Revista ARACÊ
ISSN: 2358-2472

TRAPS OF COMPUTATIONAL THINKING: CRITICAL ANALYSIS

https://doi.org/10.56238/arev7n1-121

Submission date: 12/13/2024 **Publication date: 01/13/2025**

Marcelo Magalhães Foohs[1], Antinesca Joana Pissolatto Neta[2], Júlia Andressa Paes dos Santos Krüger[3] and Randerson Oliveira Melville Rebouças[4]

ABSTRACT

This article critically analyzes the evolution of the concept of computational thinking (CT) in the works of Jeanette Wing, focusing on articles published between 2006 and 2017. It is argued that, although Wing has widely promoted CT as an essential skill for several areas of knowledge, her definition presents conceptual ambiguities that compromise its delimitation and applicability. One of the main problems identified is the use of the logical operator "OR" in the initial definition of CT, which allows for overly broad interpretations and the inclusion of exclusively human processes that cannot be executed by machines. The review of later texts shows that this lack of definition was not corrected, resulting in difficulties in consolidating CT as a formally structured field. In addition, the centrality of algorithms, an essential element for the identity of CT, is discussed, and how their explicit absence in Wing's formulation contributes to the dilution of the concept. The educational implications of the lack of a precise definition are also examined, which directly impact the teaching and curricular implementation of CT. Finally, it is argued that, in order to ensure the coherence and effectiveness of computational thinking, greater conceptual rigor is needed, clearly distinguishing its scope and its relationship with other forms of cognition. The article proposes a more precise delimitation of CT, emphasizing its algorithmic basis and differentiating it from probabilistic approaches and models based on approximate optimization, in order to strengthen its identity and applicability in both education and scientific research.

Keywords: Computational Thinking. Jeanette Wing. Algorithms. Computer Science in Education. Critical Analysis.

[1] Doctor in Informatics in Education, Federal University of Rio Grande do Sul
[2] Master in Architecture and Urbanism, Federal University of Rio Grande do Sul
[3] Master in Agricultural and Environmental Microbiology, Federal University of Rio Grande do Sul
[4] Master in Health Informatics, Federal University of Rio Grande do Sul.

Revista ARACÊ
ISSN: 2358-2472

INTRODUCTION

Since the publication of Jeanette Wing's article "Computational Thinking" in 2006, the concept of computational thinking (CT) has gained increasing prominence in discussions about education and computer science. Wing (2006) argues that computational thinking is not limited to programming, but is a form of problem-solving that should be taught to everyone from an early age. Her work has had a profound impact on the way CT is understood and taught around the world, influencing both curriculum design and pedagogical practices. However, some rhetorical and conceptual choices made by the author, especially the use of the logical connector "OR" in her definition, introduce ambiguities that generate problematic understandings.

By stating that "Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine" (Wing, 2006), the author imprecisely expands the scope of the concept. The use of the "OR" connector as an inclusive operator implies that any so-called computational process, whether it can be executed only by humans, only by machines, or both, can be classified as computational thinking. This leads to a false symmetry between human and computational processes, ignoring the specificity of algorithms and systematizable strategies that are central to computational thinking. Although the author continued to explore computational thinking in the following years, at no time was there an explicit review of this definition that rigorously delimited the scope of the concept. Thus, the conceptual vagueness persists and compromises the clarity necessary for its educational and scientific application, allowing broad interpretations that can dilute the specificity of computational thinking. This imprecision, therefore, is persistent in Wing's formulation and has direct implications for educational practices. A practical example can be observed in educational activities that address trivial everyday tasks as if they were sufficient for students to develop skills associated with computational thinking. In these activities, exclusively human processes, such as streams of consciousness or intuitive problem-solving, end up being interpreted as part of the scope of computational thinking. However, such processes often lack the clarity and systematicity expected in the context of computational thinking, deviating from their algorithmic essence.

Although subjective human problem-solving strategies, such as social influences or meditation practices, can be analyzed or described approximately through natural language, their contextual complexity and dependence on subjective nuances remove

Revista ARACÊ
ISSN: 2358-2472

them from the scope of computational thinking. This scope is reserved for clear, systematic, and potentially executable processes by machines. Human strategies based on subjectivity cannot be fully translated into algorithms due to the lack of formal logic and systematicity, characteristics intrinsic to computational thinking. Thus, these strategies remain outside the domain of processes that can be precisely represented and operationalized.

Conceptual expansion compromises the formation of specific skills that are central to computational thinking, such as abstraction and creativity aimed at building algorithms. Although abstraction and creativity are general human capacities, present in a wide variety of contexts, in computational thinking these capacities need to be applied in a targeted manner to formulate clear and executable sequences. Diluting the concept by including general human processes weakens this specificity, making the teaching and practice of computational thinking less effective. The difficulties reported about new entrants to higher education, such as the lack of abstraction and creativity capacity in this context (cf. Oliveira and Pereira, 2019), reinforce the importance of specific training that aligns these capacities with the demands of computational thinking, with a focus on the creation of systematic and algorithmic solutions.

To differentiate traditional algorithms from statistical models used in artificial intelligence, the Center for Innovation in Brazilian Education (CIEB) proposes a definition of structured algorithms, highlighting their clarity and predictability:

Mathematics provides a formal and universal language. It is through it that Computing builds models called algorithms. Although they can be described at different levels of abstraction, with specific languages that focus only on the essential and relevant elements, algorithms are finite sequences of steps that lead to a certain result. There may be complications, such as conditional sequences and repetition loops, but all the steps are defined and can be understood by a machine (gr(ifo nosso) (CIEB, Technical Note #21, p. 9, 2024).

The Technical Note then differentiates structured algorithms from statistical models used in artificial intelligence, which operate based on machine learning and dynamic adaptation to new data. This distinction is fundamental, since computational thinking, as discussed in this article, is based on the construction of structured and deterministic algorithms. The probabilistic models used in AI, on the other hand, follow a different logic, dynamically adjusting themselves based on patterns extracted from data.

