Investigating Themes of Student-Generated Analogies

Colton Harper
University of Nebraska-Lincoln
Lincoln, Nebraska, United States
colton.harper@huskers.unl.edu

Ryan Bockmon
Northeastern University
Portland, Maine, United States
ryanbockmon@gmail.com

Stephen Cooper
University of Nebraska-Lincoln
Lincoln, Nebraska, United States
stephen.cooper@unl.edu

ABSTRACT

Student-generated analogies hold potential in facilitating understanding of abstract computing concepts, as they exercise valuable computing skills, such as abstraction, re-representation, and relational reasoning. Helping students develop their own analogies is difficult, and limited research exists on scaffolding for student-generated analogies in computing education and STEM education more broadly. This exploratory study examines an open-ended student-generated analogy process, for the concept of variable scope within an introductory computer science course. The primary goal of this study is to lay the foundation for scaffolding strategies to more fully realize the potential of student-generated analogy as an educational tool.

In this study, students engaged individually or in small groups to create analogies for describing variable scope. A brief training period and minimal scaffolding were provided, encouraging less restricted exploration of analogy development, while establishing a baseline for students' analogy development skills for future studies. This open-ended approach aligns with the current state of practice in STEM education for student-generated analogies. The analysis, grounded in analogical reasoning, revealed strengths, weaknesses, recurring challenges, providing a foundation for future work in developing targeted scaffolding strategies. Key findings such as difficulties in structural alignment, and strengths like high internal domain consistency, offer valuable insights for future targeted scaffolding strategies to be developed for improved analogy development in computing education.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

student-generated analogies; qualitative research; variable scope

ACM Reference Format:
Colton Harper, Ryan Bockmon, and Stephen Cooper. 2023. Investigating Themes of Student-Generated Analogies. In *Proceedings of the ACM Conference on Global Computing Education Vol 1 (CompEd 2023), December 5–9, 2023, Hyderabad, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3576882.3617914

CompEd 2023, December 5–9, 2023, Hyderabad, India

ACM ISBN 979-8-4007-0048-4/23/12…$15.00
https://doi.org/10.1145/3576882.3617914

1 INTRODUCTION

Computing education has seen increasing emphasis on fostering critical thinking and problem-solving skills, yet comprehension of abstract computing concepts remains a challenge for many students [9]. One promising pedagogical approach to address this challenge is the use of student-generated analogies, a tool that can facilitate understanding by exercising and further building essential computing skills like abstraction, re-representation, and relational reasoning [5, 8] in the process of learning computing concepts.

Research on the use of student-generated analogies in computing education has been limited, with most studies concentrating on instructor-provided analogies. Moreover, the available studies on student-generated analogies primarily focus on STEM education, revealing benefits such as increased engagement and deeper understanding of complex concepts [11, 14, 16, 18, 19].

Our study seeks to fill this gap by examining an open-ended student-generated analogy process for the concept of variable scope within an introductory computer science course. We focus on the quality and nature of student-generated analogies, aiming to discern patterns and recurring issues that may inform future improvement of the analogy development process and scaffold the design of teaching strategies. The research question guiding this investigation is: What are the inherent strengths, weaknesses, and recurring challenges observed in student-generated analogies for introductory computing concepts?

To address this question, we employed a qualitative research design, using qualitative coding and thematic analysis, grounded in analogical reasoning [8]. Our study's findings contribute to the field of computing education by providing insights into the potential benefits and challenges of using student-generated analogies. By evaluating these analogies from a course exercise, this work offers insights that can guide the refinement of the analogy generation process, and potentially lead to novel pedagogical strategies that enhance conceptual understanding in computing education. This work does not directly measure the effectiveness of these analogies in boosting understanding. Instead, it sets the stage for subsequent studies that can incorporate performance assessments, optimize design procedures, and holistically evaluate the benefits of student-generated analogies.

2 RELATED RESEARCH

This literature review provides some broader context as well as the theoretical foundations of our work. This includes exploring basic concepts and literature on analogical reasoning and constructivism, which are integral to this work—student-generated analogies within computing education. This concise review highlights some of the existing gaps in the systematic development and application of student-generated analogies in computing education.

2.1 Analogical Reasoning in Learning

Analogical reasoning, a cognitive strategy that entails drawing structural parallels between a familiar domain (the source) and a less familiar or unfamiliar domain (the target). Analogy use spans across diverse educational contexts, including computing education, where it has been key in fostering problem-solving abilities, enhancing comprehension of new information, and fostering new insights [1, 10, 17].

