Discourse Practices in Computer Science Education

Yvonne Kao
ykao@wested.org
WestEd
San Francisco, CA, USA

David McKinney
dmckinn@wested.org
WestEd
San Francisco, CA, USA

Sam Berg
sam.berg@ousd.org
Oakland Unified School District
Oakland, CA, USA

Brenda Tuohy
brenda.tuohy@ousd.org
Oakland Unified School District
Oakland, CA, USA

Courtney Ortega
courtney.ortega@ousd.org
Oakland Unified School District
Oakland, CA, USA

ABSTRACT

Rich classroom discussion, or discourse, has long been a recommended pedagogical practice in K-12 math and science education. Research shows that discourse is beneficial for all learners, but especially for English learners and minoritized students in STEM. Discourse helps develop students' agency, academic language, and conceptual understanding. Some K-12 computer science (CS) curricula incorporate student discourse, but we believe it is under-used. In this paper, we review how discourse helps students learn, discuss the use of discourse in CS and math education, share ideas for promoting discourse in CS classrooms, and call on curriculum developers, teacher professional learning providers, and researchers to support the increased use of discourse in K-12 CS education.

CCS CONCEPTS

• Social and professional topics → K-12 education.

KEYWORDS

computer science education, discourse, academic language

ACM Reference Format:
Yvonne Kao, David McKinney, Sam Berg, Brenda Tuohy, and Courtney Ortega. 2024. Discourse Practices in Computer Science Education. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024), March 20–23, 2024, Portland, OR, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630830

1 INTRODUCTION

Imagine a middle school or high school computer science (CS) teacher leading a whole-class discussion about the Scratch code in Figure 1. The discussion might go like this:

Teacher: Who can tell me what these blocks of code do?
Student A: They make the sprite move when the arrow keys are pressed.
Teacher: Close. Which arrow keys are you talking about?

SIGCSE 2024, March 20–23, 2024, Portland, OR, USA

ACM ISBN 979-8-4007-0423-9/24/03...$15.00
https://doi.org/10.1145/3626252.3630830

Figure 1: Scratch blocks for students to discuss.

Student A: Right and left.
Teacher: Good! Can someone tell me what the code (`x position + stepSize`) does?
Student B: Make the sprite go right?
Teacher: That's right! And what about (`x position - stepSize`)?
Student C: Go left.
Teacher: Good.

Or it might go like this:

Teacher: Who can tell me what these blocks of code do?
Student A: They make the sprite move when the arrow keys are pressed.
Teacher: Does anyone else have any ideas about what these blocks of code do?
Student B: The top blocks are for the right arrow and the bottom blocks are for the left arrow.
Teacher: Does anyone notice anything else?
Student C: The glide blocks are the same.
Student D: No they're not.
Student C: Yes they are.
Student D: No they're not! The top one has a plus sign and the bottom one has a minus sign.
Student C: Really? [pause] Ohhhh, I see it!
Teacher: Good observation, Student D. What do you think the plus sign and minus sign do?
Student D: Make the sprite go right and left?
Teacher: Does anyone agree or disagree with Student D?
Student E: I agree because adding to x will make it bigger and when x gets bigger, the sprite moves right. Taking away from x does the opposite.

The first statements by the teacher and Student A are the same in these hypothetical dialogues, but from there they diverge. The first dialogue is a common teacher-directed question-and-answer routine that follows the Initiate-Response-Evaluate (IRE) pattern of questioning [4, 8]. In the IRE pattern, teachers initiate by asking a question and are typically looking for a certain answer. These types of questions are called display questions since the goal is for students to display their knowledge so the teacher can check for correctness. When a student responds to the question, the teacher then evaluates the response for correctness. The pattern repeats with a new question [32]. IRE is a default pattern of questioning. All teachers use it and less experienced teachers tend to rely on it [3]. Unfortunately, the IRE pattern of questioning limits student agency. The teacher decides how the topic develops, who gets to talk, and who is "right" [18].

The second dialogue is more freeform. Instead of evaluating Student A's response, the teacher solicits more ideas from the class. In the middle of the dialogue, Students C and D have a direct exchange where they disagree with each other. The teacher then asks Student D to be more explicit about what they are noticing and why they think those differences matter, then invites the rest of the class to engage with Student D's thinking. In this dialogue, the teacher is asking open-ended questions that encourage further discussion and debate. We summarize literature to explain why this supports student learning in Section 2.

