A Structured Inventory of Tools to Unveil Teachers' Computer Science Knowledge

Agnese Del Zozzo
agnese.delzozzo@unitn.it
University of Trento
Lab. CINI Informatica e Scuola
Trento, Italy

Luca Lamanna
luca.lamanna@unimi.it
University of Milan
Lab. CINI Informatica e Scuola
Milan, Italy

Violetta Lonati
lonati@di.unimi.it
University of Milan
Lab. CINI Informatica e Scuola
Milan, Italy

Alberto Montresor
alberto.montresor@unitn.it
University of Trento
Lab. CINI Informatica e Scuola
Trento, Italy

Abstract

Teaching computer science (CS) at all school levels is an increasingly important issue that has recently attracted the attention of institutions and policymakers. However, there is a shortage of qualified teachers which highlights an urgent need for training initiatives. To design such initiatives, it is crucial to assess teachers' baseline CS knowledge, identifying possible gaps. This study aims to articulate a proposal for a classification of tools that can be employed to assess teachers' CS knowledge. A literature review was conducted to collect relevant studies that present and/or utilize tools for assessing teachers' CS knowledge and competencies. We have identified key dimensions under which the tools can be examined: the data collection method, the nature of tasks assigned, how their reflection and action are triggered, whether the tool focuses on teachers' products or cognitive processes, and whether it also assesses their pedagogical content knowledge. For each dimension, we have identified a range of possible descriptors, leading to the development of a synoptic table categorizing the tools. This framework also enables us to compare the tools based on their features, highlighting their strengths and applications. This work makes two contributions: first, it introduces a framework for describing and comparing tools to unveil teachers' CS knowledge. Second, it offers a collection of tools for assessing it. This approach not only aids in selecting suitable tools for specific contexts but also provides a foundation for designing CS-knowledge assessment tools, enriching the resources available for analyzing educational needs and developing high-quality teacher training programs.

CCS Concepts

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

ACM ISBN 979-8-4007-2256-1/2026/02
https://doi.org/10.1145/3770762.3772555

Keywords

Teachers' knowledge, Primary and secondary education, Literature review, CS knowledge

ACM Reference Format:
Agnese Del Zozzo, Luca Lamanna, Violetta Lonati, and Alberto Montresor. 2026. A Structured Inventory of Tools to Unveil Teachers' Computer Science Knowledge. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE TS 2026), February 18–21, 2026, St. Louis, MO, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3770762.3772555

1 Introduction and background

The issue of teaching Computer Science (CS) at various educational stages has emerged as a critically important but complex topic. In recent years, this field has garnered increasing attention, drawing the focus of numerous educational institutions and policymakers who recognize the growing significance of integrating CS education throughout the school curriculum. However, this effort faces a severe shortage of qualified teachers and thus an urgent need to train them [21].

In his seminal work [24], Lee Shulman distinguishes between three types of teacher knowledge. Content Knowledge (CK) that regards the of knowledge *per se*; the Pedagogical Content Knowledge (PCK) "*which goes beyond knowledge of subject matter per se to the dimension of subject matter knowledge for teaching*" (p. 9); and curricular knowledge, which concerns the understanding about curricular content and curriculum materials. Regarding the CK, he assumes that assumes that "*most teachers begin with some expertise in the content they teach*" (p. 8). In CS, this assumption is at least debatable since, in many countries, it is not yet a subject of structured and institutional education [3, 22].

Given the cumulative nature of meaningful learning processes [23], developing suitable proposals for professional development (PD) requires, at least, first assessing the existing CS knowledge of teachers. This task becomes increasingly complex when considering that teachers' familiarity with CS is generally quite low. In fact, apart from isolated professional development courses, most in-service school teachers have received no formal training, either in CS or in CS teaching methods [14]. The overarching challenge, therefore, is: what considerations about teachers' CS content knowledge can we make? The question is far from trivial. As a preliminary step toward addressing it, we decided to focus on how to detect and assess teachers' knowledge of CS, an aspect that can already be particularly problematic, especially when the purpose of the assessment is not to evaluate teachers, but rather to diagnose possible formative needs or initial knowledge gaps. Detecting such knowledge is essential not only to describe teachers' preparedness but, more importantly, to design professional development (PD) initiatives that build on their existing understanding and address specific gaps.