Revista ARACÊ
ISSN: 2358-2472

Although some documents and proposals use the term "algorithm" to refer to these models, this nomenclature broadens the scope of the term without adequate delimitation and can generate conceptual confusion, since statistical models do not follow the deterministic, explicit and systematic structure of traditional algorithms. In the context of artificial intelligence, traditional algorithms and statistical models coexist and complement each other, but play distinct roles within computational systems.

Attempting to frame distinct realities under the same algorithmic terminology, even with different adjectives, can compromise the conceptual precision of computational thinking and hinder its educational application. This distinction, however, does not reduce the importance of statistical models, but rather strengthens the conceptual clarity of computational thinking, allowing both fields to develop in a complementary manner without diluting their identities.

It is important to recognize that statistical models and machine learning go beyond the traditional scope of computational thinking, as they operate under distinct principles, based on probabilistic inference and dynamic adjustment from data. This does not mean that these methodologies are incompatible, but rather that they should not be indistinctly absorbed within computational thinking without a rigorous operational definition. Maintaining this conceptual distinction strengthens both the teaching and the applicability of computational thinking, ensuring that it preserves its identity based on algorithmic logic and the systematic construction of computable solutions.

Indiscriminate simplifications or extensions distort the very nature of the PC and can compromise students' education, resulting in deficiencies such as difficulties in solving computational problems in a structured way. This highlights the need for a critical reassessment of Wing's ideas in the educational context. Throughout this article, we argue that the construction of algorithms is the distinctive factor of computational thinking. Although concepts such as abstraction, decomposition, and pattern recognition are essential, it is their application in the process of planning algorithms that establishes computational thinking as something unique. This centrality of the algorithm reflects the essence of computational thinking as a bridge between human cognition and automatable processes.

Given the discussions on the conceptual delimitation of computational thinking, we propose the term Probabilistic-Algorithmic Computational Thinking (PCAP) to distinguish its traditional conception, based on structured and deterministic algorithms, from that which

Revista ARACÊ
ISSN: 2358-2472

integrates probabilistic methodologies and machine learning. This distinction avoids the indiscriminate fusion of concepts and enables a more precise analysis of their applicability in teaching and scientific research. PCAP maintains the algorithmic identity of traditional computational thinking while interacting with statistical models without confusing them with algorithmic logic. This differentiation is not merely terminological, but central to preserving the specificity of computational thinking and ensuring its educational and scientific application with precision and coherence.

Although several authors have contributed to the debate on computational thinking, this article focuses exclusively on Wing's work, due to his influence, analyzing six representative texts (2006, 2008, 2011, 2014, 2016, 2017). The objective is to critically reflect on the conceptual bases of traditional computational thinking, hereinafter simply computational thinking, highlighting central points that can contribute to its consolidation and aligning them with contemporary demands of education and effective pedagogical practices.

Although computational thinking is intrinsically linked to education, this article aims exclusively to clarify fundamental concepts that define the nature of computational thinking. Issues related to learning theories and pedagogical implementation models will be addressed in future works. This delimitation allows for a critical deepening necessary to resolve ambiguities that directly impact the way computational thinking is understood and, consequently, practiced, serving as a basis for more detailed pedagogical discussions in subsequent publications.

THEORETICAL FOUNDATION

In this section, we will explore the theoretical foundations of computational thinking, with an emphasis on the concepts of abstraction, decomposition, and pattern recognition. Although these elements are essential, we argue that it is in the application of these strategies to the construction of algorithms that computational thinking finds its identity.

Jeanette Wing (2006) highlights the importance of these concepts as pillars of computational thinking:

Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system [...] Computational thinking is the ability to think at multiple levels of abstraction [...] An algorithm is an abstraction of a step-by-step procedure for taking input and producing some desired output (WING, 2006).

Revista ARACÊ
ISSN: 2358-2472

Wing's view of abstraction resonates with Piaget's (1977) theory, especially about the progression from empirical to reflective abstraction, applied to the context of algorithm construction. However, although abstraction, decomposition, pattern recognition, and, implicitly, creativity are central elements highlighted by Wing, it is the application of these elements in the construction of algorithms that identifies computational thinking. As the author highlights, computational thinking should not be confused with programming, which presupposes the mastery and use of a programming language with its syntax and resources. On the contrary, computational thinking refers to the strategies used to organize the algorithm, which can subsequently be translated into any programming language. Thus, the process of planning and structuring algorithms, sequences of unambiguous steps with the potential to be translated and executed by a machine, is the element that differentiates computational thinking from other forms of human thought.

PIAGET'S CONTRIBUTIONS TO THE CONCEPT OF ABSTRACTION

According to Piaget (1977), abstraction is one of the central processes in cognitive development, allowing the construction of more complex and generalized knowledge from concrete experiences:

Abstraction is an intellectual operation that consists of isolating certain elements of reality, ignoring the others, to concentrate on particular characteristics that are relevant to the objective of analysis." (PIAGET, 1977, p. 17).

Jean Piaget (1977) describes two main types of abstraction: empirical abstraction and reflective abstraction. These types of abstraction play different roles in cognitive development and are fundamental to understanding how individuals construct knowledge throughout their lives.

Empirical abstraction refers to the process of extracting characteristics of objects based on their observable physical properties. This type of abstraction is directly linked to sensory and motor interactions with the outside world: "Empirical abstraction consists of extracting properties of material objects from their directly perceptible characteristics, such as shape, color and texture" (PIAGET, 1977, p. 25).

