Promising Approaches to Integrating Computer Science and English Language Arts for Early Elementary Students (Fundamental Research)

Dr. Julie M. Smith, Institute for Advancing Computing Education

Dr. Julie M. Smith is a senior education researcher at the Institute for Advancing Computing Education. She holds degrees in Software Development, Curriculum & Instruction, and Learning Technologies. Her research focus is computer science education and AI education, particularly the intersection of learning analytics, learning theory, and broadening participation. She was a research assistant at MIT's Teaching Systems Lab; she is also co-editor of the SIGCSE Bulletin.

Monica McGill, Institute for Advancing Computing Education

Monica McGill is President & CEO of the Institute for Advancing Computing Education (formerly known as CSEdResearch.org). Have previously worked in industry and academia, McGill is using her experiences as a computer scientist, professor, and researcher to enable others to build a strong foundation of CS education research focused on all children while also honoring their backgrounds and experiences.

Rebekah Degener, Minnesota State University, Mankato

Promising Approaches to Integrating CS and ELA for Early Elementary Students (Fundamental Research)

Julie M. Smith[1], Monica M. McGill[2], and Rebekah Degener[3]

[1,2]Institute for Advancing Computing Education
[1]julie@csedresearch.org
[2]monica@csedresearch.org
[3]Minnesota State University, Mankato
[3]rebekah.degener@mnsu.org

Abstract

Background. This fundamental research paper explores integrated computer science (CS) instruction. One promising approach to expanding access to CS is to integrate CS into other content areas. Integration may be particularly appropriate for the earliest grades, where it is difficult for schools to carve out the time and resources to support a new content focus. Thus, recent years have seen efforts to integrate CS into other subjects in early elementary school, with an increasing body of research exploring these attempts to interweave content areas. While math and science are more common avenues for CS inclusion, some efforts to integrate CS into English Language Arts (ELA) instruction offer preliminary evidence of success.

Research Questions. This paper explores two research questions related to CS+ELA integration for grades K-2: (1) *What is known about integrating CS and ELA (i.e., standards, curriculum, pedagogy, student outcomes)?* and (2) *What are promising ways to integrate CS and ELA?*

Methodology. To answer the first research question, we conducted a systematic literature review using the SALSA approach. To answer the second research question, we used the findings from RQ1 to map (including many-to-many mapping) the K-5 CS standards for Louisiana to the state's first grade ELA standards. We also identify which mapped ELA and CS standards present the most potential for serving as the basis for bridging CS and ELA instruction.

Findings. From the systematic review, we identified promising avenues for embedding CS within ELA instruction. For example, some studies explore how integrated instruction might focus on programming in the service of storytelling. Others leverage the idea that programming languages share conceptual features with natural languages and both permit students to exercise creativity and to communicate. We identified five CS concepts (using data, analyzing data, creating and using algorithms, applying decomposition, and describing programming choices) with a high potential for integrated instruction. For example, one CS standard focuses on decomposing a problem into subproblems (E.AP.3A), and two ELA standards focus on beginning readers' decoding of words (RF.1.3b, RF.1.3c); these require similar pattern recognition and decomposition skills and therefore may be easier to teach in an integrated manner than more divergent skills.

Implications. Our results provide a useful starting point for future efforts to integrate CS with ELA for young learners and highlight promising avenues for continued research.

AI Use in This Study. AI was used to support creating code to generate data visualizations.

1 Introduction

Computer science (CS) developed as a distinct discipline from math and engineering [1, 2], a lineage reflected in the fact that CS is still often housed within colleges and school of engineering at the post-secondary level. Thus, CS education is part of the broader field of engineering education. Many CS skills are key to the development of engineering competency, including, for example, computational thinking (CT), with significant research focused on best practices for developing CT skills in the content of engineering education [3–6]. As AI (a subset of CS) becomes increasingly central to the workflow of professional engineers [7], the need for engineering education – for all subfields of engineering – to incorporate CS instruction will only increase.

CS, as with other subfields of engineering, has been introduced at the earliest grade levels, including as early as Pre-kindergarten in early childhood education [8]. Introducing CS to young students holds promise in not only improving executive function [9, 10], but it also may help introduce girls and students from historically marginalized groups early to CS [11], potentially boosting their comfort and confidence in the subject.

At these early grades, it is common for teachers to teach multiple subjects, since early elementary teachers are often responsible for teaching all of their students across the curriculum. It's been a recognized practice to integrate CS into one or more core subjects in these early grades for a variety of reasons. Many studies have surfaced teachers' lack of comfort and knowledge starting to teach CS to their students [8, 12], and integrating CS into another subject with which they are already familiar can boost their confidence and make them more comfortable [13]. It can also somewhat increase CS instruction without decreasing the amount of time in the core subject CS is integrated into [14]. Studies have also reported a variety of positive impacts of integrating CS into their core subjects [15–17].

While CS has often been integrated into science and math – which are viewed more as natural fits for CS – integrating CS into English Language Arts (ELA) has been less explored.[1] Nevertheless, for an upcoming project, we wanted to dig deeper into this curriculum strategy to understand what existing research may say about integration and its impacts on students. Our research questions for this study were:

RQ1. *What is known about integrating CS and ELA (i.e., standards, curriculum, pedagogy, student outcomes)?*
RQ2. *What are promising ways to integrate CS and ELA?*

[1] Throughout this paper, we refer to English Language Arts as *ELA* for simplicity, while recognizing that sometimes the natural language in question is not English.

Table 1: Coding as Another Language: Overlap of CS and ELA Ideas; table adapted from [24]

Computer Science Concepts	Literacy Concepts
algorithms	sequencing
the design process	the writing process
representation	letter-sound correspondence
debugging	editing
control structures	literary devices
modularity	phonological awareness
hardware and software	communication tools

2 Background

In the context of the research questions, we provide background on integrated CS education as well as the Scarborough's Reading Rope.

2.1 Integrated CS Education

The research corpus related to integrated CS education continues to grow across K-12, with the majority of papers focused on K-5. In a recent systematic literature review, Mabie et al. (2023) found common themes related to positive impacts of integration on students [18], including an increase in each of the following:

- Problem-solving skills,
- Logical thinking,
- Collaboration skills, and
- Engagement and excitement learning the subjects being taught.

More specifically, math and ELA are the two subjects most tested in the U.S., and research integrating CS/CT into these subjects shows evidence of impacts. Results from studies integrating CS into math found an increase in student engagement and student learning [19], improvements in students' sequential thinking [20], and increases in motivation to learn math. [21] also found that students' ability to decompose number problems improved [20]. And, after professional development for teaching CT, teachers' confidence in teaching math also significantly increased [21–23].

