6-2023

Pedagogical Prisms: Toward Domain Isomorphic Analogy Design for Relevance and Engagement in Computing Education

Briana Christina Bettin
Michigan Technological University, bcbettin@mtu.edu

Linda Ott
Michigan Technological University, linda@mtu.edu

Recommended Citation

Bettin, B. C., & Ott, L. (2023). Pedagogical Prisms: Toward Domain Isomorphic Analogy Design for Relevance and Engagement in Computing Education. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, 1*, 410-416. http://doi.org/10.1145/3587102.3588830
Retrieved from: https://digitalcommons.mtu.edu/michigantech-p2/44

Follow this and additional works at: https://digitalcommons.mtu.edu/michigantech-p2
Part of the Computer Sciences Commons

Pedagogical Prisms: Toward Domain Isomorphic Analogy Design for Relevance and Engagement in Computing Education

Briana Bettin
Michigan Technological University
Houghton, MI, USA
bcbettin@mtu.edu

Linda Ott
Michigan Technological University
Houghton, MI, USA
linda@mtu.edu

ABSTRACT

Analogy is a frequently leveraged pedagogical tool used across many disciplines, with computing being no exception. Computing education researchers, however, have raised concerns regarding the limitations of analogy. One obvious concern is the relevance of any given analogy to learners. Designing relevant analogies can greatly increase student engagement with the problem space by centering examples on their lived experiences. Relevant analogies can also facilitate learners in building appropriate connections as they explore novel concepts.

Designing relevant analogies is an ongoing process which requires understanding the learners' context. It is unlikely that any given analogy will be "universally" relevant across learners, problems, and decades. This poses an interesting problem for instructors: how can we adapt analogies to learners so that they are engaging and relevant, while maintaining the desired pedagogical value?

This position paper presents a framework for analogical design in computing education. We leverage what is described here as *domain isomorphism*: the ability to modify the domain in which an analogy is based while still maintaining the intended analogical structure. Through this design approach, we suggest that instructors and researchers may confidently, and in a timely fashion, redesign analogies to be relevant and engaging for a given group of learners.

CCS CONCEPTS

• **Social and professional topics** → **Computer science education**;

KEYWORDS

analogy, metaphor, relevance, engagement, culturally responsive teaching, domain isomorphism, contextualization, design tool, computing education research, cognitive and learning sciences

ACM Reference Format:
Briana Bettin and Linda Ott. 2023. Pedagogical Prisms: Toward Domain Isomorphic Analogy Design for Relevance and Engagement in Computing Education. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2023), July 8–12, 2023, Turku, Finland.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3587102.3588830

ITiCSE 2023, July 8–12, 2023, Turku, Finland

ACM ISBN 979-8-4007-0138-2/23/07.
https://doi.org/10.1145/3587102.3588830

1 INTRODUCTION

Analogy is a common pedagogical communication tool used by instructors and researchers alike to elicit comprehension of often challenging concepts. While analogies may be leveraged to aid understanding, this understanding relies on the ability to reason about the generalized abstraction the analogy represents. This can be exceptionally difficult if the domain the analogy is drawn from is not fully understood by, or relevant, to the learner. Being able to develop analogies which are relevant to learners is important for successful learner engagement and understanding. Given diverse cultures, prior knowledge, and lived experiences, instructors may find that previously successful and pedagogically sound analogies are not useful with different learners or in new contexts. An instructor's ability to take useful analogies and change their domain to one which is more applicable may allow these analogies to find new relevance and value among learners. This may also foster greater inclusion of diverse perspectives and experiences within analogies used in the classroom by centering the focus on if the abstraction mapping holds, regardless of domain. The ability to "shift" the domain of an analogy is termed by the authors as *domain isomorphism.* We present domain isomorphism itself as a potentially valuable tool within computing education. We will ground discussion of domain isomorphism in examples using the analogy design framework presented.

2 BACKGROUND

2.1 Computing Education and Analogy

Computing Education Research (CER) has had a somewhat fraught relationship with analogy. Despite the established value of analogy as a cognitive reasoning process [2, 26, 29, 30, 32, 33, 36] and a means of communicating educational concepts [11, 12, 19, 28, 34, 45, 49], research specifically targeting analogy within CER has been rather limited. While CER studies focusing on computational analogy do exist [7, 10, 16, 24, 39, 54], broader discussion is often targeted at perceived deficits of analogy as a pedagogical tool [17, 21, 37, 53].

Shifting tides have advocated benefits of analogy in spite of these perceived difficulties. Guzdial's Communications of the ACM post [35] argued for re-evaluating our perception of Dijkstra's [17] historic opposition to analogy in computing education. Bettin and Ott [4] provided a review of the cognitive value of analogy, analogy use across STEM disciplines, and proposed "roots" of analogy reservations within CER, and also explored how students in an upper level computing course leveraged analogy to understand two major course programs [5]. Harper [38] investigated approaches to help scaffold and evaluate student-generated CS1 analogies. Methods for exploring pedagogical viability of comics, which are an example of a visual analog communication, have also been explored within computing contexts [3, 57].