Reflective abstraction refers to the process of abstracting characteristics that are not directly linked to sensory perception, but that emerge from the coordination between different mental actions. This type of abstraction is more sophisticated and is linked to the capacity for reflection and generalization:

Revista ARACÊ
ISSN: 2358-2472

Reflective abstraction is a form of abstraction that does not refer to the material characteristics of objects, but to the mental operations that the subject can perform on these objects, such as grouping, ordering, and classification operations" (PIAGET, 1977, p. 32).

These concepts have profound implications for education, especially in the apprehension of complex concepts involved in strategies related to computational thinking. The differentiation of the types of abstraction provides a basis for understanding how students can progress from a concrete understanding to a more abstract and generalized understanding, essential for the construction of algorithms that organize solutions in clear and executable steps.

THE ARISTOTELIAN PERSPECTIVE ON ABSTRACTION

In addition to Piaget's (1977, 1950) psychological approach, Aristotle's philosophy offers another relevant perspective on abstraction, highlighting its basis in universal human cognition: The soul never thinks without a mental image... hence an object of thought can't exist without the senses, and without an image, we do not learn, or understand anything. (Aristotle, De Anima, Book III, Chapter 7)

The mental image to which Aristotle refers is an internal representation derived, initially, from the perception of the external senses in contact with reality. While Piaget (1977) describes abstraction in terms of cognitive development, Aristotle offers a more complementary, positioning abstraction as a universal ability of the human mind. In other words, abstraction for the philosopher is an intrinsic ability of human nature, that is, a transversal characteristic of human cognition, and not something exclusive to a type of thinking.

This perspective reinforces that, in computational thinking, abstraction is deeply rooted in universal human abilities, but gains specificity when applied to the formulation of algorithms that organize knowledge for problem-solving.

INFORMATION AND KNOWLEDGE

This reflection on abstraction leads us to explore how external information can be transformed into systematized knowledge. Although abstraction is common to all forms of human cognition, it acquires a unique specificity in computational thinking in the construction of the algorithm. In this context, the captured and organized elements are

Revista ARACÊ
ISSN: 2358-2472

directed to the creation of algorithms, which structure solutions in a logical and executable manner. This approach distinguishes computational thinking by translating scattered information into organized systems, allowing the solution of complex problems through algorithms.

The process of transforming information into knowledge is an intrinsically human phenomenon, based on the dynamic interaction between external and internal senses and external reality. As suggested by Locke (1996), the external senses are the gateways to perceptions of the world, providing the initial basis for thought. These senses capture specific information from the environment, which is subsequently organized and processed by the internal senses. Piaget (1977) highlights that the continuous interaction between cognitive schemes and the external environment enables the progressive adaptation of thought, favoring the development of more sophisticated cognitive structures. In this context, the external senses play the role of primary channels of perception, while the internal senses reorganize these perceptions into structured knowledge, allowing advances in human cognition.

EXTERNAL SENSES: GATEWAYS TO REALITY

The external senses - sight, hearing, touch, smell, and taste - are the primary channels for capturing information from the environment. Each sense has a specific and limited role in the process of perception. As Aristotle (350 BC) pointed out in De Anima, "each sense perceives a specific property of the sensible object", connecting the external world to the mind through specialized channels. Locke (1996) reinforces this perspective by arguing that the external senses are the primary source of ideas in the mind:

• Vision: Captures shapes, colors, movements, and depth, allowing spatial perception and recognition of visual patterns. Marr (1982) proposes a computational model of vision, demonstrating how the visual system transforms sensory stimuli into structured representations, essential for understanding space and objects.

• Hearing: Captures sounds, rhythms, and tonal variations, being fundamental for communication and the interpretation of auditory patterns.

• Touch: Perceives textures, temperatures, and pressures, connecting the bodily experience to the physical environment.

• Smell and Taste: Sensitive to chemical particles, they play an important role in the identification of flavors and smells. Gibson (1979) argues that perception occurs directly,

Revista ARACÊ
ISSN: 2358-2472

without the need for complex cognitive inferences, and that the senses, in general, capture affluence, environmental information that guides practical interaction with the environment.

These external senses act as "gateways" for information from the environment and are incapable of storing or processing information by themselves. This initial perception is later transformed by the internal senses, which organize and systematize the information captured.

INTERNAL SENSES: TRANSFORMATION OF INFORMATION

Although the division of the internal senses into distinct categories is a theoretical construction to understand the cognitive process, which is a dynamic and recursive interaction between the human senses, it is important to recognize the limitations inherent in this approach. Ellis (1980), in his contribution to psycholinguistics, criticizes the idea of the black box as an absolute model to describe the internal processes of cognition. He argues that although the internal mechanisms of the mind are complex and largely inaccessible directly, it is possible to investigate and represent them through theoretical models that are based on empirical data. This view supports the use of categories such as memory, intelligence, imagination, will, and cognition, not as separate entities, but as functional representations of an integrated system. Such categories are analytical tools that allow us to capture nuances of the cognitive process, even though in reality these elements act inseparably and dynamically. Therefore, the categorization of internal senses in this article does not aim to describe a fragmented reality, but rather to offer an interpretative model that facilitates the understanding of the roles attributed to different dimensions of the cognitive process. This positioning is consistent with Ellis's (1980) critique of the black box, as it recognizes that, although the divisions are artificial, they provide valuable insights into the functioning of human thought.

Memory operates at different levels: Baddeley and Hitch (1974) propose a model in which working memory is not a unitary system, but composed of multiple components, including the central executive, which coordinates attention and integrates information from the phonological loop (responsible for retaining verbal information) and the visuospatial sketchpad (which processes mental images). Tulving (1972) differentiates long-term memory into episodic memory, which stores specific events, and semantic memory, which contains factual and conceptual knowledge.