In general, as pointed out by McGill et al. [16], most research on teachers' CK relies on instruments of self-assessment, highlighting an overall lack of quality instruments for the study of teachers' CS knowledge. On the other hand, using direct, explicit assessment tests could be perceived negatively by the teachers, who might feel they are being evaluated and become defensive. This reaction could, in turn, compromise the teacher/learner relationship that should instead be nurtured in the context of professional development. In light of this, an important question arises: are there alternative methods for assessing teachers' CS knowledge that mitigate the mentioned risks, while providing reliable data?

We are interested in identifying any instrument or methodology that can be used to assess or uncover teachers' CS knowledge, either directly or indirectly, and we use the term *knowledge detection tool* (KDT) to refer to any such tool. While the problem of assessing knowledge is a general one and occurs across various disciplines and contexts, we are focusing here in particular on *teachers' CS knowledge* since, to the best of our knowledge, a similar analysis has not been conducted in the literature yet. CS knowledge is specific in two ways: first, assessing teachers requires special care in order to preserve their self-confidence and a positive attitude towards the discipline; second, CS knowledge encompasses both foundational CS concepts and higher-order competencies such as programming and debugging, which require specialized assessment approaches.

Our definition of KDT, in line with the considerations above [16], excludes self-assessment instruments, despite their widespread use in the literature to evaluate the CS knowledge of teachers and students. Self-assessment tools do not provide evidence of the effective CS knowledge of teachers, but only on their perception. Although we acknowledge that such tools can be undoubtedly useful and interesting in formative contexts with training support, or for assessing attitudes, beliefs, confidence, or other subjective constructs, we do not consider them suitable for assessing knowledge, as they lack reliability and validity [5, 6, 26].

Our study is therefore guided by the following research questions:

- **RQ1** What KDTs are presented, used, or discussed in the CS education literature?
- **RQ2** What are the characteristics of the KDTs available to detect teachers' CS knowledge?

The contribution of this paper is twofold. First, we identify a diverse set of KDTs used in CS education research and teacher professional development after conducting a literature review on teachers' CS knowledge. Second, we analyze these tools based on key dimensions—such as data collection methods, the nature of tasks assigned to teachers, and their ability to capture different aspects of CS knowledge—leading to the development of a structured framework for describing, organizing, and comparing KDTs. This framework highlights the strengths, limitations, and applicability of these tools in various educational contexts.

The remainder of this paper is organized as follows. In Section 2, we detail our research methodology, including the criteria and process used to select relevant studies. Section 3 presents our findings, describing the key dimensions that we developed to characterize KDTs. Section 4 discusses the implications of our results, considering their relevance for both research and teacher professional development. Finally, Section 5 concludes the paper by summarizing our contributions, outlining limitations, and suggesting directions for future work.

2 Methods

To address our RQs, a four-step study was conducted and detailed as follows:

Literature review The first step of our work consisted of an analysis of existing research in CS education, aiming to identify KDTs used in studies addressing teachers' CS knowledge. To this end, we conducted a literature review to select a sample of papers considered relevant for the use and/or description of KDTs (see Section 2.1). It is worth emphasizing that this review was not the central goal of our study, but rather the initial research method we employed to identify KDTs and use its results as a basis for their description.

Analysis of selected papers The selected papers were analyzed by focusing on how KDTs are used by researchers, what type of information about teachers' knowledge is identified, their strengths and weaknesses, and their potential–beyond how they were actually used in the papers–in terms of a tool's ability to reveal CS knowledge. This step of the research involved reading and analyzing the selected papers. The analysis was first conducted individually and then triangulated by authors in pairs to ensure the reliability of the data extracted from the papers.

Analysis of each KDT Focusing on each KDT retrieved from the selected papers, the following information was extracted: the type of data collected; the administration method (*e.g.*, interviews, surveys); the questions and tasks proposed to teachers (*e.g.*, answer a multiple-choice question, write a definition, debug a program); the materials possibly used with the tool (*e.g.*, a code segment to be analyzed, classroom vignettes, a technical CS text); and whether the tool also supports the identification of PCK elements.

Definition of key dimensions and modalities Proceeding bottom-up from the collected data, we developed a uniform approach to describe the KDTs, with the aim of organizing and comparing them. The key dimensions and modalities were formulated with an inductive approach and then iteratively refined during the analysis of the identified KDTs. The refinement occurred through sessions of discussion in which all four authors participated simultaneously. This process led to the construction of a synoptic table summarizing the KDTs and their features.

2.1 Literature review

In conducting the literature review on research concerning teachers' CS knowledge, our work was inspired by the PRISMA guidelines [18].

