Middle School CS Curriculum and Standards Alignment

Daisy Rutstein
dazerw@yahoo.com
edCount, LLC

Satabdi Basu
satabdi.basu@sri.com
SRI International

Hui Yang
hui.yang@sri.com
SRI International

Arif Rachmatullah
arif.rachmatullah@sri.com
SRI International

Carol Tate
carol.tate@sri.com
SRI International

ABSTRACT

The development of the CS content standards underscores the importance of curricula aligned with the standards, ensuring equitable coverage of CS concepts for all students. Because standards are broad, we emphasize the need for CS curricula to specify not only the standards they align with but also which aspects of the standards they align with and how. We map one common middle school CS curriculum to a few standards to demonstrate this need.

ACM Reference Format:
Daisy Rutstein, Satabdi Basu, Hui Yang, Arif Rachmatullah, and Carol Tate. 2024. Middle School CS Curriculum and Standards Alignment. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2024), March 20–23, 2024, Portland, OR, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3626253.3635546

1 INTRODUCTION

As the demand for CS professionals and computationally literate citizens grows, the number of CS offerings in K-12 schools has significantly expanded. Quality CS curriculum is a cornerstone of equitable CS education and determines what CS constructs students will learn and how they will learn the target constructs. However, there are significant disparities in access to quality CS curricula in the U.S. due to various factors including the fact that curriculum decisions are being made at district, school, or even classroom levels. CS standards alignment can provide an objective way to compare different CS curricula and determine their quality. However, mapping CS standards [1] to curricula is complex as the standards are often too broad and open to interpretation [2]. Standards outline the target content and practices and, in theory, support a coherent approach across grade levels, but their high-level descriptions make it challenging to use them as a guide for curriculum selection.

When CS curricula specify standards alignment, it is not very meaningful without a description of how lessons align with the standard and which parts of the standard lessons align with. Towards this end, our paper seeks to explore the standards alignment specified by a well known middle school CS curriculum and how well it actually enables student learning of concepts emphasized in the middle school CS standards for algorithms and programming.

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

ACM ISBN 979-8-4007-0424-6/24/03.
https://doi.org/10.1145/3626253.3635546

2 BACKGROUND

The landscape of middle school CS education currently includes several CS curricula, some common ones being Code.org's CS Discoveries, Google's CS First, CodeHS's middle school CS pathways, Project STEM's CS Explorations, and CS Unplugged by Colorado School of Mines. Most of these curricula describe alignment with either the K-12 CS Framework or the Computer Science Teachers Association (CSTA) CS standards [1]. However, the descriptions do not include which aspects of the standards their curricula address or how their curricula address the standards. While standards alignment may not be the only criterion education leaders and teachers use for curriculum selection, this lack of detailed information makes the selection process particularly challenging.

In this paper, we focus on code.org's Computer Science Discoveries (CSD) curriculum [5], designed to captivate and engage middle school students with no to limited prior CS background. The Code.org 2021 Annual Report [4] mentions that 788,377 students began using CSD that academic year (approximately 41% were female and 48% belonged to racial and ethnic groups underserved in STEM/CS). Based on these statistics, CSD is considered as one of the most widely embraced middle school CS curricula in the US, and thus seems like an appropriate choice for starting to explore standards alignment of CS curricula.

3 METHODS

In this paper, we focus on five middle school CSTA standards from the 'Algorithms and Programming' (AP) strand (2-AP-10 through 2-AP-14) covering the concepts of algorithms, variables, combined control structures, decomposition, and procedures, respectively. We used an Evidence-Centered Design approach [7]to break down each standard into meaningful and granular learning targets (LTs). These LTs were based on current literature related to the standards and prior documents that unpacked the concepts in the standard [3, 6]. The LTs went through a round of expert review by 4 external K-12 CS education experts and refinement based on the expert feedback.

Once the LTs were finalized, we examined the alignment between our five target standards and the CSD curriculum (Units 1-6A) by focusing on the alignment of LTs to individual CSD lessons. For each standard, we studied its alignment with the CSD curriculum along three dimensions: (1) the number of lessons that aligned with the standard in some way, (2) the mapping between specific LTs for the standard and specific CSD lessons, and (3) the strength of alignment between aligned LTs and lessons.

