and fifth TE sessions, Palmer indicated that the new

material was starting to become more challenging as we were putting everything together

to solve specific tasks related to set theory:

I feel like today was a little more challenging. Especially with the week off, I'm kind of forgetting all the different options I have now. At the beginning it was only one at a time and now it's like all of the different- you have to apply everything now. So, it's kind of overwhelming.

There was a definite shift in Palmer's attitude towards the material after Session 4 in which

we started to solve problems related to set theory using Python. Figure 6.3 shows his pre

and post ATMLQ scores.





My interpretation of his results is that he maintained his identity as a confident learner and doer of mathematics (even though he dropped two points) and gained a deeper appreciation for the complexity of computers and how they can be used to program and solve problems. As noted previously, 26 is the highest one could score on the computers and programming block, it is likely that after his experience in the TE, he realized how much he didn't know about computers and programming, which dropped his confidence down 10 points. However, his score of 16 on the post-study questionnaire on the computers and programming block was still the highest out of any of the students in the study.

Even though Palmer seemed to experience a shift in his perspective on computers and programming, his reflection after the last session shows how much he valued his time as a research participant: I'm really glad I signed up for this. I think at first I just kind of saw it as another thing to do throughout the week for school, But it was really interesting to kind of dive into a new topic I really have no background in and learn a little bit about Python. And maybe in the future I'll take another class or something to learn more.

The sentiment in his reflection was echoed in his description of his favorite part of participating in the study, "My favorite part of the study was learning about python and how it works, I hope to learn more and use it in the future." In line with my interpretation of Palmer's experience, his least favorite part of the study was, "when at the end we had all of these terms and coding techniques all coming together and it all became very overwhelming and confusing near the end." As for his experience working virtually, Palmer said, "I think Zoom works best when it's small groups of people collaborating like we did during this study, so I feel it didn't cause any trouble for me and I enjoyed working with everyone." Lastly, even though Palmer thought that the end was overwhelming and confusing, he still found interest in the material and he was motivated to continue learning. He attributes his success to his "background of using computers for most of my life and my interest in mathematics, that allowed for me to succeed in this study." Similar to Judith, Palmer attributes part of his success to his interest in mathematics, a direct predictor of positive mathematical identity. Interpreting Palmer's total score difference from pre-study ATMLQ to post-study ATMLQ of -8 points, I see this negative shift as more of a reflection of his appreciation and understanding of how much can be done with computers and programming. As noted, his post-study ATMLQ on the computers and programming block of questions was still the highest out of all the students in the study, even after it had dropped 10 points.

Group 2

The students in Group 2 were Saul, Eugene and Leo. All three identified as men, Eugene and Leo identified as Hispanic or Latinx, first-generation, commuter students and Saul identified as East Asian and white. All three of the students kept their cameras turned on throughout the duration of the study, except for when I asked Leo to turn his camera off three times about halfway to three quarters of the way through the second, third and fourth sessions due to poor connectivity issues. Generally, I would describe Group 2 as a group of students with a predisposed interest in STEM as they all brought an energy to each session that was unlike the other groups in the study. More specifically, there were instances in which we were all laughing, and it seemed that the students were genuinely having a good time. First I present a characterization of Saul's affective experiences, followed by Eugene and Leo.

Saul

Saul was the quietest participant in Group 2 in that he was hardly ever the first to speak up unless I asked for his thoughts. However, he wasn't afraid to voice his ideas when asked to. His mathematical identity during the study can be described as a high positive belief in his mathematical ability with a newly developed interest in computing and programming. I would characterize Saul's approach to the material throughout the study as positive, reflective and confident. He did not seem phased when he answered something wrong, instead he would often sit back, smile, and think about why he was wrong. In the moment, and now upon analysis, I interpret his smile as enjoyment for trying

246

to solve a difficult task, not as embarrassment or anything else. Saul's affective

experiences characterization can be found in Table 6.4.

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics	+1		Not bothered	Positive
Computers and Programming	+3	Confident in his ability to solve the tasks	wrong Enjoyed working on difficult tasks	attitude Interested
Programming to Learn Math	+6			Нарру

Table 6.4: Saul's Affective Characterization

Even though Saul was pretty quiet compared to his group I knew that he was intently listening because of his ability to rephrase or clarify his partners' ideas into his own words. For example, in one instance, the students were given the sets A and D where each set contained 12 elements and I asked what would happen if we added set A to set D. The group came to two possible scenarios, one is that we can write a For Loop to add the elements of set A to set D, as follows:

for x in A: D.add(x)

Another scenario was the one in which we added the entire set A (as an element) to set D by making A a 'frozenset' in Python and adding it to D by using 'D.add(A)' which resulted in the following exchange:

Interviewer: What's different about set D, as compared to what we did earlier where we did 'for x in A?'
Leo: You don't have to do another For Loop?
Interviewer: We don't have to do another For Loop, but also let's take a look at the length of set D, the length is being reported as 13.
Leo: Oh it's not adding the actual- it's just like adding one. Like set A would just be value one.
Interviewer: Yeah... Saul, can you rephrase that? When Leo is saying it's adding "value one," what is he referring to?
Saul: I think it's counting set A as one element so it's counting all 12 [elements] of B and then it's adding basically one element which is set A to make it 13.

As the above exchange demonstrated, Saul was often able to not only rephrase another

student's ideas, but also provide additional rationale as to why certain code produced a

specific output. As for the learning of Python, Saul was particularly in favor of the PRIMM

method of instruction, which he described after the second TE session:

I was going to say the 'and' and 'or' made a lot more sense today because we were practicing it and then the parentheses [using parentheses in a logical statement], once you start thinking about it compared to math where you do the parentheses first [order of operations] then I started getting that too. So I think it really helps that we kind of guess what it's going to be first and then we run it and see what it is, and then we alter our predictions or whatever.

The fact that Saul explicitly addressed this method of instruction shows me that he was

acutely aware of his own learning process which included making mistakes and having the

opportunity to find the reasons why he made those mistakes. Saul echoed this sentiment in

his answer to the free-response question on the final survey about his favorite aspect of

participating in the study:

I like the way we learned how to program. It was a lot of testing to figure things out rather than just given a list of what everything does. For that it made the learning process a lot more interesting and fun. During the rollout of the study and the TE sessions with Group 2, I wasn't completely sure how Saul was doing in terms of his interest and enjoyment participating in this study. That is, during the first two sessions Saul was often quiet and leaned back in his chair, which was an indication for me that he might be experiencing a lack of interest with the material. However, after the third session, I figured that Saul was just deep in thought and preferred to sit quietly with his own ideas and listen to others. Now, having revisited the data, I would certainly say that this is true. After the third session, Saul reflected on how the material was getting harder and how he was dealing with this fact:

Since we are adding more and more to our knowledge, then we're having more steps to follow for each [activity]. So, kind of trying to figure out why it's computing some things takes a little bit longer. Like the last [activity], I didn't really realize why until the very end, but before that I was kind of-I didn't really even have a guess. But the more I looked at it, then I was like, "Ohh." So, it just took me a little longer.

Saul described his experience of progressing through the TE as "having more steps to follow" for the tasks that I was giving them, which I am interpreting as "more things to consider" rather than "steps." There wasn't a step-by-step approach that I was encouraging the students to follow. It is likely that Saul was having difficulty managing all of the different concepts at one time and thus needed more time to reflect on what he knows and how that could help him solve the problem. In contrast, this idea of using what they know to solve the problem is something that I stressed often with the research participants. Saul's free-response answer about his least favorite aspect of participating in this study addresses this idea of managing all of the different concepts at once:

There wasn't anything really bad. The only thing I would say is it was sometimes a bit hard to remember at first exactly what we learned in the previous meeting since it was a week apart, but after a little refresher it all came back.

After the final session, Saul's reflection on his whole experience alluded to the desired outcome with the push for computing/programming and computational thinking in

mathematics:

I would just say after this session especially, I could see how you have to come from a pretty [good] mathematical background to understand how the [programming] language kind of- not like specific parts of the [programming] language, but how to incorporate it into what you're doing.

Saul's perspective is in line with the goal of infusing programming into the mathematics

curricula; understanding how a given piece of mathematics fits together with other ideas is

one thing, having to construct an algorithm to solve a mathematical problem using Python

(or any other programming language) is a completely different concept. This requires

intimate knowledge of the mathematics and understanding the goal of the solution to a

mathematical problem. In terms of Saul's experience working collaboratively over Zoom,

he seemed to warm up to the idea and understood how it can be useful:

At first I was a little skeptical but now I see it is possible. Although I do favor working in person and talking with others personally, I do see how a virtual setting has some potential to be used more often for convenience.

In response to how his identity contributed to his success, Saul said:

I really enjoy math and learning new topics in math. I see a lot of similarities in programming with math as some of it is kind of like a practice/method. There's always some sort of problem and you must find the solution. However, sometimes there are different ways to solve it, and that's why I think I enjoyed it. Like others in the study, Saul's interest in mathematics helped him succeed in this study, as he was determined and made a concerted effort to understand the material, even if it took him longer than he was used to or longer than his peers. His positive sense of mathematical identity is reflected in the ATMLQ results in Figure 6.4 as he came into the study reporting strong positive mathematical confidence and interest and finished the study reporting almost no change.



Figure 6.4: Saul's Pre and Post ATMLQ Results

Lastly, Saul's interest in problem solving and investigating the output to understand why his ideas were initially incorrect led to a shift in 6 points from the pre-study questionnaire to the post-study questionnaire on the programming to learn mathematics block of questions. Overall, Saul experienced a positive shift of 10 points from his pre-study ATMLQ to the post-study ATMLQ, which shows a growth in his own sense of mathematical and computational confidence and interest.

Eugene

Eugene was by far the most vocal student in Group 2 in that when he talked he gave longer descriptions of his thought processes and tended to answer the question in as much detail as possible. However, Eugene also created space for the other students as he was typically faster to catch on to the material, but would give Leo or Saul an opportunity to answer the question first before he did. When the group was stuck, I would often go to Eugene to ask his thoughts to get us back on track, and if Eugene didn't know the answer, then I knew that we had to go back or clarify something else. I would describe Eugene's mathematical identity development as one that came into the study confident and remained stable in terms of his ability to do math and understand computers and programming. His mathematical identity was also reflected in his pride and his confidence to catch on quickly to new material. Eugene's affective experiences are characterized in Table 6.5.

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics	0		Quickly	Positive
Computers and Programming	-1	Confident in his ability to understand material	corrected his own errors or and had a positive outlook	learn new mathematics
Programming to Learn Math	+2	quickly	in the value of mistakes	Enjoyed explaining his ideas to his peers

Table 6.5: Eugene's Affective Characterization

What was great about Eugene was that he was always excited and brought a positive attitude to learning new material. Even after the first session, which was overwhelming for a lot of students, Eugene was ready to move on and get into the more complicated topics:

I had some experience with coding but I've never worked with Python and I'm definitely enjoying it. It seems really really cool and interesting, but I'm excited to get into more discrete math as well. Excited to get to that part.

In the third session, Eugene really started to showcase his confidence and understanding of the programming material and was a great source of knowledge for the other students to work through the material. For example, when I introduced For Loops, Eugene was able to catch on quickly and help explain what was happening to the other students. In one instance, Saul and Leo were confused about what the 'for x in A' was doing, specifically what the function of the 'x' was in the For Loop. In the following exchange, Eugene was explaining how he was thinking about it to the other students:

Eugene: It's a placeholder, right. Yeah it's a placeholder that runs through whatever you're trying to run it through, in this case it's set A right? So I guess that little 'x' is going to be running through each element in that set and running through that code and picking out elements in the set. Then it's going to print- in the line that's in the bottom, which is a print. So whatever it picks it's going to print, and whatever picks it's going to print, so it keeps going back. It's like a little guy that runs around. That's how I picture 'x' [laughs]. And I guess it can be anything, 'element,' 'character,' 'letter.' **Leo**: Oh so it's basically an arbitrary value for any of the elements? **Eugene**: Mm hmm. **Leo**: I see it now, so 'x' and 'element' are the same thing. You can put like, 'for leaf,' or 'for dog.'

In this instance, Eugene was able to explain the function of the variable 'x' that served as the object that pulled the elements from the set and ran through the For Loop. Below I provide the code that we were looking at, in which I used Eugene's idea of using 'element'

instead of 'x:'

With the above code, the output will produce each element on its own line (in random order) and if the element is 'e' or the element is 'o,' then the element is added to set D and the set is printed after the element 'e' or 'o' with the set D printed one final time outside of the For Loop as the last line of the code. One potential output is provided below:

h t s y r o {'o'} e {'o', 'e'} {'o', 'e'}

I highlight the above code to preface another example of Eugene explaining his thought

process:

I would say that in this case it's kind of going to show you how the computer is thinking at the moment. Because it's going to print every element. Running the For [Loop] in this case, it's going to do all the letters, right. Except the ones that are repeated, it's only going to do them once. So, it's going to go through that and it's going to show you how it's thinking and once it runs into an 'e' or an 'o,' I guess it won't show you this, but it's going to add it into the element- or into the set of D. It's just showing you how it's "thinking."

Before running the code, I asked the students to consider Eugene's comments and think

about what the output will look like. After running the code, I asked the students to reflect

on the output and Eugene's comment highlights his confidence in his understanding:

Yeah it ran exactly how I was thinking how it was going to go, once you added the 'print D' inside the loop. Because- yeah, yeah, yeah, like I said, it's showing you how it's doing the process of adding those specific elements into the set.

Eugene's ability to catch on quickly likely stemmed from his confidence and interest in

computers and programming, which was high before the study began and remained high

once the study concluded. His results from the ATMLQ are presented in Figure 6.5 below.



Figure 6.5: Eugene's Pre and Post ATMLQ Results

As for Eugene's perspective on making mistakes in this study, his reflection after Session 2 sums up his approach to the material and solving the difficult problems, "It's just reverse engineering it. Because you know [laughs], it's a lot of trial and error in this. Mistakes after mistakes, but I guess that's how you learn." In his reflection after the third session, Eugene commented on how the tasks were getting more involved:

I feel like in today's session we learned a little more. It was a little different because we kind of used the previous knowledge and applied it to what we were doing in the moment. Before we were just learning new stuff individually. And this time, for example, we were running [loops] like For an If, and inside those [loops], we were using propositions like 'and' and 'or.' So, we had to understand how those worked in order to understand how the function as a whole worked. And whether it would run or not.