A key difference between the two dialogues is in the chain of speakers. In the first dialogue, students only respond to direct questions from the teacher. In the second dialogue, several student responses are chained together. The second dialogue does a better job of centering student thinking and giving students voice and agency in their computer science class.

In this position paper, we argue for the increased use of classroom discourse in K-12 computer science classes because it helps broaden participation and improves student learning. When we refer to "discourse" or "rich classroom discussion", we are referring to a discussion like the second dialogue in which students interact directly with each other to share and analyze their ideas about the content. First, we will provide an overview of the positive effects of classroom discourse and the mechanisms that drive these effects. Then we will review some examples of how K-12 CS and math curricula support classroom discourse. We conclude this paper with some ideas for bringing more rich discourse into K-12 CS classrooms and how curriculum developers, teacher professional learning providers, and researchers can support this effort.

2 HOW DISCOURSE HELPS STUDENTS LEARN

Computer science and other science, technology, engineering, and math (STEM) disciplines place great demands on students' language skills. Students must read and understand problem prompts and make sense of new words like "boolean" or discipline-specific uses of words like "variable" or "assignment". Supporting students' language development in STEM disciplines is an important aspect of building student success in these fields [36].

A key purpose of rich classroom discourse is to model sense-making and reasoning for students. While teachers often feel inclined to call on students who are likely to have the right answer, rich classroom discourse makes space for students to share their incomplete or incorrect ideas. Teachers treat errors as opportunities to further their students' understanding, asking follow-up questions that prompt students to think more deeply about the problem. Through this type of discourse, students can come to understand their teacher isn't the arbiter of "correctness". Instead, they learn there is a logic and structure to the discipline and they are capable of reasoning through it [22].

Discourse also helps build student agency and motivation and turns classrooms into communities of learners. Research suggests five principles when designing instructional materials to motivate learners: 1) creating a sense of belonging, 2) building students' confidence, 3) developing a mastery orientation towards learning, 4) supporting students' autonomy, and 5) creating personal relevance to students [13, 15]. Discourse supports the first four of these principles [14]. When students, particularly minoritized students, are encouraged to talk more in class and share their individual solutions to problems, they feel their ideas are more valued [22, 35].

There is strong evidence of the power of student discourse and other methods of eliciting deep student explanations, especially for English learners [8]. This evidence comes from laboratory and classroom studies, experimental studies and meta-analyses, and multiple academic subject areas. When students engage in rich discussion and deep questioning they improve their comprehension of the material [9, 24]. Because of this strong evidence base, classroom discourse has been widely promoted as a pedagogical technique in math and science classes throughout the United States [18, 19, 21, 23]. However, discourse is less common in CS education.

3 DISCOURSE IN K-12 CS EDUCATION

3.1 Discourse in CS frameworks and standards

The CS education community has recognized the importance of collaboration and communication around computing, but there is little mention of rich classroom discourse like the second dialogue in the introduction. The K-12 CS Framework describes seven core practices for computational literacy, three of which explicitly address the social aspects of computing [12]:

- Practice 1. Fostering an inclusive computing culture
- Practice 2. Collaborating around computing
- Practice 7. Communicating about computing

Each of these practices asks students to engage with others and solicit their perspectives, feedback, and ideas. While students can engage with these practices through the written word, students could also talk with others.

The Computer Science Teachers Association (CSTA) K-12 standards, which inform many states' CS standards, link each standard to a computing concept area and a practice. However, the detailed descriptions of Practices 1, 2, and 7 in the K-12 CS Framework [12] and the standards linked to these practices [7] largely focus on having students capture and address diverse user perspectives (Practices 1 and 2), develop working relationships and team norms (Practice 2), present data (Practice 7), and create documentation and give attribution (Practice 7). Other standards that connect to Practice 7 focus on using terminology and describing or explaining aspects of computation, all of which can be achieved through a one-way communication. Only one standard, 3B-IC-28[1], explicitly requires an exchange of ideas. There is little to suggest, in the framework or the standards, that students should share and discuss their ideas as a way to build their conceptual understanding.