2.1.1 Eligibility criteria. According to our research questions, we focused on works that explicitly use or mention a KDT. As discussed in the introduction, we primarily excluded papers that involved only self-assessment tools. We considered papers published in CS education venues (i.e., ACM SIGCSE conferences and journals), including both research papers and experience reports, but excluding posters, panels, and other short contributions. We selected papers that explicitly focused on teachers and their CS knowledge and skills, as we expected to find references to potential KDTs in studies investigating this topic or in papers concerning teachers' professional development (PD).

2.1.2 Search strategy. As we focused on the main SIGCSE publication outlets, whose proceedings are published by the ACM, we used the *ACM Digital Library*[1] as the source for scientific publications. We applied an additional filter to include only those papers tagged as "research articles" in the ACM Digital Library.

Our focus of interest is not identified by a standard terminology in the CS education literature, nor supported by a codified structure. In particular, terms such as "knowledge" or "competence" are not specific, have broad meanings, and are widely used in educational research. This makes it difficult to formulate a concise query to retrieve relevant papers. Hence, we constructed a query composed of three components:

Teachers knowledge = `"teacher knowledge" OR "teacher competence" OR "teacher competency"`
Teacher training = `"teacher training" OR ("teacher" AND "professional development")`
PCK = `"PCK" OR "pedagogical content knowledge"`

The first component of the query includes both "competence" and "competency" as alternatives to "knowledge", given their common usage in educational research. The goal is to capture works that directly address the main focus of our study. The second component is motivated by the expectation that considerations relevant to our research may emerge in papers addressing a PD context, even when they do not include the explicit expressions targeted by the first sub-query. The third component identifies papers that mention the PCK framework [24]; we chose to include this component in a non-restrictive way, based on the idea that works engaging with this framework might also contain remarks or statements related to content knowledge (CK) itself. These sub-queries are combined into the following query:

Teachers knowledge `OR` **Teacher training** `OR` **PCK**

The query was run in April 2024, resulting in 2774 papers. We restricted the list down to 685 papers, by considering only the following CSE venues:

- TOCE - Transactions on Computing Education (journal);
- ACE - Adult Centered Education Conference;
- CSERC - Computer Science Education Research Conference;
- CompEd - ACM Global Computing Education Conference;
- ICER - ACM International Computing Education Research Conference;
- ITiCSE - ACM Conference on Innovation and Technology in Computer Science Education (Proceedings and Working Groups Reports);
- Koli Calling - International Conference on Computing Education Research;
- SIGCSE TS - ACM Technical Symposium on Computer Science Education;
- WiPSCE - Workshop in Primary and Secondary Computing Education.

[1] https://dl.acm.org/

2.1.3 Selection process. The pool of 685 papers was filtered by reading their abstracts. To set up and refine the filtering process, we started by randomly sampling 45 papers (1 out of 15). Three authors independently read the abstracts and marked those they deemed relevant or potentially relevant. According to their abstracts, 7 papers were deemed relevant by at least two of the three authors. Among these, two papers had been selected by exactly two of the authors. The content of such papers was subsequently skimmed by the three authors, discussed, and ultimately excluded by unanimous agreement. This helped us refine the inclusion/exclusion criteria and the selection process.

The remaining 640 papers' abstracts were analyzed in a similar way, with all four authors assigned a different subset of abstracts in such a way that each abstract was read by exactly three authors. Each author independently marked their assigned abstracts as relevant or not. We then selected the 39 papers that were marked as relevant by all three authors that read their abstract. A sample of the 61 papers with only two approvals was checked and discussed. As a result of the discussion, none of these papers was included as relevant. Thus, we proceeded by analyzing only the 39 papers already identified as relevant.

During the analysis of their full content, 21 were further excluded because they either relied solely on self-assessment tools, or they provided too little information about the KDTs they mention, or they focused primarily on pedagogical aspects or on teachers' attitudes, beliefs, and self-confidence rather than their CS knowledge.

3 Findings

The 18 publications resulting from the literature review address different school levels, all within the K–12 range. All papers, except for two, refer to a PD context. In some cases, the KDT is used to assess the PD initiative; in others, the PD initiative is the opportunity to gather information about teachers' preparation. In yet other cases, learning activities are proposed during PD that could potentially be used to reveal teachers' knowledge.

From these papers, we identified 24 different KDTs (researchers often use more than one KDT within a single study). In Section 3.1, we describe the key dimensions and related modalities that we propose to use for describing, organizing, and comparing the KDTs. These dimensions and modalities were developed through an inductive approach, and iterative refinement during the analysis of the KDTs. In Section 3.2, we present the details of the synoptic table that summarizes the KDTs according to this classification.