Eugene's reflection shows how well he was able to articulate his understanding of how the different tools worked with one another in Python and how these tools were being used to

solve problems. He described his favorite aspect of participating in the study in the following way:

I was able to test ideas of how to solve a problem and realize that there were many different ways to achieve a solution. I could then compare my way to solve the problem with the ideas of the other participants, then combining ideas we could eventually conclude in a more efficient way to solve the problem.

With his favorite part being the ability to solve a problem in multiple ways, it makes sense that his least favorite part about the study was that he did not fully have the opportunity to write his own code, "Not being able to write the code myself, gave me a sense of helplessness and prevented my ideas to be executed correctly, due to not being able to articulate the process in the best way." The reason why I shared my screen instead of asking the students to write their own code was due to the pilot study in which the students were asked to write and run their own code. Given how long it took during the pilot study to get through the material, and the logistical complications of monitoring what each student was writing, I knew that it wasn't feasible for me to get through the material that I wanted to get through in five one-hour sessions during the main study. I will discuss more on this point in the final chapter. As for Eugene's experience working virtually with other students he said, "I enjoyed seeing other's ideas and thoughts on a specific problem. By listening to what other participants said I could usually understand and find a solution to the problem." Lastly, as to how his identity helped him succeed in the study, he said, "I think that my ability to understand new topics rather quickly helped me understand the concepts of sets and how they functioned with each other." Indeed, Eugene was able to understand the material quickly, but he also allowed space for others to participate. He

used his partners' ideas to help shape his understanding, which ultimately resulted in Eugene being able to explain to his peers how something worked, in case they were stuck. *Leo*

At the beginning of the study, Leo participated less and did not seem as confident with his answers compared to his partners. As mentioned before, he struggled with poor internet connection which could partly be the reason for his comparatively reduced level of engagement compared to his two peers. However, I see Leo's mathematical identity as a story of success, as one of a growing sense of confidence and interest as the TE sessions progressed. At the beginning of the study, he reported the third lowest score of mathematical confidence on the ATMLQ with a score of five, where the average of all 10 participants on the mathematics block of questions was 8.3. By the end of the study, Leo was coming up with his own solution methods to solve problems, speaking up when he needed clarification and participating just as much, if not more than his peers. A summary of his affective characterization is provided in Table 6.6.

Table 6.6: Leo's Affective Characterization

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics	+3		A strad quastions	
Computers and Programming	+2	Growing sense of mathematical	Asked questions when he needed clarification which led to	Positive attitude
Programming to Learn Math	+1	confidence	valuable discussions	Determined

Leo's reflections after the first two sessions stood out to me after looking back at the data that support my conclusion that Leo started to build his confidence from day one. In both reflections, he mentions the pre-study mathematical content survey and how he didn't understand the material at the time, but he felt that he now had a better sense of what was happening. In the first reflection he talked about his expectations coming into the study:

I was expecting some of the coding, and I didn't really know much of it, so I was expecting that we were going to go over a little bit of the basics. But yeah, the first quiz I did in the beginning, there were some questions in there that had sets inside of sets, that's kind of where I got confused a little bit. So [today's session] explained a little bit more of how that works.

After the second session, Leo discussed specifically what he learned:

I guess the most important thing that I learned was the parentheses - the internal parentheses [of a logical statement]. That makes a lot more sense. Going back to the first quiz that we took in the beginning, or the survey, there were internal parentheses in there, and I was kind of confused, but now I get it more. I'm not sure exactly if that's what it was about but I guess that's where it can function.

In both instances, Leo is comparing how he felt when taking the survey (confused), to how he was feeling after the session took place, which was a sense of more understanding. Note that Leo did not say that he completely understands the material, but there is a sense of growth and progress in his reflections as with the first session he says the activities "explained a little bit more" and in the second session, "but now I get it more."

Leo often found himself slightly behind his partners in terms of how quickly Saul and Eugene got a grasp of the material, but one of Leo's greatest strengths throughout the study was his comfort in asking questions for clarification. For example, during the fourth session we were discussing how to find the union of three sets, by using For Loops to add elements to a new set. One of the students wanted to use 'D.add(B)' to add the set B to set D, but the '.add()' function does not allow one to add mutable objects to a set. We found that if we make B a frozen set (making it immutable), then we can. Leo stepped in at this point and asked for clarification:

Leo: I have a question. So we added set B, I get that part. But how come before, when it wasn't a frozen set, it still worked? When you do the For Loop for set A? I don't get what you guys are doing right now.
Interviewer: What's different about set D, as compared to what we did earlier where we did 'for x in A?'
Leo: You don't have to do another For Loop?
Interviewer: We don't have to do another For Loop, but also let's take a look at the length of set D, the length is it's being reported as 13.
Leo: Oh it's not adding the actual- it's just like adding one. Like set A would just be value one.

This exchange is a good example of how one of Leo's questions led to clarification for his own learning process, which might have been beneficial for the other students as well. Leo was very direct about not understanding the material when he did not understand something. I take this as his desire to want to learn the material rather than being okay with not understanding 100% of what we were covering.

In his final reflection of the study, Leo seemed to enjoy himself so much that he asked if I had any other studies that he could be a part of:

It was pretty interesting. If you have any extra- any other studies, you know, [it would be] pretty cool to do some more. I thought it was pretty cool, learning and seeing how things that you wouldn't think [of] would affect certain stuff. I like learning how something can affect other things.

During the study, I would not have guessed that Leo would want to participate in future studies given how much harder he was working compared to his peers by asking questions and needing clarification. Being vocal about being confused is not always easy in front of one's peers, so I commend Leo for his growth in participation and showing up to each session with a desire to learn and embrace the uncomfortable nature of not knowing the answer. Leo's scores from the ATMLQ are presented below in Figure 6.6.



Figure 6.6: Leo's Pre and Post ATMLQ Results

Leo's scores reveal a growth in each block of questions, with the greatest positive difference in the mathematics block of questions of three points. In his free-response answers, Leo said that his favorite part of participating in the study was:

Collaborating with my teammates on how to solve certain problems. In the last 2 sessions, we applied our previous gained knowledge to try to solve specific questions. The application of code to these problems such as the for loop was very interesting.

It stands out to me that Leo is once again portraying a growth or progression in his knowledge in that they used what they learned in the first three sessions to solve tasks in the last two sessions. This portrays a progression and development that seems to be a sense of pride for Leo in being able to point to something that he learned. As for his least favorite aspect of participating in the study Leo said, "There really wasn't anything bad about the study. I would say that going more into what set theory is and exploring it more could be done. Other than that everything else was great." It is encouraging that Leo's least favorite part of the study was that he wanted to learn more than he did about set theory. This is additional evidence that he enjoyed himself in this study. His reflection on virtual participation alluded to his poor internet connection, "I feel like the zoom app is an ok virtual setting. The connection at times would drop for a few seconds but [Zoom] allows us to collaborate without being there in person." I do think that Leo would have significantly benefited from attending in person compared to being virtual due to his connection. It is impossible to know exactly what he missed, but given his mathematical content survey results (see Chapter 5) he could have missed crucial pieces of information, which likely led to Leo needing to ask more clarifying questions than his peers. Lastly, in terms of the aspects of his identity that led to his success, Leo said, "I would say my willingness to ask questions allowed me to succeed. If I was confused I would ask questions. I was also interested in the subject so that helped me too." Like other participants in the study, Leo referenced his interest in mathematics, but he also refers to his willingness to ask questions. Leo's questions not only helped him succeed in this study, but his questions led to discussions with his peers (specifically giving Eugene an opportunity to explain his thought processes) and ultimately helped his peers' learning processes as well.

Group 3

There were two students in Group 3, Delia and Juliana. Delia identified as a Middle Eastern or North African woman, and Juliana identified as a Hispanic or Latinx woman as well as a first-generation college student. Juliana also identified as a current or former English language learner. I intentionally chose this grouping because both students are

263

women and they both identify as belonging to historically marginalized groups. Although Delia and Juliana kept their videos turned on throughout the entire study, I had the most trouble fostering productive and prolonged discussions with the students in Group 3 compared to the other groups in the study, for which I have two hypotheses. The first is that Delia and Juliana were naturally just not as talkative as compared to the students in Groups 1 and 2, so it took me longer to understand what worked in terms of inquiring about their thought process. The strategy that seemed to work best for Delia and Juliana was to allow for more wait time as well as offer hypothetical student reasoning to get a conversation started. I figured that since they did not have a third partner, there was more room for me to step in and offer some hypothetical ways of reasoning that I often took from what I learned working with the students in Groups 1 and 2. The second hypothesis that could potentially explain their levels of engagement was that each student came into the study with lower levels of confidence. For Delia, she came into the study with the second lowest score on the mathematics block of questions and Juliana came into the study tied for second lowest score in the computers block of questions and the lowest score on the programming to learn math block of questions. The students not being confident at the beginning of the study could explain why it was more difficult to engage in discussions with the students, especially at the beginning.

Delia

Delia had the greatest negative shift from her pre-study ATMLQ score to her poststudy ATMLQ score with a total difference of -14 points. I was quite surprised to see this result because Delia was a strong student throughout the study, she seemed confident in

264

her answers, and I would often rely on Delia to rephrase her own thinking or rephrase Juliana's thinking in order to start a discussion. However, as referenced in Chapter 2, Cribs et al. (2015) found that competence does not have a direct effect on one's mathematical identity. Instead, interest and external recognition are the two main factors that have a direct effect on one's mathematical identity. This finding seems to describe Delia's experience as there never seemed to be a point that Delia was very excited about the material, she remained fairly stoic and matter of fact throughout the entire study. Table 6.7 summarizes her affective experiences.

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics	-6	Confident in		
Computers and Programming	-8	her answers Waning sense of confidence as the TE progressed	Vocal about not understanding	Matter of fact Neutral about the tasks
Programming to Learn Math	+0			

Table 6.7: Delia's Affective Characterization

As mentioned, Delia had the greatest negative shift out of all the students in the study. In the mathematics block of questions there was a negative shift of six points and in the computers and programming block of questions there was a negative shift of eight points. The one question that stood out to me on the mathematics block of questions was the following: "I have never felt myself able to learn mathematics." On the pre-study ATMLQ, Delia's answer to this question was "Strongly Disagree" while her answer on the poststudy ATMLQ was "Agree." Her response to this question alone was responsible for a negative shift of three points. For reference, there was only one other instance in which a student's pre/post answers to a question resulted in a negative shift of three or more points. As for the qualitative results, there is not much either during the study, or in her freeresponse questions that could really explain her negative shift in her confidence and interest in mathematics. While she may not have been as excited about the material as some of the other students from other groups, her lack of excitement could just be a part of her personality and does not necessarily mean that she was having a bad time.

The only piece of evidence that may explain her negative shift in confidence could be in the comparison between her reflections after the first two sessions and after the third. After the first two sessions Delia did not think that the material was too difficult, but the third session was quite frustrating for her and left her confused going into the fourth session. When asked whether the material that we covered in the first session is what she expected, Delia said:

I expected it because you told us in the beginning that you are going to teach us about Python and how to use Python with math, so I would think you would have to show us the beginnings of Python to help us understand how to apply it to the math. And I don't think it was that bad.

The one aspect of this quote that is impossible to capture through text is the tone in which she said it. Throughout the entire study she was very pragmatic, and if the reader can imagine it, she spoke in a very direct way as if her reasoning was obvious. This tone did not come off as condescending, but instead rather steadfast. That is, there were not many
"um"s or "like"s in her speech, she just seemed very sure in herself and her reasoning. As for her reflection after the second session, she said, "I had fun, I didn't think it was too difficult. The only part I thought was difficult was when we're trying to figure out where the parentheses go for when you have multiple proposition statements." Thus, after the first two sessions she did not think that the material was too difficult. However, the third session is when I introduced For Loops to the students, and this caused a lot of difficulty for Delia and Juliana. In fact, after the third session I decided to switch from Google Colab to the IDE on my own computer so that I can debug a For Loop and show the step-by-step process of the loop grabbing an element from an iterable object and stepping through the loop line by line. One For Loop that we spent a lot of time discussing was the following:

Below is an example of what an output might look like from the above For Loop:

y h e {'e'} s t {'t', 'e'} r o

- {'o', 't', 'e'}
- {'o', 't', 'e'}

Delia's comment is just one example of her frustration from the third session, "I honestly can't come up with any... I don't know why. I don't know why one has just one 'o,' the next that has 'o' and then 't' and then so on and so forth. I'm quite confused." When we ran the For Loop during the third session, the element 'o' was grabbed first which is why 'o' was the only element in the set D when it was printed. Her reflection after the third session shows a shift in how she felt compared to after the first two sessions:

I thought it was harder than the last couple of sessions. And where Juliana understood what was going on, I only understood it up to the 'o.' I still don't understand why 'o' and 't' would appear in the same set and then 'o,' 't,' and 'e' when you're just looking at that one element, which in this case is 't.'

After Delia made this comment in her final reflection I went back to try and explain the process of the For Loop in the last few minutes of the session, but there definitely was no closure to the third session. In the fourth session we used the IDE which definitely helped both Delia and Juliana solidify their understanding of For Loops, but I worry that perhaps the third session was just too challenging and frustrating for Delia. In her final reflection, Delia said, "I had a lot of fun, to be honest. And going into a [computer science] class next semester, I thought this was quite interesting. Delia's ATMLQ results are presented in Figure 6.7.



Figure 6.7: Delia's Pre and Post ATMLQ Results

Unfortunately, the stand-out result is the negative shift of six units for the mathematics block of questions, dropping Delia's confidence and interest in mathematics below zero. Delia's pre-study score on the computers and programming block of questions was the second highest out of all the participants in the study, and her post-study score was still above the average post-study score of 10.2.

True to her matter-of-fact nature (or maybe just her desire to be done participating in my study), her answers to the free-response questions were quite brief. Her favorite part of participating in the study was that she "liked starting to learn about one section of python and how that can be used to look at math problems," and her least favorite aspect of the study was that she "didn't like the use of the jamboards because I find them a hassle to use." While she found the jamboards to be a hassle, she did say that "Zoom can show how easy and productive working collaboratively can be with other people." Lastly, with respect to aspects of her identity that led to her success, she said, "I don't believe my identity had anything to do with my success in this study." Given the data, I would unfortunately say that Delia had an overall negative experience participating in this study. Even after conducting the TE with Groups 1 and 2, there were aspects of the instructional sequence that still needed to be ironed out, which likely led to some degree of frustration and lack of interest in mathematics.

