https://doi.org/10.7551/mitpress/14041.003.0009

6

COMPUTATIONAL THINKING ASSESSMENT: A DEVELOPMENTAL APPROACH

Marcos Román-González and Juan-Carlos Pérez-González

INTRODUCTION

In a previous work (Román-González, Moreno-León, and Robles 2019), we argued for the need to build inclusive systems of assessments in order to perform comprehensive evaluations of educational interventions that involve computational thinking (CT). To do so, we started by proposing a taxonomy of CT assessment tools regarding their evaluative approach. Thus, we could differentiate between seven types of instruments: (1) *diagnostic tools* (i.e., those that measure the CT aptitudinal level of the subject); (2) *summative tools* (i.e., those that evaluate if the learner has achieved enough content knowledge—and/or if he is able to perform properly—after receiving some instruction or training in CT skills); (3) *formative-iterative tools* (i.e., those that provide feedback to the learner to improve his/her CT skills); (4) *data-mining tools* (i.e., those that retrieve and record the learner activity in real time while acquiring computational concepts and practices); (5) *skill transfer tools* (i.e., those that assess to what extent the students are able to transfer their CT skills onto different kinds of problems, contexts, and situations); (6) *perceptions–attitudes scales* (i.e., those that are aimed at assessing the perceptions, such as self-efficacy perceptions, and attitudes of the subjects not only about CT, but also about related issues such as computers, computer science, computer programming, or even digital literacy); and (7) *vocabulary assessment tools* (i.e., those that intend to measure several elements and dimensions of CT when they are verbally expressed by the subjects).

Moreover, in that previous work (Román-González, Moreno-León, and Robles 2019), we established three possible criteria that could provide guidance on how to properly combine the aforementioned types of CT assessment instruments. Hence, we first explored to what extent each type of tool is suitable for measuring distinct CT dimensions (i.e., computational concepts, computational practices, and computational perspectives). Second, we exposed how to use each of those kinds of instruments regarding the different phases and chronological points within a CT educational evaluation (i.e., before, along, just after, or sometime after the intervention). Finally, we attempted to state what level(s) of Bloom's (revised) taxonomy of cognitive processes each type of CT assessment tools is addressing. However, nothing was said about how to align this myriad of CT assessment tools with the various ages and developmental stages of the individuals along K–12 education. Indeed, there is a notorious theoretical and empirical gap regarding CT assessment from a developmental perspective, and only occasional research has been undertaken in this vein.

In sum, the main contribution of that previous chapter (Román-González, Moreno-León, and Robles 2019) was to define an inclusive framework that enlightens how to combine different CT assessment tools to conduct comprehensive evaluations *at a certain moment*. In other words, our previous work was written from a *cross-sectional perspective* since its goal was to show how to set up comprehensive research designs *at a certain moment*. Nevertheless, we did not address how to articulate CT assessment from a developmental approach; that is, how to conduct CT assessments along the successive stages of human development. This is precisely the aim of the present chapter. To achieve our goal, we will intersect some of the current corpus of scientific knowledge on CT assessment with Piagetian and neo-Piagetian developmental theories, along four school stages (kindergarten, elementary school, middle school, and high school). For each of the four stages, we will try to answer the following questions:

- Which CT cognitive subprocesses are mainly involved and may be assessed at each stage of development? Which other basic cognitive skills may support CT at each stage of development?
- Consequently, which computational concepts, practices, and perspectives may be specifically addressed at each stage of development?
- Finally, which kind of CT assessment instruments may be more appropriate at each stage of development? Can we find good examples of CT assessment tools at each stage?