3.1 Dimensions and modalities

Overall, we have identified five dimensions, each of which can be articulated through different modalities. Their specific role with respect to the CS discipline will be discussed in Section 4.

3.1.1 Data collection methods. This first dimension addresses the question: "How is a KDT administered?". It concerns the methods of administration and includes four modalities: direct observation, written deliverables, learning analytics, and interviews.

Direct observation. In this modality, information about teachers' CK is collected through first person observation. Direct observation may occur in person or via video recordings, and is typically conducted by researchers or teacher educators [12, 13] or by their collaborators—for example, in [15], where observations are carried by advanced CS students acting as coaches.

Written. In this case, information about teachers' CK is collected by asking them to produce some written materials. These may result from questionnaires which include open-ended questions, as in [28], or only multiple-choice questions, as in [8]; worksheets to complete, such as those proposed in [25] for a debugging activity; complex tasks to perform, such as preparing a lesson plan [4]; performing tasks that require writing or modifying a program [10]. This category thus includes a variety of tools (such as standardized tests, ad-hoc surveys, and worksheets), all of which result in written responses from teachers. While standardized tests provide validated measures, ad-hoc surveys and worksheets are often used to explore teachers' reasoning and pedagogical thinking. These tools, as components of KDTs, allow researchers to gather insights into teachers' CS knowledge in both structured (*e.g.*, standardized tests) and flexible (*e.g.*, open-ended worksheets) formats.

Learning analytics. This administration method involves the use of learning platforms that collect data on users' behavior, including logs and interactions [10].

Interview. In this case, information about teachers' CK is collected by involving them in an oral interview (*e.g.*, semi-structured, in-depth, etc.), which may be conducted in person, as in [13], or online. In the field of CS, an interview can serve as a strategic data collection method to enrich the understanding of what motivates teachers to act in certain ways when faced with a CS problem or a teaching and learning situation.

3.1.2 Stimulus. This second dimension addresses the question: "What different types of stimulus is presented to teachers to initiate the activity"? It refers to the nature of the input used to trigger the teacher's response in the context of the KDT, such as a task, question, or prompt. We identified three modalities, which are described below.

CS Artifact. In this case, the stimulus consists of one or more given artifacts with CS content, which teachers are asked to consider when responding to a proposed question or task. For example, in [17], the artifact is a technical CS text excerpt in which teachers are asked to highlight unfamiliar words while in [10] the artifact is a given program to analyze or debug.

Didactic. In this case, the stimulus consists of a teaching/learning situation accompanied by an implicit or explicit question or task. The situation may be real and ongoing–as in [13], where the teacher is in her classroom and conducts a self-designed lesson on inheritance and polymorphism–or hypothetical, as in [27], where teachers are asked to choose their reaction, from a list of options, to vignettes illustrating students' reasoning.

Only question/task. In this case, the stimulus presented to teachers consists solely of a direct question/task to answer/solve–such as in [15], where teachers are asked to write a definition of computational thinking, or in [9], where the stimuli are items from an exam in a general CS course.

3.1.3 Open/closed. This third dimension defines the type of tasks teachers are asked to engage with, addressing the question: "Do teachers have the opportunity to express themselves freely, either verbally or through actions?" It includes two modalities: *open*, if the answer is yes, and *closed*, otherwise. These terms should be interpreted broadly. For example, "open" does not refer only to open-ended questions, but also includes activities such as conducting a classroom lesson or designing a task.

3.1.4 Products/process. This fourth dimension specifies whether the KDT focuses on the final product of teachers' activity (*e.g.*, in [20], where researchers analyze the outcomes of learning activities - for instance, questions composed by teachers - during a PD program) or on the process itself (*e.g.*, in [12], where the analysis focuses on observing teachers during PD lectures and lab practice).

3.1.5 PCK. This final dimension examines whether KDTs can support the identification of PCK elements (such as in [4], where lesson plans are analyzed), or not (such as in [2], where teachers are asked to write a definition of algorithm, starting from written descriptions of candidate algorithms).

3.2 Synoptic table

The analysis of the identified KDTs was summarized into a synoptic table. Each KDT was accompanied by a brief description and organized according to the dimensions presented in the section above. An interactive version of the table is entirely available at https://cricca.disi.unitn.it/montresor/sigcse26/, where users can filter and sort its entries. Figure 1 exemplifies its structure.