Juliana

Juliana had the second greatest positive shift from her pre-study ATMLQ score to her post-study ATMLQ score with a total difference of +21 points. I would not have predicted this result as Juliana was often hesitant about her reasoning and not always confident in her answers. However, as mentioned in Delia's section, competence alone does not necessarily predict one's sense of a positive mathematical identity. One possible contributing factor to Juliana's positive shift was the fact that she never seemed worried about getting the wrong answer. In fact, Juliana was often the first to throw out an idea, and I either built off that idea or asked Delia to rephrase in her own words what her interpretation was of Juliana's reasoning. A summary of Juliana's affective experiences is presented in Table 6.8 below.

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics	+12		Okay with	Overwhelmed in the beginning
Computers and Programming	+1	Growing sense of self confidence as the TE progressed	Important that her wrong answers were not evaluated in the moment	Moments of frustration
Programming to Learn Math	+8			at the end and felt supported by Delia

There were two questions in the mathematics block of questions on which Juliana reported a complete shift from the most negative disposition to the most positive disposition. The first was in response to the following question: "I don't understand how some people seem to enjoy spending so much time on mathematics problems." On the pre-test questionnaire Juliana answered "Strongly Agree" and on the post-test she answered "Strongly Disagree" which resulted in a four point positive shift. The second question was the following: "I have never been very excited about mathematics." Similar to the previous question, Juliana's answers resulted in a four point positive shift. For reference, no other student's answers from the pre-test to the post-test resulted in a four point shift on a single question. In the programming to learn mathematics block of questions, there were four questions that Juliana answered that resulted in a positive three point shift:

 Using computing power/programming for the calculations makes it easier for me to do more realistic applications

- I like the idea of exploring mathematical methods and ideas using programming.
- 3. I want to get better at using computers to help me with mathematics.
- 4. Having programming to do routine work makes me more likely to try different methods and approaches.

Out of all of the students, there were only eight responses from pre-test to post-test that resulted in a positive shift of three or more points. Juliana accounted for six of these eight responses.

As previously noted, Juliana was not the most confident, at least initially. After the first session she mentioned that the material was pretty overwhelming for her, "Yeah it wasn't that bad, but I guess being introduced to new stuff we were- I guess for me it was totally overwhelming. But I guess if we try practicing again I feel like we'll get better at it." During the second session it seemed that she was slowly getting more comfortable, but she was experiencing some frustration with the tasks, as well as with my questioning. For example, after questioning her about what the output would be for a compound propositional statement she said, "Are we supposed to give you an answer right away? Or can we get a little time to think about it?" I took the hint and gave Juliana and Delia a couple of minutes to work out the solution on their own, which worked really well in terms of fostering a productive discussion with the students. After the second session Juliana's reflection was that she "thought it was pretty easy. When we had several 'or's and 'and's together in one sentence, but I thought it got harder when 's' and 't' weren't defined- or they were unknown." From her perspective, things were getting better, but still

challenging. This trend continued in the third session as she described her experience as "kind of hard because we were introduced to new things. But after you explained the For Loop statements, I guess it was pretty easy." Juliana's final reflection shows the growth that she experienced over the course of the TE:

I honestly found working with Python very interesting because you're able to use numbers but also words and come up with an answer using Python in those words and I thought it was really cool honestly.

The change in her response from the first session of being "totally overwhelming" to the last session being "really cool," is evidence that Juliana experienced a positive growth in her mathematical identity, which is supported by her ATMLQ results shown in Figure 6.8.



Figure 6.8: Juliana's Pre and Post ATMLQ Results

The greatest difference from the pre-study questionnaire to the post-study questionnaire was in the mathematics block of questions where there was a difference of 12 points, and the programming to learn mathematics block of questions showed a positive difference of

eight points. My hypothesis as to why there was not a dramatic shift in the computers and programming block of questions is that those questions did not directly ask about mathematics, which is where I see Juliana's shift in her perspective of herself as a learner and doer of mathematics using programming.

As for Juliana's answers to the free response questions at the end of the study, her favorite aspect of participating in the study was the following:

My favorite part about participating in this study was applying mathematics to python. Although I feel like I used more of my thinking than actual calculations, the math was still there which made it enjoyable although confusing at times.

Her reflection on her favorite aspect of the study addresses my last hypothesis, that Juliana was more excited and interested in the mathematics, rather than just working with computers. Her least favorite aspect of participating in the study was, "being asked to solve a question when I did not know how to answer haha." While Juliana's response is light-hearted, this idea of "cold calling" on students when they don't know the answer has come up before with Haven's free-response answer, and is something that I will have to think about moving forward with my research (more in Chapter 7). Juliana's answer to the question about her participation in a small group over Zoom was one that had not come up before and important to consider, "I think I have gotten used to it, I find it easy and less intimidating than it would be working face to face." I am not entirely sure why being in person would be more intimidating, but perhaps the comfort of being at home has something to do with it. In terms of aspects of her identity that helped her succeed, she said the following:

274

I feel like my knowledge and my ambition to learn some aspects of python contributed to my success in this study. Also it was very important that Antonio never told me my answer was wrong, rather we worked from my answer to generate a more sophisticated answer. My partner Delia also contributed to my success in this study because many times she agreed with my answer and she never put me down if my answer was wrong. Overall, I felt confident speaking up even if I knew my answer would be wrong.

Once again referencing Cribbs et al.'s (2015) findings related to mathematical identity, Juliana commented that Delia contributed to her success because Delia agreed with her answers (recognition from others). Additionally, she said that I never said that her answer was wrong, which likely helped her maintain a positive sense of mathematical identity as a contributor to more advanced mathematical ideas. Lastly, her strong positive shift in mathematical identity is likely a reason why she was able to maintain her confidence even if she knew that her answer was going to be wrong.

Group 4

There were two students in Group 4, Alonso and Julian. Alonso identified as Hispanic or Latinx and as white. Julian identified as Southeast Asian. Both students also identified as men, and both were Mechanical Engineering majors at the time of data collection. Both students kept their cameras turned on throughout the duration of the study. As mentioned in Chapter 5, Group 4 made it the furthest in the instructional sequence and I attribute their success to their interest in the material as well as my being able to streamline the instructional tasks. From the very beginning, both students seemed to feel comfortable participating and sharing their thoughts, which led to productive conversations and opportunities for the students to explain their reasoning. For example, in some cases the students would not agree with one another's reasoning, and their answers compared to the actual output generated a discussion about aspects of their answers that were right and how their thinking has changed once seeing the output. I present Julian's results first, then conclude with Alonso's.

Julian

Julian's mathematical identity development can be described as a process of continual growth in his confidence and interest as the study progressed. In the beginning he seemed unsure in his reasoning, which could be attributed to nerves in being a participant in a research study. Also, Julian was usually behind Alonso in terms of how quickly they understood the material. However, as the study progressed, he became more and more comfortable describing his thought processes and was not shy about vocalizing his thought processes out loud. This did lead to some difficulty in my interpretation of Julian's reasoning as he would often waver back and forth between multiple ideas, but this also provided an opportunity to ask Alonso to rephrase what he understood of Julian's reasoning. Julian's affective experiences are summarized in Table 6.9.

Tuble 0.7. Julian S Aneenve Characterization

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics	+2	Growing sense of confidence as the TE progressed	Comfortable acknowledging that he was confused Quick to correct or go back to where he went wrong in his original thought process	Positive attitude to
Computers and Programming	+4			Нарру
Programming to Learn Math	+0			Problem solving was more interesting

Julian did also struggle with all the vocabulary that we were using (e.g., using "variables" instead of "elements" or "propositions"), but it did not get in the way or stop him from explaining his thought processes. Additionally, when Julian answered a question incorrectly, he consistently did a good job of explaining why his original thought process was incorrect or what he learned after seeing the correct answer:

So, in the beginning I was kind of confused [between] regular text and the actual boolean value. I was just mixing them up. But I realized that now, it was actually the true or false statements that are set. I was kind of confused a little bit with the variables and everything, but now I understand that it's really just simplifying the longer code into just variables, which then you just put into print and it just runs it along. So, for 'p,' it will just be 'dog in setA' which is obviously true because in setA 'dog' is there.

In this excerpt Julian was referring back to some of the propositions that we were exploring and what the difference was between running 'print(p)' and 'print(dog in setA),' which were the same thing. The important aspect of his reflection is that he was able to identify specifically what he was confused about and how he now understood the material. As mentioned, Julian and Alonso were moving through the material quickly, which Julian alludes to in his reflection after the first session in terms of his expectation for what his participation in the study would look like:

It's not really what I expected. I didn't really know what we were getting into. I didn't know set [theory] and I don't really know a lot about Python. So, seeing how you can integrate it and how you can use different codes to read the sets and have True and False values was pretty interesting. I also like how we had the 'and' and 'or' and how you can do different ways of reading the different texts of the set and finding values into values. Like how we did the variables and we did the 'and' and 'or.' It kept getting more complicated as we went on.

There was a lot that we covered in the first session, which is likely why Julian said that he "didn't really know what we were getting into." He also mentioned that the material "kept getting more complicated" which shows that he experienced some struggle with the material, but was still interested. An additional strength of Julian's was that he let me know when he was confused. For example, in the second session I introduced For Loops to the students and asked for them to reflect on the different outputs that we were getting. Julian was hesitant in saying that he was confused with the process, but he still let me know:

I'm kind of thinking about it in a different way. I'm kind of a little bit confused. Not confused, but kind of...[long pause]...a little bit confused in how it still works. Because of the print thing. So I'm not quite sure how it actually works.

By saying out loud that he was confused, we were able to have a discussion about what was confusing and go back to his point of confusion after working out more examples and debugging the For Loops. A third strength of Julian's was his comfort in disagreeing with Alonso when he had a different idea of what the output would be given a certain code block. For example, in the second session Julian rephrased Alonso's reasoning and described how his reasoning was slightly different:

[Alonso] said that it would be printing the elements, one and then add another one until it has the same elements as A. I was thinking that it would just be printing each single element, along the lines. So I think my hypothesis is kind of different from his.

The two students disagreeing with each other was not uncommon, and led to many

productive conversations that gave me a chance to inquire into their thinking.

Julian's reflection after the third session shows a slight shift in his interest

compared to his first reflection (we ran out of time to do reflections after the second

session):

I also think it's pretty fun. I like going through all the processes. I think, especially this session, finding all the unique approaches, especially with finding all the variables and everything and having everything being used. Like the For [Loop] and having to find all that stuff and then just interpreting how to actually use the code to then add things or like equal things, all that stuff. I think it's very you know- I like how open minded this concept is.

By "open minded," perhaps Julian meant to say "open ended." After this reflection I asked

him whether the material was more or less difficult than after the first two sessions:

I think it's maybe less difficult because we're kind of using more of what we already know, just trying to find the pathway. Where before we're kind of like learning new things, and it was kind of more finding out more information and how to use it, rather than here we're just interpreting- we're also trying to find how to use it, but more interpreting how we can implement our thoughts with the code.

For Julian, he preferred problem solving and finding a solution path, which he thought was

less difficult than learning about all of the syntax and different functions in Python. As his

reflection after Session 4 demonstrates, his perspective remained the same in terms of his

interest in problem solving:

I'm going to be honest, I don't really remember much from last week [laughs], but I think today was definitely more fun. I think it was putting it together and just finding how everything fit mostly with the subsets and just putting together those different statements. Everything was pretty nice.

In his reflection with the group after the fifth and final session, he summarized his

experience participating in the study:

I thought this was pretty fun. You know, I definitely think that the beginning was basically just learning vocabulary and all the basic stuff like all the functions and what elements were and everything. And that was kind of just learning the steps. And then we got into more of problem solving and just putting everything together and just basically going deeper into what the idea of coding and all these processes were. Then today we didn't use Python, it was all mostly just an idea of how it would work and how we would need to go through this process of proving this thing. But yeah, I think it was definitely a nice experience, a first time for me, you know seriously coding, and just learning this.

Julian saw the first couple of sessions as basic, and found more interest as the study

progressed as they were required to do more problem solving with the tools at their

disposal. This could explain some of Julian's ATMLQ results which are presented in

Figure 6.9 below.



Figure 6.9: Julian's Pre and Post ATMLQ Results

Julian maintained a strong positive mathematical identity but experienced a significant growth in his interest and confidence with computers. I would have expected to see a similar growth in the programming to learn mathematics block of questions, but it could be that Julian found more interest in the idea of working with computer science and programming instead of mathematics. As for Julian's answers to the free-response questions, his favorite aspect of participating in the study was, "Creating new lines of code to solve certain coding problems" and his least favorite aspect was "Learning through the initial phase of vocab, code phrases, and technical things." Given that his least favorite part of the study was the introductory sessions, I am confident in saying that his interest, and thus mathematical identity, grew positively as the study progressed. As for working collaboratively over Zoom, Julian said, "I think that a collaborative setting allowed me to look at certain problems in a different perspective and allowed me to have a greater insight on certain problems and code." As mentioned, the discussions between Julian and Alonso were really productive for each to better understand the material as they were able to describe their thought processes in different ways. Lastly, in terms of the aspects of his identity that helped him succeed he said, "I think being initially interested in coding and having coded a little previously on other languages helped me grasp the concept a little better." As with some of the other students, Julian referenced his interest as a main factor that helped him succeed. Additionally, he referenced some experience with coding, which could be a reason why the coding aspect of this study was more salient for Julian than the mathematics.

Alonso

Alonso's mathematical identity development was one of maintaining a strong positive sense of mathematical identity, especially with respect to computers and programming to learn mathematics. Alonso came into the study with the highest score for the programming to learn mathematics block of questions on the pre-study ATMLQ. He also had higher-than average scores for the mathematics and computers and programming blocks of questions on the ATMLQ. Throughout the study, Alonso was quicker to understand the material than Julian, but he did not dominate the space and provide his answer before Julian had an opportunity to speak. There were some moments where I could see that Alonso had an answer and was waiting for Julian to finish thinking. Additionally, if Julian had an answer that Alonso knew wasn't right, he would disagree with Julian, but do so in a way that would foster a back-and-forth discussion. For example, in one instance Alonso said, "I am thinking slightly differently," and followed with his explanation. This allowed room for me to ask Julian what his thoughts were on Alonso's reasoning which often led to an opportunity for both students to rephrase their thought processes and come to a common understanding. Overall, Alonso's ATMLQ results show a total decrease of three points, which was slightly surprising because he showed so much confidence, interest and excitement throughout the study. Even though his total score was negative three points, it is difficult to say that Alonso had a negative experience participating in this study, especially since there were no questions from Alonso's ATMLQ responses that resulted in a negative shift of greater than two points. Alonso's summary of affective experiences is presented in Table 6.10 below.