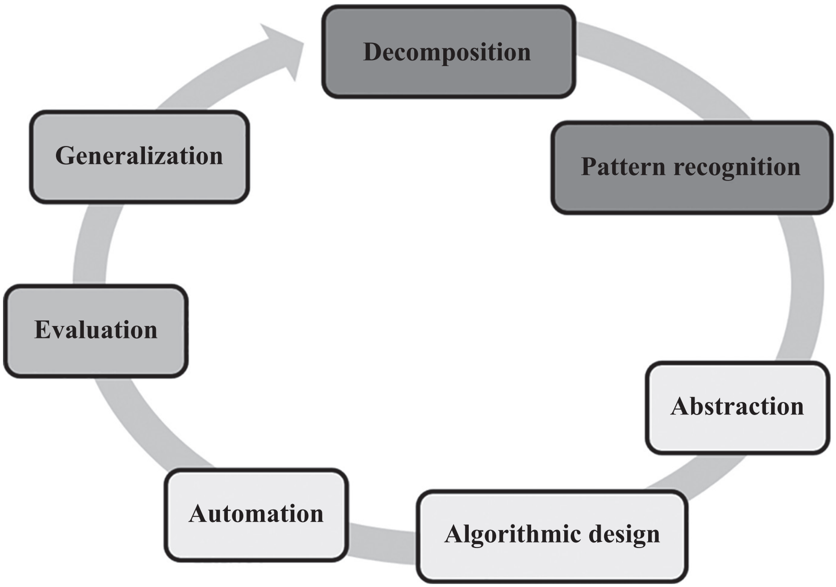
Ultimately, when answering the prior questions, the goal of the present chapter is to sketch a possible framework that could guide the configuration of CT longitudinal research designs. Therefore, the present manuscript is written from a *longitudinal perspective*.

BACKGROUND: CT FROM A COGNITIVE PERSPECTIVE

In previous papers, we have offered several definitions of computational thinking (CT). For instance, we have stated that CT involves the ability to formulate and solve problems by relying on the fundamental concepts and practices of computing (Román-González, Pérez-González, and Jiménez-Fernández 2017). We have also defined CT as the ability that allows the subject to effectively solve problems and express ideas by using the power of computers (Moreno-León, Robles, Román-González, and Rodríguez García 2019). However, none of the aforementioned definitions is enough to face the target of the present chapter. Since we aim to intersect CT with Piagetian developmental theories, it is necessary to define CT in cognitive terms.

In this vein, CT can be understood as a problem-solving ability that is composed by the following series of successive cognitive subprocesses (figure 6.1), namely:

i. **Decomposition**: to break down a problem into smaller and simpler elements.
ii. **Pattern recognition**: to perceive and detect regularities between those elements.
iii. **Abstraction**: to remove/ignore nonrelevant details and information of the problem to highlight the critical variables that will enable one to represent it properly (i.e., often called "internal generalization").



6.1 Cognitive subprocesses involved in CT.

iv. **Algorithmic design**: to build step-by-step instructions that, when followed, will allow for solving the problem.
v. **Automation**: to implement the aforementioned instructions within a digital device by means of some kind of computer programming.
vi. **Evaluation**: to assess the efficacy and efficiency of that implementation in order to debug and/or to improve the programming code.
vii. **Generalization**: to transfer the achieved solution to a wider range of analogous problems (i.e., often called "external generalization").

We can exemplify the previous figure trying to think computationally on the following problem: "what clothes should I wear today?" If we want to project CT on that problem, firstly we must *decompose* the broad category "clothes" into simpler elements (e.g., underwear, bottom clothes, top clothes, shoes, and accessories). Then, we must recognize some *patterns* along those elements (e.g., long trousers are usually worn with shirts, while short trousers do so with t-shirts; umbrellas are only used while raining, while coats only do so under certain temperature; blue and white clothes tend to fit together, while red and green do not). Afterwards, it becomes essential to detect and *abstract* the critical variables that permit to internally represent the space-problem (e.g., the clothes I should wear today will probably depend on “the day of the week” (working day versus weekend), “the weather” (e.g., temperature or presence or absence of rainfalls), “the range of colors” I wish to wear, and so on. Then, it becomes possible to *design an algorithm* based on those critical variables, which could state the instructions to answer the problem in different conditions. For example:

If today is a working day, I should wear long trousers, a shirt, and a jacket; else [if temperature is over 20°C, I should wear short trousers and a t-shirt; else I should wear a tracksuit].

Once the algorithm has been designed, it can be *automated* and implemented in a digital device through programming and executing some computer code (e.g., we can implement the aforementioned algorithm that solves “what clothes should I wear today” as a mobile app written in App Inventor language, which could be connected to the calendar and to the weather forecast to give an answer to that question). Furthermore, the algorithm and its corresponding programming code can be *evaluated* and refined to improve their efficacy and efficiency (e.g., the algorithm may better fix the values of some parameters and/or may include new variables such as “the current mood” of the subject who is using the clothing app). Finally, we can perform a second-level abstraction to find some communalities between our specific problem and a wider family of analogous ones, so it will become possible to transfer and externally *generalize* some elements of our specific solution (e.g., we may generalize our algorithm and its corresponding programming code to a similar problem such as “what food should I eat today?”).