Revista ARACÊ
ISSN: 2358-2472

Sweller's (1994) cognitive load theory argues that working memory has limited capacity and can only process a limited amount of information simultaneously. When the extrinsic (unnecessary elements) or intrinsic (complexity of the material) cognitive load exceeds this capacity, learning is compromised. In contrast, long-term memory stores organized schemas, allowing for reduced cognitive load and making information retrieval more efficient.

In the context of computational thinking, formulating algorithms requires the simultaneous manipulation of multiple pieces of information, such as variables, conditional structures, and logical patterns. This process overloads working memory if it is not well structured, making reasoning and problem-solving difficult. To mitigate this overload, strategies such as problem decomposition, use of pseudocode, and visual representation of algorithms help distribute the cognitive load, allowing learners to internalize computational concepts more efficiently. In this way, by reducing the demand on working memory, learning computational thinking becomes more accessible and structured, facilitating the retention and application of algorithmic concepts.

According to Piaget (1950), intelligence manifests itself through the processes of assimilation and accommodation, which allow the individual to adapt to the environment and build cognitive schemes to understand and interact with different contexts. Assimilation occurs when the individual incorporates new information into existing cognitive structures, using their previous schemes to interpret the world. For example, a child who has learned the concept of a dog as a four-legged animal may initially classify a cat as a dog. This categorization demonstrates the process of assimilation, in which the child tries to accommodate a new experience within familiar schemes.

On the other hand, when new information cannot be accommodated within existing schemes, the process of accommodation occurs. This process requires the individual to modify or reorganize their cognitive schemes to deal with the novelty. In the previous example, as the child realizes that the cat does not share all the characteristics of a dog, such as barking or being a certain species, he or she creates a new schema for the cat, adjusting his or her understanding of the world. Thus, intelligence is not limited to accumulating information but manifests itself in the ability to reorganize and transform schemas to better adapt to the complexities of the environment. In the context of computational thinking, these processes of assimilation and accommodation play a central role. For example, when a student first approaches the concept of algorithms, he or she

Revista ARACÊ
ISSN: 2358-2472

may try to assimilate this concept within existing schemas, such as step-by-step instructions for performing an everyday task. However, when he or she realizes the need for precision and formal logic in algorithms, he or she must accommodate this knowledge, adjusting his or her initial understanding to include elements specific to computational thinking, such as the need for clarity and lack of ambiguity. Furthermore, the balance between assimilation and accommodation is directly reflected in how students develop the capacity for abstraction and decomposition in computational thinking. Initially, they tend to approach complex problems based on already known schemes (assimilation); however, when faced with obstacles, they need to reorganize their strategies, adjusting their mental models to incorporate new ways of solving the problem (accommodation).

Will is the engine that drives motivation, guiding the intellectual effort needed to transform information into systematized knowledge. According to James (1890), will play an essential role in sustaining attention and perseverance in the face of cognitive challenges. Additionally, the Self-Determination Theory (Deci & Ryan, 1985) offers a contemporary perspective on the factors that drive motivation, highlighting that autonomy, perceived competence, and a sense of belonging play a fundamental role in the quality of engagement and learning. These authors argue that when individuals perceive that they have control over their actions (autonomy), feel effective when facing challenging tasks (competence), and experience meaningful connections with others (relatedness), their intrinsic motivation is strengthened. This strengthening of intrinsic motivation sustains the continuous effort required to build systematized knowledge and is therefore essential for learning and the creation of algorithms in the context of computational thinking.

Imagination combines existing elements in new ways, enabling innovation and creative problem-solving. Aristotle (350 BC) positions imagination as a fundamental capacity for thought, linking it to the generation of internal images that make reasoning possible. In the modern context, Csikszentmihalyi (1990) argues that creativity emerges in states of flux, in which imagination plays a central role in enabling innovative associations between distinct elements, resulting in new ideas and solutions. Sawyer (2012) discusses the importance of imagination in collaborative creativity, highlighting how social interactions can enhance the generation of innovative ideas. He argues that creativity does not occur in isolation, but is influenced by cultural practices and contextual stimuli. Kaufman and Gregoire (2015), in turn, emphasize the relevance of imagination in everyday creativity, suggesting that it is accessible to everyone and can be cultivated through openness to new experiences and the exploration of intellectual challenges. In the context of computational thinking, imagination acts as a catalyst for the creation of innovative algorithms, enabling the modeling of solutions to complex problems. Root-Bernstein (2013) highlights the importance of scientific imagination in the formulation of hypotheses and experiments, by establishing bridges between theoretical abstractions and practical applications. From this perspective, it can be inferred that the same ability to transcend the obvious and generate new ideas is fundamental to computational thinking and creative computing, which combines logic and innovation in problem-solving.

Creative computing is an approach that explores how computational tools can be used to solve problems and for creative expression. This perspective transcends the conventional application of algorithms to focus on the creation of interactive digital artifacts, such as stories, games, animations and music. Creative computing is directly linked to the development of computational thinking, as it encourages abstraction, experimentation and iteration as part of a creative process.

One of the most significant advances in creative computing was led by Mitchel Resnick and his team at the Lifelong Kindergarten Group at the MIT Media Lab. In 2007, Resnick and his collaborators launched Scratch, a visual programming platform designed to enable children and young people to create and share digital projects. Scratch was developed with the goal of making programming accessible and intuitive, using blocks of code that can be dragged and dropped, eliminating the need to memorize complex syntax (cf. Resnick et al., 2009). In addition to Scratch, creative computing has been widely explored by researchers such as John Maeda (2004), who emphasizes the fusion of art, science, and technology as a means to expand expression and innovation in design and digital media. Similarly, Reas and Fry (2001) developed the Processing programming language, aimed at designers and visual artists, consolidating it as an essential tool at the intersection of computing and visual arts. These tools and philosophies established creative computing as an interdisciplinary area that connects computer science, design, and the arts. In the context of computational thinking, creative computing contributes to the construction of efficient algorithms and computational models and to the stimulation of innovation and creativity. This relationship highlights that computational thinking goes beyond being a technical skill, and also presents itself as an intellectual approach capable of enriching various areas of knowledge and human expression.