	Pre/Post Survey Change	Sense of Self- Efficacy	Response to Errors / Difficulty of Tasks	Beliefs, Attitudes, Emotions
Mathematics	+1		Slightly hard on	
Computers and Programming	-3	Confident in his answers	himself for getting a question wrong Enjoyed working on difficult tasks	Positive attitude Serious
Programming to Learn Math	-1			

Table 6.10: Alonso's Affective Characterization

Alonso was one of the most serious participants out of all of the students as he almost always had his eyebrows scrunched together in concentration. I could tell that he was really trying hard to understand 100% of the material at all times. He would even ask clarifying questions about the syntax or functionality of some functions in Python that were not the direct focus of the material, he just wanted to know everything that we were doing and why we were doing it. With his seriousness also came a sense of pressure that he put on himself to have all of the right answers, and when he didn't he would sometimes be hard on himself for not knowing. For example, in one instance his guess did not match up with the output and he said the following:

I should have been able to predict this because I just realized we did the exact same thing when we had 'for element in A' and there the length was outputted. So I should have been able to guess that. And yeah, I was not expecting it to be random.

However, other times when he was not as confident in his answer he would still give his best guess which was really helpful moving forward and comparing his guess to the final output. One example of this was in the second session when I first introduced For Loops to the students and they were just giving their initial interpretations of what the For Loop might be doing.

I think there will be repeats because you're just printing the elements and not the length...I think. I'm not completely sure. I'm not sure how we would deal with that. So, my best guess is that it will go 's' dash dash dash, 'e' dash dash dash, 't' dash dash dash all the way through.

Like Julian, Alonso tended to give complete descriptions of his thought processes which demonstrated a level of confidence and comfort for both of the students that not many other participants in my study showed. In the previous section I discussed how I would often find myself inserting some ideas into the discussion with Delia and Juliana to foster more productive conversations, but with Julian and Alonso I rarely had to provide a similar level of support and encouragement to garner their engagement. After the first session, I asked Alonso and Julian to reflect on whether this was what they expected, and the following was what Alonso said:

I didn't know what to expect because I had no clue what set theory was. So, I can't say this is different from what I expected because I don't know, I was just completely clueless. I was kind of surprised that my philosophy course actually helped [laughs]. Didn't really expect to ever use that again. Yeah, I mean Python, I guess it just makes sense, you have to just understand what everything actually does.

During the first session Alonso commented that in his philosophy course they spent a small amount of time discussing logical operations and propositional statements, which is why he referenced his philosophy course in his reflection. He also had a very sensible outlook on understanding Python in terms of it making sense if you understand all of the individual components. However, as he soon realized, sometimes it isn't that easy. His reflections after the third and fourth sessions reflect this point. His reflection after the third session alluded to the difficulty, "I find this harder for sure. I'd say harder, but more fun. The other was easier, but less fun." Unlike Julian, Alonso thought that the material in the third session was more difficult than the first two sessions. However, he also thought that the material in the third session was more enjoyable. He echoed this idea in his reflection after the fourth session as well:

I found this time definitely more difficult for sure. I'm not sure if that's because I'm a little sleep deprived but for some reason that just wasn't clicking, the print statement thing. I don't know why. But yeah, I like seeing how this stuff is applied.

Alonso's comment about "the print statement thing" was referring back to a solution method that Julian had proposed and Alonso was not understanding why Julian suggested that we check the cardinality of two sets to check if they were equal (see Chapter 4). This one instance was one of the only times in our five sessions that Alonso was behind Julian in terms of their understanding of the material and, as evident in his reflection, it was weighing on him that he didn't understand. In his last reflection, after the final session, he commented that he enjoyed participating in the study, but it was difficult to keep track of everything that they were being asked to do.

I really enjoyed it. It was enjoyable to learn about this, I thought it was really interesting. The last few sessions were a lot more challenging. I guess I mean a lot more confusing. The first sessions were crystal clear and then it just became hard to keep track of everything that was going on. To keep track of all the variables. It was just a lot of stuff to remember.

Even though Alonso was one of the most competent students that participated in my study (see Chapter 5 pre/post content survey results), it seemed that there was a sense of being overwhelmed with all of the material. This, or his pressure on himself to get all the right answers, could explain the slight decrease in his overall score from the ATMLQ results, which are presented below in Figure 6.10.



Figure 6.10: Alonso's Pre and Post ATMLQ Results

Alonso scored above average in each category for the pre-study ATMLQ as well as the post-study ATMLQ. He also scored the highest in the programming to learn mathematics block of questions on the pre-study ATMLQ.

Even though Alonso did comment that with each session the material was getting more difficult, his answers to the free response questions reveal that he did enjoy his time participating in the study. He had the following to say about his favorite aspect of the study, "I really liked how much I learned in such a short amount of time. Both python and set theory were really interesting to learn about." As for his least favorite aspect, he said, "Nothing was bad." His reflection about the virtual setting was in line with Juliana's from the previous section:

In this study working virtually went really well. I felt like it might actually be better for this activity. In general working collaboratively with other people on zoom is more of a hassle and worse than in person settings. As the one that conducted the study, I also consider the virtual setting to have worked particularly well for this activity. In fact, it is not entirely clear to me if being in person would have greatly added to this study (more in Chapter 7). As to the factors of his identity that contributed to his success, Alonso said that, "All I needed to succeed was an interest in the subjects being taught. Everything was clear and manageable." As we have seen with the other students, interest is a main component for his perceived success. Lastly, after the study Alonso emailed me to follow up and said, "I really enjoyed learning python and set theory. You were a great teacher and I wish you luck with your study. You inspired me to continue learning python on my own time." His total shift between the two ATMLQ scores was negative, but he enjoyed his time participating in the study and found something that he wanted to pursue further, evidence of positive mathematical/computer science identity growth and interest.

All Students

Out of the ten students that participated in the study, seven reported a positive change in their overall confidence and interest with respect to the three categories on the ATMLQ. Additionally, using a paired sample t-test to compare the scores between the prestudy ATMLQ (mean = 25.4, standard deviation = 16.27) and the post-study ATMLQ (mean = 31.6, standard deviation = 8.13), the results indicate a statistically significant improvement (p = 0.0043 < 0.01). While it is not possible to generalize given the small sample size, comparing the scores of the pre-study ATMLQ and the post-study ATMLQ by gender reveals that the women's total score improved by 50 points and the men's score improved by 12 points. Given the different sized groups of the men (6) and women (4), it

288

was striking to see such a large difference for the women. Dividing the total difference for the women compared to the total difference for the men results in a shift of +12.5 for the "average woman" and a shift of +2 for the "average man." Due to the small sample size, the assumptions to run a two-sample t-test could not be met, but I do present the statistics of the two groups' normalized gains. The normalized gain represents the amount that a student (or groups of students in this case) improved compared to the maximum amount they could have improved. For example, on a 100-point test, a student that scores 90 on the pretest and 95 on the post test has a normalized gain of 50%. The same is true for a student that scores 50 on the pretest and 75 on the post test. For the ATMLO, the maximum score possible is 70 points (representing extreme confidence and interest in mathematics, computers and programming, and programming to learn mathematics). The average ATMLQ pre-test score for the women was 18.75 and the average ATMLQ pre-test score for the men was 29.83. The average ATMLQ post-test score for the women was 31.25 and the average ATMLO post-test score for the men was 31.83. As a group, the women came into the study with lower levels of confidence and interest compared to the men. However, the two average post-test scores are roughly the same. As for the normalized gains, the women had a normalized gain of 24.39% and the men had a normalized gain of 4.98%. Given the large difference reported by gender, future investigation regarding the differential impact that a similar learning situation may have will be necessary. All the point shifts from pre-test to post-test for the students in the study are presented below in Table 6.11.

Haven	+32	
Judith	+11	
Palmer	-8	
Saul	+10	
Leo	+6	
Eugene	+1	
Juliana	+21	
Delia	-14	
Julian	+6	
Alonso	-3	

 Table 6.11: Total Scores Across all Three Categories

As for the three students whose total scores were negative, I consider only Delia's case to be truly representative of her experience. For Palmer, his pre-study ATMLQ score on the computers and programming section was so high that there wasn't any potential for positive growth on that section. That is, even though his score on the post-study ATMLQ was the highest out of all the students, there was still a difference of negative 10 points on that block of questions. For Alonso, he was the most proficient student in the group, enjoyed his time during the study, and maintained fairly stable scores across all three blocks of questions from the pre-study ATMLQ to the post-study ATMLQ. It is not impossible that Alonso experienced a negative shift in his mathematical identity, but I consider it highly unlikely due to his explicit interest in the material and his positive sense of self-efficacy. In Delia's case, she came into the study with the second lowest level of mathematical confidence and interest, which was surprising given her confidence and proficiency during the TE sessions. There isn't one factor that I can attribute her negative experience to, although she was very stoic which I sometimes interpreted as her being bored. There is a real possibility that the material was too easy for Delia as she was often correct in her reasoning, or she had different expectations for what the study would be. This could explain her matter-of-factness and lack of engagement at times. Additionally, all the students except for Delia referenced some aspect of positive mathematical identity development in their answers to the free-response question about aspects of their identity that helped them succeed. For most of the students, it was an interest in mathematics or computer science. For others, it was the ambition or ability to learn new material. For Delia, she didn't believe that any aspect of her identity helped her succeed.

Chapter 7: Conclusion

The purpose of this teaching experiment was to address the following conjecture: Programming can not only be leveraged as a processing tool, but also serve as an experientially real context in which students will be able to connect mathematical logic and set theory that also positively influences their identities as mathematicians. In an era where machine learning, data science, and computer science more broadly, are becoming the most influential levers of change for our society, why are we still teaching mathematics as if computers don't exist? Of course, it is not literally true that all mathematics courses are taught without the use of computers, but the implementation of computer programming to teach and learn mathematics is not widespread and has considerable room for growth (Hickmott et al., 2018). I conducted this study to make progress in addressing this concern, to better understand the impact that computing and programming may have on students' learning of mathematics. This is particularly important as advances in technology are leading to STEM occupations with the highest rates of worker turnover and decline of old skills (Deming & Noray, 2018). I begin this chapter with a summary of my findings. I then highlight the implications of this work. I then present a discussion on the various limitations of my work. I follow by discussing areas of future research, and conclude the chapter with some final remarks.

Summary of Findings

The findings summarized in this section are divided by each research question, as each research question had its own unique focus. As a reminder, the first research question focused on the in-the-moment ways of reasoning by the students and how Python influenced these ways of reasoning. The second research question focused on what the students learned over the course of their actual learning trajectories. The third question addressed the students' affective experiences and changes in their mathematical identities.

Research Question 1

Using the instrumental approach as the analytical framework, which considered the confluence of an artifact (often a piece of technology) and the human mind, I identified six instrumented action schemes emerging from various ways of student reasoning. Before addressing each of the six schemes, it is important to note that the instrumental approach has two major strengths. The first is that through analysis, we can better understand how an artifact, typically a piece of technology, can be used to strengthen or enhance one's understanding of a mathematical idea. The second major strength is the opposite, understanding how a piece of technology may in fact hinder one's thoughts about a mathematical idea. The six schemes emerged as groups of two, grouped by three main mathematical concepts: (a) propositional statements, (b) set intersection, and (c) subsets.

The first two schemes, Impossible-to-Answer scheme and Flexible Propositional Statement scheme related to the students' conceptions of propositional statements and the truth value of "unknown" propositions. By an unknown proposition, what I intended was for the students to reason about *any* proposition that could take on either a True or False value. For example, in the case of the propositional statement '*s* or *t*,' there are four potential cases that represent two outcomes: (a) True, (b) True, (c) True, and (d) False. The only situation which results in a False outcome is the one in which both *s* and *t* are False. In Haven's case, her reasoning reflected a perspective that an unknown proposition was

one that no one could possibly know the answer to, such as "there are more wheels in the universe than there are doors." So, a propositional statement such as 's or t,' a propositional statement with two unknown propositions, cannot be evaluated because the truth values of s and t are undeterminable. This is what I referred to as an Impossible-to-Answer scheme. The Flexible Proposition scheme describes an understanding that any proposition can take on either a True or False value at the time of evaluation. For Adeline and Kristal this meant that the propositional statement 's and t' would produce two outputs depending on the truth value of the propositions themselves. In both cases, Python influenced the students' reasoning in that the students understood that when evaluating a print statement in Python, one should receive a single output. For Haven, given a scenario with unanswerable propositions, it is impossible to determine a single output of a propositional statement such as 's or t.' In contrast, for Adeline and Kristal, they thought that Python must make a choice between True or False with the probability of Python choosing False 75% of the time when evaluating the statement 's and t.' While both student conceptions are incorrect, the use of Python provided an opportunity for the students to face these misconceptions and engage in a discussion about the functionality of propositions and logical operators.

The Filter Every Element scheme and Monitor Change in the Cardinality scheme related to finding set intersections. I presented Group 2's work related to the Filter Every Element scheme in which they used the idea of For Loops and If Statements as the artifacts in the construction of an instrument to solve a problem. That is, the use of For Loops and If Statements are computational tools that were not originally designed to find the

294

intersection of sets, but were co-opted by Leo to solve the mathematical task of finding the intersection of three sets. This is a nuanced elaboration of how I initially interpreted the role of Python as the primary artifact when I designed this study. The nuance is that when I originally conceptualized this study, I considered Python itself as the primary artifact in the students' instrument development. I was thinking about text-based code in general, not specific computational tools like For Loops and If Statements, which is what emerged through the students' work. The Monitor Change in Cardinality scheme represented by Alonso's work also utilized For Loops as the artifact in the construction of his instrument to solve the mathematical task. In wanting to find the intersection of sets A, B, and C, his idea was to add each element from A to both B and C. If the cardinality of B and C each did not change, then the element added from A must belong to all three sets and can be added to a new set, D. Once this process is complete, the new set D would contain all of the common elements from A, B and C, which is the intersection of the three sets.