Integrating CS/CT into ELA can be a natural fit, given the syntax and sequencing that exists within both storytelling and programming, as well as other possible areas of overlap. Bers created a pedagogical approach for early elementary CS/CT integration referred to as Coding as Another Language (CAL) [15]. Using Piaget's levels of cognitive development, Bers aligned progressions of learning in ELA with CS/CT. Impacts of CAL on learning both CS/CT and ELA have been investigated (details of which are shared later in this paper). The foundation of CAL is the idea that there is substantial overlap between key concepts in CS and in ELA; see Table 1.

In other studies, researchers directed students through a storytelling process, then taught CS concepts that enabled the students to create an original narrative that they then programmed within a block-based programming language [16, 25]. The students showed high engagement and excitement [15–17].

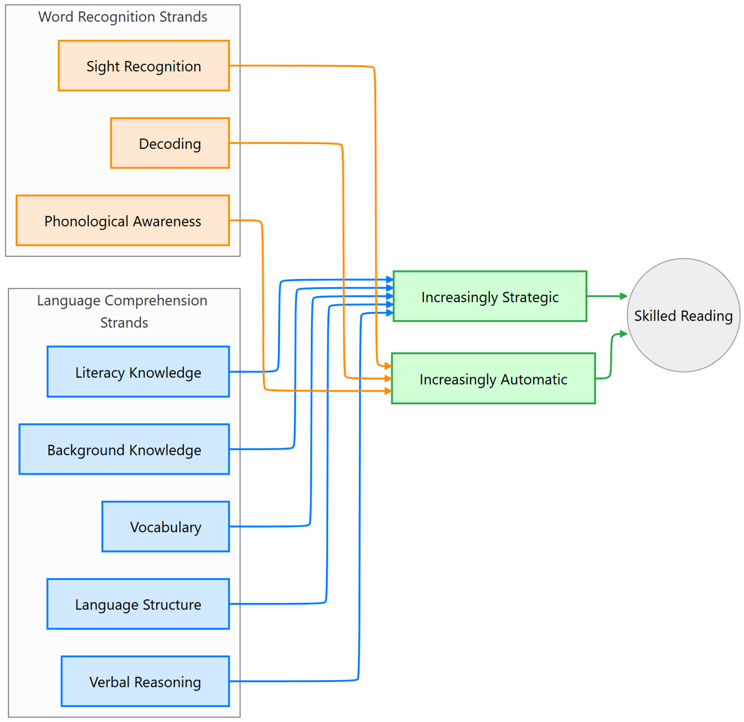

Figure 1: Scarborough's Reading Rope (adapted from [26])

2.2 Scarborough's Reading Rope

Scarborough's Reading Rope [26] is a conceptual framework for understanding the core components of skilled reading and how they work together to support reading development (see Figure 1.

The framework emphasizes that reading proficiency emerges from the integration of multiple, interrelated strands rather than from any single skill in isolation. These strands include word recognition processes (i.e., phonological awareness, decoding, and sight word recognition) and language comprehension processes (i.e., background knowledge, vocabulary, language structures, verbal reasoning, and literacy knowledge). The framework allows for educators to conceptualize the core components of reading requiring high-quality explicit instruction. By illustrating the reciprocal relationship between word recognition and language comprehension, Scarborough's Reading Rope underscores the importance of attending to all dimensions of reading instruction, including language development, vocabulary, and comprehension. The model posits that these strands must be intentionally and simultaneously supported so that reading processes become increasingly automatic and coordinated over time.

3 Methodology

To answer our research questions, we engaged in a systematic literature review followed by a mapping of one state's (Louisiana) first grade ELA standards to their CS standards.

3.1 Systematic Literature Review

This systematic literature review followed the SALSA (search, appraisal, synthesis, and analysis) approach [27]. Specifically, we conducted the search via a citation chaining (also known as 'snowballing') approach [28] to identify literature for this systematic review. This approach was selected due to (1) the known paucity of relevant literature, (2) the limitations of a search-string based approach given the number of search terms that would likely result in papers not ultimately germane to this topic, and (3) its ability to leverage the expertise of scholars (via their decision of which papers to cite). The search process was conducted in October 2025. We examined, as needed, the title, abstract, and text of each paper to determine if it met our criteria for inclusion or exclusion:

- we included papers if they focused on CS+ELA (or another natural language) integration; the paper may also have involved the integration of other school subjects
- we included papers if they focused on K-2 students and/or their teachers; the paper may also have involved students of other grades/ages
- we excluded systematic literature reviews, posters, abstract-only publications, book reviews, book chapters, white papers, position papers, theses, and dissertations
- we excluded papers not written in English
- we included the paper if the abstract mentioned CAL

We began the chaining process with the paper "A systematic literature review examining the impacts of integrating computer science in K-5 settings" [18], the only systematic literature review of integrated CS at the elementary school level of which we are aware. We used both Google Scholar and Semantic Scholar to identify the papers that cite each included paper; we also identified papers cited by each included paper. We repeated this process for two iterations. The result was a set of 18 papers.

Each paper was coded by two of the authors, with the following items recorded:

- research approach (e.g., case study, randomized control trial)
- part(s) of Scarborough's Reading Rope targeted, if any
- context (e.g, grade or age of students, geographic location)
- intervention if any
- what was measured (e.g., CT skills)
- what data was gathered (e.g., coding artifacts)
- findings

Each author completed this coding independently. Then, the authors identified any inconsistencies in the results of the coding and resolved them by consensus [29] during a meeting. We also identified the type of venue (CS/CT, ELA, education, educational technology, or other) for each publication.

3.2 Standards Mapping

We mapped Louisiana's first grade ELA standards to the state's K-5 CS standards. To perform this mapping, one author – with a background in ELA, CS, standards-based curriculum development, and CS standards research – iteratively compared each ELA standard to each CS standard to identify any conceptual overlap between the two. Conceptual overlap was deemed to occur if the same underlying skill or activity was evident in both standards. For example, decoding two-syllable words (RF.1.3e) is an instance of analyzing data for patterns and relationships (E.DA.5B).

Then, the other two authors (with, respectively, CS expertise and ELA expertise) reviewed and, as needed, refined the mapping.

4 Results

4.1 Systematic Literature Review

In this section, we first present an overview of the findings and then discuss them in more detail. Table 2 summarizes the studies included in this systematic literature review.

Of the studies included in this dataset, 6 were published in general education venues (e.g., the AERA annual meeting), 9 appeared in educational technology venues (e.g., *Journal of Technology and Teacher Education*), and 3 in computer science venues (e.g., *Transactions on Computing Education*). None appeared in ELA (or other natural language) venues.

Many studies occurred outside of the US and/or in locations where English was not the language of instruction.