Notional machines describe expert conceptual models, which as conceptual models used for instruction, are by nature analogies. Literature on notional machines is increasingly popular within CER [13, 14, 20, 41, 56]. However, notional machines being analogical tools is inconsistently noted within the research. As such, relevant methods and approaches from analogical research may not be applied to notional machines and similarly, perceptions of analogy can be somewhat disjoint from that of notional machines with CER.

2.2 Culturally Responsive Teaching

A student's culture shapes their pedagogical environment, their perceptions, and their motivations within the classroom [52]. Culturally responsive teaching [25, 48] requires that instructors consider how culture and lived experiences can impact pedagogical outcomes and perspectives. Additionally, culturally responsive teaching provides diverse examples, perspectives, and praxis throughout the curricular material. Work in culturally *relevant* pedagogy [43] is also considered foundational within a shared, ever-growing landscape and history [1, 9, 15, 46] of culturally conscious approaches.

The National Academies Press expands on designing "authentic experiences" in order to foster interest and competency in computing [50]. This suggests the importance of "real world" situated and culturally responsive approaches. Cultural responsiveness within computing has been explored through a variety of learning environments and targeted activities [18, 40, 47].

Pedagogy which centers relevant examples to student experiences and culture requires understanding. Cultural competency has been found to be historically lacking within computing education [59], though this space is growing [42, 51, 55, 58, 60]. Within all spaces, communication can help improve understanding. Dialogic methods, in particular, can aid in understanding, synthesis, and perspective as conversation participants refine their understanding of each other through this "gap closing" [44]. Thus, instructors within computing who foster greater communication and discussion with and among students may find better success in understanding their students and being able to develop relevant examples. STEM classrooms more generally have seen effectiveness in dialogic methods creating greater understanding of the communicated concepts [22, 23].

3 AN ANALOGY DESIGN WORKSHEET TOOL

The **Outlining Programming Analogy Layout (OPAL)** design worksheet was developed to help computing educators and education researchers focus on the purpose of their communication when creating or using an analogy. OPAL helps specify programming analogies centered on the structure rather than on elements alone. An example of a completed OPAL design worksheet is shown in Table 1. OPAL can be leveraged across computing courses to plan and review analogy design.

3.1 OPAL's Theoretical Underpinning

OPAL's scaffolded approach to analogy design is guided by seminal work in cognitive and learning sciences. Gentner's work on the cognitive process of structure mapping for analogical reasoning focuses on relational structure over elements [26–29, 31, 32]. Through Gentner's work, it can be seen that a "good" analogy establishes strong *relationships* between consistent elements (often referenced as entities in the literature). Analogies mapping key relationships across their elements elevate their relational *systematicity* and preserve an overall structural schema. Through this lens, discussion of whether an analogy is "good" or "bad" should focus on the in-context structure it establishes. A design tool such as OPAL, therefore, should encourage paralleling relationships across the source and base domains.

To summarize, a well-structured analogy is designed with:

(1) recognition of a concept to be communicated
(2) identification of the target domain's entities
(3) identification of the source domain's entities
(4) knowledge of the relations to be represented by the domains
(5) comparable relation and entity representation across both domains

OPAL has five key components to guide analogy design:

(1) Identification of the context for an analogy
(2) Target domain (programming) procedure
(3) Source domain (analogous) procedure
(4) Clearly defined relationships (relational arc) comprising each domain's procedure
(5) General structure relating the domains of both procedures

3.1.1 How Can a Component Based Approach Help? Gick and Holyoak [33] provided valuable groundwork for choosing to frame analogy "segments" across relationships. They adapted a Means-Ends Analysis (MEA) utilized by Carbonell [8] to compare analogous components within their study design. While this approach was used to evaluate an existing analogy (not create one), this provided guidance on analogy form analysis.

OPAL is focused on describing some **observable process** occurring across the source and target domains. OPAL's approach considers *actions* and their *results*. This helps achieve development of systematic relations: *"If X action results in Y in the target domain, what action achieves comparable results to the entities in the source domain?"* Explanatory analogies within education often compare sources to targets in order to explain expected or resultant phenomena in the target: *"Here, [target] A acts sort of like [source] B [...]"* or *"What happened here is kind of like [source] C [...]"*. This focus also requires analogy to be situated. Programming actions and computing phenomena do not occur in a vacuum — some context exists that motivates the use of analogy in educational settings.

OPAL's focus on "action and reaction" scenarios fits programming contexts: an action we engage with and observe the results of. Instead of focusing on element-based analogy claims such as "a variable is like a box", OPAL re-frames the discussion as *"what specific process am I explaining in which a variable and a box behave similarly?"* This framing requires appropriate *relational structure* be present in the analogy's design. Not every scenario we may analogize with variables translates to "a box" (such as, to use another entity-based claim, when "a variable is like a label"). By centering the process of specific contexts, solid analogies *for that phenomena* are more easily identified through their relational structure.