3.2 Elementary Computing for All

The Elementary Computing for All (ECforAll) curriculum is a CS curriculum that intentionally includes discourse practices. There are multiple activities that encourage students to talk with each other [27]:

(1) Turn and Talk. During these activities, students pair up with a classmate and discuss their responses to a problem prompt, such as "Which scripts do the same thing?"
(2) Structured Pair Programming. The curriculum's pair programming activities are accompanied by sentence frames. These sentence frames provide linguistic support for students who are English learners and model appropriate language for navigating the social aspects of pair programming. For example, the sentence frame "It's my turn to type. What should I add?" reinforces the roles and expectations for the pair programming exercise. A quieter student could use this frame to speak up when it's their turn to be the "driver" who inputs the code.
(3) Presentation and Reflection. There are a variety of activities in the curriculum that give students the opportunity to present their work and reflect on what they learn. The curriculum also includes sentence frames to support students' reflections, like "We used loops in our project in order to ______" or "I had a hard time ______."

The developers of the CSforAll curriculum created these discourse elements with the goal of supporting English learners [27], but prior research on discourse practices suggest they could be beneficial for English proficient students as well (e.g., [9]).

3.3 Other Curricula

There are other CS curricula that include discussion elements, but they seem less intentionally developed than what we see in the ECforAll curriculum or the math curricula described in Section 4. For example, many lessons from Code.org's curricula include at least one discussion segment. This is the guidance for the warm-up discussion in Lesson 23 of Code.org's CS Fundamentals Express Course [6]:

> Discuss: How do computer programs ask us for information?
> Discussion Goal: Students should think about their own experiences as users and times when a computer asks them for information. There are lots of ways to input information into a computer, but focus on ideas where something is typed into a prompt for now.

This guidance could yield a rich classroom discussion if the teacher is skilled with facilitation. However, it is easy to imagine an IRE-style question-and-answer session in which the teacher sequentially calls on students, each of whom responds with a single word or a short phrase.

[1] 3B-IC-28: Debate laws and regulations that impact the development and use of software

4 DISCOURSE IN K-12 MATH AND SCIENCE EDUCATION

In this section we provide two concrete examples of how discourse is supported in specific K-12 math curricula. These are far from the only STEM curricula which support discourse practices, but we have chosen to describe a small number in more detail in order to provide readers a deeper understanding of how discourse can be integrated into lessons. These examples inform our suggestions for supporting discourse in K-12 CS curricula in Section 5 and our calls to action in Section 6.

4.1 Illustrative Mathematics

Illustrative Mathematics (IM) is a problem-based K-12 math curriculum that emphasizes inclusive teaching practices [10]. A key feature of IM is the incorporation of eight Math Language Routines (MLRs) to support teachers and students in their use of mathematical language [36]. In this paper, we focus on MLR7: Compare and Connect.

In Compare and Connect, students compare and contrast different representations and approaches to the same problem. Students explain when or why a particular approach should be used. An example of Compare and Connect appears in Unit 1, Lesson 19 of the sixth grade IM curriculum [11]. This lesson focuses on surface area. Students first work with a partner to study pictures of different tent styles and their specifications. Then they individually design their own tent and calculate its surface area to estimate the amount of fabric needed to construct the tent.

The teacher then pairs up students who created tents with similar overall capacity but different designs (e.g., a tent shaped like a triangular prism vs. a tent shaped like a pentagonal prism). The students then do the following:

(1) Explain your tent design and fabric estimate to your partner or partners. Be sure to explain why you chose this design and how you found your fabric estimate.
(2) Compare the estimated fabric necessary for each tent in your group. Discuss the following questions:
   - Which tent design used the least fabric? Why?
   - Which tent design used the most fabric? Why?
   - Which change in design most impacted the amount of fabric needed for the tent? Why?

This lesson also includes prompts for whole-class discussion that could be used for wrap-up.

The Math Language Routines can be adapted for specific content and classroom needs. The routines are designed to help students learn how to describe and communicate their thinking to others, make sense of concepts and representations, cultivate conversations between students, and reflect on their own ideas and reasoning [31, 36]. In section 5.1, we provide an example of how Compare and Connect can be adapted for a computer science context.