One study examined ELA-related outcomes [32], finding that – despite the fact that the integrated CS instruction occurred during time normally devoted to literacy instruction, students' literacy performance was not harmed.

The authors coded studies according to which portion of Scarborough's Reading Rope were targeted. However, the various portions of this framework were not well-represented in this dataset. Many studies (e.g., [35] and [33]) did not explicitly focus on specific reading-related skills but rather on skills such as digital storytelling or language development more generally.

Many studies examined the impact of integrated CS instruction on CT skills. Most studies found evidence of improved CT skills [24, 30, 40, 41, 43], but some did not [31, 32]. For CS skills, many studies found evidence of improvement [31, 32, 40, 43, 44].

To organize our detailed discussion of the findings, we divide included studies into two categories: those that used CAL and those that used another intervention or no intervention at all.

Out of the 19 papers composing the dataset, 11 used CAL for teaching integrated CS+ELA. While some mixed methods were used in these studies, quantitative approaches and randomized control trials (RCT) were more commonly described. Study outcomes tended to focus on CT and/or CS skills and usually found at least some statistically significant improvement in these skills when CAL was used, although improvement was not universal (see above).

Studies that did not involve CAL either used another curricula or approach or had no intervention. Interventional studies found that students using ScratchJr could be successful with digital storytelling [35]. Similarly, robotics instruction could lead to expressive and communicative activities by students [34] and to cognitive, language, and creative skill development [33] but also required careful design of activities [39].

One study without an (explicit) intervention found that integrated CS curricula was highly focused on the most introductory of CS concepts, to the exclusion of even moderately advanced

Table 2: Summary of Included Studies

Citation	Context	Description	Intervention	Findings
[15]	4-7 year olds	composite case studies	CAL	students progress through six stages as they learn to code
[30]	1st-2nd grade students in the US	quasi-experimental	CAL with KIBO	students using CAL had improved CT skills
[31]	2nd grade students in the US	cluster RCT	CAL with ScratchJr	students' coding skills – but not CT skills – improved after using CAL
[32]	K-2 students in the US	pilot study and cluster-RCT	CAL	coding skills improved but CT skills did not; there were increases in math and literacy scores (pilot study)
[33]	kindergarten students in Turkey	quasi-experimental	Productive Children: Coding and Robotics Education Program	students' cognitive development, language development, and creativity improved with the intervention
[34]	3-6 year olds in Portugal	case study	robotics	many student activities were related to expression and communication; children were enthusiastic but there were classroom management challenges
[35]	summer camp for 6-12 year olds in Cyprus	mixed methods	ScratchJr for storytelling	students were successful in coding for storytelling
[36]	K-8 teachers in the US	qualitative, descriptive	N/A	teachers described the importance of integrated CS and mentioned students engaged in literacy-related activities
[37]	K-2 students in the US	task-based interviews	N/A	students used storytelling to perform CT tasks
[38]	K-8 students in the US	content analysis	N/A	integrated CS curricula focuses on introductory concepts
[24]	K-2 students in the US	mixed methods case study	Coding Playground, with CAL and ScratchJr	students' CT skills improved
[39]	1st-4th grade students at a festival in the Azores	observation and surveys	Botley robot	children found the robot easy to use, but the activities could be improved
[40]	K-2 students in Argentina	mixed methods and cluster-RCT	CAL	improved coding and CT skills with use of CAL curriculum
[41]	2nd grade students in Greece	quantitative, semi-experimental	CAL	students' CT skills improved
[42]	K-2 teachers	quantitative, surveys	CAL	teachers' coding and self-efficacy improved
[43]	kindergarten students in Israel	mixed methods	CAL with KIBO	students' coding and CT skills improved
[44]	K-2 students in the US	RCT	CAL	students' coding skills improved, but there was no significant difference for CT skills with use of CAL
[45]	1st-2nd grade students in the US	experimental with observation	CAL with KIBO	students with and without disabilities had similar CT learning

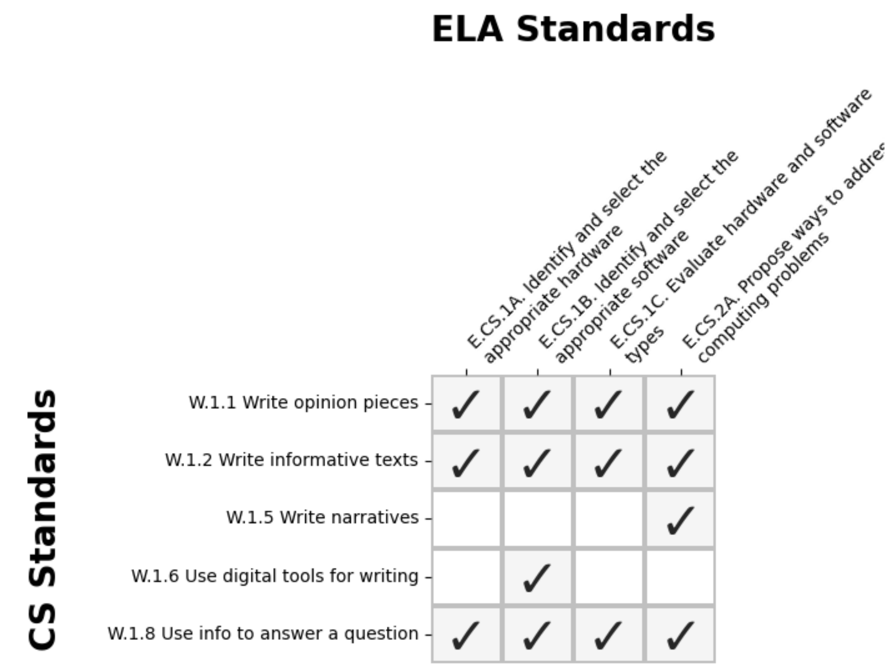

Figure 2: Mapping of Computing Systems Standards to ELA Standards

topics [38]. In another non-interventional study, task-based interviews with students found that students were likely to choose to use storytelling as they completed CT tasks [37]. Similarly, a study of teachers found that students used storytelling with block-based coding [36].

4.2 Standards Mapping

We divide the standards mapping into two parts. The mapping for the first part can be found in Table 3. This table lists the ELA standards that would be relevant to any CS standard taught via an informational text (i.e., a reading passage covering the content of the standard) and the CS standards that are amenable to being taught via an informational text.

The second part of the mapping directly maps ELA standards to relevant CS standards; see Figures 2, 3, and 4. These figures are organized by Louisiana's categories for CS standards. Two of Louisiana's categories are not included in these visualizations. First, *Networks and the Internet* is not included because the standards in this category are entirely covered in Table 3. Second, *Impacts of Computing* is not included because the standards in this category are also covered in Table 3, with the exception of "describe how computing has changed the ways people live and work" (E.IC.1A), which is mapped to "use verbs to convey a sense of past, present, and future" (L.1.1e).