Revista ARACÊ
ISSN: 2358-2472

Finally, the cognitive aspect is fundamental for the processing and articulation of thoughts, especially through language, which Vygotsky (1986) describes as the central mediator between thought and communication. For him, the development of language organizes ideas and structures abstract thought, enabling the creation of coherent narratives and complex arguments. Contemporary studies, such as those by Tomasello (2014), reinforce the view that human cognition is closely linked to the social use of language, being fundamental for the coordination of collective efforts and the sharing of knowledge.

In addition, Pinker (2007) highlights that language is one of the most advanced evolutionary innovations of humanity, allowing the cultural transmission of ideas, technologies and values. From this perspective, it can be inferred that language acts as an essential cognitive tool, assisting in the formulation and structuring of abstract thought. In the context of computational thinking, this function manifests itself in the conversion of abstract concepts into precise linguistic descriptions, such as pseudocodes and algorithm specifications, which can be shared and interpreted by both humans and machines.

Ferrari and Sternberg (1998) emphasize that the processes of self-reflection and metacognition are fundamental in problem formulation and creative resolution, promoting a continuous interaction between analysis and synthesis. This interaction is essential in computational thinking, as it allows the refinement of ideas and the transformation of complex reasoning into structured and communicable solutions.

Therefore, the cognitive is not limited to the internal organization of thought, but also acts as the bridge that connects cognitive processes to clear and effective expression, both in interpersonal communication and in the creation of computational systems. Its importance transcends the linguistic domain, being a component of innovation and problem-solving in the contemporary world, especially in a context where computational thinking requires structural clarity and conceptual precision.

Thus, the internal senses operate in an integrated manner, transforming the information captured by the external senses into structured knowledge. Each one plays an indispensable role in the cognitive process, contributing in an interactive and adaptive way to the construction of knowledge. Below, we briefly present the roles attributed to each internal sense as a way of better understanding their contribution to this process, recognizing that this division is an abstraction to facilitate analysis:

Memory (working and long-term):

Revista ARACÊ
ISSN: 2358-2472

• Working memory: Acts as a transitory space for the immediate processing of information, allowing its connection with broader concepts and specific contexts (Baddeley and Hitch, 1974).

• Long-term memory: Consolidates and organizes information in a structured and accessible way, ensuring that it can be retrieved and applied in future situations (Tulving, 1972).

II. Intelligence:

• Identifies patterns and regularities in stored information, promoting a deeper understanding (Piaget, 1950).

• Constructs generalizations and rules applicable to multiple contexts, based on deduction and induction processes.

III. Willpower:

• Acts as the driving force behind motivation, directing cognitive and emotional efforts toward the consolidation of knowledge (Ryan and Deci, 2000).

• Supports continuity in learning, especially in contexts that require perseverance and dedication.

IV. Imagination:

• Produces the "mental images" that make thinking possible, as described by Aristotle (350 BC).

• Combines existing elements in innovative ways, creating new solutions and contributing to scientific and practical creativity (Csikszentmihalyi, 1990; Kaufman and Beghetto, 2009).

V. Cogitative:

• Articulates thoughts through language, essential for communication and the formulation of complex concepts (Vygotsky, 1986).

• Integrates the results of cognition into coherent narratives, promoting the refinement of arguments and the resolution of structured problems (Tomasello, 2014; Pinker, 2007).

THE SCOPE OF COMPUTATIONAL THINKING: CONSTRUCTION OF ALGORITHMS

Although all of these internal senses are fundamental to the process of transforming information into knowledge, they are universal and act in all areas of human cognition, therefore, they should not be confused with one or another type of thinking. Piaget (1950)

Revista ARACÊ
ISSN: 2358-2472

observes that mental mechanisms such as assimilation and accommodation operate in a generalized way, but their practical application is shaped by specific contexts. Therefore, it is not the isolated presence or functioning of each of them that defines computational thinking, but rather the specific scope in which they operate. In the context of computational thinking, memory retains and structures relevant information, intelligence identifies patterns and logical relationships, imagination combines elements to create innovative solutions, cognition organizes and articulates thought into formal representations, and will drives the effort required to solve problems. Thus, computational thinking is distinguished by the targeted application of these processes. s to the creation of algorithms, logical and unambiguous structures that organize information in a way that can be executed by machines. By defining this scope, it becomes possible to characterize computational thinking as a unique form of human cognition that is not confused with other forms.

Pseudocode is not necessarily an algorithm

A possible confusion in the teaching of computational thinking is the idea that pseudocode is equivalent to an algorithm. Although pseudocode is a useful textual representation for expressing the logic of an algorithm, it does not necessarily have the essential characteristics that define an algorithm.

An algorithm must be clear, systematic, and potentially executable by a machine. To illustrate this distinction, consider the following examples:

Pseudocode that is not an algorithm: an exaggerated example

1. Remember something good your mother told you.
2. Reflect on it in your heart.
3. Tell how you felt.

This pseudocode, although executable by a human, demonstrates how a sequence of vague and subjective steps fails to meet the criteria necessary to be considered an algorithm. The dependence on human judgment, the lack of specificity, and the impossibility of practical execution by a machine highlight its inadequacy in the context of computational thinking.

Pseudocode that is an algorithm

1. Read an integer.

2. If the remainder of the integer division of n by 2 (n % 2) is equal to 0, display "Even".

3. Otherwise, display "Odd".

This second example meets the criteria of clarity, systematicity, and executability by a machine. The logic is clearly defined, and the steps can be implemented directly in any programming language.

The lack of distinction between pseudocode and algorithm in educational practices reflects and reinforces the ambiguity present in Wing's (2006) definition of computational thinking. By accepting general processes as part of the scope of computational thinking, including vague textual representations, the definition dilutes the algorithmic specificity central to the concept.