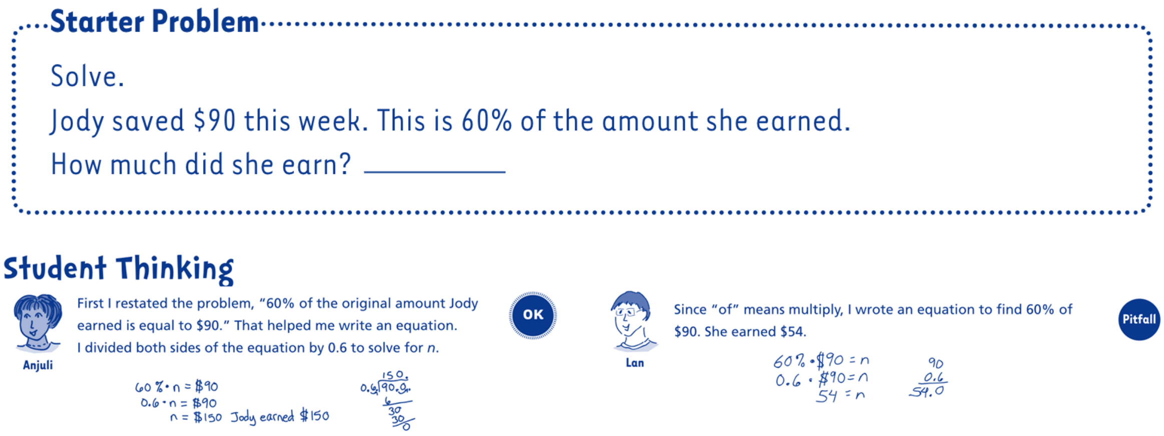

Figure 2: The Starter Problem from Unit 6, Lesson 9 of Math Pathways & Pitfalls—Percents, Ratios, and Proportions with Algebra Readiness: Lessons and Teaching Manual, Grades 6-8 (2010) by Carne Barnett-Clarke, Alma Ramirez, and Debra Coggins [1]. Used by permission from WestEd.

Table 1: Student and teacher discussion supports provided in Math Pathways & Pitfalls [33].

Support for...	Examples
	Discussion Builders (sentence frames for students)
Presenting alternative ideas	I have an idea... I wonder what would happen if...
Expanding on others' ideas	I have a question about ______'s idea.... I agree/disagree with ______'s idea because....
Posing additional questions	Would that be true if ...? Is there another way?
	Clipboard Prompts (discussion prompts for teachers)
Understanding the problem	Who can explain what this problem means —not how to solve it —what does it mean?
Understanding the solution process	Can someone else help us clarify this idea? Who has another way to help us understand it?
Reflecting on and extending the problem	Explain how you know this answer makes sense. How could you check? How can we prove it is correct?

From Math Pathways & Pitfalls—Percents, Ratios, and Proportions with Algebra Readiness: Lessons and Teaching Manual, Grades 6-8 (2010) by Carne Barnett-Clarke, Alma Ramirez, and Debra Coggins [1]. Used by permission from WestEd.

4.2 Math Pathways & Pitfalls

Math Pathways & Pitfalls (MPP) is a K-8 supplementary math curriculum [33]. Each MPP lesson opens with a Starter Problem that is designed to elicit a common misconception, or pitfall. Students first solve the problem individually. Then students talk with a partner to make sense of two solutions from fictional students: an "OK" solution that demonstrates correct reasoning and a "Pitfall" solution that includes an error or a common misconception (see Figure 2). The teacher then facilitates a whole-class discussion, selecting and highlighting important points of discussion from the pairs. The whole-class discussion is intended to draw out key mathematical ideas in the Starter Problem. Students record these ideas in their lesson packets and then apply them during more problem-solving later in the lesson. To help support teachers and students in creating rich classroom discussions, the MPP curriculum also includes a series of sentence frames, called Discussion Builders, and a series of "clipboard prompts" that teachers can use to probe students' thinking and direct the discussion (see Table 1). A large-scale randomized trial of the MPP curriculum with a geographically, ethnically, and linguistically diverse student sample found it improved students' performance on standardized mathematics assessments, especially for Latine students and English learners [9].

4.3 Key Takeaways from the Examples

We provided these examples of discourse-focused curricula in order to highlight three key ideas. First, rich discussion requires a system of supports. Both Illustrative Mathematics and Math Pathways & Pitfalls have structured approaches to initiating and facilitating student talk. These structures are built into lessons and repeat throughout the curriculum. Successful classroom discourse does not happen spontaneously. In order to maintain the rigor of the discussions, teachers should establish recurring routines so all students know how to participate. Otherwise, both students and teachers might struggle with finding the right words or not knowing who should get a turn to speak [18]. Second, rich tasks are needed to elicit rich discussion. The tasks and prompts from these examples do not have obvious short answers. Each of the example tasks requires students to first make sense of the problem and context and then to think deeply about mathematical concepts. Third, rich discussion uses incomplete and incorrect ideas as learning opportunities. Many of the structures to support discourse use incorrect ideas as a way to get students to think more deeply about the underlying concepts and their problem-solving strategies. Other times, students' ideas are incomplete because they struggled with the problem and ran out of time. Productive discourse often requires a mindset shift from both students and teachers. Teachers must be comfortable with highlighting ideas and work that are not completely correct. Students must feel safe in order to share their ideas freely without fear of embarrassment. Margulieux et al. have argued for this type of pedagogy in CS education [16]. In the next section, we present some ideas to illustrate how discourse can be used to further CS student learning.