Table 1: Design Worksheet Example: Final Array Position at Length-1—A Hallway

Identification of Analogy Context		
Misconception	Index values are zero to the length of the array	
Desired Knowledge	Index values are zero to the length of the array - 1	
Comparison of Analogy Procedure Across Domains		
	Source Domain	Target Domain
Domain	A Hallway	Programming (Java, Arrays)
Precondition	Leave end-of-hall hotel room and stand in hallway with rooms are along one side.	Have a defined array to iterate over inside a looping control flow in your code.
Required Action	Your door is zero steps away. Each step takes you to another door. Count as you move.	Start at element position zero and increment by one each iteration.
Postcondition	It will have taken doors-1 steps to reach the end of the hallway.	The last element in the array will be iterated to at index position length-1.
Constraints	None.	None.
Analysis of Common Structural Elements		
Precondition	Being at start point of a collection of elements.	
Required Action	Incrementing to each element one movement at a time.	
Precondition	The amount of movements required is number of elements - 1.	
Constraints	None.	

4 OPAL AND DOMAIN ISOMORPHISM

An analogy maps "source" (known) knowledge to "target" (to learn) knowledge. These sources and targets are referred to as "domains" for the analogy. The concept of **domain isomorphism** then is used to refer to the ability to change one source domain to another source domain (or target to another target) while still preserving the analogical mappings of the remaining target (or source).

Given that OPAL centers analogies with a programming target, target domain isomorphism is not relevant to this paper. However, source isomorphism would allow for instructors to still leverage the same programming concept as target while changing the source knowledge to more relevant (or simply additional) examples. Being able to generate new sources may allow for instructors to incorporate greater cultural responsiveness, and to adapt to learner experiences and culture when a source is unknown to the student. Beyond finding some single source that is relevant, leveraging multiple source domains when thinking analogically has demonstrated pedagogical value [28, 45, 49]. In several instances, this value appears to stem from the necessary comparison and abstraction used for analogical reasoning. Additional source domains allow confounding details that may exist within a singular source domain to be more easily abstracted away through attempts to find commonality across the examples.

The "Analysis of Common Structural Elements" portion of OPAL is the component which can best encourage and validate domain isomorphism. With a well-reasoned analogical structure, additional source domains can be generated by validating them against this structure. This allows "pivoting" source domains with confidence that the analogy is still well-formed. If the critical target entities and relationships for understanding intended meaning are consistent with the new source, that source is able to effectively map to this analogical context.

4.1 Common Structure Analysis

The final design step in using OPAL is an abstraction exercise, asserting the **general structure** of the analogy by characterizing the relationship between the target and the source. This process encapsulates how reasoning about the source domain is expected to generalize to understanding in the target domain. Arrays and hallways in Table 1 both correlate to the general concept of numbered and ordered locations where the numeric ordering starts from some fixed beginning. In both of these cases, if we start from the element at the beginning, we do not need to move to view the item. If each "movement" allows us to see one more item, we will thus see all items with "number of items minus one" movements.

4.2 Domain Isomorphism was Identified With, But is Not Exclusive to, OPAL

OPAL is described and leveraged in this paper in order to provide concrete examples which showcase common structural elements that promote domain isomorphism. Finding isomorphic analogies was of interest when developing OPAL, but it was through using OPAL that a path forward was identified. As a result, OPAL is the central tool in how we came to and illustrate domain isomorphism and its benefits. Despite the presentation using OPAL, domain isomorphism is ultimately a property of the *analogy* itself.

5 AN OPAL ISOMORPH EXAMPLE

A relevant example of domain isomorphism's value occurred while we were working to apply OPAL developed analogies within our CS1 course's labwork. Example analogies had been developed for instructional use during each week, with the week on File I/O being no exception. While the examples that had been developed for this week posed no notable structural issues for the instructional team, some source domains posed significant issue at the time of use. The original example source domains for analogies from the File I/O week referenced a Pandemic, Lockdown, and Family Illness. Each

Table 2: Example of a File I/O Based Domain Isomorphism: Pandemic to Nuclear Radiation

Domain Isomorphic Original: Try Blocks - Pandemic		
Identification of Analogy Context		
Misconception	Only first instance of possible exception needs to be in a try block	
Desired Knowledge	Any code that could be affected by the exception must be in a try block	
Comparison of Analogy Procedure Across Domains		
	Source Domain	Target Domain
Domain	Pandemic	Programming (Java, File Exceptions)
Precondition	A contagious disease is identified.	Code that can throw an exception.
Required Action	Anything that could spread the disease is quarantined and sanitized for safety.	All code the exception can affect is in a try block.
Postcondition	The disease should not be able to spread. The situation is contained and able to be handled.	The exceptional circumstance should be safely contained for catching and handling.
Analysis of Common Structural Elements		
Precondition	A potentially dangerous situation.	
Required Action	Anything that could be affected has cautionary measures enacted.	
Postcondition	The situation should be appropriately contained for handling.	
Constraints	None.	