The Determine Set Equality scheme and Verify Each Element scheme characterized two ways students determined that one set is a subset of another set. The Determine Set Equality scheme emerged through Julian's work in Group 4 as well as with Eugene and Saul's work in Group 2. Both groups came up with solutions to determine that the set of integers between 1 and 1000 which are divisible by 21 (A) is a subset of the sets of integers between 1 and 1000 which are divisible by three (B) and seven (C). Both solution methods used For Loops and If Statements as artifacts in the construction of their instruments, but they solved the task in slightly different ways. Julian's approach was to check each element in set A and add it to a new set, D, if it was also an element in set B. After the For Loop was run, Julian set A equal to D to check whether the sets were the same. If the sets were the same (which they were) then every element in A was an element in B which means that A was a subset of B. Eugene and Saul took a similar approach, but their For Loop used an If Statement to check that every element in A belonged in both B and C. If the element in A was an element in B and C, then the element was added to a new set, D. After the For Loop was run, the students checked the cardinality of A and D and determined that since the cardinalities were the same, A was a subset of both B and C. The Verify Each Element scheme emerged as Leo came up with a different solution than his partners Eugene and Saul. Leo's approach was similar to his work that contributed to the Filter Every Element scheme in that he constructed a For Loop to check each element in A. His For Loop passed each element of A through the Union of B and C and if it was in the union, the code would produce the output "A is a subset." There were 47 elements in A, which resulted in the output of 47 lines of "A is a subset." Of course, finding that A is a subset of the union is a slightly different problem, but the central idea is that Leo's goal was to verify that each element in A was an element in the other set. In the next section I discuss the extent to which the students in my study achieved the learning outcomes that I set when I designed this study. As a reminder, the first learning goal was for the students to develop operational definitions of the logical operators, 'and' and 'or.' The second learning goal was for the students to be able to determine what it means for one set to be a subset of another set.

Research Question 2

The goal of Research Question 2 was to characterize the students' increasingly sophisticated ways of reasoning about set theory and logic. To do this I utilized four components of a hypothetical (and actual) learning trajectory:

- 1. Established learning goals related to set theory and logic that would prepare students going into advanced mathematics.
- Developed tasks that would help support the advancement of the students' mathematical reasoning.
- 3. Analyzed the advancing mathematical sophistication of the students as evidenced by the actual learning trajectory of the students.
- 4. Monitored the specific moves that I made as the researcher to help support the students' mathematical thought processes and reasoning.

There were two learning outcomes that I wanted the students to be able to take away from participating in this study. The first was to be able to develop operational definitions of the logical operators 'and' and 'or' and be able to flexibly reason about these logical operators to solve problem tasks. The second learning outcome was for the students to be able to utilize Python and their conceptions of union and intersection to determine that one set is a subset of another set. The extent to which the students achieved these goals varied by each group as some groups progressed further in the instructional task sequence than other groups. Specifically, Group 4 made it the furthest and had the most success in achieving both stated goals. As for Group 1, their actual learning trajectory focused more on the first goal, as well as an added focus on the functionality of a For Loop.

As is the case with most endeavors, the more one does something, the better they get. This was certainly the case with this study as I learned what aspects students struggled with and how I can streamline the tasks to best support the students in their learning of the material related to the two goals. For example, with Group 1 when I introduced the For Loop, I included a running index to track the iterations of the loop. This aspect was more of a topic of confusion and distraction than anything else, so I removed it for the other groups. I also made the decision to switch from Google Colab to an IDE on my own computer for Groups 3 and 4, which helped with a more detailed understanding of the processes of a given block of code. Making these changes allowed for more time, which led to richer, more meaningful conversations about the mathematical ideas. The tasks that I created were also developed utilizing two instructional theories known as Realistic Mathematics Education and PRIMM (Predict, Run, Investigate, Modify, and Make). Both theories start with the underlying premise that one can design tasks in such a way that students are able to engage in rich and meaningful problems from the moment they are introduced to the new material. Moreover, each of these theories support an approach that encourages students to develop an ownership of the material in ways that straightforward instruction or lecture cannot do. The overarching goal with the tasks that I created was to support students in their development of algorithms to solve problems related to set theory and logic.

To monitor the advancing mathematical sophistication of the students, I used the four levels of mathematical activity that fall under the emergent modeling heuristic of Realistic Mathematics Education. These levels of activity are situational, referential,

298

general, and formal. Typically, these levels of mathematical activity are used to inform the design of the tasks to support students in their construction of models, or ways of understanding mathematical material. However, they have also proved to be helpful in documenting the developmental learning progression of students as one is able to monitor the shift between students' models-of ways of reasoning about a mathematical idea, and then utilizing that model-of reasoning as a model-for reasoning about a more advanced topic. In this study, I highlighted the emergence of what I called the 'proposition/operator/proposition' model that all the students in my study were able to construct. This model addressed both aspects of the first learning outcome in the following ways. First, the students developed a model-of reasoning in being able to determine the truth value of a given propositional statement such as 'True or (False and True)' and general propositional statements such as 's and t' where s and t are any two propositions. These two propositional statements are examples of the situational and referential levels of activity, respectively. The next step in their model development addressed the second part of the first learning outcome, being able to flexibly reason about the 'and' and 'or' operators to solve more complicated problem tasks. This next step was the students' general mathematical activity in which the students were able to use If Statements inside of a For Loop that contained a logical operator such as 'and' or 'or' to filter for certain elements in sets to reinvent set union and set intersection. As for the second learning outcome, the students did not achieve this goal to the extent to which I was hoping. The main reason for this is that the learning opportunities afforded to the students on the idea of subsets was minimal compared to those on logical statements and other ideas such as

elements of a set. I was focused more on the idea of using a For Loop to reason about sets that I lost sight of the mathematics itself. If I were to redo the introductory task on subsets, I would likely construct a situation for the students in which they were tasked with investigating certain properties and relationships between defined sets and their subsets. One potential example would be a task in which the students were asked to reason about sets $A = \{1, 2, 3\}$ and $B = \{1, 2, 3, \{1, 2, 3\}\}$. In this example, A is both an element of B and subset of B. This task would provide the students with an opportunity to engage more deeply with the idea of a subset, and as a result, would have more of an opportunity to construct a model-of way of reasoning that could be developed for more complex mathematical problems. As a result of the oversight, a model did not emerge from the students' work on subsets. However, some students such as Alonso were able to internalize a working definition of a subset, evidenced by his performance on the post-study mathematical content survey which I will summarize in the last paragraph of this section.

The last component of the learning trajectory was a focus on the different instructional strategies that I utilized to inquire into the students' thinking and encourage productive conversations. I identified the following four instructor moves:

- 1. I did not evaluate the students' answers.
- 2. I asked for each student's thoughts and probed for more details of their understanding.
- 3. I "told" the students certain pieces of information when necessary.
- 4. I wrote code that would provide an opportunity for the students to investigate and compare outputs.

I found the most helpful instructor moves to engage students in productive conversations were items 2 and 4. By asking each student what they thought about a certain problem task (and not evaluating their answers), I was able to facilitate a back and forth between the students by asking them to reflect on their partner's ideas. Additionally, by providing complete code, the students were able to compare the output with their initial thoughts and interpretations of what the code would produce.

Lastly, as a measure to document what the students may have known coming into the study compared to what they learned by the end of the study, I administered a pre and post-study mathematical content survey. Every student in the study improved their mathematical content survey score from pre-study to post-study. Additionally, the results of the surveys support my finding that the students achieved more success related to the first learning outcome compared to the second learning outcome. For the three questions related to subsets, there was only an improvement of four correct answers on the first question (from zero correct to four correct), no improvement on the second question (three correct to three correct), and only one additional student answered the third question correctly (one correct to two correct). Comparing this to the three questions that contained components of both set theory and logic, we see that there was a difference of six correct answers (from zero to six) for question nine, a difference of seven correct answers (from zero to seven) for question 10, and a difference of seven correct answers (from one to eight) for question 11. Given these findings, I realize that my approach to the content related to subsets was flawed and requires more attention. Specifically, developing tasks that would foster the development of a model related to subsets, or a task that would foster

301

their situational activity, is needed. In the next section I summarize the results related to the students' affective experiences and their mathematical identities.

Research Question 3

The third research component of my study was to investigate the effect that a collaborative online research study using Python to learn mathematics might have on students' affective experiences, and in turn, their mathematical identities. There are two fundamental aspects related to my study that are important to consider when I discuss students' mathematical identities. First, one's mathematical identity is composed of multiple factors including interest, recognition by others, self-efficacy, beliefs, attitudes, and competence. The second aspect is that one's mathematical identity is not a static characteristic. That is, one's identity is fluid and context-specific and may change from one task to the next. While the complexity of identity cannot be entirely captured by implementing a pre/post survey on confidence and interest (such as the ATMLQ), these data points can serve as a blunt measure of students' sense of self before and after a teaching experiment. Studying students' affective experiences and understanding their shifting mathematical identities is important since the STEM disciplines are dominated by white men; understanding how we can support students of color, women, and other marginalized groups in their development of positive mathematical identities is crucial to diversifying the future STEM workforce. Treating our research participants as human beings and understanding their mathematical identity development can tell us so much more than whether or not the students were able to solve some problem tasks.
The data analyzed to address this component of my research consisted of debrief discussions with the students after each teaching experiment session, comments that the students made as they worked through the tasks, and the pre and post-study surveys called the ATMLQ (Attitudes to Technology in Mathematics Learning Questionnaire). The ATMLQ measured the students' confidence and interest related to three categories: (a) mathematics, (b) computers and programming, and (c) programming to learn mathematics. To capture a general description of the students' affective experiences and mathematical identity, I analyzed the students' pre-study ATMLQ score compared to their post-study scores, evidence of the students' sense of self-efficacy, their response to errors, and their general beliefs, attitudes, and emotions.

The difference between the pre-study ATMLQ score and their post-study ATMLQ score was the main factor that I measured to understand the students' experiences participating in the study. On the pre-study ATMLQ, the average score on the mathematics block of questions was 6.9, 10.2 for the computers and programming section, and 8.3 for the programming to learn mathematics section. For the post-study ATMLQ, the average score on the mathematics block of questions was 9.3, 10.5 for the computers and programming section, and 11.8 for the programming to learn mathematics section. Note, any number above zero on the ATMLQ indicates a positive sense of confidence and interest. Generally, the students that participated in my study experienced a positive shift in their confidence and interest related to mathematics and programming. However, three out of the ten students reported an overall negative shift in their confidence and interest. I summarize these three students' experiences below.

Palmer experienced a negative shift of eight points from the pre-study ATMLQ to the post-study ATMLQ, with the largest negative shift occurring in his confidence and interest related to computers and programming (negative 10 points) but a small positive shift with respect to programming to learn mathematics (positive four points). As evident through his participation, Palmer was confident in his ability, but my hypothesis is that the complexity of the computer programming opened his eyes to what was possible with computers. That is not to say that Palmer was completely defeated, instead I conjecture that Palmer gained an appreciation for the power of computers. In support of this conjecture are his ATMLQ scores. His score on the pre-study ATMLQ was as high as one could score on the computers and programming block of questions, with the post-study ATMLQ it dropped 10 points, and yet his score was still the highest out of all of the research participants.

Alonso had an overall difference between the pre-study ATMLQ and post-study ATMLQ of negative three points, but I do not consider this negative shift to be very descriptive of his overall experience. If anything, Alonso maintained his identity as a capable mathematician. Alonso understood the material quicker than any other student in my study, and answered every question correctly on the post-study mathematical content survey. Moreover, after the last session Alonso emailed to inform me that his participation in the study motivated him to study Python on his own time. There were moments where I thought that Alonso was being slightly too hard on himself for not knowing the right answer, but this was likely just a product of his desire to learn the material because of his genuine interest in mathematics and programming. So, like Palmer, I do not consider his

negative shift on the ATMLQs to be compelling evidence of a negative shift in his mathematical identity.

The third student that reported a negative shift from the pre-study ATMLQ to the post-study ATMLQ was Delia, with a difference of 14 points. Throughout the study, Delia seemed neutral in her disposition toward the tasks, she even seemed bored in some moments. It could be that the material we were covering was not challenging enough for her as she was a competent mathematician, and she was confident in her reasoning. However, as stated previously, one's mathematical identity is not inextricably linked to competence. One's mathematical identity is also composed of one's interest and other affective measures. Another explanation could be that she had a different expectation of what this study was going to be like, which could have led to a lack of engagement and interest as the study progressed.

Additional evidence that supports the claim that Delia had a different experience participating in this study related to the other students was in the students' answers to the free-response questions. All the students, except for Delia, noted some aspect of positive mathematical identity growth that contributed to their success in this study. Many students said that their interest in mathematics helped them succeed, other students mentioned their comfort in asking questions or a general desire to learn new material. For Delia, she did not believe that any aspect of her identity helped her succeed. In the next section I discuss the various implications that the findings related to my three research objectives may have in the years to come.

Implications

There are three main implications of this work. The first has to do with curriculum design and development in that the instructional sequence that I designed for this study showcases one example of how set theory and logic can be taught strictly using Python. More generally, this study is an existence proof that computer programming can serve as an accessible onramp to mathematics in a way that empowers learners in their understanding of the material. Therefore, one direct implication of this work is that educators may find ways to utilize computing and programming in traditional mathematics classrooms as a way to introduce a concept and encourage students to explore the material for themselves. Moreover, all the students in my study had little to no previous experience coding, and all of them were either enrolled in differential or integral calculus, courses that we traditionally view as "introductory." What this implies then is that we don't need to wait until students have some type of background in computing or advanced mathematics to encourage them to think computationally. Therefore, this study is particularly relevant for those teaching Introduction to Proofs courses, as set theory and logic are two ideas which are often central to the curriculum.