5 IDEAS FOR DISCOURSE PRACTICES IN K-12 CS EDUCATION

5.1 Discourse on Data

The authors modified and piloted a lesson from Code.org's CS Discoveries course to help middle school students engage in discourse (see [17] for a detailed discussion of this experience). In this lesson, which is based on CS Discoveries Unit 5, Lesson 10: Structuring Data [5], students learn strategies for cleaning data and explore the trade-offs involved when cleaning a data set.

The modified lesson begins by introducing students to the questions of the day: 1) What do data scientists need to do to make data usable? 2) What are the tradeoffs in cleaning data? Then students work in pairs to create graphs of data from a fictitious survey on students' pizza preferences. The data set includes many responses that are "Pepperoni" and "Cheese," but also misspellings, variations in capitalization, and unique non-standard responses, such as "I'm allergic to mushrooms," "Anything is good," and "I'll be absent next week." Students must decide how to categorize these responses in the data set. After creating graphs, students follow the Compare and Connect language routine as each pair of students teams up with another pair to share their respective graphs. Students discuss "What's the same?" and "What's different?" between the two graphs. Teachers offer, model, and practice the sentence frame, "We chose to put _______here because _______." For example, students may respond with, "We chose to put 'Mushrooms' with 'Vegetarian' since vegetarian pizza usually has mushrooms on it," or "We chose to put 'Mushrooms' in a category called 'Other' since only one person said 'Mushrooms.'" Following this activity, students engage in a second activity with a different data set and again use the Compare and Connect language routine to discuss their work. Students reported high engagement with this lesson during the pilot [17].

The original lesson from Code.org does include several opportunities for class discussion. For example, one suggested prompt is "What changes did you need to make to the data? Was there any that you just needed to throw away completely? Why?" However, we did not feel these prompts provided enough support for rich classroom discourse instead of an IRE-style question-and-answer. One of the key changes we made to the lesson was to routinize the discussion and use Compare and Connect, which was familiar to students from their math classes. Another key change we made was to provide students with a sentence frame, which is a commonly-used pedagogical technique to support English learners in classroom discussion (e.g., [9, 27]). Sentence frames will also help English-proficient students focus on content of their responses instead of the way to frame it [18].

5.2 Discourse on Debugging

Discourse can help students learn to read, comprehend, and debug code, which are all prerequisite skills for fluent programming [34]. This example (see Figure 3) is modeled after the Starter Problem from Math Pathways & Pitfalls. Students first see a programming prompt: Write code so the sprite will count down from 10 to 1 and then stop. Students can spend some time trying to program a solution on their own. Then, students see two snippets of code. The code on the left produces the expected output. The code on the right is buggy. The snippets are labeled so there is no question as to which is which. Students explain to partners and/or the whole class why one solution works and the other does not. In this specific example, students could discuss the differences between the `repeat (10)` block, which is a definite loop, and the `forever` block, which is an infinite loop. Teachers could select alternate solutions that students programmed and display them next to the code in Figure 3 and ask students to compare the different solutions.

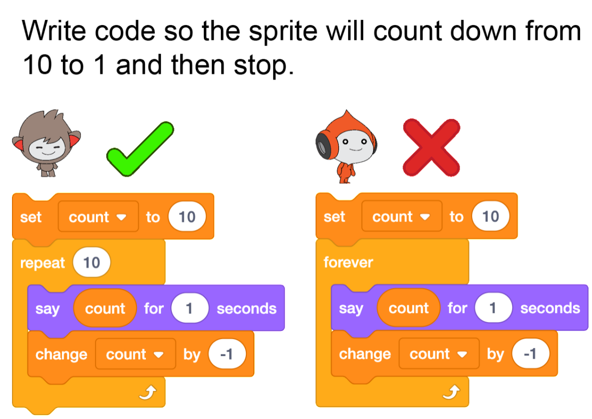

Figure 3: Example of a programming prompt and correct (left) and buggy (right) code in response.