Gentner's Structure Mapping Theory (1983), asserts that the crux of analogical reasoning resides in a process called structural alignment, wherein the relational structure shared between the source and target domains is brought into focus [7]. Structural alignment emphasizes the relationships and correspondences between elements in the source and target domains rather than relying solely on superficial or attribute-based similarities. Structural alignment includes a one-to-one mapping constraint and the systematicity principle. The one-to-one mapping constraint suggests a unique match for every core element in the source with a core element in the target domain. The systematicity principle prefers interconnected relationships (systematic structures) over isolated ones, promoting the mapping of a system of relations that construe a coherent whole.

In the context of this study, a quality analogy would achieve a nearly one-to-one mapping and systematicity. It would not only align individual elements of variable scope with elements in the analogy domain but also capture congruent levels of abstraction across each comparison that makes up the overall analogy for variable scope.

2.2 Student-Generated Analogies in Computing Education

While the use of analogies in computing education is not new, most of the focus has been on instructor-provided analogies [6], such as the classic dining philosophers analogy for concurrency and synchronization [3]. However, recent observations show students naturally employing their own analogies in a concurrency course [2].

2.3 Theoretical Basis for Student-Generated Analogies in Computing Education

Notional machines, a term introduced in the context of computing education, refer to simplified, abstract models that assist learners in conceptualizing how a piece of code operates [4, 6]. They help students form a mental model of the inner workings of code execution, demystifying complex computational processes. Analogously, student-generated analogies can function as a form of personal notional machine, aiding students in their efforts to visualize and comprehend abstract computing concepts such as variable scope.

This pedagogical approach aligns with the principles of constructivist learning theory [15]. Constructivism posits that learning is an active, constructive process where learners are not passive recipients of information but active constructors of their knowledge. Accordingly, the act of creating analogies enables students to actively engage with the computing concept at hand, facilitating a deeper and more personalized understanding.

2.4 Variable Scope in Computing Education

In the realm of computing education, the concept of variable scope warrants our attention due to its crucial role in programming. It fundamentally influences the organization, readability, and correctness of code, thereby laying the groundwork for learning more advanced computing concepts. Despite its foundational importance, the layered complexity of variable scope, coupled with its potential for variation across different programming languages, often constitutes a significant learning challenge for students [12].

The inherent complexity of variable scope, along with its wide-ranging application, makes it an ideal subject for exploration using analogical reasoning. We have chosen this concept as the subject of our study, aiming to strike a balance between the complexity of the topic and our primary focus on the process of analogy development. Introduced relatively early in programming education, variable scope is a nuanced topic that provides students with a degree of familiarity while challenging them to engage in thoughtful exploration.

3 METHODOLOGY

This study employs a qualitative research approach, focusing on the exploration of student-generated analogies of the concept of variable scope in a college introductory computer science course. This work primarily aims to identify some initial strengths, weaknesses, recurring challenges observed in these analogies.

3.1 Study Context and Participants

The data collection for this study took place during the Fall semester of 2022 in an introductory computer science (CS1) course at a research-intensive university in the United States Midwest region. The section of the course in which this data collection took place in was made up of entirely non-computing majors. This CS1 course used Python and covered traditional CS1 concepts such as variables, variable types, loops, control structures, functions, data structures, basic algorithms, and an introduction to object-oriented programming. The course was conducted in-person over a 16-week semester, with two lectures and one lab session each week, each lasting 75 minutes. The enrolled students consisted of 28 individuals, 21 of whom consented to participate in this study. The participants had diverse academic backgrounds, ranging from freshman to senior level, and their majors spanned from humanities to STEM fields.

A neutral third party was responsible for all recruitment and data collection procedures, ensuring that participant information remained confidential from the instructor (and study first author) until the final course grades were submitted. Students were explicitly informed that their participation would not affect their grades or academic standing in any way and were included in the study on an opt-in bases. To maintain participant anonymity, the neutral third party organized students into groups of 2 or 3 individuals. Out of the 10 total groups in the course, 8 of the groups were made up exclusively of participants. The university's Institutional Review Board granted approval for conducting this study. This exercise was conducted as a regular graded course activity. Participation in the study did not result in additional work except for the consent form.