Domain Isomorphic Revision: Try Blocks - Nuclear Radiation		
Identification of Analogy Context		
Misconception	Only first instance of possible exception needs to be in a try block	
Desired Knowledge	Any code that could be affected by the exception must be in a try block	
Comparison of Analogy Procedure Across Domains		
	Source Domain	Target Domain
Domain	Nuclear Radiation	Programming (Java, File Exceptions)
Precondition	An unusual substance that emits toxic nuclear radiation is located.	Code that can throw an exception.
Required Action	Any objects the substance touched that could also have radiation must be contained.	All code the exception can affect is in a try block.
Postcondition	The radiation should not be able to spread. The situation is contained and able to be handled.	The exceptional circumstance should be safely contained for catching and handling.
Constraints	None.	None.
Analysis of Common Structural Elements		
Precondition	A potentially dangerous situation.	
Required Action	Anything that could be affected has cautionary measures enacted.	
Postcondition	The situation should be appropriately contained for handling.	
Constraints	None.	

of these analogies was developed prior to the COVID-19 pandemic, which — apart from the terrible loss of life that occurred on a global scale — caused mass stay-at-home expectations and isolation to prevent spread. Using the original analogies with students after campus closures would have certainly been "relevant" and "understandable", but incredibly insensitive to the gravity of the then lived experiences of some of the students.

The common structural elements of the previously developed analogies provided a general structure which made modifying the originals rapid and rather effortless. These analogies were quickly reworked, to source domains of Nuclear Radiation, A Bank Robbery, and Home Repairs. The Pandemic example and its isomorphic shift to the Nuclear Radiation example is shown in Table 2. Tables for the other two examples were also created, but in the interest of space cannot be included here. In reviewing Table 2, one can see that the "Common Structural Elements" segments of both analogies remains the same, and similarly, that the programming target to be communicated has also stayed the same. The new isomorphic analogy still fits the information we wish to communicate with the analogy, but the source has shifted. However, this source can still be compared to the generalized schema and seen to "fit", which is assurance that the analogy presents a relevant entity-relation structure. This assurance gives the ability to pivot source domain rapidly with confidence that the analogy has structural relevance.

6 MORPHISM AS PART OF DIALOGUE

Analogy as a communication tool can be (and often is) used to facilitate dialogue. Given the displayed success of dialogic methods in scientific classrooms in the research, dialogue is a pathway worth exploration when considering analogical methods. Dialogue and communication are by nature morphic processes. Communication is conducted with the intent to be understood. When we receive feedback on our communication and engage with that feedback, it becomes dialogic. This cycle of "gap closing" through dialogue often results in the original conversation landscape shifting.

OPAL's structure can provide a useful grounding when dialogue morphs non-isomorphically. OPAL's design allows conversation participants to impose contextual boundaries in the entities, relations, and targeted knowledge as dialogue shifts. If a learner and facilitator are engaging in a dialogue which includes an analogy, the learner may showcase some misapplication of the analogy through this dialogue. This may be due to the facilitator misunderstanding the context of the problem the learner is trying to understand (thus using an ill-fit analogy), or the learner "overextending" the analogy beyond the context it is being used to model. Considering OPAL, an instructor may be able to recognize that the learner is identifying a problem that does not fit the intended analogy design, or is extending the scope beyond the analogy fit. This recognition can allow the conversation to morph non-isomorphically and bridge knowledge by tending to "the gap" that was seen through the non-isomorphic transition. This dialogue is also not one way - a facilitator misunderstanding a learner may be "course corrected" back on track by the learner implicitly recognizing these disconnects as well.

The value of isomorphic shifts may also be seen in dialogue. A misunderstanding (or lack of awareness) of the source domain may become apparent during a conversation. This can prompt a natural flow to rectify the misunderstanding and "close the gap", in this case through an isomorphic shift or clarification of the existing source domain.

6.1 An Example of Morphic Dialogue in Action

One analogy-based morphic dialogue is employed quite often by Author One when working with students on understanding objects in Intro Programming. Students can struggle with how to approach object-oriented design, and understanding the difference between objects and classes. When students visit office hours with these questions, Author One inquires first about the student's interests and hobbies. From Pressed Pennies to Hockey Players, Stamps to Dogs, Author One has seen a fair amount of interesting topics shared by students. Despite the difference in source domains, the same dialogue with students can occur, guided conversations where students co-create providing relevant information through questions and prompts like:

- "What properties would we want to represent for any [Dog, Hockey Player, Stamp]?"
- "What actions would we want those things to do, or to be able to do with those things, if we were programming [A Dog Park, a Hockey Game, Our Collection]?"
- "These actions and these properties represent things each [Dog, Hockey Player, Stamp] should have and do, but it's just a template - this isn't [Your Dog, The Star Hockey Player, An Antique Stamp], but a way for us to organize information to create that thing."

These dialogues are clearly morphed to fit for the student attending office hours, but maintain a common structure and approach that addresses the similar nature of the questions. One can see how the components of this conversation could be transformed to fit into OPAL's flow. The knowledge of common analogical structure elements allows for these conversations to quickly pivot to new source domains in order to adapt to student needs, interest, and understanding.