6 CALLS TO ACTION

6.1 Developing K-12 CS Instructional Materials that Support Discourse

Instructional materials need to include explicit supports for rich classroom discussions. Simply including discussion prompts is not suficient, as many teachers will default to an IRE-style question-and-answer routine [18]. It is important to create consistent, recurring routines like those presented earlier in this paper. In addition, instructional materials should help teachers purposefully cultivate a classroom culture of trust and respect, especially for students who may be most familiar with IRE-style questioning where the focus is on correct answers. Students take a risk when sharing their thinking with others and often feel vulnerable [18].

Illustrative Mathematics embeds Math Language Routines into every lesson starting with Unit 1, Lesson 1. The introductory lessons include additional teacher guidance and time so students can learn the structure of the routines with basic math concepts. By Unit 2, students and teachers will have internalized the routine, allowing them to focus on the mathematical ideas that students are building and connecting. Math Pathways & Pitfalls introduces the practices and norms of class discussion to students over three days, before they begin to discuss any math problems. Over these three days, students practice with the different types of Discussion Builders (Table 1), watch a video of a real student discussion during an MPP lesson, and then discuss how the students in the video explain their ideas and show respect for each other even when disagreeing. Every lesson after this three-day introduction follows the same format, beginning with a Starter Problem and discussion of the OK and Pitfall solutions.

6.2 Preparing K-12 CS Teachers to Facilitate Classroom Discourse

Facilitating classroom discourse requires teachers to understand the subject matter, how student knowledge of that subject matter develops, and be able to quickly react to student ideas with follow-up prompts to create rich discussion that moves to a conclusion and newly-developed shared understanding of the content. Many teachers find facilitating discourse to be a daunting task, but there are methods to help teachers plan for and structure student discussion. Smith and Stein designed five practices to help teachers facilitate classroom discourse [28, p. 9-10]:

(1) anticipating likely student responses to challenging tasks and questions to ask students who produce them;
(2) monitoring students' actual response to the tasks (while students work on the tasks in pairs or small groups);
(3) selecting particular students to present their work during the whole-class discussion;
(4) sequencing the student responses that will be displayed in a specific order; and
(5) connecting different students' responses and connecting the responses to key ideas

Using these practices requires significant knowledge of the content and of teaching. CS teacher professional learning should help teachers understand common student errors and what kinds of responses might reflect more or less mastery of different concepts. Professional learning can help CS teachers develop different "talk moves" to shape and guide discussions while still giving students autonomy. For example, prompts like "Who wants to add on?" can invite students to link their ideas together while "Why do you think that?" or "Take your time; say more" can help draw out deeper explanations from students [18]. MPP's Clipboard Prompts (Table 1) provide teachers with a series of talk moves that they can literally carry with them on a clipboard if they wish.

It can also be helpful for CS teachers to practice engaging in this type of discourse themselves. When teachers learn how to use the Math Language Routines in Illustrative Mathematics, they first participate in the routines through the lens of a student. They also watch a classroom video to see how students might engage with the routine. Teachers then reflect on the purpose of the routine, how it happens, and how the routine supports students to develop language and math understanding.

6.3 Further Research to Support Classroom Discourse in K-12 CS Education

Researchers also have a role to play in supporting classroom discourse. In order to prepare teachers so they can successfully anticipate, select, sequence, and connect their students' ideas in a discussion, we need a better understanding of how students learn computing concepts. K-12 CS teachers often need further support in understanding how CS content standards break down into smaller learning targets and how to use students' work to identify their level of mastery [2]. Researchers should continue to do work like the Learning Trajectories in Elementary Computing (LTEC) project [25, 26] so we know more about how students' understanding of computing concepts builds. Work to map out students' learning trajectories should also be integrated with work on student misconceptions (e.g., [29, 30]). Often, students' misconceptions are not wholesale misunderstandings, but an incomplete understanding of when certain principles apply or when certain procedures should be followed. Misconceptions can be an important marker on a student's learning trajectory—an indicator that, with the right instructional push, a student can improve their understanding [20].