Relatedly, the second main implication is the idea that set theory and logic do not necessarily need to be introduced in the context of proof, nor do they need to be introduced as independent units of instruction. Traditionally, set theory and logic are independent units taught with methods utilizing proofs. Granted, proof writing is one of the most, if not the most, important skill for mathematicians, and undergraduate students need to practice this skill. However, in this study I demonstrated how the fundamental ideas of set theory and logic can be introduced together, in the context of Python. This implies that educators teaching introduction to proofs courses can utilize some aspects of Python to help students reason about set theory and logic, if only to introduce the material and ground the problem tasks in a realizable context. One example of a learning outcome for a student in my study that might not necessarily be true for a student in a standard introduction to proofs course would be the strong connection between the 'and' operator and set intersection as well as the 'or' operator with set union.

The third implication is that conducting a research study in the virtual environment can not only be possible, but can be an ideal method of data collection. For this study, I conducted all the teaching experiment sessions over Zoom, using the screen record feature to capture student work in one window and the software that ran the code in another window. This format worked particularly well as the students were able to annotate their work in Jamboard, monitor the output of a given block of code, and easily communicate with one another all at the same time. If we were in person, navigating back and forth between one student's code to another student's code would end up being more of a logistical and physical challenge than anything else. That is, the students would end up needing to type their code and share it to a shared document like Google Docs or Jamboard anyways, thus I do not see the logistical added benefit of conducting the study in person. Additionally, some of the students in the study commented that being able to attend these sessions virtually was convenient for them, and in Juliana's case, she believed that working online was less intimidating than working in person. An additional added benefit of collecting data via Zoom is that recording to the cloud will produce an automatic transcript

of the session, something that is a major time saver compared to writing up the transcript from scratch. In the next section I address the limitations of this study.

Limitations

There are three main limitations of this study that are important to address, presented in decreasing levels of significance. First, after the pilot study I realized that asking students to write their own code, think deeply about ideas related to set theory and logic, communicate their ideas out loud, and engage in discussions with their partner(s) was too much to ask for a short five-session teaching experiment like the one I designed. For this reason, I decided that I would primarily write the code, and the students would evaluate the code, predict the output, etc. My writing of the code had two benefits. First, the students did not have to worry about the syntax of Python which allowed them to instead concentrate on the ideas. Secondly, with me writing the code, we were able to progress through the material much quicker than if the students were required to write their own code. The students were given some opportunities to write their own code, and in most cases I would not write something unless they had verbalized their code out loud, but in general the students were not tasked with developing their own code. While there were some benefits, my hypothesis is that not asking the students to write their own code had a negative effect in that the students were not able to take ownership over the code that was being run. In retrospect, if I were to conduct this study again, I would only focus on the first learning outcome (on the use of logic to solve more complex tasks related to set theory) and spend the extra time to support students in their writing of code.

A second limitation is related to the opt-in nature of the students' participation in this study. As a reminder, I recruited students from differential and integral calculus classes by asking the instructors of the courses to encourage their students to participate in my study which focused on the use of programming to learn mathematics. If a student was interested, they were required to take a screening survey which asked about their experience with set theory, logic and programming as well as other demographic information. While I selected students with minimal experience in these three areas of study, the students that opted-in to participate likely had higher levels of confidence and interest compared to their peers. This is reflected by the ATMLQ data in which all but one of the students (Haven) scored positively on the pre-study questionnaire asking about their confidence and interest related to mathematics, computers and programming, and programming to learn mathematics. Therefore, the students' general positive disposition towards mathematics and programming produced an ideal teaching scenario where the students were interested and wanted to learn about mathematics and programming. Given a random sample of students, I would not be so sure that this instructional task sequence would have gone as smoothly as it did.

Lastly, there were some aspects of the virtual environment that limited the true potential of this study. Poor connectivity was the first issue. During some sessions I would ask students to turn their cameras off to try and establish a better connection, but it is not entirely clear how much that helped them. Listening back to the video recordings I also realized that I experienced some connectivity issues without knowing it (the students did not say anything when it happened), which certainly led to missed information.

Additionally, when screen sharing, the screen record function on Zoom only captured the screen that I was sharing, and a mini window of whoever was speaking. Due to only being able to see the speaker, I was not able to incorporate any additional data such as gestures or facial expressions by those that were not highlighted as the speaker.

Future Research

One important aspect of research is that one typically generates more questions about their work compared to the number of questions they were able to answer. This is certainly true about the work presented in this manuscript as I see five important areas for future research. The first is related to the students' schemes in terms of the instrumental approach. In Chapter 4 I only highlighted the schemes of select students, but how well do these schemes characterize the other students' work on the same tasks? That is, are there slight variations of each scheme that would capture each student's thought processes in more detail? A more detailed analysis of each student's work (and additional students in a follow up study) is necessary to better understand the prevalence and scope of the schemes presented in Chapter 4.

A second area of future research would focus on the task sequence on subsets. The students in my study were not provided with learning opportunities that supported constructing a model-of what it means for one set to be a subset of another set in the way that I originally intended. Future research would entail the development of a new task sequence, or a modification of the introductory task sequence presented in this manuscript. More generally, future research to investigate how to infuse programming into any

instructional task sequence using Realistic Mathematics Education and PRIMM is needed, which brings me to the next area of future research.

As alluded to previously, this study portrays the enactment of an ideal scenario in which the students were interested and wanted to learn how to use programming to learn mathematics. How would other students not as interested in programming or mathematics receive this material? Would it push them away from STEM? Or would it encourage and empower them to pursue STEM? Moreover, this study was conducted as a series of teaching experiment sessions in which the only focus was on programming to learn set theory and logic. Finding a way to integrate computing into an intact mathematics course with homework, quizzes and exams is the next step in bringing computing into the classroom and into the student learning experience. There has been some progress on this front with Kaplan's development of *Computer-Age Calculus with R* (2020) and Brownlee's *Basics of Linear Algebra for Machine Learning: Discover the Mathematical Language of Data in Python* (2018) to name two of the more recent attempts to integrate programming with introductory mathematics.

Another area of future research could be on the differential impact that programming to learn mathematics may have based on students' social-marker identities such as gender. As seen in my study, the top three greatest shifts in terms of positive mathematical identity development were three women with shifts of +32, +21, and +11. Delia was the only other woman in this study and her ATMLQ results produced a shift of -14 points, but the overwhelming difference between the total scores of the women compared to the total scores of the men (50 compared to 12, respectively) deserves further

investigation. Relatedly, two of the students that identified as women mentioned that their least favorite aspect of the study was being "cold-called." I imagine that most students prefer other methods of eliciting participation over cold-calling, but I wonder if the online environment had anything to do with a heightened-sense of discomfort when being cold-called. Additional research on cold-calling in a virtual teaching environment would be beneficial to understanding student engagement and how that may affect their sense of mathematical identity development.

Lastly, there were many instances in my study in which the students would anthropomorphize Python or the written code itself. For example, some students would say that the 'or' operator "favors True" or "wants it to be True" to produce a True output. There were also instances of students using "see" as if Python itself was reading the code, "if it *sees* an 'e' or an 'o' in set A, it's going to add the 'e' and the 'o' to the empty set." This kind of language is natural when one is describing a process, but I wonder what correlation exists between one's anthropomorphizing of Python and their confidence and ability to construct an algorithm to solve a mathematical task. My hypothesis is that there exists a negative correlation between the two ideas. That is, for the students that do not anthropomorphize and see computer programming as a simple set of instructions that are carried out by a machine, then those students would understand that they are in charge of what the output will be. If on the other hand students see computers or programming languages as things that have their own agency and/or preferences, then the students may be less likely to find a sense of ownership over the material.

Final Thoughts

I begin this section with some thoughts on completing a dissertation during the height of a global pandemic. In March 2020 I was finishing my last set of courses required for my doctoral degree and preparing some thoughts on what I wanted to do for my dissertation. As we all know, it was then that the world shut down due to COVID-19 and we were required to conduct operations under guarantine. Now, restrictions have almost entirely been lifted (e.g., no mask requirement for flying, no proof of vaccine necessary at concerts or other indoor venues). For me, that meant that the entire dissertation process from proposal to data collection, to analysis and writing occurred during the COVID-19 pandemic. Research has shown that the COVID-19 pandemic and preventative health measures such as quarantining and school/work closures have resulted in elevated levels of stress, anxiety, and depression which has had deleterious effects on cognitive functioning and overall affect (e.g., Boals & Banks, 2020; de Figueiredo et al., 2021; Nogueira et al., 2021; Schwartz et al, 2021, Vannorsdall, 2022; Wirkner et al., 2022). With that said, and with the lessons learned from completing this work, I encourage those reading this to allow yourself grace, find a moment to reflect on what matters to you most, call your loved ones, and do your best to support those around you, because we all need it.

To conclude this manuscript, I revisit the need for computing and programming in mathematics education. Integrating programming and computational thinking into the mathematics curriculum goes beyond just preparing undergraduate STEM majors for their future careers. It is a matter of economic mobility for the underserved and marginalized groups of our society. Computer science should not just be for those that can afford to

attend private institutions or live in the right zip codes and as a result attend the best public schools, nor should it just be for undergraduate students in STEM - by that time we are already too late. Moving forward, my hope is that we find ways to introduce computing at an earlier age, which means teaching future K-12 educators with methods that integrate computing and investing in the resources needed to properly support all students.

Appendix A: Screening Survey

Welcome! Thank you so much for your interest in participating in my dissertation study. The following questions will help me in selecting participants for the pilot study of my dissertation. The overall focus of my dissertation work is to better understand how the programming language, Python, can be leverage to connect ideas related to Set Theory and Mathematical Logic.

The pilot study will occur mid-to-late January 2021. Each one of you has something important to offer, and thus I am so appreciative of your time. If selected, you will be modestly compensated for your time.

To participate in this survey, you must be 18 years old, or older. Are you?

o Yes

o No

Thank you for participating! Your progress will be automatically saved every time you move to the next page. If you experience any difficulties, please contact Antonio Martinez at antonio.aemartinez@gmail.com

On a scale from 1-10, where 1 represents little to no knowledge or experience and 10 represents extremely knowledgeable and experienced, how would you rate your knowledge and experience with the following:

0 1 2 3 4 5 6 7 8 9 10

Mathematical Set Theory ()	
Mathematical Logic ()	
Computer Programming (e.g., Java, C++, Python) ()	

What do you know about Mathematical Set Theory? In what context have you seen it?

What do you know about Mathematical Logic? In what context have you seen it?

What do you know about programming? In what context have you seen it?

The purpose of the following questions is to ensure that a wide range of student voices will be heard and learned from throughout this study. Some of this information may be sensitive, and I will handle it responsibly. Your responses will be kept private and secure, and will not be used for any other purposes outside of this study. Only members of the research team will have access to this data.

Do you plan to enroll in Math 245 (Discrete Mathematics) in the future?

o Yes

o No

o I'm not sure

o I am currently taking or have already passed Discrete Mathematics (4)

When do you plan on taking Math 245 - Discrete Mathematics?

o Spring 2021

o Fall 2021

o Other

What is your class standing?

- o First Year
- o Second Year
- o Third Year
- o Fourth Year
- o Other (please specify)_____
- o Prefer not to disclose

Which major(s) have you declared, or do you intend to declare?

(Select all that apply) Do you consider yourself to be:

Alaska Native or Native American
Black or African American
Central Asian
East Asian
Hispanic or Lantinx
Middle Eastern or North African
Native Hawaiian or Pacific Islander
Southeast Asian
White
Not listed (please specify)
⊗Prefer not to disclose

(Select all that apply) Do you consider yourself to be:

□ International student

□ First-generation college student (i.e., neither parent nor guardian completed a Bachelor's degree)

- Commuter student
- □ Transfer student

- □ Student with a disability
- □ Student athlete

Current or former English language learner (i.e., the primary language spoken in your childhood home was not English)

- □ Parent, guardian or care giver
- \Box \otimes Prefer not to disclose

(Select all that apply) Do you consider yourself to be:

	Man
	Gender fluid or gender diverse
	Transgender
	Woman
	Not listed (please specify)
0	⊗Prefer not to disclose

Please confirm your participation in my study:

o Yes, I would like to participate. Here is my contact email:

o No, I would not like to participate in your study.

Thank you for being willing to participate! Antonio will reach out to you if you are selected for participation.

If you would like to revisit any of your responses, please use the back button on this page. Submitting this page will finalize your responses and complete your submission. If you have any questions about the project or this survey, please contact Antonio Martinez at antonio.aemartinez@gmail.com.

Appendix B: List of Codes by Research Question

RQ1:

- Scheme
 - Goals
 - Rules
 - Operational Invariants
 - Inference
- Concept
 - Set theory
 - Logic
 - Python
- Techniques
 - Diagram
 - \circ Run the code
 - Write a For Loop
 - Add elements to a set
- Tech-Elements
 - Written Code
 - Annotation in Jamboard
 - Verbalized code

RQ2:

- Goals
 - Sets as objects
 - Use of 'and'
 - Use of 'or'
- Tasks
 - Unhelpful Bored
 - Unhelpful No discussion
 - Helpful Productive Discussion
 - Helpful Aha moment
 - Confusing Lots of Questions
- Mathematical Activity
 - Situational Activity
 - Referential Activity
 - General Activity
 - Formal Activity
- Instructor Moves
 - 'Telling' Students
 - Not evaluating answer
 - Inquire into student thinking
 - Writing code

RQ3:

- Self Efficacy
 - Confident
 - Not confident
 - Waning confidence
 - Gaining confidence
- Errors
 - Give space
 - Shut down
 - Signal understanding
 - Smile out of interest or curiosity
 - Revise error
 - Preface answer with "not sure" or "I know this isn't right"
- Beliefs
 - Math/programming is interesting
 - Math/programming difficult
 - Collaboration over Zoom worked well
- Attitudes
 - Positive Attitude
 - Negative Attitude
 - Neutral Attitude
- Emotions
 - Нарру
 - Excited
 - Nervous
 - Anxious
 - Embarrassed
 - Frustrated
 - Serious
 - Tired/burnt out
 - Joy in explaining answer
 - Curious
 - \circ Overwhelmed
 - Determined
 - Matter-of-fact

Appendix C: Mathematical Content Survey

Please answer the following questions to the best of your abilities. If you do not know how to answer the question, that is okay!

What is your name?

 $A = \{ \{1, 5, 7\}, \{3, 4, 8\} \}$. Are any odd numbers between 1 and 10 elements of set A? Please provide details with your answer.

o Yes _____

o No _____

o I'm not sure

A = { $\{1, 5, 7\}, \{3, 4, 8\}$ }, B = $\{1, 5, 7, \{3, 4, 8\}$ }. If we define C to be the set of elements that exist in A and B, what elements are in C?