Paper	Professional Development (PD)	KTD description	Data collection method	Stimulus	Type of task	Product or process	PCK
Reference	Whether the paper regards a PD	Sentence to describe the KTD	• Written deliverables • Direct observation • Interview • Learning analytics	• Only question/task • Specific artifact with CS content + question/task • Specific CS-related teaching/learning situation + question/task	• Open • Close	• Product • Process	• Yes • No
EXAMPLE							
[15]	X	Semi-structured classroom observations of teachers in their classes, by advanced CS students (coaches)	Direct observation	Specific CS-related teaching/learning situation + question/task	Open	Process	Yes

Figure 1: Structure of the synoptic table. Each column with a blue header represents a dimension. The modalities for each dimension are listed in the corresponding column.

Together with a short description of each KDT and a reference to the related paper, the table also includes the following:

- the data collection method (direct observation, interview, written deliverables, learning analytics);
- whether teachers are provided with a specific stimulus;
- whether the assigned tasks are open, closed, or both (if not specified, the task is considered open);
- whether the tool also investigates teachers' cognitive processes (if not specified, only teachers' products are considered);
- whether the tool allows to investigate PCK as well.

The majority of the identified KDTs (21 out of 24) rely on written deliverables; direct observations, interviews, and learning analytics are used rarely.

Half of the KDTs offer stimuli to trigger the teachers' answer. In 8 cases the offered stimuli are artifacts with CS content of various types–from excerpts of technical texts in which teachers are asked to identify unknown terms [17] to programs containing bugs to be analyzed [25]; in the remaining 3 cases, CS-related teaching and learning situations are used as stimuli, prompting teachers to write a comment or reaction. The remaining half KDTs use only questions or tasks without extra stimuli. Specifically, in 3 cases the questions are taken from student exams, in 2 cases they are part of standardized tests, and in the remaining 5 cases they are ad hoc tasks/questions created by the researchers. Among the latter, three KDTs require teachers to define or describe a concept–namely, computer science in [15], and computational thinking in [11, 28].

Most KDTs use only closed questions, whereas seven KDTs use open tasks that offer teachers the opportunity to express themselves more freely. Finally, only 4 KDTs analyze teachers' cognitive processes, the other rely only on products.

4 Discussion

In this section, we first outline the limitations of our search process; we then discuss our findings considering the role of KDTs' dimensions and modalities in the CS discipline in Section 4.2 and we present the implications for research and teacher education in Section 4.3.

4.1 Limitations

While SIGCSE-family venues represent a large and influential portion of the computer science education research community, they do not capture the full breadth of relevant scholarship—particularly studies published in general education, teacher education, or discipline-based education research outlets. Moreover, despite our efforts to define an inclusive search strategy, we likely missed papers in which potential KDTs were not highlighted in the abstract or were embedded in studies with different primary goals. Additionally, our search was constrained by the absence of standardized terminology in this area.

Considering the full set of papers (including also the excluded ones), we observed several cases in which the research, although outside the scope of our specific focus, made use of instruments that–while not primarily intended to assess teachers' CS knowledge–showed potential for collecting relevant data and could therefore qualify as KDTs. These limitations suggest the need for a second, broader cycle of literature analysis, which could refine the current inventory and expand its scope.

Extracting information related to KDTs from the selected papers was not always straightforward, since their primary focus is not typically to present KDTs; rather, KDTs are used instrumentally to address other research questions. As a result, we sometimes had to rely on information derived from comments or considerations involving the KDTs. For this reason, the features described in our synoptic table may not be entirely accurate in some cases.

4.2 KDTs in CS Education

Data collection methods. The task of writing definitions [11, 15] emerges as both significant and informative for at least two reasons: on the one hand, it provides information about teachers' CK, offering evidence consistent with findings from other studies, including those outside our sample–such as [7], where an analysis of the terms "coding" and "programming" sheds light on primary school teachers' conceptions and misconceptions. On the other hand, it can provide insights into teachers' "knowledge *about* CS", which, drawing a parallel with Ball and colleagues' reflections in the field of mathematics, is the other critical dimension to be considered together with CK when talking about subject knowledge for teaching [1].

In general, the use of written deliverables among KDTs resonates with the strong link between linguistic skills and performance in CS [19], making such kind of protocols an effective method for collecting information about disciplinary knowledge in this area.

Despite its under-representation in the analyzed sample, direct observation is a data collection method that can be particularly valuable in the CS field. For instance, in laboratory sessions, it enables the collection of information about the interaction between the teacher and the compiler (see [13] for an example).

Regarding learning analytics, in the CS domain, some basic form of it is almost always present, making it a particularly effective data collection approach for gathering information about disciplinary knowledge. Digital platforms are inherently part of CS education–at least in plugged activities–and these environments either generate, or can be easily adapted to generate, detailed logs.