CONDITIONS ON THE DEVELOPMENT OF COMPUTATIONAL THINKING

Although this article has emphasized the cognitive mechanisms that underlie computational thinking, it is essential to recognize that these mechanisms do not operate in a vacuum. They are deeply influenced by a series of social, cultural, biological, and economic constraints that shape both the opportunities and challenges faced by individuals in the construction of computational thinking.

Recognizing the constraints that shape cognitive mechanisms is very important for a more comprehensive view. These influences highlight the importance of creating educational and social policies that mitigate inequalities and promote equitable access to learning computational skills.

Thus, this section serves as an acknowledgement that, although cognitive mechanisms are universal, their manifestation and development are deeply conditioned by contextual factors. Exploring these interactions is an important task for future research and educational practices that aim at the inclusion and equitable development of computational thinking.

The role of social and cultural conditioning

Human cognitive development is influenced by social interactions and cultural tools. Vygotsky (1986) emphasizes that learning and development occur in a process of mediation between the individual and the sociocultural environment, being shaped by language, cultural practices and available tools. In the case of computational thinking,

Revista
ARACÊ
ISSN: 2358-2472

inequalities in access to educational resources, digital technologies and stimulating environments can significantly impact its development.

However, the lack of technological infrastructure is not, in itself, an insurmountable barrier to learning computational thinking. Unplugged activities, which do not depend on computers or digital devices, offer ways to develop skills such as abstraction, pattern recognition and decomposition in the construction of algorithms. What becomes essential, in this context, is the preparation and training of teachers to mediate this process efficiently. Furthermore, cultural differences in the value placed on different forms of structured reasoning can influence students' interest and motivation, highlighting the importance of educational practices that make computational thinking accessible and relevant to different realities.

Biological and individual conditioning

Biological factors also play a relevant role in the development of computational thinking. Brain maturity, for example, is a determining factor for the capacity for abstraction and complex problem-solving. Piaget (1950) describes that cognitive development occurs in successive stages, in which more advanced skills emerge as the child reaches specific levels of maturation and interaction with the environment.

In addition, individual differences, including variations in cognitive functioning, learning styles, and neurodiversity, influence the way each person internalizes and applies concepts related to computational thinking.

Economic conditioning

Human cognitive development occurs in constant interaction with the social and cultural environment. Vygotsky (1986) emphasizes that learning is mediated by language, cultural practices, and the tools available in the environment. In the case of computational thinking, inequalities in access to educational resources and digital technologies can impact learning.

EVOLUTION AND CHALLENGES: A CRITICAL ANALYSIS OF WING'S WORK

Since 2006, Jeanette Wing has played a fundamental role in the consolidation of computational thinking (CT) as an essential skill in education and in interdisciplinary areas. Through his articles, Wing has presented core definitions of the concept and explored its

Revista ARACÊ
ISSN: 2358-2472

applications and challenges in various contexts. This section critically analyzes Wing's main contributions in six key articles (2006, 2008, 2011, 2014, 2016, 2017) in light of five categories: definition of computational thinking, centrality of algorithms, universal cognitive elements, educational impact and pedagogical practices, and conceptual expansion and ambiguities.

DEFINITION OF COMPUTATIONAL THINKING

Wing introduced computational thinking as a skill to be cultivated in his seminal 2006 article, defining it as computational processes "performed by a human or a machine". However, the use of the logical connector "OR" in the expression "human or machine" generates important conceptual ambiguities. According to the truth table of propositional logic, the "OR" operation is inclusive, which means that the processes referred to by Wing (2006) can be executed exclusively by humans, exclusively by machines, or by both. This formulation broadens the scope of computational thinking, allowing purely human processes, not executable by machines, to be considered part of the concept. In subsequent texts, Wing does not explicitly revisit this definition, nor does she establish rigorous criteria to differentiate exclusively human processes from those that are essentially computational. In Wing (2008), for example, computational thinking continues to be presented as a universal skill, without a clear delimitation that avoids confusion with other types of human reasoning. In the 2011 article, the author emphasizes that CT involves "fundamental concepts of computer science" and should be treated as an essential pillar of modern education, but still without resolving the issue of its scope. In Wing (2014), the focus shifts to the implementation of CT in teaching, but once again, without a redefinition that eliminates the ambiguity of the conceptual scope.

Even in her most recent article analyzed (Wing, 2017), the lack of definition persists. The author reiterates that computational thinking can be applied to all areas of knowledge and reinforces its importance in the formation of critical and creative citizens. However, this approach, without a clear distinction between computational and human processes, maintains the possibility of overly broad interpretations that dilute the identity of computational thinking and hinder its consolidation as a formally delimited field.

Revista ARACÊ
ISSN: 2358-2472

CENTRALITY OF ALGORITHMS

Throughout her work, Wing recognizes the importance of algorithms as an inseparable part of computational thinking. However, by including in the concept of computational thinking the ability to “solve problems, design systems and understand human behavior based on fundamental concepts of computer science” (Wing, 2006), the author reinforces the ambiguity already present in the initial definition and unduly expands the scope of the concept. The inclusion of the understanding of human behavior in computational thinking suggests a shift from a technical focus to a broader cognition, without the requirement of clear and systematizable algorithms.

In later texts, this issue is not resolved. In Wing (2011), for example, there is reinforcement of the idea that computational thinking involves “the formulation and resolution of problems in a way that can be executed by an information processing agent”, but the relationship between this agent and the need for structured algorithms is not explored in depth. In Wing (2014), when discussing the implementation of computational thinking in the school curriculum, the author defends the need to teach children to think computationally from an early age, but without clearly defining what distinguishes this ability from other forms of logical or creative thinking.