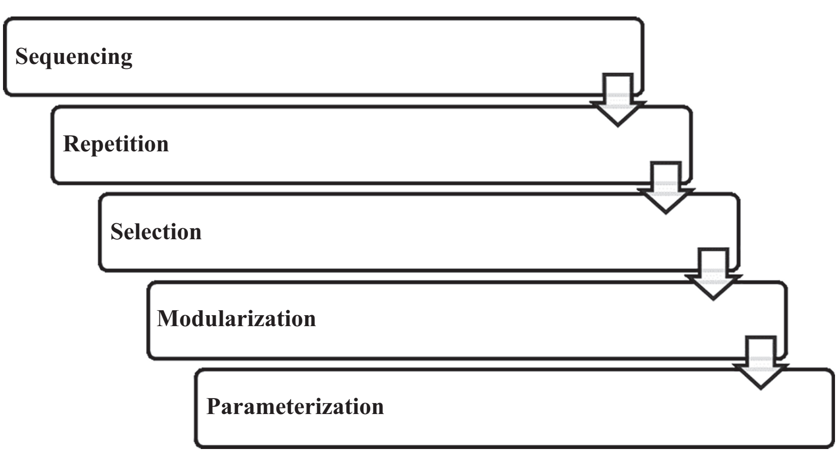
It is worth noting that not all the aforementioned cognitive subprocesses stand at the same level of importance and hierarchy within the whole CT process. On the one hand, *decomposition* and *pattern recognition* (see the green boxes in figure 6.1) are common elements in almost every problem-solving task. In other words, “decomposition” and “pattern recognition” are not specific to CT, and they could even be considered just as prerequisites to think computationally.

On the other hand, *abstraction*, *algorithmic design*, and *automation* (see the blue boxes in figure 6.1) can be located at the core of the CT process. Thus, relevant authors have stated that abstraction (e.g., Grover and Pea

2013; Wing 2006) and/or algorithmic thinking (Aho 2012; Shute, Sun, and Asbell-Clarke 2017) are the main cognitive subprocesses for thinking computationally. For example, Jeanette Wing affirms that CT "requires thinking at multiple levels of abstraction" (Wing 2006, 35), and/or Alfred Aho defines CT as "the thought process involved in formulating problems so their solutions can be represented as computational steps and algorithms" (Aho 2012, 832). Nevertheless, if we only take into account "abstraction" and "algorithmic design" to define the core of CT, then it could become indistinguishable from mathematical thinking (Stephens and Kadijevich 2020). According to these authors and from our point of view, it is indispensable to add "automation" as a third element to the aforementioned pair in order to clearly characterize the essence of CT.

Finally, *evaluation* and *generalization* (see the red boxes in figure 6.1) may be considered rather as consequences, implications, or applications of CT than as central elements of it.

Furthermore, if we focus again on *algorithmic design* as the most specific cognitive subprocess of CT, then it comes very relevant to point out what structures can be progressively learned and used by the individual to build better and more complex algorithms. These structures are (Mühling, Ruf, and Hubwieser 2015; Román-González 2016): (a) *sequencing* structures; (b) *repetition* structures; (c) *selection* structures; (d) *modularization* structures; and (e) *parameterization* structures (figure 6.2).



6.2 Algorithmic design structures.

The aforementioned structures for algorithmic design involve increasing levels of abstraction, they can be progressively incorporated and nested along algorithmic solutions, and they can all be automated by means of computer programming.

In the following sections, we will take figures 6.1 and 6.2 as a reference to describe CT and its assessment from a developmental approach.

CT ASSESSMENT FROM A DEVELOPMENTAL APPROACH

Once we have defined CT in cognitive terms, it becomes possible to intersect CT with Piaget's theory of cognitive development. The goal of this intersection is to explore and describe CT assessment from a developmental approach, along the following four educational stages: kindergarten, elementary school, middle school, and high school.

CT ASSESSMENT IN KINDERGARTEN

Kindergarten mainly concurs with the so-called preoperational stage in Piagetian theory (Ginsburg and Opper 1988). During this stage, young children are not able to perform logical operations, nor onto mental concepts nor even onto physical elements that surround the subjects. Consequently, kindergarteners cannot abstract or think in algorithmic terms, so these core elements of CT should not be assessed during this stage.