7 CONCLUSION

In this position paper, we have summarized literature that shows how rich classroom discussion supports student learning in STEM subjects, particularly for English learners. We have provided examples of how different curricula support these types of discussions and suggested ways to fit discourse into K-12 CS instruction. We encourage curriculum developers, professional learning providers, and researchers to embrace the use of classroom discourse as a potent pedagogical technique to broader participation and increase equity in K-12 CS education.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2122485. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Carne Barnett-Clarke, Alma Ramirez, and Debra Coggins. 2010. Math Pathways & Pitfalls—Percents, Ratios, and Proportions with Algebra Readiness: Lessons and Teaching Manual, Grades 6-8. WestEd, San Francisco, CA. https://mpp.wested.org/

[2] Satabdi Basu, Daisy Rutstein, Carol Tate, Arif Rachmatullah, and Hui Yang. 2022. Standards-Aligned Instructional Supports to Promote Computer Science Teachers' Pedagogical Content Knowledge. In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (Providence, RI, USA) (SIGCSE 2022). Association for Computing Machinery, New York, NY, USA, 404–410. https://doi.org/10.1145/3478431.3499403

[3] Kristy Elizabeth Boyer, William Lahti, Robert Phillips, Michael D. Wallis, Mladen A. Vouk, and James C. Lester. 2010. Principles of Asking Effective Questions during Student Problem Solving. In Proceedings of the 41st ACM Technical Symposium on Computer Science Education (Milwaukee, Wisconsin, USA) (SIGCSE '10). Association for Computing Machinery, New York, NY, USA, 460–464. https://doi.org/10.1145/1734263.1734417

[4] Courtney B. Cazden. 1988. Classroom Discourse: The Language of Teaching and Learning. Heinemann, Portsmouth, NH.

[5] Code.org. 2023. Lesson 10: Structuring Data. https://studio.code.org/s/csd5-2023/lessons/10

[6] Code.org. 2023. Lesson 23: Text and Prompts. https://studio.code.org/s/express-2023/lessons/23

[7] Computer Science Teachers Association. 2017. CSTA K-12 Computer Science Standards, Revised 2017. https://csteachers.org/k12standards/

[8] Pauline Gibbons. 2009. English Learners, Academic Literacy, and Thinking: Learning in the Challenge Zone. Heinnemann, Portsmouth, NH.

[9] Joan I. Heller, Thomas Hanson, Carne Barnett-Clarke, and Kate D. Darling. 2010. The Impact of Math Pathways & Pitfalls on Students' Mathematics Achievement and Mathematical Language Development: A Study Conducted in Schools with High Concentrations of Latino/a Students and English Learners. https://mpp.wested.org/supporting-research/research-findings/#IES-1

[10] Illustrative Mathematics. [n.d.-a]. Illustrative Mathematics. https://illustrativemathematics.org/

[11] Illustrative Mathematics. [n.d.-b]. Unit 1, Lesson 19: Designing a Tent. https://curriculum.illustrativemathematics.org/MS/teachers/1/1/19/index.html

[12] K-12 Computer Science Framework Steering Committee. 2016. K-12 Computer Science Framework. Technical Report. Association for Computing Machinery, New York, NY, USA.

[13] Lisa Linnenbrink-Garcia, Erika A. Patall, and Reinhard Pekrun. 2016. Adaptive Motivation and Emotion in Education: Research and Principles for Instructional Design. Policy Insights from the Behavioral and Brain Sciences 3, 2 (2016), 228–236. https://doi.org/10.1177/2372732216644450

[14] M-PLANS. [n.d.-a]. M-PLANS Toolkit. http://m-plans.org/toolkit

[15] M-PLANS. [n.d.-b]. Motivation Design Principles. http://m-plans.org/mdps

[16] Lauren Margulieux, Paul Denny, Kathryn Cunningham, Michael Deutsch, and Benjamin R. Shapiro. 2021. When Wrong is Right: The Instructional Power of Multiple Conceptions. In Proceedings of the 17th ACM Conference on International Computing Education Research (Virtual Event, USA) (ICER 2021). Association for Computing Machinery, New York, NY, USA, 184–197. https://doi.org/10.1145/3446871.3469750

[17] David McKinney, Chloe Morton, Brenda Tuohy, Sam Berg, Audrey Karlstad, Courtney Ortega, Zelda Allison, Grifin Munzel, Max Washburn, and Yvonne Kao. 2024. Iterative Design of a Socially-Relevant and Engaging Middle School Data Science Unit. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education (Portland, OR, USA) (SIGCSE '24). Association for Computing Machinery, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630886

[18] Sarah Michaels, Mary Catherine O'Connor, Megan Williams Hall, and Lauren B. Resnick. 2010. Accountable Talk Sourcebook: For Classroom Conversation That Works. https://ifl.pitt.edu/publications/

[19] Sarah Michaels and Cathy O'Connor. 2012. Talk Science Primer. Technical Report. TERC. https://inquiryproject.terc.edu/shared/pd/TalkScience_Primer.pdf

[20] Jim Minstrell. 2001. The role of the teaching in making sense of classroom experiences and effecting better learning. In Cognition and Instruction: Twenty-Five Years of Progress, Sharon M. Carver and David Klahr (Eds.). Lawrence Erlbaum Associates Publishers, Mahwah, NJ, Chapter 4, 121–149.