Stimulus. In general, CS stimuli can take many forms–not only due to the diversity of objects of study in computer science (e.g., programming languages and paradigms, software systems, algorithms, and data structures), but also because of its distinctive processes and phases, such as debugging and testing. We argue that the proposed distinction–artifact with CS content, CS-related didactic situation, only task or question–can support the identification and analysis of teachers' CK, while acknowledging its inherent complexity and helping make that knowledge more accessible for research and reflection.

Open/closed. This distinction helps to specify tasks that can also reveal the creative dimension of teachers' disciplinary knowledge. This is an aspect that plays a crucial role in computer science, not only in problem solving, but also in the design, optimization and implementation of computational solutions. Programming itself is not merely a technical activity; it often requires creativity in structuring code, designing algorithms, and adapting solutions to various constraints. Recognizing this creative component is especially relevant when analyzing how teachers approach open-ended tasks, as it provides deeper insight into how they engage with core disciplinary practices and express creativity within the field.

Product/process. The distinction between product and process reflects two fundamental aspects of computer science, which–when addressing a problem–can be understood both as a science of products (concerned with the final, effective solution) and as a science of processes (concerned with the strategies, methods, and decisions that lead to the solution). Therefore, in characterizing KDTs, considering whether they emphasize products or processes can offer insight into the depth and nature of the knowledge the tool is able to reveal.

PCK. Tracing the possibility of identifying elements of PCK can offer valuable insights for future research in CS education. Given that CS education is still a relatively young research field, having an inventory of KDTs that also support the identification or investigation of teachers' PCK can be especially beneficial.

4.3 Implications

Drawing on our results, we observe that, when it comes to studying teachers' CS knowledge, researchers predominantly rely on static KDTs–namely, written deliverables. This suggests an underlying view of knowledge as a fixed body to be measured, rather than as something dynamic that can be observed through different facets. However, a small portion of the recorded KDTs does attend to this dynamic aspect, focusing on the processes through which teachers' CS knowledge is activated and expressed. In such cases, where researchers relied primarily on direct observation [12, 13, 15], it becomes possible to collect data that are particularly valuable to understand how knowledge manifests in action. Moreover, this approach allows researchers to capture instances in which knowledge is dynamically put into use–a key feature when moving toward the analysis of teachers' PCK in CS.

Among the KDTs analyzed in our study, an interesting case is presented in [10], where the authors collect snapshots related to the problem-solving process. In this case, while the main focus of the analysis is the final output of the interaction with the task, the data also provide insight into the process that led to that output. This temporal information supports the deconstruction of possible thinking patterns or mistakes, offering a glimpse into the cognitive path followed by the teacher.

Overall, beyond its descriptive aspects, our work provides a structure for an inventory of methods that may be useful for both teachers and researchers. In particular, we see several possible implications. On the one hand, for teacher educators, this framework offers a perspective on tools to unveil teachers' CK and that can support the design of targeted, reasoned, and fine-tuned professional development initiatives and didactic resources. On the other hand, the analysis conducted on the identified KDTs introduces a structure of dimensions and modalities that may assist researchers in framing the methodology of their studies or, through a bottom-up approach, in constructing data collection instruments and designing situations that can generate more focused and meaningful data.

In general, the process of exploration we conducted on existing literature on teachers' CS knowledge depicts a very heterogeneous set of papers that appears scattered among different focuses of research, thus not identifying a stable and well-established strand of research. In this sense, it appears that the study of knowledge, at least within CS education, mostly corresponds to the measurement of knowledge–either for assessing the effectiveness of an intervention (mostly PD course) or to assess some specific knowledge. Nevertheless, in spite of the use of KDTs that emerges from our analysis, the kind of data that is gathered appears to be qualitatively rich, offering potential for a further, more in-depth, analysis and opening up to more structured investigations on teachers' CS knowledge.

5 Conclusions and future work

This study explored the detection and assessment of teachers' CS knowledge by identifying and analyzing the methodological tools currently used in the literature. We reviewed relevant publications and selected a set of papers for deeper investigation. Our focus on the tools employed in these studies allowed us to compile an inventory of what we define as Knowledge Detection Tools, thereby addressing our first research question (RQ1).

We then examined these tools through a structured framework composed of key dimensions and modalities—such as data collection methods, types of stimuli, openness of tasks, and attention to cognitive processes or PCK. This analysis enabled us to respond to RQ2 by characterizing the nature of the KDTs and the kinds of data they elicit.