A few ELA and a few CS standards appear in both parts of the mapping.

We also identified which CS standards present the most potential for serving as the basis of bridging CS and ELA instruction; see Table 4. Note that standards E.DA.1A and E.DA.4A are very similar.

Table 3: CS and ELA Standards Related to Informational Texts

(a) CS Standards

E.NI.1A. Explain how networks connect computers to other computing systems and the internet.
E.NI.2A. Describe personally identifiable information (PII) and identify practices for when and where sharing PII is appropriate.
E.NI.2B. Identify ways to maintain data security when using networks.
E.DA.1B. Classify types of data and describe the attributes used to sort data.
E.AP.4D. Identify intellectual property rights and apply the appropriate attribution when creating or remixing programs.
E.IC.1A. Describe how computing has changed the ways people live and work.
E.IC.2A. Identify and describe examples of appropriate versus inappropriate computer communications.
E.IC.2B. Identify examples of cyberbullying with age-appropriate responses.
E.IC.3A. Explain how online actions have real-world consequences and that laws and rules may also apply online.
E.IC.3B. Describe the safe versus unsafe uses of computing systems at age-appropriate levels.
E.IC.3C. Explain how the school and school system's computing rules and policies keep students safe.

(b) ELA Standards

RI.1.1 Ask and answer questions about key details in a text.
RI.1.2 Identify the main topic and retell key details of a text.
RI.1.3 Describe the connection between two individuals, events, ideas, or pieces of information in a text.
RI.1.4 Ask and answer questions to help determine or clarify the meaning of words and phrases in a text.
RI.1.5 Know and use various text features (e.g., headings, tables of contents, glossaries, electronic menus, icons) to locate key facts or information in a text.
RI.1.6 Distinguish between information provided by pictures or other illustrations and information provided by the words in a text.
RI.1.7 Use the illustrations and details in a text to describe its key ideas.
RI.1.8 Identify the reasons an author gives to support points in a text.
RI.1.9 Identify basic similarities in and differences between two texts on the same topic (e.g., in illustrations,descriptions, or procedures).
RI.1.10 With prompting and support read informational texts appropriately complex for grade 1.
RF.1.4a Read on-level text with purpose and understanding.
RF.1.4b Read on-level text orally with accuracy, appropriate rate, and expression on successive readings.
RF.1.4c Use context to confirm or self-correct word recognition and understanding, rereading as necessary.
W.1.1 Write opinion pieces . . .
W.1.2 Write informative/explanatory texts . . .
W.1.3 Write narratives in which they recount two or more appropriately sequenced events . . .
W.1.7 Participate in shared research and writing projects . . .
W.1.8 . . . recall information from experiences or gather information from provided sources to answer a question.
SL.1.2 Ask and answer questions about key details in a text read aloud or information presented orally or through other media.
SL.1.3 Ask and answer questions about what a speaker says in order to gather additional information or clarify something that is not understood.
SL.1.4 Describe people, places, things, and events with relevant details, expressing ideas and feelings clearly.
SL.1.5 Add drawings or other visual displays to descriptions . . .
L.1.4a Use sentence-level context as a clue to the meaning of a word or phrase.
L.1.6 Use words and phrases acquired through conversations, reading and being read to, and responding to texts . . .

ELA Standards

CS Standards	E.DA.1A. Organize and present data visually	E.DA.1B. Classify types of data	E.DA.2A. Select the appropriate data collection tool and technique	E.DA.2B. Describe and collect data	E.DA.3A. Compare and contrast ways to store data using technology	E.DA.3B. Explain how to [use] data using a computing device	E.DA.4A. Organize and present data visually >2 ways	E.DA.5A. Utilize data to create models . . .	E.DA.5B. Analyze data for patterns and relationships
L.1.1c Use singular and plural nouns with matching verbs									✓
L.1.5b Define words by category		✓							✓
RF.1.1a Recognize sentence features						✓			
RF.1.1a Recognize the distinguishing features of a sentence									✓
RF.1.3a Common consonant digraphs spelling and sound									✓
RF.1.3c Know final -e and common vowel team conventions									✓
RF.1.3e Decode two-syllable words									✓
RF.1.3g Recognize and read irregularly spelled words								✓	
SL.1.2 Sort words in categories		✓							✓
SL.1.5 Add visual displays to descriptions	✓						✓	✓	
W.1.1 Write opinion pieces	✓				✓		✓		✓
W.1.2 Write informative texts	✓			✓	✓	✓	✓		✓
W.1.5 Write narratives				✓		✓	✓		✓
W.1.6 Use digital tools for writing	✓		✓			✓	✓	✓	
W.1.8 Use info to answer a question	✓		✓	✓	✓	✓	✓	✓	✓

Figure 3: Mapping of Data and Analysis Standards to ELA Standards

Figure 4: Mapping of Algorithms and Programming Standards to ELA Standards

Table 4: Most Promising Integration Opportunities

CS Standard	ELA Standard(s)
E.DA.1A. Organize and present data visually to highlight relationships and support claims.	W.1.1 Write opinion pieces W.1.2 Write informative texts W.1.6 Use digital tools for writing W.1.8 Use info to answer a question SL.1.5 Add visual displays to descriptions
E.DA.4A. Organize and present data visually in at least three ways to highlight relationships and evaluate a claim.	W.1.1 Write opinion pieces W.1.2 Write informative texts W.1.5 Write narratives W.1.6 Use digital tools for writing W.1.8 Use info to answer a question SL.1.5 Add visual displays to descriptions
E.DA.5B. Analyze data for patterns and relationships.	RF.1.3a Know the spelling-sound correspondences for common consonant digraphs RF.1.3c Know final -e and common vowel team conventions RF.1.3e Decode two-syllable words W.1.1 Write opinion pieces W.1.2 Write informative texts W.1.5 Write narratives W.1.8 Use info to answer a question L.1.1c Use singular and plural nouns with matching verbs in basic sentences SL.1.2 Sort words in categories L.1.5b Define words by category RF.1.1a Recognize the distinguishing features of a sentence
E.AP.1B. Create, use, and apply an algorithm.	RF.1.3a Know . . . common consonant digraphs RF.1.3c Know final -e and common vowel team conventions RF.1.3d Determine the number of syllables in a printed word RF.1.3e Decode two-syllable words W.1.5 Write narratives L.1.1c Use singular and plural nouns with matching verbs L.1.2e Spell untaught words phonetically L.1.2d Use conventional spelling for words with common spelling patterns RF.1.2a Distinguish long from short vowel sounds RF.1.2b Orally produce single-syllable words
E.AP.3A. Define and apply decomposition.	RF.1.2c Isolate and pronounce phonemes RF.1.2d Segment spoken single-syllable words RF.1.3b Decode regularly spelled words W.1.5 Write narratives W.1.6 Use digital tools for writing W.1.8 Use info to answer a question RF.1.1a Recognize the distinguishing features of a sentence
E.AP.4E. Describe and justify the steps taken and choices made during a program's development.	W.1.1 Write opinion pieces W.1.2 Write informative texts W.1.5 Write narratives W.1.8 Use info to answer a question