In other words, during kindergarten, we should not expect to develop and assess the whole cycle of CT cognitive subprocesses depicted in figure 6.1. At most, we may focus on fostering and assessing so-called prerequisites of CT, namely "decomposition" and "pattern recognition." In this vein, some basic cognitive abilities that become critical to support CT development at this stage are attentional and perceptual skills (Georgiou and Angeli 2019; Marinus et al. 2018; Urlings, Coppens, and Borghans 2019) (see figure 6.11).

Referring to figure 6.2, on the one hand, just some sequencing and repetition protostructures may be found during this stage. These protostructures rely on and are limited by one main characteristic of preoperative thought: irreversibility (i.e., when children are unable to mentally reverse a sequence of events). Thus, kindergarteners are only able to break down, serialize, and sequence elements in one single direction. On the other hand, selection structures are unexpected to appear during kindergarten, since precausal thinking is another main feature of children in the preoperational stage.

Finally, we can state some principles (and cite some good examples) for addressing CT assessment in kindergarten:

- Although kindergarteners are not able to perform logical operations, they can represent and understand symbols, mainly by means of so-called symbolic play. In this vein, CT assessment should rely on playing symbolic games that involve CT prerequisites and sequencing/repetition protostructures (e.g., when assessing CT within the context of playing with Bee-Bots or KIBO robots, which in addition align with the “animism” of kindergarteners) (Critten, Hagon, and Messer 2021; Kotsopoulos et al. 2021; Relkin, de Ruiter, and Bers 2021).
- Since these children are not able to abstract information, paper-and-pencil assessment or testing are not recommended along this stage. Instead of that, it may be more appropriate to assess CT through observation grids and templates while kids are directly manipulating physical objects to solve problems that involve CT prerequisites (Angeli and Valanides 2020; Diago, Arnau, and González-Calero 2018).
- Since kindergarteners are keen on learning by means of narratives and tales, one relevant and natural way to develop and to assess CT along this stage is through decomposing and sequencing stories (Kazakoff, Sullivan, and Bers 2013; Terroba, Ribera, and Lapresa 2020).

CT ASSESSMENT IN ELEMENTARY SCHOOL

Elementary school mainly concurs with the so-called concrete operational stage in Piagetian theory (Ginsburg and Opper 1988). During this stage, children are able to solve problems and perform logical operations onto concrete and specific objects/events that are within their reach. In other words, these individuals are capable of inductive reasoning based on concrete and specific elements around them, often by means of trial-and-error strategies. Conversely, elementary school students have not yet acquired or consolidated deductive reasoning, which involves using general principles to hypothesize and predict further results. Retrieving the terms already used in our background section, we may say that elementary school students can perform first-level abstractions (often called "internal or inductive generalizations"), but not second-level abstractions (often called "external or deductive generalizations").

Referring to figure 6.1, during elementary school, children start to deal with the core elements of CT (namely "abstraction," "algorithmic design," and "automation"). Since elementary school students are able to perform first-level abstractions, they are ready to start designing and automating simple algorithms, which will become more complex along the stage. Then, those core elements of CT should be developed and assessed during elementary school. Conversely, since these children are not able to perform second-level abstractions, the CT cognitive subprocess named "generalization" can hardly be developed or assessed within this stage. Furthermore, since elementary schoolers have not yet acquired enough metacognitive skills, it also does not seem appropriate to assess the "evaluation" subprocess.

Referring to figure 6.2, during elementary school, children can learn and properly use sequencing and repetitions structures (corresponding with computational concepts such as "repeat times-loop" or "repeat until-loop"). Moreover, since children at this age have already acquired causal thinking, they can also understand and apply selection structures (corresponding with computational concepts such as "if-conditional" or "if/else-conditional"). In contrast, it is not to be expected that elementary schoolers use modularization and parameterization structures properly due to the high degree of formalization of these structures, which correspond with computational concepts such as "functions" and "variables."