7 RELEVANT & ENGAGING ISOMORPHS

Domain isomorphism appears to be a key method by which analogy can move from a general tool for pedagogical engagement to a *culturally responsive* tool for pedagogical engagement, provided this is grounded in communication and dialogue.

Isomorphic source domains allow for the potential impact an initial analogy has to expand in unique directions. Analogies are often developed from our own lived experiences or what we perceive as relevant examples - but these may not be relevant, engaging, or understood by students. A "well-formed" analogy does no good to a student who is not able to find relevance with the source domain. If a student is struggling to understand the source domain presented, they will be unable to analogically reason and find no value in the analogy. However, if the source domain of that analogy can isomorphically changed to *meet the student and their relevant prior knowledge*, the "pedagogical value" of that analogy is preserved while relevance to the student is heightened.

Engaging students with the analogy making, sharing, and reasoning process can also offset some difficulty instructors may face in identifying relevant analogs. While learners may not always come up with viable domains, and instructors should always strive to create positive relevant examples, the ability to incorporate student voice, choice, and perspectives can be one path to minimize missteps and maximize value for both learners and teachers.

Culturally responsive teaching centers prior learner knowledge and experiences alongside the importance of diverse examples [25]. Isomorphism can allow pedagogically valuable analogy structures to be designed to a learner's lived experiences, prior knowledge, and cultures. Examples grounded in the lives and interests of learners are also likely to be more engaging: both in that learners are *able to engage with* and understand the analogy, as well as learners *feeling that the material is interesting and relevant to them.*

Methods aligned with culturally responsive teaching are not only in the best interest of diverse groups of learners. These methods can create what is known as a "curb cut effect" (a United States term for sidewalk concrete grading down to smoothly meet the street) [6]. "Curb cuts" build accessibility into the design, which can generate new value and (in this context) learning fit for *all* students, including those considered historically privileged within the classroom environment. Further, the process of finding culturally and experientially relevant examples is also likely to benefit overall classroom climate and engagement through greater understanding. Communications and dialogue aimed at finding these examples can engage instructors in understanding learners, learners in understanding each other, and results in a greater quantity of increasingly relevant analogies within the context. Further, dialogue via discussion in and of itself is an engaged active learning method.

7.1 Feasible Instructional Applications

Leveraging the common structural elements for specific programming contexts can allow instructors greater flexibility in designing analogical pedagogy activities. Instructors may find new activities which encourage creative analogical reasoning via guidance on the common structural elements central to the lesson.

Instructors might, for example, guide students to develop their own analogies related to the current programming concept and its properties in group or classroom-wide discussion activities. Such activities could encourage collaborative and creative dialogue centering the targeted programming knowledge and guided by the overarching "common analogical structure".

Identified relevant source domains that students generate can continue to be leveraged and can be used to guide dialogue and examples throughout the course. In addition, student generated domains might serve as examples in future courses, adding to the "possible analogies" derived from a common structure. These student examples can provide additional source domains to help other students further their analogical reasoning and development of generalized schema, promoting understanding.

8 FUTURE WORK

Further work on OPAL is planned, including more exploration of usability by instructors, researchers, and students. Studies leveraging OPAL analogies for classroom effectiveness are also planned. We are considering the possibility that any resulting effectiveness may be seen in less traditional metrics such as engagement [36] as opposed to a performance measure. Further, the dialogic application that is encouraged by the authors is likely to somewhat complicate the analysis, and greater exploration into appropriate methods for this will need to be explored.

The relational structure present in OPAL somewhat parallels the nature of a typical story (setup-action-conclusion). Thus, storytelling and narrative use in computing pedagogy may also be relevant. Using story and narrative as a vehicle for analogy can encourage conjuring appropriate context even if one lacks prior knowledge (such as how works of fiction encourage conjuring of scenarios we have never been in). This may result in context generation by learners through the analogy's particular form, as well as potentially promoting engagement. Storytelling is also a knowledge passing tradition among cultures, suggesting a key intersection of analogy and cultural relevance.

9 CONCLUSION

Activating relevant prior knowledge for *any* learner can help them reason about and assimilate new knowledge. Analogy is a powerful tool in promoting this activation, provided that the sources used are relevant to the learner and that the analogical structure is relevant to the problem space. The OPAL framework tool described in this paper provides scaffolded elements which can help to facilitate pivots in pedagogical dialogue to ensure that analogies have impactful outcomes. Specifically, the common structural elements identified using OPAL can allow for confidence in isomorphic source domain shifts. The ability to shift domains quickly in order to address gaps and promote analogical reasoning among learners can promote development of more examples with greater cultural relevance to the learners. This in turn may increase both understanding and engagement. This may also allow the classroom community to connect more meaningfully through better recognition and inclusion of each member of the learning community's lived experiences and knowledge.