3.2 Analogy Development Training

To prepare students for the analogy generation exercise, a training session was conducted, encompassing an introduction to analogies, analogy structure, an analogy creation process, a practice session, and a group discussion.

3.2.1 Introduction to Analogies. The training began by motivating the purpose and role of analogies in learning complex concepts. This session was developed with the intent to help students understand how analogies can simplify and enhance the understanding of abstract concepts, but can also as a double-edged sword that must be crafted mindfully so as not to overextend or misuse them.

Students were next introduced to the anatomy, or structure, of an analogy. The aim of this was to familiarize students with the fundamental components of an analogy, including the source domain (the familiar concept), the target domain (the recently introduced computing concept), and the mapping between the two. Students were guided on how to identify these components in a given analogy from a set of common examples. Students were also made aware of the importance of maintaining a clear, consistent, and accurate structure throughout the analogy.

Subsequently, students were then introduced to a basic step-by-step process of creating analogies. The process was broken down into five core steps:

(1) **Identifying the target concept and its key characteristics:** This involves understanding the new or complex concept that the analogy intends to explain.
(2) **Brainstorming potential source domains that share similarities with the target concept:** This encourages divergent thinking, aiding the students in exploring various familiar concepts that could be used in the analogy.
(3) **Selecting the most appropriate source domain:** This emphasizes the need for careful selection of the source domain that best aligns with the target concept.
(4) **Mapping the similarities between the source and target domains:** This step focuses on creating a clear and accurate correlation between the two domains.
(5) **Identifying and addressing any differences or limitations in the analogy:** This step fosters critical thinking, helping students to recognize the boundaries of the analogy and address its limitations.

The practice session involved creating an analogy between phones and factories. This seemingly unusual pairing was chosen for its structural parallels, mirroring the cognitive demands students would face when developing analogies for abstract computing concepts. Both phones and factories offer numerous components and processes that can be related, providing a rich ground for analogical thinking. This analogy challenges students but not to the point of overwhelming them. This allowed students to practice identifying structural similarities and mapping concepts between different domains. At the end of the training a group discussion and reflection was facilitated by the instructor.

3.3 Variable Scope Lecture

Before the main analogy generation exercise, a lecture was held to help students understand the concept of variable scope in Python. The lecture covered essential subtopics to provide a comprehensive view of variable scope. The following subtopics were included:

(1) **Variable Accessibility:** This concept explains where a variable can be accessed within a program. It underscores how the scope of a variable determines where it can be read or modified. As part of this, the **LEGB Rule** was introduced. This rule outlines the hierarchy Python follows when a variable is referred to in a program. It stands for Local, Enclosing, Global, and Built-in, denoting the order in which Python searches for variable references.
(2) **Variable Lifetime:** The lecture also covered how long a variable exists in memory during program execution, which is known as the variable's lifetime.

To help the students better understand these concepts and to help them apply them practically, the lecture was interspersed with Python-based code examples specifically designed to challenge the students' understanding of variable scope. These hands-on activities required students to apply their knowledge of variable scope in real coding situations. The students worked on these code examples following a "think-pair-share" approach. This approach was integrated to provide immediate practice of the newly introduced concept and also to help foster a more collaborative learning environment, setting the stage for the following analogy generation task.

3.3.1 Analogy Generation Exercise. After the lecture on variable scope, the students were provided their primary task, to develop an analogy for the concept of variable scope. The analogy generation exercise was conducted in a lab session held on the same day. This exercise was designed to engage students in a practical application of their understanding of variable scope, creating analogies related to the concept. The task was purposely left open-ended, allowing students to draw from a wide array of source domains for their analogies based on their own experiences and knowledge.

Students were organized into their previously assigned groups for the analogy generation exercise. Within these groups, students could decide whether they wanted to work collaboratively to develop a single group analogy or individually create their own. Regardless of the choice, all students were required to actively participate in group discussions throughout the development process, presenting their initial analogies, and receiving feedback from their peers.

The assigned groups were created to help foster collaborative learning, encouraging students to give and receive constructive feedback and to articulate their thoughts clearly. This collaborative process was also intended to help students help each other identify any misconceptions early on, ideally helping to enhance the quality of the analogies created.