Furthermore, Wing (2017) continues to emphasize the transversality of computational thinking across different disciplines. inas, without rigorously defining the centrality of algorithms. This lack of a specific definition compromises the clarity of the concept and allows interpretations that distance computational thinking from its algorithmic and systematic basis.

UNIVERSAL COGNITIVE ELEMENTS

Wing consistently highlights the importance of cognitive elements such as abstraction, decomposition and pattern recognition in computational thinking. However, her approach does not clearly differentiate these elements from general forms of cognition. In Wing (2008), for example, the author states that CT involves "thinking abstractly", but does not explain how this abstraction is distinguished from that used in other disciplines, such as mathematics or philosophy.

In Wing (2011), the emphasis on abstraction remains, but there is little differentiation between the abstraction necessary for the formulation of algorithms and abstraction as a broad cognitive process. In Wing (2016), the author reinforces the need to develop these

Revista ARACÊ
ISSN: 2358-2472

skills in teaching, but does not address how their specific application in computational thinking can be differentiated from other educational practices.

The lack of a clear criterion to delimit the application of cognitive elements in computational thinking contributes to conceptual confusion. By presenting abstraction, creativity and pattern recognition as essential components of computational thinking without establishing their specific relationship with the creation of algorithms, Wing broadens the concept to the point of making it indistinguishable from other forms of human reasoning.

EDUCATIONAL IMPACT AND PEDAGOGICAL PRACTICES

Wing (2014) and Wing (2017) emphasize the need to include computational thinking in education from the early years. The author argues that computational thinking can be taught through unplugged activities and interdisciplinary teaching strategies. However, the conceptual uncertainty of computational thinking directly impacts its educational implementation.

For example, in Wing (2014), there is a concern with training teachers to teach computational thinking, but without a clear definition of what needs to be taught. In Wing (2017), the author reinforces the importance of the transversality of CT, but does not establish how to differentiate it from other forms of problem-solving. The lack of clarity leads to school curricula that can misinterpret computational thinking, diluting its identity and resulting in practices that barely reflect its algorithmic essence.

CONCEPTUAL EXPANSION AND AMBIGUITIES

Since its introduction in 2006, computational thinking has been presented as a skill applicable to all areas of knowledge. However, Wing (2011) reinforces the difficulty of establishing its limits by stating that "everyone should learn computational thinking". This proposal, although interesting from an educational point of view, does not establish objective criteria that differentiate CT from other cognitive processes.

In Wing (2014), there is an attempt to connect computational thinking to different disciplines, but without defining exactly which elements are exclusive to this form of thinking. Wing (2017) further expands this view, suggesting that the CT may be a general mental model applicable to science, engineering, and even the humanities. This approach can be interpreted in an overly broad way, making computational thinking indistinguishable from other forms of analytical reasoning and problem-solving.

Thus, the issue is not the interdisciplinarity of computational thinking, but the lack of an operational model that establishes its conceptual limits. For its application to be effective, it is essential to define its fundamental principles, especially with regard to the centrality of algorithms and the systematicity of their solutions.

FINAL CONSIDERATIONS

A detailed analysis of Jeanette Wing's publications over more than a decade revealed significant contributions to the dissemination of the concept of computational thinking, but also highlighted ambiguities that need to be addressed to ensure its theoretical and practical application. Since its initial formulation, the concept of computational thinking has been presented as a broad and essential competence, but with unclear conceptual limits, such as the inclusion of exclusively human processes that go beyond the computational scope. The use of the logical operator "OR" in the definition "Computational thinking builds on the power and limits of computing processes, whether they are executed by a human OR by a machine" (Wing, 2006) generates ambiguities that make it difficult to delimit computational thinking in relation to other forms of cognition. Although Wing expanded the discussion on computational thinking in later works, the absence of an explicit revision of the initial definition maintained the conceptual ambiguity present in its original formulation.

In addition, A review of his work highlights a persistent lack of centrality in the role of algorithms, an element that defines the specificity of computational thinking. Although concepts such as abstraction, decomposition and pattern recognition are essential, it is their application in the process of building algorithms that establishes computational thinking as something unique. The conceptual dilution of CT results in educational difficulties, since the lack of definition of what should be taught compromises its effective implementation.

In addition to the conceptual lack of definition identified in Wing's work, another contemporary issue that requires greater clarity in the delimitation of computational thinking concerns the relationship between probabilistic models and the traditional algorithmic structure. Although probabilistic models can be powerful auxiliary tools, they do not

Revista ARACÊ
ISSN: 2358-2472

eliminate the need for structured algorithms, which continue to be the operational basis of CT. Computational thinking and probabilistic models are not competitors, but rather complementary, since CT provides the logical and formal structure on which probabilistic models can operate to deal with uncertainty.

While computational thinking emphasizes the construction of structured, deterministic, and systematic solutions, probabilistic models deal with uncertainty, variability, and emergent patterns. In many cases, deterministic algorithms and statistical techniques work together to optimize processes, such as in data filtering, machine learning, and artificial intelligence. For example, a recommendation system may combine traditional algorithmic strategies for organizing and retrieving information, while probabilistic models refine suggestions based on statistical patterns and adaptive inferences.

This relationship does not detract from computational thinking, but rather complements it, offering new layers of complexity and efficiency. However, it is important to emphasize that algorithmic logic remains essential, as probabilistic models and deep learning function as complementary analytical tools within a broader algorithmic framework. Ultimately, computational thinking provides the logical framework necessary to interpret, validate, and integrate probabilistic models in a reliable and understandable way.

Thus, rather than replacing algorithms, probabilistic models become an additional element within a broader computational ecosystem, in which computational thinking remains the fundamental foundation. This interaction between deterministic logic and statistical inference illustrates the need for a clear definition of CT, which recognizes its capacity to integrate multiple concepts without losing its central algorithmic identity.