Nevertheless, in some other works, we have qualified what is said in the previous paragraph:

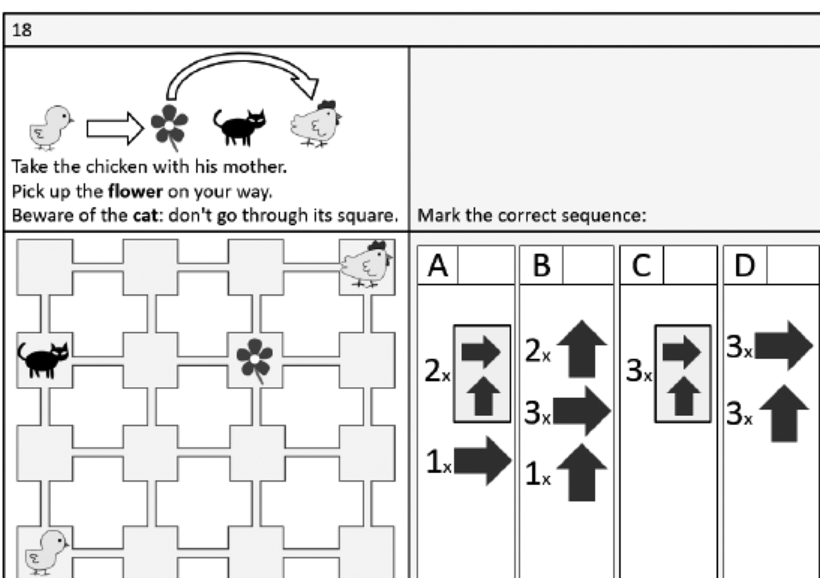
- Due to the maturational limitations of working memory, elementary school students may struggle when mentally sequencing long series of objects/events (a number of four to five elements seems to be the upper limit). These children may also have difficulties when using the "while-loop" since it requires them to apply a repetition structure while a certain condition is met, which is consequently a very demanding computational concept for the working memory of elementary schoolers (Zapata-Cáceres, Martín-Barroso, and Román-González 2021).

- There is evidence that numerical factor/ability is critical to support the development of repetition structures during elementary school (Tsarava et al. 2022) (see figure 6.11).
- There is also evidence that visual elements and colors can scaffold the acquisition of difficult repetition and selection structures in elementary school students.

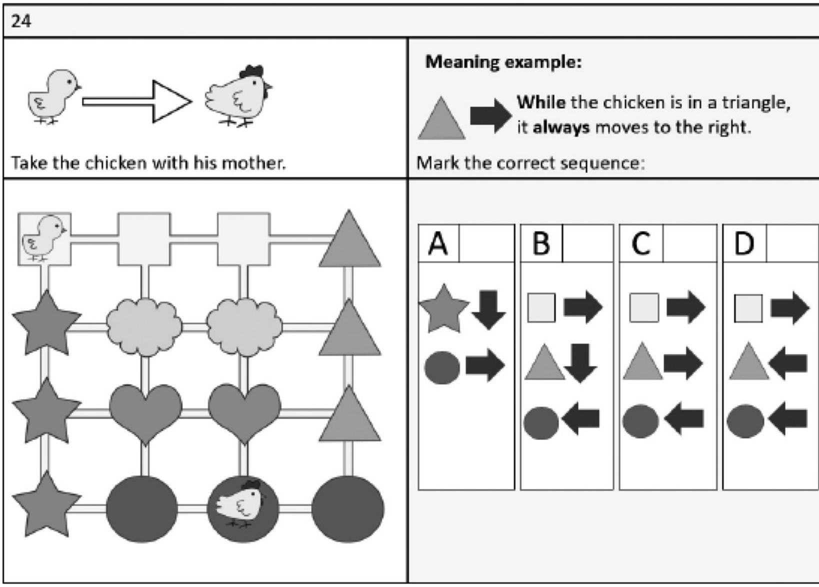
An excellent example of a CT assessment tool aimed at elementary school is the *Beginners Computational Thinking Test (BCTt)* (Zapata-Cáceres, Martín-Barroso, and Román-González 2020). This test consists of twenty-five items and is aligned with all the guidelines exposed along this subsection (figures 6.3 and 6.4).

CT ASSESSMENT IN MIDDLE SCHOOL

Middle school coincides with the beginning of the "formal operational stage" in Piaget's theory of cognitive development (Ginsburg and Opper 1988). During middle school, individuals start to perform logical operations onto symbols related to abstract concepts. Consequently, hypothetical



6.3 *BCTt* item example (item #18).