[21] National Council of Teachers of Mathematics (NCTM). 2014. Principles to Actions: Ensuring Mathematical Success for All. National Council of Teachers of Mathematics, Reston, VA. https://www.nctm.org/PtA/

[22] National Research Council. 2001. Adding It Up: Helping Children Learn Mathematics. The National Academies Press, Washington, DC. https://doi.org/10.17226/9822

[23] National Research Council. 2007. Taking Science to School: Learning and Teaching Science in Grades K-8. The National Academies Press, Washington, DC. https://doi.org/10.17226/11625

[24] Harold Pashler, Patrice M. Bain, Brian A. Bottge, Arthur Graesser, Kenneth Koedinger, Mark McDaniel, and Janet Metcalfe. 2007 Organizing Instruction and Study to Improve Student Learning (NCER 2007-2004). Technical Report. National Center for Education Research, Institute of Education Science, U.S. Department of Education, Washington, DC. http://ncer.ed.gov

[25] Kathryn M. Rich, Carla Strickland, T. Andrew Binkowski, and Diana Franklin. 2019. A K-8 Debugging Learning Trajectory Derived from Research Literature. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (Minneapolis, MN, USA) (SIGCSE '19). Association for Computing Machinery, New York, NY, USA, 745–751. https://doi.org/10.1145/3287324.3287396

[26] Kathryn M. Rich, Carla Strickland, T. Andrew Binkowski, Cheryl Moran, and Diana Franklin. 2017. K-8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals. In Proceedings of the 2017 ACM Conference on International Computing Education Research (Tacoma, Washington, USA) (ICER '17). Association for Computing Machinery, New York, NY, USA, 182–190. https://doi.org/10.1145/3105726.3106166

[27] Dana Saito-Stehberger, Leiny Garcia, and Mark Warschauer. 2021. Modifying Curriculum for Novice Computational Thinking Elementary Teachers and English Language Learners. In Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (Virtual Event, Germany) (ITiCSE '21). Association for Computing Machinery, New York, NY, USA, 136–142. https://doi.org/10.1145/3430665.3456355

[28] Margaret S. Smith and Mary Kay Stein. 2018. 5 Practices for Orchestrating Productive Mathematics Discussions, 2nd Edition. Corwin, Thousand Oaks, CA.

[29] Juha Sorva. 2012. Visual Program Simulation in Introductory Programming Education. Ph. D. Dissertation. Aalto University, Department of Computer Science and Engineering.

[30] Alaaeddin Swidan, Felienne Hermans, and Marileen Smit. 2018. Programming Misconceptions for School Students. In Proceedings of the 2018 ACM Conference on International Computing Education Research (Espoo, Finland) (ICER '18). Association for Computing Machinery, New York, NY, USA, 151–159. https://doi.org/10.1145/3230977.3230995

[31] Kristen Taylor. 2022. Math Language Routines: Discourse with a Purpose. https://illustrativemathematics.blog/2022/02/17/math-language-routines-discourse-with-a-purpose/

[32] Steve Walsh. 2011. Exploring Classroom Discourse: Language in Action. Taylor & Francis, New York, NY. https://www.routledge.com/Exploring-Classroom-Discourse-Language-in-Action/Walsh/p/book/9780415570671

[33] WestEd. [n.d.]. Math Pathways & Pitfalls. https://mpp.wested.org/

[34] Jacqueline L. Whalley and Raymond Lister. 2009. The BRACElet 2009.1 (Wellington) Specification. In Proceedings of the Eleventh Australasian Conference on Computing Education - Volume 95 (Wellington, New Zealand) (ACE '09). Australian Computer Society, Inc., AUS, 9–18.

[35] Dorothy Y. White. 2003. Promoting productive mathematical classroom discourse with diverse students. The Journal of Mathematical Behavior 22, 1 (2003), 37–53. https://doi.org/10.1016/S0732-3123(03)00003-8

[36] Jeff Zwiers, Jack Dieckmann, Sara Rutherford-Quach, Vinci Daro, Renae Skarin, Steven Weiss, and James Malamut. 2017. Principles for the Design of Mathematics Curricula: Promoting Language and Content Development. Technical Report. Stanford University, UL/SCALE. http://ell.stanford.edu/content/mathematics-resources-additional-resources