3.3.2 Analogy Refinement and Presentation. Subsequent to the lab session, as a homework assignment, students were tasked with refining their analogies based on the feedback received and their reflections on the exercise. Students that developed an individual analogy completed the out-of-class portion independently. Students that co-developed an analogy during lab continued working with their assigned peer group to refine their analogy. All students were required to develop a 5-10 minute digital presentation. The presentations were structured to cover five key aspects: (i) an overview of the analogy, (ii) similarities between the source and target domains, (iii) differences, (iv) limitations of the analogy, and (v) the process of analogy formation.

Each student or group was responsible for recording their presentation outside of class time. The presentations were not delivered live, allowing students to craft a thoughtful, well-structured explanation of their analogy without time pressure. This resulted in a collection of 4 individual and 6 group analogies. One group contained two students. The other groups had three students each.

These recorded presentations were transcribed verbatim, providing the primary data set for this study. These transcriptions were prepared for further analysis, setting the stage for the extraction of key statements and subsequent coding processes.

4 DATA ANALYSIS

4.1 Qualitative Coding

A qualitative coding process was employed to systematically analyze the transcripts. This task was shared between two independent coders to promote reliability and reduce bias. Coding was conducted at two different levels: (i) individual statement level and (ii) aggregate transcript level.

The coders identified and extracted relevant statements from the transcripts. These statements were then classified into two categories: comparison statements and statements of limitations. For individual statement-level coding, four primary criteria were considered. The "precision of mapping" evaluated the clarity, specificity, and potential ambiguity of how a statement related to the subconcept of variable scope. The "completeness of mapping" checked how extensively the statement addressed the critical facets of variable scope. "Clarity of mapping" assessed the understandability and distinctness of the statement's connection to the subconcept. The "accuracy of mapping" determined the correctness of a statement in its representation of the subconcept of variable scope.

Aggregate level coding involved assessing the depth of coverage of each subconcept of variable scope, and the overall structural alignment, completeness, and clarity of the entire analogy. A set of codes defined on a 5-point Likert scale was used during the coding process to capture the variations in the quality of the analogies.

Codes were split into two categories: content codes and structural codes. Content codes addressed the analogy's content regarding the target concept, variable scope, while structural codes examined the internal consistency and potential contradictions within each comparison statement. The collective set of comparison statements was used to assess the overall structural alignment of the analogy.

4.2 Inter-rater Reliability

Prior to open coding of the transcripts, the authors performed a calibration exercise by jointly coding one transcript. This initial collaboration allowed for any discrepancies to be discussed and resolved, establishing a consistent approach for the subsequent coding process. Inter-rater reliability for the ordinal data was then evaluated using the Intraclass Correlation Coefficient (ICC), with reference to the guidelines suggested by Koo and Li [13]. In the aggregate-level analysis, where each transcript was coded as a whole, the inter-rater reliability for the average ratings of the first two authors, as assessed by ICC(3,k), was 0.841. This falls within the range of 0.75 to 0.90, indicating 'good' reliability according to the guidelines [13].

In the statement-level analysis, where each comparison and limitation statement was independently extracted and coded, the inter-rater reliability, as assessed by ICC(3,k), was 0.784. This also indicates 'good' reliability, reflecting a consistent level of agreement between the authors in their coding of individual statements. These results highlight the consistency and reliability of the coding process at both the aggregate and detailed statement level, reinforcing the robustness of our findings.

4.3 Thematic Analysis

Building on the initial qualitative coding, the thematic analysis was conducted to discern prevalent patterns within the student-generated analogies. This process was essentially a deeper dive into the coding results, seeking patterns of high and low scores that might point to recurring strengths and weaknesses in the students' analogy creation process.

The initial step of the thematic analysis involved a comprehensive review of the coding scores. This step facilitated an overview of the data landscape, which was instrumental in gaining a more holistic understanding of the results.

Subsequently, the process focused on identifying patterns within the coding scores. High scores were interpreted as strengths, reflecting areas where students showed a clear understanding and effective usage of analogies. Low scores, on the other hand, were seen as potential areas of weakness, where students might have struggled with analogy creation. Both types of patterns were recorded and analyzed for their significance and frequency.

The next phase involved consolidating the identified patterns. Patterns that occurred more frequently across multiple analogies were considered as major themes. These themes represented common trends in students' analogy creation process, and were thus considered of primary importance. Throughout this process, the thematic analysis was guided by the data derived from the initial coding process. This ensured that the themes were grounded in empirical data.