REFERENCES

[1] Kathryn Au, Cathie Jordan, et al. 1981. Teaching reading to Hawaiian children: Finding a culturally appropriate solution. *Culture and the bilingual classroom: Studies in classroom ethnography* (1981), 139–152.

[2] Paul Bartha. 2019. Analogy and Analogical Reasoning. In *The Stanford Encyclopedia of Philosophy* (spring 2019 ed.), Edward N. Zalta (Ed.). Metaphysics Research Lab, Stanford University.

[3] Briana Bettin, Michelle Jarvie-Eggart, Kelly S. Steelman, and Charles Wallace. 2021. Developing a Comic-Creation Assignment and Rubric for Teaching and Assessing Algorithmic Concepts. In *2021 IEEE Frontiers in Education Conference (FIE)*. 1–5. https://doi.org/10.1109/FIE49875.2021.9637190

[4] 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"*. Association for Computing Machinery, New York, NY, USA, 359–365. https://doi.org/10.1145/3430665.3456381

[5] 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*. ACM. https://doi.org/10.1145/3502718.3524796

[6] Angela Glover Blackwell. 2016. The Curb-Cut Effect. , 28–33 pages. Issue 1. https://doi.org/10.48558/YVMS-CC96

[7] Yingjun Cao, Leo Porter, and Daniel Zingaro. 2016. Examining the Value of Analogies in Introductory Computing. In *Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16)*. Association for Computing Machinery, New York, NY, USA, 231–239. https://doi.org/10.1145/2960310.2960313

[8] Jaime G. Carbonell. 1983. Learning by Analogy: Formulating and Generalizing Plans from Past Experience. In *Machine Learning: An Artificial Intelligence Approach*, Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell (Eds.). Springer Berlin Heidelberg, 137–161. https://doi.org/10.1007/978-3-662-12405-5_5

[9] Courtney B Cazden and Ellen L Leggett. 1976. Culturally responsive education: a response to LAU guidelines II. *Austin, Texas: Southwest Educational Development Laboratory* (1976).

[10] Yam San Chee. 1993. Applying Gentner's Theory of Analogy to the Teaching of Computer Programming. *Int. J. Man-Mach. Stud.* 38, 3 (March 1993), 347–368. https://doi.org/10.1006/imms.1993.1016

[11] John Clement, David E. Brown, and Aletta Zietsman. 1989. Not all preconceptions are misconceptions: finding 'anchoring conceptions' for grounding instruction on students' intuitions. *International Journal of Science Education* 11, 5 (1989), 554–565. https://doi.org/10.1080/0950069890110507 arXiv:https://doi.org/10.1080/0950069890110507

[12] John J. Clement. 1998. Expert novice similarities and instruction using analogies. *International Journal of Science Education* 20, 10 (1998), 1271–1286. https://doi.org/10.1080/0950069980201007

[13] John Clements and Shriram Krishnamurthi. 2022. Towards a Notional Machine for Runtime Stacks and Scope: When Stacks Don't Stack Up. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1 (ICER '22)*. Association for Computing Machinery, New York, NY, USA, 206–222. https://doi.org/10.1145/3501385.3543961

[14] Kathryn Cunningham, Shannon Ke, Mark Guzdial, and Barbara Ericson. 2019. Novice Rationales for Sketching and Tracing, and How They Try to Avoid It. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '19)*. ACM, New York, NY, USA, 37–43. https://doi.org/10.1145/3304221.3319788

[15] New York State Education Department. 2018. *Culturally Responsive-Sustaining Education Framework*. New York State Education Department, Albany, NY. 64 pages. http://www.nysed.gov/crs/framework

[16] Barbara Di Eugenio, Nick Green, Omar AlZoubi, Mehrdad Alizadeh, Rachel Harsley, and Davide Fossati. 2015. Worked-out Examples in a Computer Science Intelligent Tutoring System. In *Proceedings of the 16th Annual Conference on Information Technology Education (SIGITE '15)*. Association for Computing Machinery, New York, NY, USA, 121. https://doi.org/10.1145/2808006.2808011

[17] Edsger W. Dijkstra. 1988. On the cruelty of really teaching computing science. (Dec. 1988). http://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1036.PDF Circulated privately.

[18] Colin G. Dixon, Sherry Hsi, and HyunJoo Oh. 2022. Seamfulness and Culturally Responsive Computing. In *2022 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE, Philadelphia, PA, USA, 5. https://doi.org/10.1109/RESPECT55273.2022.00014

[19] Reinders Duit, Wolff-Michael Roth, Michael Komorek, and Jens Wilbers. 2001. Fostering conceptual change by analogies - Between Scylla and Charybdis. *Learning and Instruction* 11 (08 2001), 283–303. https://doi.org/10.1016/S0959-4752(00)00034-7

[20] Sally Fincher, Johan Jeuring, Craig S. Miller, Peter Donaldson, Benedict du Boulay, Matthias Hauswirth, Arto Hellas, Felienne Hermans, Colleen Lewis, Andreas Mühling, Janice L. Pearce, and Andrew Petersen. 2020. Capturing and Characterising Notional Machines. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '20)*. Association for Computing Machinery, New York, NY, USA, 502–503. https://doi.org/10.1145/3341525.3394988