For each lesson, we started by noting the standards that the curriculum developers identified as aligned with the lesson. Next, we went through the set of activities students engage with during

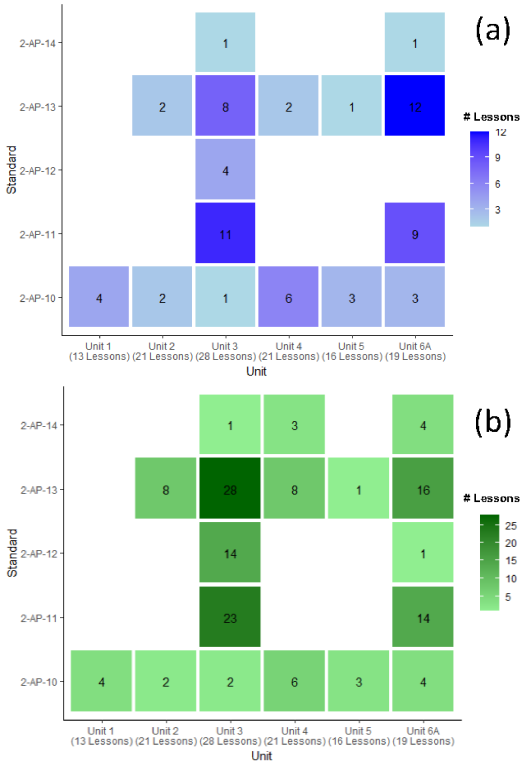

Figure 1: Number of CSD lessons aligned with the target standards: a) Only directly aligned lessons; b) Directly and indirectly aligned lessons

the lesson to determine if any of the activities relate to any of the LTs across all five standards. We identified two levels of alignment, direct and indirect. A direct alignment indicates that students received instruction that was directly related to the LT. For example, when students had an activity where they identified problems with variable names, this was directly aligned with our 2-AP-11 LT related to the naming of variables. For an indirect alignment, students may have opportunities to engage with the LT but it may depend on how the teacher frames the activities or how the students design their projects. For example, alignment with particular LTs that focus on control structures for lessons in which students are given the freedom to pick their own projects may depend on which control structures, if any, students decide to use. We used a lesson's alignment to LTs to determine its overall alignment to a standard. If a given lesson was directly aligned to any LT for a standard, we stated that the overall standard was directly aligned to the lesson. If the only LT alignments for a standard were indirect alignments, we called the overall standard indirectly aligned to the lesson. If no LTs for a standard were specified as either direct or indirect alignment, the standard was deemed not aligned to the lesson.

4 FINDINGS AND DISCUSSION

Figure 1 illustrates the alignment of CSD units 1-6 with our five target standards. CSD units that did not focus on programming concepts (units 1, 2, 4, and 5) were only aligned with 2-AP-10 (algorithms) and/or 2-AP-13 (decomposition) and were not aligned with the programming standards. Overall, we found that all CSD units covered at least one of our five target standards, though alignment was limited to specific aspects of standards. For example, Unit 1 focused only on recognizing relevant information to create an algorithm, but did not cover other aspects such as creating flowcharts or comparing or testing algorithms. Unit 3 covered all standards but focused only on specific aspects such as knowledge that algorithms need to be precise, simple variables (did not include compound variables such as lists), compound conditionals (did not include nested loops or nested conditionals which are called out in the standard), and creating and calling procedures without input parameters.

Discussion. Currently, most CS curricula indicate their standards alignment. We point out that the usefulness of this approach is limited if curricula do not specify how they align with standards and which aspects of standards lessons align with. Using the CSD curriculum, we demonstrate what detailed standards alignment can reveal about the content coverage of a curriculum. Examining alignment with only five CS standards, we find that while CSD covers all the standards to varying degrees, it does not provide complete coverage of all components of the standards. This is not an indication of the quality of the curriculum, but more a recognition that the CS standards are complex and cover a lot in one standard [2]. Additionally, CS curricula typically need to cover some precursors to the grade-level standards, increasing the number of concepts to cover. A detailed and more transparent account of standards alignment can help school leaders make informed choices when selecting curricula and can indicate to teachers that they need to supplement student learning on certain concepts. Our paper demonstrates the value of expanding this work to unpack more CS standards and examine standards alignment for additional curricula.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. DRL-2010591. 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] Computer Science Teachers Association. 2017. CSTA K-12 computer science standards. Revised 2017. http://www.csteachers.org/standards
[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-Volume 1*. 404–410.
[3] Satabdi Basu, Daisy W Rutstein, Yuning Xu, Haiwen Wang, and Linda Shear. 2021. A principled approach to designing computational thinking concepts and practices assessments for upper elementary grades. *Computer Science Education* 31, 2 (2021), 169–198.
[4] Code.org. 2021. 2021 Annual Report. https://code.org/files/code.org-annual-report-2021.pdf
[5] Code.org. n.d. Computer Science Discoveries. https://code.org/educate/csd
[6] Shuchi Grover and Satabdi Basu. 2017. Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*. 267–272.
[7] Robert J Mislevy and Geneva D Haertel. 2006. Implications of evidence-centered design for educational testing. *Educational measurement: issues and practice* 25, 4 (2006), 6–20.