6.4 *BCTt* item example (item #24).

and deductive reasoning emerges during middle school. Moreover, three cognitive milestones can be highlighted in middle school students:

- The emergence of abstract thinking occurs, which goes beyond concrete and specific objects and events.
- Middle schoolers develop metacognitive skills, which allow the subject to consciously observe, to reason about, and to supervise his/her own thinking.
- Problem-solving in middle schoolers becomes systematical, since they begin to solve problems in a logical and methodical way (not just by means of trial-and-error strategies).

Overall, middle school students begin to distance themselves from concrete reality and from their own cognitive processes, which is also supported by an increase of their working memory and processing capacity. All of the above have obvious consequences on CT development along this stage. Referring to figure 6.1, middle school students become capable of evaluating their own algorithmic solutions (and their corresponding programming codes). Thus, "evaluation" is a CT cognitive subprocess that should be assessed from this stage onwards, mainly by means of debugging computational practices.

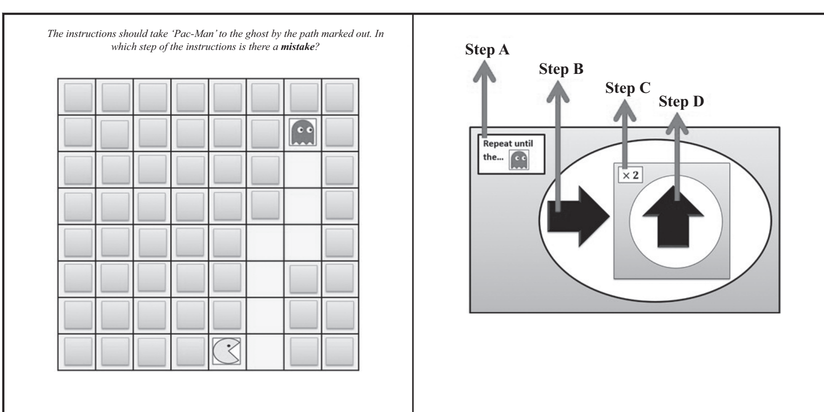
Referring to figure 6.2, the increased working memory of middle school students allows them to deal with long sequences and to master the aforementioned "while-loop." Furthermore, since these children begin to think "out of the box," they become also capable to use modularization structures (corresponding with computational concepts such as "procedures" or "functions") and nesting.

In another vein, there is recent evidence that verbal ability is crucial and critical to support the appearance of these emergent formal-logical thinking skills (Tsarava et al. 2022) (see figure 6.11). In other words, syntactic structures from natural language seem to scaffold the advent of formalization skills along this stage, which are so central to CT development (Howland and Good 2015).

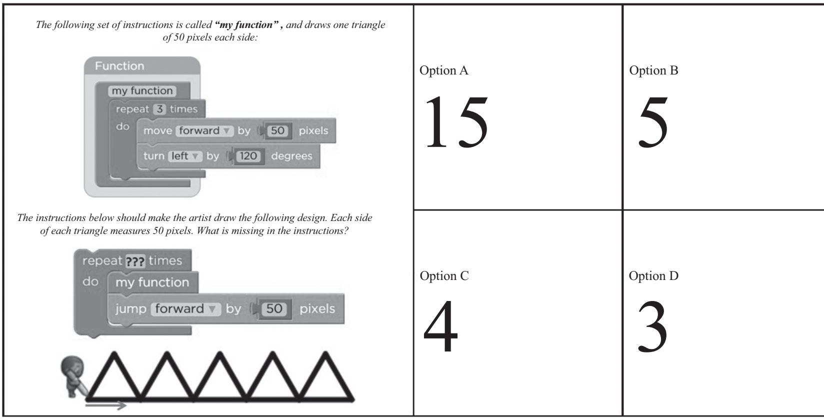
A remarkable example of a CT assessment tool aimed at middle school is the *Computational Thinking Test (CTt)* (Román-González, Pérez-González, and Jiménez-Fernández 2017; Román-González, Pérez-González, Moreno-León, and Robles 2018a). This test consists of twenty-eight items and is aligned with all the ideas exposed along this subsection (figures 6.5 and 6.6).