5 Discussion
5.1 Systematic Literature Review
5.1.1 Overview of Findings
The most important finding of this SLR is the lack of research on CS+ELA integration for the youngest students. We were able to identify fewer than 20 relevant studies. Further, many of these studies were only minimally focused on CS+ELA integration for K-2 students. For example, in one study, only 1 of 4 stations (at a festival) was targeted at 1st and 2nd grade students, and only 1 of 3 tasks at that station was focused on a literacy-related concept [39]. Thus, overall, only a small portion of this study's findings were germane to the topic of this review.

The prevalence of non-US and non-English based research is notable, especially since some articles were excluded from the dataset since they were not written in English. Studies that measured CS and/or CT skills usually found improvement based on exposure to integrated CS+ELA instruction. However, since some studies did not identify improvements in CT skills, more research is warranted to determine the conditions under which instruction is most likely to lead to improved CT skills.

The emphasis on CS/CT skills as a measure is appropriate, in order to better understand the outcomes of integrated instruction. However, other outcomes are also important but are less well-represented in this dataset. One scale for measuring the engagement of 3rd-5th grade students when they participate in learning activities that integrate coding with writing covers four dimensions of engagement (affective, behavioral, cognitive, and social) [46] and suggests the variety of constructs that might be measured in studies of integrated CS+ELA instruction.

As described above, by far the most common intervention was the CAL curriculum, and this curriculum usually led to gains in CT and/or CS skills in students who experienced it. However, improvements were not universal, so more research is warranted. Similarly, few approaches to CS+ELA curricula other than CAL have been studied, so it is unknown how CAL might compare with similar curricula in terms of student learning outcomes. Nonetheless, the findings from studies of CAL suggest that (1) integrated CS+ELA instruction is likely to lead to improvements in CS and/or CT skills and (2) CAL is an effective approach to integrating CS+ELA. Additional research into CAL might more closely examine its impact on ELA skills (since only one portion of one study investigated ELA learning outcomes for CAL; see below).

5.1.2 Gaps in the Literature
Scarborough's Reading Rope. As described above, very few interventions focused explicitly on specific aspects of reading skills; rather, most papers in this dataset emphasized other ELA topics such as digital storytelling or language development more generally - or considered literacy by broad definitions without naming specific aspects of literacy skills to be targeted with integrated CS instruction. More research into reading-specific skills (generally and in terms of the specific skills identified in Scarborough's Reading Rope) is warranted to advance the understanding of integrated CS+ELA instruction. Similarly, other ELA skills not included in Scarborough's Reading Rope (e.g., writing, oral communication) could also be investigated for their potential for integrated CS+ELA instruction. Additionally, many of the studies examined conceptualized CS skills as a form of literacy within broader definitions of literacies. For example, Zourmpakis framed CT as a vital twenty-first century literacy for learners [41]. However, this emphasis on positioning CS as a distinct literacy often resulted in integration efforts that foregrounded the

inclusion of CS alongside ELA, rather than examining how CS practices intersect with and support existing literacy skills. As a result, fewer studies explicitly or comprehensively addressed the conceptual and instructional connections between CT and foundational literacy processes. Especially in light of the research indicating teachers' success with integrating CS when they are incorporating it into already familiar content instruction, connections to specific foundational literacy skills that are already a part of educators' current curriculum aligned with the Reading Rope may be particularly pragmatic and realistic for integrating CS successfully into the classroom instruction.

ELA-related outcomes and publication venues. The study of integrated CS+ELA appears to skew toward CS, per the choice of measures for research studies and the choice of publication venue.

Only two studies examined ELA-related outcomes. In one of those studies, only in the pilot study – not in the RCT – were ELA-related measures assessed [32], finding that ELA development was not harmed by the CS instruction, even when the integrated instruction occurred during time normally devoted to literacy. In the other study, (Turkish) language development was measured [33], finding improvement for students exposed to integrated CS instruction. While these findings are important, and point to the possibility that integrated CS+ELA might improve ELA learning outcomes, they also indicate the overall paucity of the assessment of the impact of integrated CS+ELA instruction on students' ELA development.

Similarly, the majority of the dataset was published in venues focused on educational technology, with others in general education and computer science education venues; none were in ELA venues.

The lack of literature in ELA-related venues and lack of ELA-related measures suggest an opportunity to expand research into integrated CS education in ways that may enrich the field and encourage more interdisciplinary collaboration, as well as elevating the voices and expertise of ELA education experts. One barrier to CS+ELA integration that has been identified are concerns by educators (e.g., lack of professional development, lack of coaching and resources, lack of teacher buy-in) [8]; better integration of ELA venues and ELA measures may provide the foundation for mitigating these concerns.

Implementation guidance and research practices. In this set of studies, there was little guidance for implementing CS+ELA instruction. One exception to this general pattern was the observation that stories might be productively used to introduce CS/CT concepts [37] in order to make the concepts more relatable and understandable and to leverage the ability of CS/CT and ELA to support each other.

5.2 Standards Mapping

As Table 3 and Figures 2, 3, and 4 show, there are many ways to integrate CS and ELA instruction.

An important finding from the process of mapping CS and ELA standards is that appropriate mapping often depends on *how* a concept is taught. For example, one ELA standard is "ask and answer questions about key details in a text" (RI.1.1). This standard can map to many CS standards if the standard is taught via an informational text. For example, the CS standards "describe how computing has changed the ways people live and work" (E.IC.1A) and "explain how networks connect computers to other computing systems and the internet" (E.NI.1A) could easily involve asking and answering questions about details of a text. Of course, if these CS concepts were taught via another modality (e.g., watching a video, playing an educational game), then the ELA standard would not map well to the CS standard. To an extent, any CS standard might be taught via an informational text, but certainly some standards (e.g., learning how networks connect computers) are more amenable to this approach than others (e.g., organizing and presenting data visually), especially for young students. We believe that our two-fold approach to mapping provides the most utility and flexibility for those seeking to integrate CS and ELA (e.g., teachers, curriculum developers).