While many studies rely on written deliverables, we observed considerable variation in the types of stimuli used and, in some cases, an interesting focus on the dynamic processes through which knowledge is enacted, not merely evaluated.

We believe our contribution can serve multiple purposes. For researchers, the proposed framework can guide methodological decisions and support the development of new tools or research designs. For teacher trainers, the inventory of KDTs may be used to inform the design of PD initiatives and instruments of assessment tailored to educators' needs. More broadly, by introducing and structuring the notion of KDTs, our work contributes to initiating a conversation around the assessment of CS knowledge in teaching—an area that remains largely underexplored and offers substantial opportunities for further inquiry.

Future work will focus on extending the literature review and collecting other KDTs. Additionally, we aim to investigate more in depth how insights from KDTs can inform both the design of PD programs and the development of more robust models of teacher knowledge in computer science.

Acknowledgments

We acknowledge financial support under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.1, Call for tender No. 104 published on 2.2.2022 by the Italian Ministry of University and Research (MUR), funded by the European Union – NextGenerationEU– Project Title "Learning Informatics" – CUP E53D23007720006- Grant Assignment Decree No. 959 adopted on 22/04/2022 by MUR.

References

[1] Deborah L. Ball. 1990. The Mathematical Understandings That Prospective Teachers Bring to Teacher Education. *The Elementary School Journal* 90, 4 (1990), 449–466. doi:10.1086/461626

[2] Carlo Bellettini, Violetta Lonati, Mattia Monga, and Anna Morpurgo. 2024. To Be Or Not To Be . . . An Algorithm: The Notion According to Students and Teachers. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland (OR)) *(SIGCSE '24)*. ACM, New York, NY, USA, 102–108. doi:10.1145/3626252.3630950

[3] Stefania Bocconi, Augusto Chioccariello, Giuliana Dettori, Anusca Ferrari, Katja Engelhardt, Panagiotis Kampylis, and Yves Punie. 2022. *The Integration of Computational Thinking and Coding in Compulsory Education: A Review of National Curricula and Initiatives in Europe.* JRC Science for Policy Report. European Commission, Joint Research Centre. doi:10.2760/52317

[4] Heather Bort and Dennis Brylow. 2013. CS4Impact: measuring computational thinking concepts present in CS4HS participant lesson plans. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (Denver, Colorado, USA) *(SIGCSE '13)*. ACM, New York, NY, USA, 427–432. doi:10.1145/2445196.2445323

[5] Gavin Brown, Heidi Andrade, and Fei Chen. 2015. Accuracy in student self-assessment: Directions and cautions for research. *Assessment in Education Principles Policy and Practice* 22 (01 2015). doi:10.1080/0969594X.2014.996523

[6] Travis J. Carter and David Dunning. 2008. Faulty self-assessment: Why evaluating one's own competence is an intrinsically difficult task. *Social and Personality Psychology Compass* 2, 1 (2008), 346–360.

[7] Isabella Corradini, Michael Lodi, and Enrico Nardelli. 2018. Coding and Programming: What Do Italian Primary School Teachers Think?. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. ACM, New York, NY, USA, 1074. doi:10.1145/3159450.3162268

[8] Karla Hamlen, Nigamanth Sridhar, Lisa Bievenue, Debbie K. Jackson, and Anil Lalwani. 2018. Effects of Teacher Training in a Computer Science Principles Curriculum on Teacher and Student Skills, Confidence, and Beliefs. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. ACM, New York, NY, USA, 741–746. doi:10.1145/3159450.3159496

[9] Karla Hamlen Mansour, Debbie K. Jackson, Lisa Bievenue, Adam Voight, and Nigamanth Sridhar. 2023. Understanding the Impact of Peer Instruction in CS Principles Teacher Professional Development. *ACM Trans. Comput. Educ.* 23, 2, Article 24 (apr 2023), 21 pages. doi:10.1145/3585077

[10] Viraj Kumar and Amey Karkare. 2021. Instructor Performance on Progressively Complex Programming Tasks: A Multi-Institutional Study from India. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (Virtual Event, Germany) *(ITiCSE '21)*. ACM, New York, NY, USA, 561–567. doi:10.1145/3430665.3456384

[11] Anna Lamprou and Alexander Repenning. 2018. Teaching how to teach computational thinking. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) *(ITiCSE 2018)*. ACM, New York, NY, USA, 69–74. doi:10.1145/3197091.3197120