A = { $\{1, 5, 7\}, \{3, 4, 8\}$ }, B = $\{1, 5, 7, \{3, 4, 8\}$ }. If we define C to be the set of elements that exist in A or B, what elements are in C?

Consider any two sets, A and B. What does it mean for an element to not be an element of A and B?

Please do not change your answers to the previous questions. We define an "element" as a member of a set. We define a subset in the following way: A is a subset of B given that every element in A is an element of B.

 $S = \{1, 3, \{3, 4\}\}$. Is 3 an element of S?

o Yes

o No

o I'm not sure

- $S = \{1, 3, \{3, 4\}\}$. Is 3 a subset of S?
 - o Yes
 - o No
 - o I'm not sure
- $S = \{1, 3, \{3, 4\}\}$. Is $\{3\}$ a subset of S?
 - o Yes
 - o No
 - o I'm not sure
- $S = \{1, 3, \{3, 4\}\}$. Is $\{3, 4\}$ an element of S?
 - o Yes
 - o No
 - o I'm not sure
- $S = \{1, 3, \{3, 4\}\}$. Is $\{3, 4\}$ a subset of S?
 - o Yes
 - o No
 - o I'm not sure

Is the following statement true or false?

"Given an integer number x, x is even or x is odd"

o True

- o False
- o I'm not sure

Is the following statement true or false?

"The integer 15 is even or 15 is odd"

- o True
- o False
- o I'm not sure

[Question on the Pre-Study Survey]

What would be the output of the following code in Python?

```
city = "San Diego"
```

```
for x in city:
print(x)
```

[Question on the Post-Study Survey]

What would be the output of the following code in Python?

```
prop = "proposition"
```

for x in prop: print(x) if ((x == "o") or (x == "p")): print(x)

Appendix D: Confidence and Interest Survey

Five-point Likert scale for the following questions: 1 = Strongly agree; 2 = Agree; 3 = Neutral; 4 = Disagree; 5 = Strongly disagree

The following statements refer to your confidence when learning mathematics.

- 1. I have less trouble learning mathematics than other subjects.
- 2. When I have difficulties with mathematics, I know I can handle them.
- 3. I do not have a mathematical mind.
- 4. It takes me longer to understand mathematics than the average person.
- 5. I have never felt myself able to learn mathematics.
- 6. I enjoy trying to solve new mathematics problems.
- 7. I find mathematics frightening.
- 8. I find many mathematics problems interesting and challenging.
- 9. I don't understand how some people seem to enjoy spending so much time on mathematics problems.
- 10. I have never been very excited about mathematics.
- 11. I find mathematics confusing.

The following statements refer to your confidence when using computers.

- 12. I have less trouble learning how to use a computer than I do learning other things.
- 13. When I have difficulties using a computer I know I can handle them.
- 14. I am not what I would call a computer person.
- 15. It takes me much longer to understand how to use computers than the average person.
- 16. I have never felt myself able to learn how to program.
- 17. I enjoy trying new things on a computer.
- 18. I find having to use computers frightening.
- 19. I find many aspects of using computers interesting and challenging.
- 20. The idea of being asked to program is frightening.
- 21. I don't understand how some people can seem to enjoy spending so much time using computers.
- 22. I have never been very excited about programming.
- 23. I find using computers confusing.
- 24. I'm nervous that I'm not good enough with computers to be able to use them to learn mathematics.

The following questions refer to the way you feel about computers in the learning of mathematics.

- 25. Computing power makes it easier to explore mathematical ideas.
- 26. I know programming is important but I don't feel I need to use it to learn mathematics.
- 27. Computers and graphics calculators are good tools for calculation, but not for my learning of mathematics.

- 28. I think programming is too new and strange to make it worthwhile for learning mathematics.
- 29. I think programming wastes too much time in the learning of mathematics.
- 30. I prefer to do all the calculations and graphing myself, without using a computer or graphics calculator.
- 31. Using computing power/programming for the calculations makes it easier for me to do more realistic applications.
- 32. I like the idea of exploring mathematical methods and ideas using programming.
- 33. I want to get better at using computers to help me with mathematics.
- 34. The symbols and language of mathematics are bad enough already without the addition of programming.
- 35. Having programming to do routine work makes me more likely to try different methods and approaches.

Post-Study Free-Response Questions

- 1. What was your favorite part about participating in this study?
- 2. What was your least favorite part about participating in this study?
- 3. What are your thoughts on working collaboratively with others in the Zoom virtual setting?
- 4. What about you or your identity contributed to your success in this study?

References

- Adiredja, A. P., & Andrews-Larson, C. (2017). Taking the sociopolitical turn in postsecondary mathematics education research. *International Journal of Research in Undergraduate Mathematics Education*, *3*(3), 444-465.
- Andrews-Larson, C., Wawro, M., & Zandieh, M. (2017). A hypothetical learning trajectory for conceptualizing matrices as linear transformations. *International Journal of Mathematical Education in Science and Technology*, 48(6), 809-829.
- Artigue, M. (2002). Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work. *International Journal ofComputers for Mathematical Learning*, 7(3), 245–274.
- Avigad, J. (2019). Learning logic and proof with an interactive theorem prover. In G. Hanna, D. Reid, M. de Villiers (Eds.), *Proof technology in mathematics research* and teaching (pp. 277–290). Springer.
- Bagni, G. T. (2006). Some cognitive difficulties related to the representations of two major concepts of set theory. *Educational Studies in Mathematics*, 62(3), 259-280.
- Ball, L., Drijvers, P., Ladel, S., Siller, H. S., Tabach, M., & Vale, C. (2018). Uses of technology in primary and secondary mathematics education. Springer.
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological review*, 84(2), 191.
- Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *AcM SIGcSE Bulletin*, 39(2), 32-36.
- Boals, A., & Banks, J. B. (2020). Stress and cognitive functioning during a pandemic: Thoughts from stress researchers. *Psychological Trauma: Theory, Research, Practice, and Policy, 12*(1), 255-257.
- Bogusevschi, D., Muntean, C., & Muntean, G. M. (2020). Teaching and learning physics using 3D virtual learning environment: A case study of combined virtual reality and virtual laboratory in secondary school. *Journal of Computers in Mathematics and Science Teaching*, *39*(1), 5-18.
- Brownlee, J. (2018). *Basics of linear algebra for machine learning: Discover the mathematical language of data in python*. Machine Learning Mastery.

- Buteau, C., Gueudet, G., Muller, E., Mgombelo, J., & Sacristán, A. I. (2020). University students turning computer programming into an instrument for 'authentic' mathematical work. *International Journal of Mathematical Education in Science* and Technology, 51(7), 1020-1041.
- Clements, D. H., & Sarama, J. (2004). Learning trajectories in mathematics education. *Mathematical Thinking and Learning*, 6(2), 81–89.
- Cobb, P. (1994). Constructivism in mathematics and science education. *Educational Researcher*, 23(7), 4.
- Cobb, P., & Hodge, L. L. (2011). *Culture, identity, and equity in the mathematics classroom.* In A. J. Bishop (Managing Ed.), E. Yackel, K. Gravemeijer, & A. Sfard (Vol. Eds.), Mathematics education library: Vol. 48. A journey in mathematics education research: Insights from the work of Paul Cobb (pp. 179-195). New York, NY: Springer, doi: 10.1007/978-90-9729
- Cobb, P., & Yackel, E. (1996). Constructivist, emergent, and sociocultural perspectives in the context of developmental research. *Educational Psychologist*, *31*(3-4), 175-190.
- Confrey, J., & Lachance, A. (2000). Transformative teaching experiments through conjecture-driven research design. In R. Lesh & A. E. Kelly (Eds.), *Research design in mathematics and science education* (pp. 231-265). Hillsdale, NJ: Erlbaum.
- Cooper, K. M., & Brownell, S. E. (2016). Coming out in class: Challenges and benefits of active learning in a biology classroom for LGBTQIA students. *CBE—Life Sciences Education*, 15(3), 1-19.
- Cretchley, P., Harman, C., Ellerton, N., & Fogarty, G. (1999). Computation, exploration, visualisation: Reaction to MATLAB in first year mathematics. *2nd Australian Symposium on Modern Undergraduate Mathematics*.
- Cribbs, J. D., Hazari, Z., Sonnert, G., & Sadler, P. M. (2015). Establishing an explanatory model for mathematics identity. *Child development*, 86(4), 1048-1062.
- Darragh, L. (2016). Identity research in mathematics education. *Educational Studies in Mathematics*, 93(1), 19-33.
- David, E. J., & Zazkis, D. (2019). Characterizing introduction to proof courses: A survey of US R1 and R2 course syllabi. *International Journal of Mathematical Education in Science and Technology*, *51*(3), 388-404.

- Dawkins, P. C. (2017). On the importance of set-based meanings for categories and connectives in mathematical logic. *International Journal of Research in Undergraduate Mathematics Education*, *3*(3), 496-522.
- Dawkins, P. C., & Cook, J. P. (2017). Guiding reinvention of conventional tools of mathematical logic: students' reasoning about mathematical disjunctions. *Educational Studies in Mathematics*, 94(3), 241-256.
- Dawkins P. C., Zazkis, D. & Cook, J. P. (2020): How do transition to proof textbooks relate logic, proof techniques, and sets? *PRIMUS*, DOI: 10.1080/10511970.2020.1827322
- de Figueiredo, C. S., Sandre, P. C., Portugal, L. C. L., Mázala-de-Oliveira, T., da Silva Chagas, L., Raony, Í., ... & Bomfim, P. O. S. (2021). COVID-19 pandemic impact on children and adolescents' mental health: Biological, environmental, and social factors. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 106, 110171.
- Deming, D. J., & Noray, K. L. (2018). STEM careers and the changing skill requirements of work (No. w25065). National Bureau of Economic Research.
- Design-Based Research Collective (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, 32(1), 5-8.
- Dogan-Dunlap, H. (2006). Lack of set theory relevant prerequisite knowledge. International Journal of Mathematical Education in Science and Technology, 37(4), 401-410.
- Drijvers, P., Godino, J., Font, V., & Trouche, L. (2013). One episode, two lenses. *Educational Studies in Mathematics*, 82(1), 23–49.
- Dubinsky, E. (1995). ISETL: A programming language for learning mathematics. *Communications on Pure and Applied Mathematics*, 48(9), 1027-1051.
- Dubinsky, E. & Leron, U. (1994). Learning abstract algebra with ISETL. Springer Science & Business Media.
- Durand-Guerrier, V. (2020). Logic in mathematics education. In S. Lerman (Ed.), *Encyclopedia of mathematics education*, (pp. 478-481). Springer.
- Ekici, C., Mehrubeoglu, M., Palaniappan, D., & Alagoz, C. (2020, October). Coherence and transfer of complex learning with fourier analysis learning trajectories for engineering mathematics education. *IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.

- Fey, J. T. (1989). Technology and mathematics education: A survey of recent developments and important problems. *Educational Studies in Mathematics*, 20(3), 237-272.
- Fischbein, E., & Baltsan, M. (1998). The mathematical concept of set and the 'collection' model. *Educational studies in mathematics*, *37*(1), 1-22.
- Fogarty, G., Cretchley, P., Harman, C., Ellerton, N., & Konki, N. (2001). Validation of a questionnaire to measure mathematics confidence, computer confidence, and attitudes towards the use of technology for learning mathematics. *Mathematics Education Research Journal*, 13(2), 154-160.
- Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410-8415.
- Freudenthal, H. (1971). Geometry between the devil and the deep sea. *Educational Studies in Mathematics*, *3*(3), 413-435.
- Freudenthal, H. (1991). *Revisiting mathematics education: China lectures*. Springer Netherlands.
- Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015). Computing education in K-12 schools: A review of the literature. 2015 IEEE Global Engineering Education Conference (EDUCON) (pp. 543-551). IEEE.
- Gee, JP. (2000). Identity as an analytic lens for research in education. *Review of Research in Education*, 25, 99-125.
- Glaser, B., & Strauss, A., 1967. The Discovery of grounded theory. Aldine Publishing Company.
- Gravemeijer, K. (1999). How emergent models may foster the constitution of formal mathematics. *Mathematical Thinking and Learning*, *1*(2), 155-177.
- Gravemeijer, K. (2020a). A Socio-constructivist elaboration of realistic mathematics education. In Van den Heuvel-Panhuizen M. (Ed.), *National reflections on the Netherlands didactics of mathematics* (pp. 217-233). Springer.
- Gravemeijer, K. (2020b) Emergent modeling: An RME design heuristic elaborated in a series of examples. *Educational Designer*, *4*(13), 1-31.

- Gravemeijer, K., Bowers, J., & Stephan, M. (2003). Chapter 4: A hypothetical learning trajectory on measurement and flexible arithmetic. *Journal for Research in Mathematics Education. Monograph*, *12*, 51-66.
- Gravemeijer, K., Cobb, P. & Bowers, J. Whitenack. (2000). Symbolizing, modeling and instructional design. In P. Cobb, E. Yackel & K. McClain (Eds.), Symbolizing and communicating in mathematics classrooms: Perspectives on discourse, tools, and instructional design. (pp. 225-273). Lawrence Erlbaum Associates.
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, *42*(1), 38-43.
- Guin, D., & Trouche, L. (1998). The complex process of converting tools into mathematical instruments: the case of calculators. *International Journal of Computers for Mathematical Learning*, *3*(3), 195–227.
- Gutiérrez, R. (2008). A" gap-gazing" fetish in mathematics education? Problematizing research on the achievement gap. *Journal for Research in Mathematics Education*, *39*(4), 357-364.
- Gutiérrez, R. (2013). The sociopolitical turn in mathematics education. *Journal for Research in Mathematics Education*, 44(1), 37. doi:10.5951/jresematheduc.44.1.0037
- Gutiérrez, R., & Dixon-Román, E. (2010). Beyond gap gazing: How can thinking about education comprehensively help us (re) envision mathematics education? In B. Atweh, M. Graven., W. Secada, & P. Valero (Eds.), *Mapping equity and quality in mathematics education* (pp. 21-34). Springer.
- Hawthorne, C., & Rasmussen, C. (2015). A framework for characterizing students' thinking about logical statements and truth tables. *International Journal of Mathematical Education in Science and Technology*, 46(3), 337-353.
- Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K–12 mathematics classrooms. *Digital Experiences in Mathematics Education*, 4(1), 48-69.
- Horton, D., & Craig, M. (2015, February). Drop, fail, pass, continue: Persistence in cs1 and beyond in traditional and inverted delivery. In A. Decker, K. Eiselt, C. Alphonce, & J. Tims (Eds.), *Proceedings of the 46th ACM Technical Symposium* on Computer Science Education (pp. 235-240).
- Jones, M. G., & Brader-Araje, L. (2002). The impact of constructivism on education: Language, discourse, and meaning. *American Communication Journal*, 5(3), 1-10.