We identified six CS standards that are particularly promising avenues for ELA integration (see Table 4). We note that two of these six standards (E.DA.1A and E.DA.4A) are very similar. Thus, there are five CS topics where integration is particularly promising:

- *organizing and presenting data visually* (E.DA.1A and E.DA.4A). Links to ELA standards stem from the fact that both of these standards use data visualization to highlight relationships and support/evaluate claims, thus creating a natural relationship between many kinds of writing if data visualizations are used to support the written text. Similarly, W.1.8 asks students to use information to answer a question, which might be done with data. Using digital tools for writing (W.1.6) is also a logical fit.
- *analyzing data* (E.DA.5B). Many reading tasks (e.g., decoding words [RF.1.3e], sorting words into categories [SL.1.2], and recognizing sentence features [RF.1.1a]) rely on analyzing data to identify patterns and relationships, thus suggesting a logical relationship between this CS standard and any ELA standard that involves identifying patterns in words or sentences.
- *creating and using algorithms* (E.AP.1B). For many ELA skills, once a pattern has been identified (see above), an algorithm (i.e., a set of steps) can be used to complete a task. For example, "determine the number of syllables in a printed word" (RF.1.3d) and "distinguish long from short vowel sounds" (RF.1.2a) can both be accomplished by learning and enacting an algorithm. Focusing on the algorithmic nature of ELA skills may help students master these skills by explicitly teaching the steps to be followed. Inductive learning approaches can be used when a student is asked to create an algorithm based on the patterns they observe; deductive learning approaches can be used when the teacher provides students with an algorithm to apply to a task.
- *applying decomposition* (E.AP.3A). In any situation where an ELA task requires breaking down a word into smaller units (e.g., syllables [RF.1.2d], phonemes [RF.1.2c]), decomposition is required. Similarly, when students write narratives [W.1.5], they will first break down a story or event into smaller units. Linking these ELA tasks to the CS concept of decomposition can help students transfer their learning between domains.
- *describing steps taken and choices making during program development* (E.AP.4E). Documenting programming decisions may be done by writing an opinion piece (W.1.1), an informative text (W.1.2), or a narrative (W.1.5). In all of these cases, there is a clear overlap between this CS topic and writing skills.

Very few CS or ELA standards were not mapped to the other content area. Only two CS standards were not mapped, and even for these standards, there are still opportunities to integrate CS and ELA. One of the unmapped CS standards is “create a simple program to achieve a goal with expected outcomes” (E.AP.4A). This standard could be integrated into ELA instruction by asking students to create a program where the content of the program is related to an ELA standard. For example, students could use block-based programming to create a game that involves one of the unmapped ELA standards, such as “use frequently occurring prepositions” (L.1.1i), with the program incorporating prepositions into gameplay (e.g., the player has to select which preposition is illustrated by the action in the game). With this approach, any of the unmapped ELA standards could be addressed. Similarly, the other unmapped CS standard (“Using an iterative process, test a program step-by-step and document areas of refinement,” E.AP.4F) could involve an activity where a student documents areas of refinement with special attention to unmapped ELA standards (e.g., “Use frequently occurring adjectives” [L.1.1f] or “identify real-life connections between words and their use” [L.1.5c]). Thus, it is possible to integrate all of the CS standards with ELA instruction.

6 Conclusions

In this paper, we present what we believe to be the first systematic literature review of studies that integrate CS with ELA instruction for very young students. We find that there is a pressing need for additional research in this area given the potential benefits of such instruction. We also mapped CS and ELA standards, concluding that – despite the lack of extensive prior research – there are ample opportunities to teach integrated CS and ELA to young students. Integration holds promise for making CS instruction more widely available to more students, and CS instruction at the earliest grade presents an opportunity to begin developing knowledge and skills foundational to all branches of engineering education.

7 Acknowledgments

This material is based upon work supported by the National Science Foundation under Award No. 2524330. 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] Claude E. Shannon. A symbolic analysis of relay and switching circuits. 57(12):713–723. ISSN 0095-9197, 2376-7804. doi: 10.1109/EE.1938.6431064. URL `https://ieeexplore.ieee.org/document/6431064/`.

[2] A. Newell, A. Perlis, and H.A. Simon. What is computer science? (157):1373–4. URL `https://www.cs.cmu.edu/~choset/whatiscs.html`.

[3] Sebastian Gross, Mischa Kim, Joachim Schlosser, Coorous Mohtadi, Dan Lluch, and Dorian Schneider. Fostering computational thinking in engineering education: Challenges, examples, and best practices. In *2014 IEEE Global Engineering Education Conference (EDUCON)*, pages 450–459, April 2014. doi: 10.1109/EDUCON.2014.6826132. URL `https://ieeexplore.ieee.org/document/6826132`. ISSN: 2165-9567.

[4] Hendra Hidayat, Zulhendra Zulhendra, Efrizon Efrizon, Vera Irma Delianti, Fitrika Kumala Dewi, Mohd Rizal Mohd Isa, Dani Harmanto, and Jem Cloyd M. Tanucan. Computational Thinking Skills in Engineering Education: Enhancing Academic Achievement Through Innovations, Challenges, and Opportunities. *TEM Journal*, pages 3454–3467, November 2024. ISSN 2217-8333, 2217-8309. doi: 10.18421/TEM134-78. URL https://www.temjournal.com/content/134/TEMJournalNovember2024_3454_3467.html.

[5] Alejandra J. Magana and Genisson Silva Coutinho. Modeling and simulation practices for a computational thinking-enabled engineering workforce. *Computer Applications in Engineering Education*, 25(1):62–78, 2017. ISSN 1099-0542. doi: 10.1002/cae.21779. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.21779. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cae.21779.

[6] Noemi V. Mendoza Diaz, Russ Meier, Deborah A. Trytten, and So Yoon Yoon. Computational Thinking Growth During a First-Year Engineering Course. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–7, October 2020. doi: 10.1109/FIE44824.2020.9274250. URL https://ieeexplore.ieee.org/document/9274250. ISSN: 2377-634X.

[7] Mohamed Khaleel, Abdussalam Ali Ahmed, and Abdulgader Alsharif. Artificial Intelligence in Engineering | Brilliance: Research of Artificial Intelligence. September 2023. URL https://jurnal.itscience.org/index.php/brilliance/article/view/2170.

[8] Monica M. McGill, Laycee Thigpen, and Alaina Mabie. Emerging Practices for Integrating Computer Science into Existing K-5 Subjects in the United States. In *Proceedings of the 18th WiPSCE Conference on Primary and Secondary Computing Education Research*, pages 1–10, Cambridge United Kingdom, September 2023. ACM. ISBN 9798400708510. doi: 10.1145/3605468.3609759. URL https://dl.acm.org/doi/10.1145/3605468.3609759.