5 RESULTS

5.1 Description of Student-Generated Analogies

In the analogy generation exercise, participants could create individual analogies (IA) or create analogies in groups (GA) of the concept of variable scope in Python. Table 1 provides a high-level overview of these analogies. Due to space constraints, this tabular overview of the analogies, while limited, is intended to provide a high-level snapshot of the core structure for the analogies students generated. Much of the richness of each analogy is not captured in this table (e.g., comparisons to variable lifetime), however the subsequent thematic analysis offers some more insights into this richness.

5.2 Content-Based Themes

Incomplete Mappings: The majority of student-generated analogies had low completeness scores, indicating a focus on only one

Table 1: Summary of Student-Generated Analogies

Analogy ID	Analogy Theme	Built-in Scope	Global Scope	Enclosed Scope	Local Scope
IA1	Personality Change	Base Personality (Consistently present)	Behavior in Different Environments	Behavior in Specific Spaces (e.g., at checkout)	Behavior Toward Specific Individuals
IA2	Information Accessibility in a School	School Rules (Accessible anytime)	School-wide Announcements	Classroom Teaching	Whispering in Class
IA3	Sine Function	Not Addressed	Sine Function	Radians (Used to define sine function)	Pi in Radians (Most specific unit)
IA4	Meal Preparation	Not Addressed	Week's Meal Plan	Meals Planned on a Given Day	Individual Meals
GA1	US Government Structure	United Nations (Constant presence)	President (Overarching control)	Governor (Larger reach, nested)	Mayor (Limited jurisdiction)
GA2	Video Game Taxonomy	Collection of Available Games	Gaming Modes (Single vs Multiplayer)	Specific Game (e.g., Roblox)	Individual Quests Within the Game
GA3	Road Trip Stops	Not Addressed	States Driven Through	Mini-stops (Nested within states)	Specific Cities/Towns Visited
GA4	Quadrilateral Classification	All Quadrilaterals	Parallelograms	Rectangles/Rhombuses	Square
GA5	Whiteboard Usage	Not Addressed	Entire Whiteboard	Sections of a Whiteboard	Notes in Specific Section
GA6	Shopping Mall Structure	Entire Mall	Entire Store	Department Within Store	Specific Counter in Department

aspect of the concept of variable scope. The analogies generally concentrated on either variable accessibility or variable lifetime, with only a single analogy meaningfully covering the concept of variable lifetime. As one group put it, "*...this represents the returning [values] from the functions... as you can see, the variable... has the longest lifetime. The rest of the variables on the board... are not returned so their lifetime is shorter.*" -GA5

Analogy Overview Abstraction Level: Many students struggled to provide a succinct, one-line description of their analogy that accurately captured the underlying concept of variable scope at an appropriate level of abstraction. "*...variable scope is like cultural norms... We have the highest level which is the built-in functionality. That in this analogy is considered to be the base personality, or how you act when nobody is around, and you are just by yourself. The next level down is your global personality. That's going to be how you act in a specific environment...* " - IA1

5.3 Structure-Based Themes

Surface-Level Limitations: Limitations tended to simply mention the general parts of variable scope their analogy does not cover. Most limitations identified by students did not address structural aspects of the analogies. "*This analogy is not perfect, obviously... The lifetime of the variable is not described accurately using this analogy; it's not described at all. This analogy only describes the accessibility*" - IA1. "*Whiteboards are also entirely manual. Requiring something or someone to take a marker and drag it across the board. Computers on the other hand are fast and automatic.*" - GA5

Confusion over "Source" and "Target": Some students demonstrated confusion over the terms "source" and "target," with some even creating analogies of these terms rather than the intended concept of variable scope. This suggests a potential misunderstanding of the role of source and target within the context of analogical reasoning. "*...the source which is the elections, and the target which is getting into office.*" - GA1. "*The source and target in the analogy road tripping are both physical places and can vary in distance in between. A lot like variable scope, distance can also vary in between and there are a lot of different routes.* " - GA3

Difficulties with Structural Alignment: Some students struggled to maintain consistent mappings at the appropriate levels of abstraction. There were also incongruencies in the structural mappings where some students incorporated strong core structural mappings, while also demonstrating a tendency to include surface-level or less related mappings indicating difficulties in maintaining conceptual coherence. "*we need other tools to clean and write on whiteboard. You can't write on a whiteboard without things such as a marker or wipes to clean off everything.*" - GA5

High Internal Consistency with Some Inconsistencies: Analogies typically showed high internal consistency and low internal conflict, indicating that while students may have had difficulties in creating a coherent analogy at a more abstract level, they were generally successful in maintaining consistency within individual comparison statements, as evident in Table 1.