[21] Sally Fincher and Anthony Robins. 2019. *The Cambridge Handbook of Computing Education Research*. Cambridge University Press. 11–39 pages. https://doi.org/10.1017/9781108654555.002

[22] Michael J. Ford. 2012. A Dialogic Account of Sense-Making in Scientific Argumentation and Reasoning. *Cognition and Instruction* 30, 3 (Jul 2012), 207–245. https://doi.org/10.1080/07370008.2012.689383

[23] Michael J. Ford and Brian M. Wargo. 2012. Dialogic framing of scientific content for conceptual and epistemic understanding. *Science Education* 96, 3 (May 2012), 369–391. https://doi.org/10.1002/sce.20482

[24] Michal Forišek and Monika Steinová. 2012. Metaphors and Analogies for Teaching Algorithms. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12)*. ACM, New York, NY, USA, 15–20. https://doi.org/10.1145/2157136.2157147

[25] Geneva Gay. 2018. *Culturally responsive teaching: theory, research, and practice* (3rd ed ed.). Teachers College, New York. 384 pages.

[26] Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science* 7, 2 (1983), 155 – 170. https://doi.org/10.1016/S0364-0213(83)80009-3

[27] Dedre Gentner and Kenneth D. Forbus. 2011. Computational models of analogy. *Wiley interdisciplinary reviews. Cognitive science* 2 3 (2011), 266–276.

[28] Dedre Gentner and Donald R. Gentner. 1983. Flowing waters or teeming crowds: Mental models of electricity. In *Mental Models*, Dedre Gentner and Albert L Stevens (Eds.). Erlbaum, 99–129.

[29] Dedre Gentner and Christian Hoyos. 2017. Analogy and Abstraction. *Topics in Cognitive Science* 9, 3 (2017), 672–693. https://doi.org/10.1111/tops.12278 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/tops.12278

[30] Dedre Gentner, Jeffrey Loewenstein, Leigh Thompson, and Kenneth D. Forbus. 2009. Reviving Inert Knowledge: Analogical Abstraction Supports Relational Retrieval of Past Events. *Cognitive Science* 33, 8 (2009), 1343–1382. https://doi.org/10.1111/j.1551-6709.2009.01070.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1551-6709.2009.01070.x

[31] Dedre Gentner and Albert L Stevens. 1983. *Mental Models*. Erlbaum.

[32] Dedre Gentner and Cecile Toupin. 1986. Systematicity and surface similarity in the development of analogy. *Cognitive Science* 10, 3 (1986), 277 – 300. https://doi.org/10.1016/S0364-0213(86)80019-2

[33] Mary L. Gick and Keith J. Holyoak. 1980. Analogical problem solving. *Cognitive Psychology* 12, 3 (1980), 306 – 355. https://doi.org/10.1016/0010-0285(80)90013-4

[34] Shawn M. Glynn, GA. National Reading Research Center, Athens, and MD. National Reading Research Center, College Park. 1994. *Teaching Science with Analogies [microform] : A Strategy for Teachers and Textbook Authors. Reading Research Report No. 15 / Shawn M. Glynn.* Distributed by ERIC Clearinghouse [Washington, D.C.]. 34 p. pages. https://eric.ed.gov/?id=ED373306

[35] Mark Guzdial. 2020. Dijkstra Was Wrong About 'Radical Novelty': Metaphors in CS Education. https://cacm.acm.org/blogs/blog-cacm/248985-dijkstra-was-wrong-about-radical-novelty-metaphors-in-cs-education/fulltext

[36] Jesper Haglund. 2013. Collaborative and self-generated analogies in science education. *Studies in Science Education* 49 (03 2013), 1–34. https://doi.org/10.1080/03057267.2013.801119

[37] Frank Halasz and Thomas P. Moran. 1982. Analogy Considered Harmful. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems (CHI '82)*. ACM, New York, NY, USA, 383–386. https://doi.org/10.1145/800049.801816

[38] Colton Harper. 2022. Developing and Evaluating Scaffolding for Student-Generated Analogies in CS1. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 2*. ACM, Dublin Ireland, 656–657. https://doi.org/10.1145/3502717.3532114

[39] Rachel Harsley, Nick Green, Mehrdad Alizadeh, Sabita Acharya, Davide Fossati, Barbara Di Eugenio, and Omar AlZoubi. 2016. Incorporating Analogies and Worked Out Examples As Pedagogical Strategies in a Computer Science Tutoring System. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. ACM, New York, NY, USA, 675–680. https://doi.org/10.1145/2839509.2844637

[40] David James. 2020. The Use of DJing Tasks as a Pedagogical Bridge to Learning Data Structures. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '20)*. Association for Computing Machinery, New York, NY, USA, 193–197. https://doi.org/10.1145/3341525.3387427