Kaplan, D. (2020). Computer-age calculus with R. https://dtkaplan.github.io/RforCalculus/

- Ko, Y. Y., & Knuth, E. (2009). Undergraduate mathematics majors' writing performance producing proofs and counterexamples about continuous functions. *The Journal of Mathematical Behavior*, 28(1), 68-77.
- Koch, A. K., & Drake, B. M. (2018). *Digging into the disciplines: The impact of gateway courses in accounting, calculus, and chemistry on student success*. John N. Gardner Institute for Excellence in Undergraduate Education.
- Labinowicz, E., & Frazee, S. P., (1980). *The Piaget primer: Thinking, learning, teaching*. Addison-Wesley Publishing Company.
- Lakoff, G., & Núñez, R. (2000). Where mathematics comes from (Vol. 6). Basic Books.
- Larsen, S. (2009). Reinventing the concepts of group and isomorphism: The case of Jessica and Sandra. *The Journal of Mathematical Behavior*, 28(2-3), 119-137.
- Larsen, S. (2018). Didactical Phenomenology: The Engine That Drives Realistic Mathematics Education. *For the Learning of Mathematics*, *38*(3), 25-29.
- Laursen, S. L., & Rasmussen, C. (2019). I on the prize: Inquiry approaches in undergraduate mathematics. *International Journal of Research in Undergraduate Mathematics Education*, 5(1), 129-146.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... & Werner, L. (2011). Computational thinking for youth in practice. *Acm Inroads*, 2(1), 32-37.
- Leng, N. W. (2011). Using an advanced graphing calculator in the teaching and learning of calculus. *International Journal of Mathematical Education in Science and Technology*, 42(7), 925-938.
- Levine, B., Mauntel, M., Zandieh, M., & Plaxco, D. (2020, February). Chase that Rabbit! Designing vector unknown: A linear algebra game. In S. S. Karunakaran, Z. Reed, & A. Higgins. Proceedings of the 23rd Annual Conference on Research in Undergraduate Mathematics Education.
- Lew, K., & Mejía-Ramos, J. P. (2019). Linguistic conventions of mathematical proof writing at the undergraduate level: Mathematicians' and students' perspectives. *Journal for Research in Mathematics Education*, 50(2), 121-155.
- Li, Y., Schoenfeld, A.H., diSessa, A.A. et al. (2020). On computational thinking and STEM education. *Journal for STEM Education Research*, *3*(2), 147–166.

- Linchevski, L., & Vinner, S. (1988). The naive concept of sets in elementary teachers. In A. Borbas (Ed.) *Proceedings of the 12th international conference, psychology of mathematics education* (pp. 471-478).
- Lobato, J., & Walters, C. D. (2017). A taxonomy of approaches to learning trajectories and progressions. In J. Cai (Ed.), *Compendium for Research in Mathematics Education* (pp. 74-101). Reston, VA: National Council of Teachers of Mathematics.
- Lockwood, E., & De Chenne, A. (2020). Enriching students' combinatorial reasoning through the use of loops and conditional statements in Python. *International Journal of Research in Undergraduate Mathematics Education*, 6(3), 303-346.
- Lockwood, E., DeJarnette, A. F., & Thomas, M. (2019). Computing as a mathematical disciplinary practice. *The Journal of Mathematical Behavior*, 54, 1-18.
- Lockwood, E., & Mørken, K. (2021). A call for research that explores relationships between computing and mathematical thinking and activity in RUME. *International Journal of Research in Undergraduate Mathematics Education*, 1-13. https://doi.org/10.1007/s40753-020-00129-2
- Martin, D. M. (2006). Mathematics learning and participation as racialized forms of experience: African American parents speak on the struggle for mathematics literacy. *Math. Think. Learn, 8*, pp. 197-229.
- McLeod, D. B. (1992). Research on affect in mathematics education: A reconceptualization. *Handbook of Research on Mathematics Teaching and Learning*, *1*, 575-596.
- Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis* (2nd ed.). Sage Publications, Inc.
- Moore, R. C. (1994). Making the transition to formal proof. *Educational Studies in Mathematics*, 27(3), 249-266.
- Namkung, J. M., Peng, P., & Lin, X. (2019). The relation between mathematics anxiety and mathematics performance among school-aged students: A meta-analysis. *Review of Educational Research*, 89(3), 459-496.
- Nasir NS, Saxe GB (2003). Ethnic and academic identities: A cultural practice perspective on emerging tensions and their management in the lives of minority students. *Educational Researcher*, *32*, 14-18.

- National Academies of Sciences, Engineering, and Medicine. (2020). Roundtable on data science postsecondary education: A compilation of meeting highlights. The National Academies Press.
- Nogueira, J., Gerardo, B., Silva, A. R., Pinto, P., Barbosa, R., Soares, S., ... & Freitas, S. (2021). Effects of restraining measures due to COVID-19: Pre-and post-lockdown cognitive status and mental health. *Current Psychology*, 1-10.
- Norton, A. H., & McCloskey, A. V. (2008). Modeling students' mathematics using Steffe's fraction schemes. *Teaching Children Mathematics*, 15(1), 48–54.
- Oates, G. (2011). Sustaining integrated technology in undergraduate mathematics. International Journal of Mathematical Education in Science and Technology, 42(6), 709-721.
- Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: Difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 1-27.
- Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic books.
- Patton, M. Q. (1990). *Qualitative evaluation and research methods*. SAGE Publications, inc.
- Perlis, A. (1962). The computer in the university. In M. Greenberger (Ed.), *Computers and the world of the future* (180–219). MIT Press.
- Prediger, S., Gravemeijer, K., & Confrey, J. (2015). Design research with a focus on learning processes: An overview on achievements and challenges. ZDM, 47(6), 877-891.
- Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, 45(5), 583-602.
- Rasmussen, C., & Blumenfeld, H. (2007). Reinventing solutions to systems of linear differential equations: A case of emergent models involving analytic expressions. *Journal of Mathematical Behavior*, 26(3), 195-210.
- Rasmussen, C., Dunmyre, J., Fortune, N., & Keene, K. (2019). Modeling as a means to develop new ideas: The case of reinventing a bifurcation diagram. *PRIMUS*, 29(6), 509-526.

- Rasmussen, C., Zandieh, M., King, K., & Teppo, A. (2005). Advancing mathematical activity: A practice-oriented view of advanced mathematical thinking. *Mathematical thinking and learning*, 7(1), 51-73.
- Renninger, K. A. (2009). Interest and identity development in instruction: An inductive model. *Educational Psychologist*, 44(2), 105-118.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, *13*(2), 137-172.
- Roorda, G., Vos, P., Drijvers, P., & Goedhart, M. (2016). Solving rate of change tasks with a graphing calculator: A case study on Instrumental Genesis. *Digital Experiences in Mathematics Education*, 2(3), 228-252.
- Rubin, B. C. (2007). Learner identity amid figured worlds: Constructing (in) competence at an urban high school. *The Urban Review*, *39*(2), 217-249.
- Sangwin, C. J., & O'Toole, C. (2017). Computer programming in the UK undergraduate mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, 48(8), 1133-1152.
- Satyam, V. R. (2020). Satisfying moments during the transition-to-proof: Characteristics of moments of significant positive emotion. *The Journal of Mathematical Behavior*, 59, 1-18.
- Sentance, S., & Waite, J. (2017, November). PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. In E. Barendsen, & P. Hubwieser (Eds.), Proceedings of the 12th Workshop on Primary and Secondary Computing Education (pp. 113-114).
- Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, 29(2-3), 136-176.
- Settle, A., Goldberg, D. S., & Barr, V. (2013). Beyond computer science: computational thinking across disciplines. In J. Carter, I. Utting, & A. Clear (Eds.), Proceedings of the 18th ACM conference on Innovation and technology in computer science education (pp. 311-312).
- Shah, N., Christensen, J. A., Ortiz, N. A., Nguyen, A. K., Byun, S., Stroupe, D., & Reinholz, D. L. (2020). Racial hierarchy and masculine space: Participatory in/equity in computational physics classrooms. *Computer Science Education*, 30(3), 254-278.

- Simon, M. A. (1995). Reconstructing mathematics pedagogy from a constructivist perspective. *Journal for Research in Mathematics Education*, 26(2), 114-145.
- Simon, M. A., Saldanha, L., McClintock, E., Akar, G. K., Watanabe, T., & Zembat, I. O. (2010). A developing approach to studying students' learning through their mathematical activity. *Cognition and Instruction*, 28(1), 70–112. doi:10.1080/07370000903430566
- Simon, M. A., & Tzur, R. (2004). Explicating the role of mathematical tasks in conceptual learning: An elaboration of the hypothetical learning trajectory. *Mathematical Thinking and Learning*, 6(2), 91-104.
- Sjøberg, S. (2010). Constructivism and learning. In P. Peterson, E. Baker, & B. McGaw (Eds.) *International Encyclopedia of Education* (pp. 485-490). Oxford: Elsevier.
- Sutherland, R. (1994). The role of programming: towards experimental mathematics. In R. Biehler, R. W. Scholz, R. Sträßer, B. Winkelmann (Eds.), *Didactics of mathematics as a scientific discipline*, (pp. 177-187). Springer.
- Schwartz, K. D., Exner-Cortens, D., McMorris, C. A., Makarenko, E., Arnold, P., Van Bavel, M., ... & Canfield, R. (2021). COVID-19 and student well-being: Stress and mental health during return-to-school. *Canadian Journal of School Psychology*, 36(2), 166-185.
- Tall, D. (1989). Different cognitive obstacles in a technological paradigm. In S. Wagner & C. Kieran (Eds.), *Research issues in the learning and teaching of algebra*. LEA.
- Tall, D. (1991). Advanced mathematical thinking (Vol. 11). Springer Science & Business Media.
- Tall, D. (1992). The transition to advanced mathematical thinking: Functions, limits, infinity, and proof. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 496-511). Macmillan.
- Theobald, E. J., Hill, M. J., Tran, E., Agrawal, S., Arroyo, E. N., Behling, S., ... & Grummer, J. A. (2020). Active learning narrows achievement gaps for underrepresented students in undergraduate science, technology, engineering, and math. *Proceedings of the National Academy of Sciences*, 117(12), 6476-6483.
- Treffers, A. (1987). *Three Dimensions: A model of goal and theory description in mathematics instruction the wiskobas project.* Springer.

- Trouche, L. (2004). Managing complexity of human/machine interactions in computerized learning environ- ments: Guiding students' command process through instrumental orchestrations. *International Journal of Computers for Mathematical Learning*, 9(3), 281–307.
- Tulis, M., Steuer, G., & Dresel, M. (2016). Learning from errors: A model of individual processes. *Frontline Learning Research*, 4(2), 12-26.
- van den Heuvel-Panhuizen, M. (2003). The didactical use of models in realistic mathematics education: An example from a longitudinal trajectory on percentage. *Educational Studies in Mathematics*, 54(1), 9-35.
- Vannorsdall, T. D., Brigham, E., Fawzy, A., Raju, S., Gorgone, A., Pletnikova, A., ... & Oh, E. S. (2022). Cognitive dysfunction, psychiatric distress, and functional decline after COVID-19. *Journal of the Academy of Consultation-liaison Psychiatry*, 63(2), 133-143.
- Vergnaud, G. (2009). The theory of conceptual fields. *Human Development*, 52(2), 83–94.
- von Glasersfeld, E. (1995). *Radical constructivism: A way of knowing and learning*. The Falmer Press.
- Voigt, M., Rasmussen, C., & Martinez, A. (2021). The refiguring of students' mathematical identities: A mixed methods study of three tailored calculus courses. *International Journal of Mathematical Education in Science and Technology*, 1-21. https://doi.org/10.1080/0020739X.2021.1940331
- Watson, C., & Li, F. W. (2014). Failure rates in introductory programming revisited. In A. Cajander, & M. Daniels (Eds.), *Proceedings of the 2014 conference on Innovation* & technology in computer science education (pp. 39-44).
- Weber, K. (2001). Student difficulty in constructing proofs: The need for strategic knowledge. *Educational Studies in Mathematics*, 48(1), 101-119.
- Weese, J. L., Feldhausen, R., & Bean, N. H. (2016, June). *The impact of STEM experiences on student self-efficacy in computational thinking*. 2016 ASEE Annual Conference & Exposition, New Orleans.
- Weston, T. J., Seymour, E., Koch, A. K., & Drake, B. M. (2019). Weed-out classes and their consequences. In *Talking about leaving revisited* (pp. 197-243). Springer, Cham.

- Willison, D. J., Keshavjee, K., Nair, K., Goldsmith, C., & Holbrook, A. M. (2003). Patients' consent preferences for research uses of information in electronic medical records: Interview and survey data. *BMJ*, 326(7385), 373.
- Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.
- Wing, J. M. (2014). *Computational thinking benefits society. 40th anniversary blog of social issues in computing*. Retrieved from http://socialissues.cs.toronto.edu/index.html%3Fp=279.html
- Wirkner, J., Christiansen, H., Knaevelsrud, C., Lüken, U., Wurm, S., Schneider, S., & Brakemeier, E. L. (2022). Mental health in times of the COVID-19 pandemic. *European Psychologist*. https://doi.org/10.1027/1016-9040/a000465
- World Economic Forum. (2016). The future of jobs: Employment, skills and workforce strategy for the fourth industrial revolution. *Global challenge insight report. Geneva: World Economic Forum.*
- Young, J. L., Young, J. R., & Capraro, R. M. (2018). Gazing past the gaps: A growthbased assessment of the mathematics achievement of black girls. *The Urban Review*, 50(1), 156-176.
- Zandieh, M., & Rasmussen, C. (2010). Defining as a mathematical activity: A framework for characterizing progress from informal to more formal ways of reasoning. *The Journal of Mathematical Behavior*, 29(2), 57-75.