Some analogies showed evidence of internal inconsistency. For instance, GA1's analogy drew a comparison between the structure of the US Government and variable scope. The mapping of global, enclosed, and local scopes to the roles of President, Governor, and Mayor respectively, fit reasonably well within the source domain. The mapping to the built-in scope had two significant issues: (i) it is not typically considered part of the US government structure, and (ii) it represents an organization (United Nations) rather than an individual leadership role. This inconsistency also indicates room for improvement with respect to structural alignment.

6 DISCUSSION

In the interest of driving future improvements in teaching and learning, the results presented in this study primarily highlight the weaknesses and challenges observed in the student-generated analogies. It is important to acknowledge that this focus does not imply the absence of strengths or successful instances of structural alignment in the student-generated analogies. Indeed, many students demonstrated a sophisticated understanding of variable scope through their analogies, with high internal consistency and effective structural alignment. However, given the goal of this research is to enhance pedagogical methods and improve student comprehension of variable scope, we have chosen to emphasize areas that present opportunities for growth and improvement.

Our thematic analysis identified several emerging patterns and recurring issues, of note the **Misinterpretation of Analogies** and **Incomplete Mappings**. These themes suggest the need for further instructional guidance that focuses on the process of abstraction and the creation of well-structured analogies. Misinterpretations and incomplete mappings indicate that students may struggle with accurately and completely translating between the source and target domains, which can lead to erroneous understanding of the target concept, in this case variable scope.

One surprising finding, **Confusion over "Source" and "Target"** theme, illuminated a distinctive complexity within the analogy-making process. In a notable example, a student constructed an analogy that seemed to be more focused on the literal interpretations of the terms "source" and "target" in the context of a road trip analogy, rather than aligning these terms with the actual abstract concept - variable scope. This meta-analogical approach could be seen as a misdirection in their analogy creation, suggesting that the student may have misunderstood the concept of source and target. This issue emphasizes the necessity for guidance on the correct identification and dissociation of the components of an analogy. One way to accomplish this could be to provide a set of small Python-specific examples, allowing students to practice identifying the source and target or other parts within a variety of analogies. This would aid students in separating the specific terms used in the analogy from the overall conceptual roles they represent.

Additionally, the **Difficulties with Structural Alignment** theme suggests a common challenge in maintaining consistent mappings at the appropriate levels of abstraction. This observation signals the necessity for assistance that supports students in aligning their analogies' structure maintaining consistent levels of abstraction throughout the mappings of their analogy to help facilitate more accurate transfer of knowledge between the source and target domains. Additionally, to take steps toward a more complete one-to-one mapping, the use of multiple source domains could be explored.

A notable strength observed in the student-generated analogies was their **High Internal Domain Consistency and Minimal Internal Conflict**. Students demonstrated an inherent aptitude for creating analogies with consistent internal mappings and minimal contradiction among individual statements. Despite other challenges, students seem to be capable of maintaining thematic coherence in their analogies. Given this strength, instructional strategies could prioritize addressing other identified challenges while leveraging students' inherent ability to maintain thematic coherence.

The use of spoken language as the primary medium for presenting analogies may have inadvertently permitted some degree of ambiguity or lack of clarity. Students may have interpreted this format as being more lenient towards unclear or ambiguous language, which, in turn, could have affected the precision of their analogy mappings. Alternate representation methods and clearer guidelines for explicit mapping may help.

7 LIMITATIONS

Demographic and Sample Size: One limitation is the limited demographic information collected and the small sample size. This makes it challenging to analyze the impact of demographic differences on the students' tendencies when generating analogies and should be explored in future work. Similarly, the potential impact of differences in students' major or academic level was not explored in this study. **Concept Choice:** The focus on variable scope as the sole concept for analogy creation could also have influenced the results. Other computing concepts might elicit different analogy creation strategies and outcomes, given varying levels of concept complexity or student familiarity.

Individual vs. Group Analogies: We observed no glaring differences in quality between individual and group-generated analogies. This could be a result of our methods, or due to the masking effects of group discussion on the final analogy output. We found similar variances in quality within both individual and group analogies. This could be a fruitful area for future investigation.