[9] Carolina Robledo-Castro, Christian Hederich-Martínez, and Luis F. Castillo-Ossa. Cognitive stimulation of executive functions through computational thinking. *Journal of Experimental Child Psychology*, 235:105738, November 2023. ISSN 00220965. doi: 10.1016/j.jecp.2023.105738. URL https://linkinghub.elsevier.com/retrieve/pii/S0022096523001145.

[10] Nikolaos Pellas. Enhancing Computational Thinking, Spatial Reasoning, and Executive Function Skills: The Impact of Tangible Programming Tools in Early Childhood and Across Different Learner Stages. *Journal of Educational Computing Research*, 63(1):3–32, March 2025. ISSN 0735-6331, 1541-4140. doi: 10.1177/07356331241292767. URL https://journals.sagepub.com/doi/10.1177/07356331241292767.

[11] Sapna Cheryan, Allison Master, and Andrew N. Meltzoff. Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in Psychology*, 6(FEB):1–8, 2015. ISSN 16641078. doi: 10.3389/fpsyg.2015.00049.

[12] Laycee Thigpen and Monica McGill. Intrinsic Barriers to Supporting the Integration of Computer Science Into Elementary School Subject Areas. In Hamid R. Arabnia, Leonidas Deligiannidis, Farzan Shenavarmasouleh, Soheyla Amirian, and Farid Ghareh Mohammadi, editors, *Computational Science and Computational Intelligence*, volume 2504, pages 12–28. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-94942-5 978-3-031-94943-2. doi: 10.1007/978-3-031-94943-2_2. URL https://link.springer.com/10.1007/978-3-031-94943-2_2. Series Title: Communications in Computer and Information Science.

[13] Aman Yadav, Hai Hong, and Chris Stephenson. Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends*, 60(6):565–568, November 2016. ISSN 8756-3894, 1559-7075. doi: 10.1007/s11528-016-0087-7. URL http://link.springer.com/10.1007/s11528-016-0087-7.

[14] Jeanne Century, Kaitlyn A. Ferris, and Huifang Zuo. Finding time for computer science in the elementary school day: a quasi-experimental study of a transdisciplinary problem-based learning approach. *International*

Journal of STEM Education, 7(1):20, December 2020. ISSN 2196-7822. doi: 10.1186/s40594-020-00218-3. URL https://stemeducationjournal.springeropen.com/articles/10.1186/s40594-020-00218-3.

[15] Marina Umaschi Bers. Coding as another language: a pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, 6(4):499–528, December 2019. ISSN 2197-9995. doi: 10.1007/s40692-019-00147-3. URL https://doi.org/10.1007/s40692-019-00147-3.

[16] Robert Whyte, Shaaron Ainsworth, and Jane Medwell. Designing Multimodal Composition Activities for Integrated K-5 Programming and Storytelling. 2020. Publisher: International Society of the Learning Sciences (ISLS).

[17] Ammar H Safar, Ali A Al-Jafar, and Zainab H Al-Yousefi. The effectiveness of using augmented reality apps in teaching the English alphabet to kindergarten children: A case study in the State of Kuwait. *EURASIA Journal of Mathematics, Science and Technology Education*, 13(2):417–440, 2016. Publisher: Modestum LTD.

[18] Alaina Mabie, Monica McGill, and Brenda Huerta. A Systematic Literature Review Examining the Impacts of Integrating Computer Science in K-5 Settings. In *2023 ASEE Annual Conference & Exposition*, Baltimore, Maryland, June 2023. URL https://nemo.asee.org/public/conferences/327/papers/39653/view.

[19] Carla Strickland, Kathryn M Rich, Donna Eatinger, Todd Lash, Andy Isaacs, Maya Israel, and Diana Franklin. Action Fractions: The design and pilot of an integrated math+ CS elementary curriculum based on learning trajectories. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 1149–1155, 2021.

[20] Feiya Luo, Wei Yan, Ruohan Liu, and Maya Israel. Elementary Students' Understanding of Variables in Computational Thinking-Integrated Instruction: A Mixed Methods Study. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1*, pages 523–529, 2022.

[21] Gashawa Ahmed, Jalal Nouri, LeChen Zhang, and Eva Norén. Didactic methods of integrating programming in mathematics in primary school: Findings from a swedish national project. In *Proceedings of the 51st ACM technical symposium on computer science education*, pages 261–267, 2020.

[22] Theodore J Kopcha, Cheryl Y Wilson, and Dayae Yang. Improving teacher use of educational robotics to teach computer science in K-5 mathematics. In *Computational Thinking in PreK-5: Empirical Evidence for Integration and Future Directions*, pages 47–54. 2022.

[23] Aman Yadav, Christina Krist, Jon Good, and Elisa Nadire Caeli. Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, 28(4):371–400, 2018. Publisher: Taylor & Francis.

[24] Marina Umaschi Bers. Computer science education as a humanistic endeavor: a model for designing technology-rich formative experiences. *Education and Information Technologies*, 30(15):21519–21550, October 2025. ISSN 1573-7608. doi: 10.1007/s10639-025-13623-4. URL https://doi.org/10.1007/s10639-025-13623-4.

[25] Robert Whyte, Shaaron Ainsworth, and Jane Medwell. Designing for Integrated K-5 Computing and Literacy through Story-making Activities. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, pages 167–175, 2019.

[26] Hollis S. Scarborough, S. B. Neuman, and D. K. Dickinson. Connecting early language and literacy to later reading (dis) abilities: Evidence, theory, and practice. *Handbook of early literacy research*, 1:97–110, 2001. URL https://books.google.com/books?hl=en&lr=&id=0HXODgAAQBAJ&oi=fnd&pg=PA97&dq=Connecting+early+language+and+literacy+to+later+reading+(dis)abilities:+Evidence,+Theory,+and+Practice&ots=VDv2WAp4ra&sig=qZW2srQU269GhZjJk58FOuAqSdQ.

[27] Maria J. Grant and Andrew Booth. A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal*, 26(2):91–108, June 2009. ISSN 1471-1834, 1471-1842. doi: 10.1111/j.1471-1842.2009.00848.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1471-1842.2009.00848.x.

[28] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, pages 1–10, New York, NY, USA, May 2014. Association for Computing Machinery. ISBN 978-1-4503-2476-2. doi: 10.1145/2601248.2601268. URL https://doi.org/10.1145/2601248.2601268.

[29] M. Ariel Cascio, Eunlye Lee, Nicole Vaudrin, and Darcy A. Freedman. A Team-based Approach to Open Coding: Considerations for Creating Intercoder Consensus. *Field Methods*, 31(2):116–130, May 2019. ISSN 1525-822X, 1552-3969. doi: 10.1177/1525822X19838237. URL https://journals.sagepub.com/doi/10.1177/1525822X19838237.