CT ASSESSMENT IN HIGH SCHOOL (AND BEYOND)

High school often involves the consolidation of the "formal operational stage" in Piaget's theory (Ginsburg and Opper 1988). During this stage, individuals sharpen their formal-logical abilities onto abstract concepts,



6.5 *CTt* item example (item #11).



6.6 *CTt* item example (item #26).

and deductive-hypothetical reasoning skills are refined. Retrieving our previous terms, we can affirm that high school students are finally capable of performing second-order abstractions (i.e., to find some communalities between a specific problem and a wider family of analogous ones), so they become able to transfer and externally generalize some elements of their algorithmic solutions. In this vein and referring to figure 6.1, the CT cognitive subprocess called "generalization" should be specially developed and assessed along this stage.

Referring to figure 6.2, refined formal-logical abilities of high school students permit them to understand and use parameterization structures (corresponding with computational concepts such as "functions with parameters" and "variables"[1]). In another vein, recent evidence suggests that nonverbal reasoning (also called visual or figurative reasoning) is critical to foster and consolidate CT along this final stage (Tsarava et al. 2022) (see figure 6.11).

When we search for good examples of CT assessment tools aimed at high school (and beyond), the results are scarce. In this regard, one work in progress is the *Computational Thinking Test for Higher Education*[2] *(CTt-H)* (Lafuente Martínez et al. 2022), which intends to assess the transfer of CT on a wide variety of problems (figures 6.7 and 6.8).

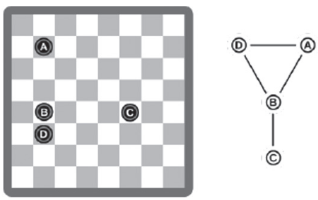
Finally, it is also worth noting that neo-Piagetian theory declares the existence of one more stage in human development called the "postformal thought stage" (e.g., Sinnott 1998). Postformal thought involves the

Below on the left you see a picture of a game board with 4 pieces placed on it.

The diagram on the right of the board represents the positions of the pieces. It is drawn in the following way:

- For each piece on the board, draw a circle.

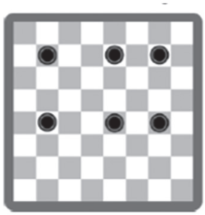
- If two pieces are in the same row on the board or in the same column on the board, then draw a line between the circles in the diagram.



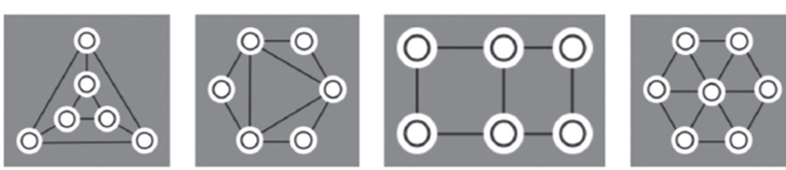
Letters have been placed and the circles so you easily check that the diagram is correct.

A new board of six pieces is shown below.

A new position diagram for this board is drawn in the same way.



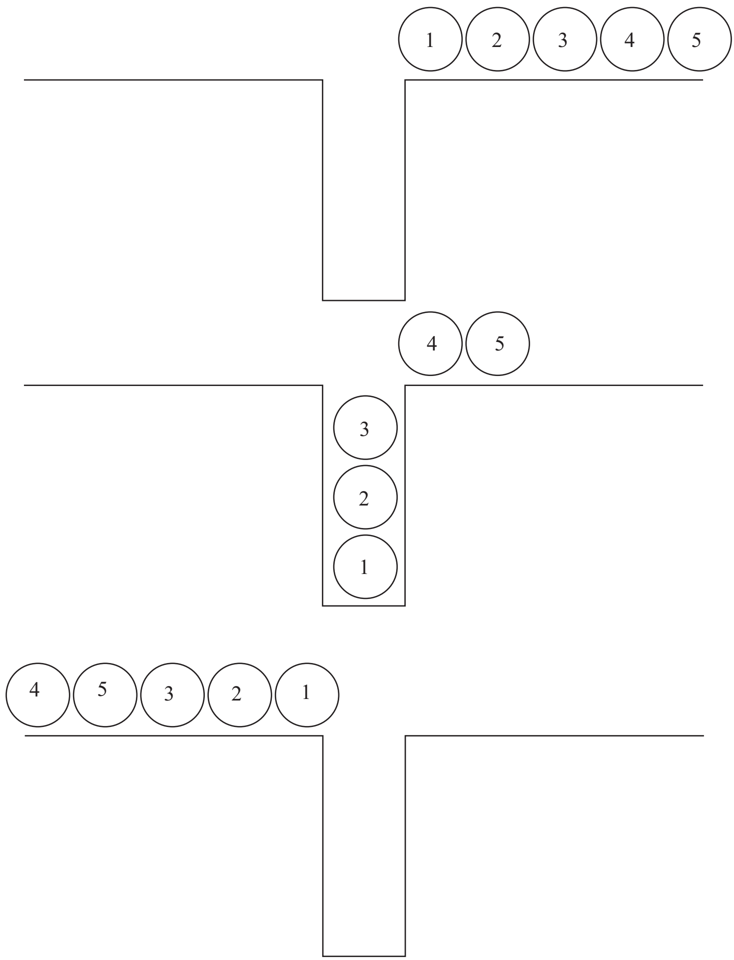
Question:
Which of the four diagrams below were drawn?