[41] Gayithri Jayathirtha. 2022. "How Does the Computer Carry out DigitalRead()?" Notional Machines Mediated Learner Conceptual Agency within an Introductory High School Electronic Textiles Unit. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1 (ICER '22)*. Association for Computing Machinery, New York, NY, USA, 189–205. https://doi.org/10.1145/3501385.3543964

[42] Amy J Ko, Anne Beitlers, Brett Wortzman, Matt Davidson, Alannah Oleson, Mara Kirdani-Ryan, and Stefania Druga. 2022. *Critically Conscious Computing: Methods for Secondary Education*. Self Publication. https://criticallyconsciouscomputing.org/

[43] Gloria Ladson-Billings. 1995. But that's just good teaching! The case for culturally relevant pedagogy. *Theory Into Practice* 34, 3 (1995), 159–165. https://doi.org/10.1080/00405849509543675 arXiv:https://doi.org/10.1080/00405849509543675

[44] Jean Lave, Michael Murtaugh, and Olivia de la Rocha. 1984. The dialectic of arithmetic in grocery shopping. In *Everyday cognition: Its development in social context*. Harvard University Press, 67–94.

[45] Cedric J. Linder. 1993. A challenge to conceptual change. *Science Education* 77, 3 (1993), 293–300. https://doi.org/10.1002/sce.3730770304 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/sce.3730770304

[46] Gerald Mohatt, Frederick Erickson, et al. 1981. Cultural differences in teaching styles in an Odawa school: A sociolinguistic approach. *Culture and the bilingual classroom: Studies in classroom ethnography* 105 (1981).

[47] Jessica Morales-Chicas, Mauricio Castillo, Ireri Bernal, Paloma Ramos, and Bianca Guzman. 2019. Computing with Relevance and Purpose: A Review of Culturally Relevant Education in Computing. *International Journal of Multicultural Education* 21, 1 (Mar. 2019), 125–155. https://doi.org/10.18251/ijme.v21i1.1745

[48] Robert P Moses and Charles E Cobb. 2002. *Radical Equations: Civil Rights from Mississippi to the Algebra Project*. Beacon Press. 256 pages.

[49] Nancy Nersessian. 2008. *Creating Scientific Concepts*. The MIT Press. https://doi.org/10.7551/mitpress/7967.001.0001

[50] Committee on the Role of Authentic STEM Learning Experiences in Developing Interest & Competencies for Computing, Board on Science Education & Computer Science, Telecommunications Board, Division of Behavioral & Social Sciences & Education, Division on Engineering & Physical Sciences, and National Academies of Sciences Engineering & Medicine. 2021. *Cultivating Interest and Competencies in Computing: Authentic Experiences and Design Factors*. National Academies Press, Washington, D.C. 214 pages. https://doi.org/10.17226/25912

[51] Maria Ong, Michael Cassidy, Sabrina De Los Santos, and Anya Carbonell. 2022. Rural Students Exploring Identity, Techno-Social Justice and Safe Spaces in a Culturally Responsive Computing Program. In *2022 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE, Philadelphia, PA, USA, 5. https://doi.org/10.1109/RESPECT55273.2022.00015

[52] Student Experience Project. 2022. *Increasing Equity in College Student Experience: Findings from a National Collaborative*. Student Experience Project. 20 pages. https://studentexperienceproject.org/report/

[53] Yizhou Qian and James Lehman. 2017. Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Trans. Comput. Educ.* 18, 1, Article 1 (Oct. 2017), 24 pages. https://doi.org/10.1145/3077618

[54] 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 (SIGCSE '14)*. ACM, New York, NY, USA, 585–590. https://doi.org/10.1145/2538862.2538945

[55] Jaemarie Solyst, Tara Nkrumah, Angela Stewart, Amanda Buddemeyer, Erin Walker, and Amy Ogan. 2022. Running an Online Synchronous Culturally Responsive Computing Camp for Middle School Girls. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*. ACM, Dublin Ireland, 158–164. https://doi.org/10.1145/3502718.3524747

[56] Juha Sorva. 2013. Notional Machines and Introductory Programming Education. *Trans. Comput. Educ.* 13, 2, Article 8 (July 2013), 31 pages. https://doi.org/10.1145/2483710.2483713

[57] Sangho Suh. 2022. Coding Strip: A Tool for Supporting Interplay within Abstraction Ladder for Computational Thinking. , 191 pages.

[58] Raechel Walker, Eman Sherif, and Cynthia Breazeal. 2022. Liberatory Computing Education for African American Students. In *2022 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE, Philadelphia, PA, USA, 5. https://doi.org/10.1109/RESPECT55273.2022.00024

[59] Alicia Nicki Washington. 2020. When Twice as Good Isn't Enough: The Case for Cultural Competence in Computing. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 213–219. https://doi.org/10.1145/3328778.3366792

[60] Aman Yadav and Marie K. Heath. 2022. Breaking the Code: Confronting Racism in Computer Science through Community, Criticality, and Citizenship. *TechTrends* 66, 3 (May 2022), 450–458. https://doi.org/10.1007/s11528-022-00734-9