[30] E. Relkin, L. E. de Ruiter, and M. U. Bers. Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, 169:104222, August 2021. ISSN 0360-1315. doi: 10.1016/j.compedu.2021.104222. URL https://www.sciencedirect.com/science/article/pii/S0360131521000993.

[31] Zhanxia Yang, Jessica Blake-West, Dandan Yang, and Marina Bers. The impact of a block-based visual programming curriculum: Untangling coding skills and computational thinking. *Learning and Instruction*, 95: 102041, February 2025. ISSN 0959-4752. doi: 10.1016/j.learninstruc.2024.102041. URL h10.1016/j.learninstruc.2024.102041.

[32] Marina Umaschi Bers, Jessica Blake-West, Madhu Govind Kapoor, Tess Levinson, Emily Relkin, Apittha Unahalekhaka, and Zhanxia Yang. Coding as another language: Research-based curriculum for early childhood computer science. *Early Childhood Research Quarterly*, 64:394–404, July 2023. ISSN 0885-2006. doi: 10.1016/j.ecresq.2023.05.002. URL https://www.sciencedirect.com/science/article/pii/S0885200623000571.

[33] Merve Canbeldek and Nesrin Isikoglu. Exploring the effects of "productive children: coding and robotics education program" in early childhood education. *Education and Information Technologies*, 28(3):3359–3379, March 2023. ISSN 1573-7608. doi: 10.1007/s10639-022-11315-x. URL https://doi.org/10.1007/s10639-022-11315-x.

[34] Ana Francisca Monteiro, Maribel Miranda-Pinto, and António José Osório. Coding as Literacy in Preschool: A Case Study. *Education Sciences*, 11(5):198, May 2021. ISSN 2227-7102. doi: 10.3390/educsci11050198. URL https://www.mdpi.com/2227-7102/11/5/198. Publisher: Multidisciplinary Digital Publishing Institute.

[35] Eleni A. Kyza, Yiannis Georgiou, Andria Agesilaou, and Markos Souropetsis. A Cross-Sectional Study Investigating Primary School Children's Coding Practices and Computational Thinking Using ScratchJr. *Journal of Educational Computing Research*, 60(1):220–257, March 2022. ISSN 0735-6331. doi: 10.1177/07356331211027387. URL https://doi.org/10.1177/07356331211027387. Publisher: SAGE Publications Inc.

[36] Yin-Chan Liao, Jiyoung Kim, Anne T. Ottenbreit-Leftwich, Michael Karlin, and Meize Guo. Voices of Elementary Computer Science Teachers: Computer Science Integration Rationales and Practices. *ACM Trans. Comput. Educ.*, 24(4):43:1–43:26, October 2024. doi: 10.1145/3688854. URL https://doi.org/10.1145/3688854.

[37] Kristina M. Tank, Tamara J. Moore, Anne Ottenbreit-Leftwich, Zarina Wafula, Lin Chu, and Sohheon Yang. Pathways to inclusive early childhood computational thinking education: unveiling young students' strategies with multiple representations. *Journal of Research on Technology in Education*, 57(1):2–25, January 2025. ISSN 1539-1523. doi: 10.1080/15391523.2024.2410194. URL

https://doi.org/10.1080/15391523.2024.2410194. Publisher: Routledge _eprint: https://doi.org/10.1080/15391523.2024.2410194.

[38] Lauren E. Margulieux, Yin-Chan Liao, Erin Anderson, Miranda C. Parker, and Brendan D. Calandra. Intent and Extent: Computer Science Concepts and Practices in Integrated Computing. *ACM Trans. Comput. Educ.*, 24(3): 35:1–35:23, August 2024. doi: 10.1145/3664825. URL https://doi.org/10.1145/3664825.

[39] Ana Isabel Santos, Dora Pereira, Nanci Botelho, Paulo Medeiros, and José Manuel Cascalho. Robotics for young people in azoresminibot. *International Journal of Information and Education Technology*, 13(1):33–41, 2023. URL https://www.academia.edu/download/118595826/IJIET-V13N1-1777-IJIET-6241.pdf.

[40] Tess Levinson, Francisca Carocca P, and Marina Bers. Coding as another language: an early childhood programming curriculum in Argentina. *Computer Science Education*, 0(0):1–21, January 2025. ISSN 0899-3408. doi: 10.1080/08993408.2025.2455339. URL https://doi.org/10.1080/08993408.2025.2455339. Publisher: Routledge _eprint: https://doi.org/10.1080/08993408.2025.2455339.

[41] Alkinoos Ioannis Zourmpakis. Developing Computational Thinking in Early Childhood Education: Long-Term Impacts on CT Skills and Motivation Using the CAL Approach, ScratchJr, and Gamification. *Advances in Mobile Learning Educational Research*, 5(2):1536–1547, August 2025. ISSN 2737-5676. doi: 10.25082/AMLER.2025.02.009. URL https://www.syncsci.com/journal/AMLER/article/view/AMLER.2025.02.009.

[42] Madhu Kapoor, Zhanxia Yang, and Marina Bers. Supporting Early Elementary Teachers' Coding Knowledge and Self-Efficacy Through Virtual Professional Development. *Journal of Technology and Teacher Education*, 30(4):461–491, 2022. ISSN 1059-7069. URL https://www.learntechlib.org/primary/p/221629/. Publisher: Society for Information Technology & Teacher Education.

[43] Avia Ben-Ari, T. Levinson, Marina Umaschi Bers, and Rinat B. Rosenberg-Kima. Nurturing Computational Thinking in an Israeli Kindergarten with the CAL-KIBO Robotics Curriculum, 2023. URL https://www.aera.net/Publications/Online-Paper-Repository/AERA-Online-Paper-Repository-Viewer/ID/2012601.

[44] Dandan Yang and Marina Bers. Enhancing Computer Science Education for K–2 Students: Insights From a Randomized Controlled Trial. URL https://convention2.allacademic.com/one/aera/aera24/online_program_direct_link/view_paper/2101340/.

[45] T. Levinson and Marina Bers. Student Centered Computational Thinking for Children with Disabilities., 2022.

[46] Amy Hutchison, Seth A. Parsons, and Samantha T. Ives. Development of the Coding Engagement Scale to measure elementary students' engagement in coding. *British Journal of Educational Technology*, page bjet.70025, October 2025. ISSN 0007-1013, 1467-8535. doi: 10.1111/bjet.70025. URL https://bera-journals.onlinelibrary.wiley.com/doi/10.1111/bjet.70025.