[12] Neomi Liberman, Catriel Beeri, and Yifat Ben-David Kolikant. 2011. Difficulties in Learning Inheritance and Polymorphism. *ACM Trans. Comput. Educ.* 11, 1, Article 4 (Feb. 2011), 23 pages. doi:10.1145/1921607.1921611

[13] Neomi Liberman, Yifat Ben-David Kolikant, and Catriel Beeri. 2009. In-service teachers learning of a new paradigm: a case study. In *Proceedings of the Fifth International Workshop on Computing Education Research Workshop* (Berkeley, CA, USA) *(ICER '09)*. ACM, New York, NY, USA, 43–50. doi:10.1145/1584322.1584329

[14] Michael Lodi. 2020. *Introducing Computational Thinking in K-12 Education: Historical, Epistemological, Pedagogical, Cognitive, and Affective Aspects.* Ph.D. Dissertation. University of Bologna, Italy. https://tel.archives-ouvertes.fr/tel-02981951

[15] María Cecilia Martinez, Marcos J. Gomez, Marco Moresi, and Luciana Benotti. 2016. Lessons Learned on Computer Science Teachers Professional Development. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (Arequipa, Peru) *(ITiCSE '16)*. ACM, New York, NY, USA, 77–82. doi:10.1145/2899415.2899460

[16] Monica M. McGill, Joseph C. Tise, and Adrienne Decker. 2024. Piloting a Diagnostic Tool to Measure AP CS Principles Teachers' Knowledge Against CSTA Teacher Standard 1. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024)*. ACM, Portland, OR, 819–825. doi:10.1145/3626252.3630905

[17] Bhagya Munasinghe, Tim Bell, and Anthony Robins. 2021. Teachers' understanding of technical terms in a Computational Thinking curriculum. In *Proceedings of the 23rd Australasian Computing Education Conference* (Virtual) *(ACE '21)*. ACM, New York, NY, USA, 106–114. doi:10.1145/3441636.3442311

[18] Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjørn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Susan McDonald, and David Moher. 2021. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *International Journal of Surgery* 88 (2021), 105906. doi:10.1016/j.ijsu.2021.105906

[19] Chantel S. Prat, Tara M. Madhyastha, Malayka J. Mottarella, and Chu-Hsuan Kuo. 2020. Relating natural language aptitude to individual differences in learning programming languages. *Scientific reports* 10, 1 (2020), 3817.

[20] Noa Ragonis. 2012. Integrating the teaching of algorithmic patterns into computer science teacher preparation programs. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*. ACM, Haifa, Israel, 339–344.

[21] Royal Society. 2017. After the Reboot: Computing Education in UK Schools. https://royalsociety.org/-/media/policy/projects/computing-education/computing-education-report.pdf Accessed: 2024-05-07.

[22] Demetrios Sampson, Panagiotis Kampylis, Jesús Moreno-León, and Stefania Bocconi. 2025. Towards high-quality informatics K-12 education in Europe: key insights from the literature. *Smart Learning Environments* 12, 1 (2025), 14. doi:10.1186/s40561-025-00366-5

[23] Thomas J. Shuell. 1992. *Adaptive Learning Environments*. Springer, Berlin, Heidelberg, 19–54. doi:10.1007/978-3-642-77512-3

[24] Lee S. Shulman. 1986. Those who understand: Knowledge growth in teaching. *Educational Researcher* 15, 2 (1986), 4–31. doi:10.3102/0013189X015002004

[25] Jennifer Tsan, David Weintrop, and Diana Franklin. 2022. An Analysis of Middle Grade Teachers' Debugging Pedagogical Content Knowledge. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1* (Dublin, Ireland) *(ITiCSE '22)*. ACM, New York, NY, USA, 533–539. doi:10.1145/3502718.3524770

[26] James H Wykowski and Helene Starks. 2024. What Type of Self-Assessment Is Best for Your Educational Activity? A Review of Pre-Post, Now-Then, and Post-Only Designs. *Journal of General Internal Medicine* 39, 11 (2024), 2045–2050.

[27] Aman Yadav and Marc Berges. 2019. Computer Science Pedagogical Content Knowledge: Characterizing Teacher Performance. *ACM Trans. Comput. Educ.* 19, 3, Article 29 (may 2019), 24 pages. doi:10.1145/3303770

[28] Aman Yadav, Chris Mayfield, Ninger Zhou, Susanne Hambrusch, and John T. Korb. 2014. Computational Thinking in Elementary and Secondary Teacher Education. *ACM Trans. Comput. Educ.* 14, 1, Article 5 (March 2014), 16 pages. doi:10.1145/2576872