Lack of Student Voice: Due to space constraints, this study did not fully capture the richness and nuance of the original student-generated analogies. The absence of students' own language and expressions in presenting the analogies may limit the depth of insights gained from this work.

8 FUTURE WORK

Refining the Analogy Development Process: This study serves as foundational work in identifying initial challenges and strengths in student-generated analogies. Our aim was to identify "low-hanging fruit" that can be addressed to help make the analogy generation process smoother and the resulting analogies more robust. Future work can build on these findings to develop formal guidance and scaffolding strategies to improve the analogy creation process. **Educational Value of Student-Generated Analogies:** This study also sets the stage for future research to focus more extensively on the actual educational value of student-generated analogies.

Our work serves as a foundational exploration into the areas for improvement for student-generated analogies in introductory computing education. Importantly, it also offers a base model for facilitating the generation of such analogies in the computing classroom. We anticipate these findings will steer future efforts in scaffolding and guiding students in creating effective analogies, thereby enhancing their conceptual understanding.

REFERENCES

[1] Briana Bettin and Linda Ott. 2021. Frozen in the Past: When it Comes to Analogy Fears, It's Time For Us to" Let it Go". In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. 359–365.

[2] Briana Bettin, Linda Ott, and Julia Hiebel. 2022. Semaphore or Metaphor? Exploring Concurrent Students' Conceptions of and with Analogy. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*. 200–206.

[3] Edsger W Dijkstra. 1971. Hierarchical ordering of sequential processes. *Acta informatica* 1 (1971), 115–138.

[4] Benedict Du Boulay. 1986. Some difficulties of learning to program. *Journal of Educational Computing Research* 2, 1 (1986), 57–73.

[5] Reinders Duit et al. 1991. On the role of analogies and metaphors in learning science. *Science education* 75, 6 (1991), 649–672.

[6] Sally Fincher, Johan Jeuring, Craig S Miller, Peter Donaldson, Benedict Du Boulay, Matthias Hauswirth, Arto Hellas, Felienne Hermans, Colleen Lewis, Andreas Mühling, et al. 2020. Notional machines in computing education: The education of attention. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*. 21–50.

[7] Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science* 7, 2 (1983), 155–170.

[8] Dedre Gentner and Linsey Smith. 2012. Analogical reasoning. *Encyclopedia of human behavior* 2 (2012), 130–136.

[9] Mark Guzdial. 2015. Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics* 8, 6 (2015), 1–165.

[10] Keith J Holyoak, Paul Thagard, and Stuart Sutherland. 1995. Mental leaps: analogy in creative thought. *Nature* 373, 6515 (1995), 572–572.

[11] Fredrik Jeppsson, Jesper Haglund, Tamer G Amin, and Helge Strömdahl. 2013. Exploring the use of conceptual metaphors in solving problems on entropy. *Journal of the Learning Sciences* 22, 1 (2013), 70–120.

[12] Maria Kallia and Sue Sentance. 2017. Computing teachers' perspectives on threshold concepts: Functions and procedural abstraction. In *Proceedings of the 12th workshop on primary and secondary computing education*. 15–24.

[13] Terry K Koo and Mae Y Li. 2016. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of chiropractic medicine* 15, 2 (2016), 155–163.

[14] Rachael Anderman Lancor. 2014. Using student-generated analogies to investigate conceptions of energy: A multidisciplinary study. *International Journal of Science Education* 36, 1 (2014), 1–23.

[15] Jean Piaget. 1970. *Piaget's theory*. Vol. 1. Wiley New York.

[16] Kim M Pittman. 1999. Student-generated analogies: Another way of knowing? *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching* 36, 1 (1999), 1–22.

[17] Joseph P Sanford, Aaron Tietz, Saad Farooq, Samuel Guyer, and R Benjamin Shapiro. 2014. Metaphors we teach by. In *Proceedings of the 45th ACM technical symposium on Computer science education*. 585–590.

[18] Lesley Spier-Dance, Jolie Mayer-Smith, Nigel Dance, and Samia Khan. 2005. The role of student-generated analogies in promoting conceptual understanding for undergraduate chemistry students. *Research in Science & Technological Education* 23, 2 (2005), 163–178.

[19] Rand J Spiro, Paul J Feltovich, Richard L Coulson, and Daniel K Anderson. 1988. *Multiple analogies for complex concepts: Antidotes for analogy-induced misconception in advanced knowledge acquisition*. Number 439. University of Illinois at Urbana-Champaign.