In this way, the criticism of the conceptual vagueness of computational thinking remains pertinent. If any problem-solving strategy is indistinctly classified as CT, the concept becomes diffuse and loses its distinctive character, compromising its applicability and its academic and pedagogical relevance. Algorithmic logic must continue to be the basis of computational thinking, as it is the structure capable of ensuring clarity, predictability and systematicity in the resolution of computable problems.

To ensure the advancement of computational thinking as a pillar of contemporary education, it is necessary to align theory and practice. This requires a coordinated effort to review conceptual definitions, strengthen teacher training, and implement pedagogical activities that reflect the centrality of algorithms.

Revista ARACÊ
ISSN: 2358-2472

Clearly defining the boundaries of computational thinking is essential to strengthen its conceptual identity, ensuring that its implementation in education and scientific research is accurate and effective.

Revista ARACÊ
ISSN: 2358-2472

REFERENCES

1. Aristóteles. (1986). De Anima (W. S. Hett, Trad.). Harvard University Press. (Original publicado em 350 a.C.).

2. Baddeley, A., & Hitch, G. (1974). Working memory. Psychology of Learning and Motivation, 8, 47-89.

3. Centro de Inovação para a Educação Brasileira. (2024). CIEB: Notas Técnicas #21 Inteligência artificial na educação básica: novas aplicações e tendências para o futuro (e-book em PDF). Disponível em: https://cieb.net.br/wp-content/uploads/2024/06/Inteligencia-Artificial-na-Educacao-Basica_2024.pdf. Acesso em: 26 ago. 2024.

4. Csikszentmihalyi, M. (1990). Flow: The Psychology of Optimal Experience. Harper & Row.

5. Deci, E. L., & Ryan, R. M. (1985). Intrinsic Motivation and Self-Determination in Human Behavior. Springer Science & Business Media.

6. Ellis, N. C. (1980). The Study of Second Language Acquisition. Pergamon.

7. Ferrari, M., & Sternberg, R. J. (1998). Self-awareness: Its Nature and Development. Guilford Press.

8. Gibson, J. J. (1979). The Ecological Approach to Visual Perception. Houghton Mifflin.

9. James, W. (1890). The Principles of Psychology (Vols. 1 & 2). Henry Holt and Company.

10. Kaufman, S. B., & Gregoire, C. (2015). Wired to Create: Unraveling the Mysteries of the Creative Mind. Perigee Books.

11. Locke, J. (1996). An Essay Concerning Human Understanding (P. H. Nidditch, Ed.). Clarendon Press.

12. Maeda, J. (2004). Creative Code: Aesthetics and Computation. MIT Press.

13. Marr, D. (1982). Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. W. H. Freeman.

14. Oliveira, C. M., & Pereira, R. (2019). Desenvolvimento do Pensamento Computacional no Ensino Superior em Ciência da Computação. In VIII Congresso Brasileiro de Informática na Educação (CBIE 2019). Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação. Universidade Federal do Paraná (UFPR). DOI: 10.5753/cbie.wcbie.2019.1502.

15. Pinker, S. (2007). The Stuff of Thought: Language as a Window into Human Nature. Viking.

Revista ARACÊ
ISSN: 2358-2472

16. Piaget, J. (1977). The Development of Thought: Equilibration of Cognitive Structures. Harvard University Press.

17. Piaget, J. (1950). The Psychology of Intelligence. Routledge and Kegan Paul. (Original em francês: La Psychologie de l'Intelligence [1947]).

18. Reas, C., & Fry, B. (2001). Processing: A flexible software sketchbook and a language for learning how to code within the context of the visual arts. MIT Media Lab.

19. Resnick, M., et al. (2009). Scratch: Programming for All. Communications of the ACM, 52(11), 60-67. https://doi.org/10.1145/1592761.1592779. Acesso em: 27 dez. 2024.

20. Root-Bernstein, R. S. (2013). Sparks of Genius: The 13 Thinking Tools of the World's Most Creative People. Mariner Books.

21. Sawyer, R. K. (2012). Explaining Creativity: The Science of Human Innovation. Oxford University Press.

22. Sweller, J. (1994). Cognitive Load Theory, Learning Difficulty, and Instructional Design. Learning and Instruction, 4(4), 295-312.

23. Tomasello, M. (2014). A Natural History of Human Thinking. Harvard University Press.

24. Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), Organization of Memory (pp. 381-403). Academic Press.

25. Vygotsky, L. S. (1986). Thought and Language (A. Kozulin, Trad.). MIT Press. (Original em russo publicado em 1934).

26. Wing, J. M. (2006). Computational Thinking. Communications of the ACM, 49(3), 33-35. https://doi.org/10.1145/1118178.1118215. Acesso em: 26 ago. 2024.

27. Wing, J. M. (2008). Five deep questions in computing. Communications of the ACM, 51(1), 58-60. https://doi.org/10.1145/1327452.1327479. Acesso em: 26 ago. 2024.

28. Wing, J. M. (2011). Computational thinking: What and why. The Link, Carnegie Mellon University. Disponível em: http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why. Acesso em: 26 ago. 2024.

29. Wing, J. M. (2014). Computational thinking [PDF]. Disponível em: http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing14.pdf. Acesso em: 26 ago. 2024.

30. Wing, J. M. (2016). Computational thinking, 10 years later. Communications of the ACM, março. Disponível em: https://www.acm.org/media/news/2016/wings-computational-thinking. Acesso em: 26 ago. 2024.

31. Wing, J. M. (2017). Computational thinking's influence on research and education for all. Italian Journal of Educational Technology, 25(2), 7-14.

Revista ARACÊ
ISSN: 2358-2472

https://doi.org/10.17471/2499-4324/922. Disponível em: https://ijet.itd.cnr.it/article/view/922. Acesso em: 26 ago. 2024.