A B C D

6.7 *CTt-H* item example #1.

Five people numbered 1, 2, 3, 4, 5 are trying to cross a road that contains a number of deep holes. All holes have in depth 3. To pass over holes with depth 3, the 3 leading people have to go into the holes in order so that the other people can safely cross the hole. Then, the last people who crossed the hole will pull the highest people from the hole up, and so on. For clarity, look at the diagram below where these people are crossing the first hole:



After which hole will they reach the intial formation again (1, 2, 3, 4, 5)?

A) 11th hole
B) 15th hole
C) 16th hole
D) 25th hole

6.8 *CTt-H* item example #2.

communication and coordination of multiple different logics in a dialectical and flexible way. Thus, postformal thought allows the individuals to coordinate multiple goals, methods, causalities, and results to reach a deeper and intersubjective knowledge. All of this resonates with the so-called computational perspectives (namely "expressing," "connecting," and "questioning") (Brennan and Resnick 2012), which consequently must have been developed and assessed during high school as an indispensable ingredient of a CT quality education.

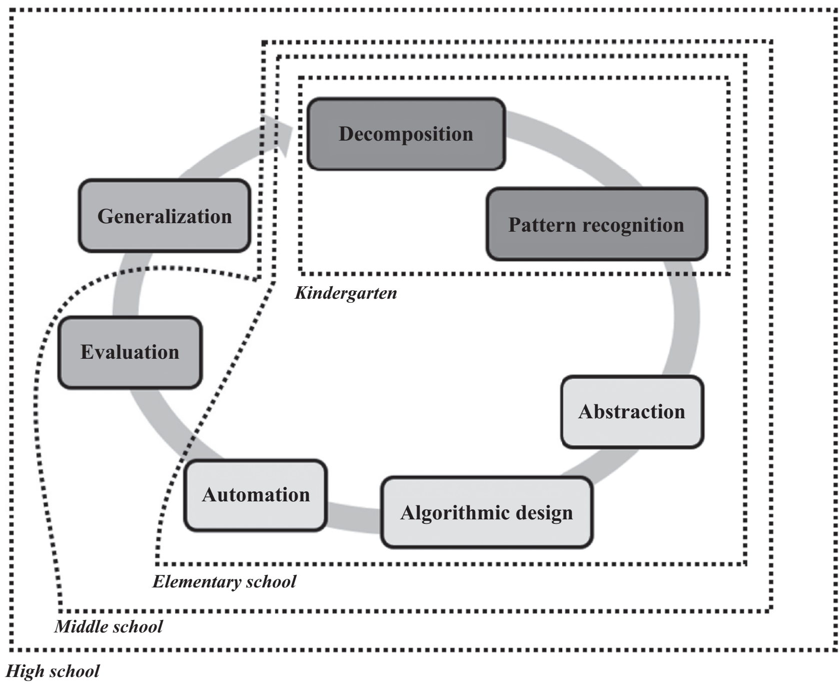
Even more, since postformal thought involves generating, communicating, and coordinating multiple solutions (i.e., more than a single "correct solution") to a certain problem, it aligns with the computational concept called "diffusion." Diffusion is an emergent, flexible, collaborative, and dynamic computational concept that has been recently proposed as a relevant topic for high school students (e.g., Repenning 2017).

CONCLUSIONS, IMPLICATIONS, AND LIMITATIONS

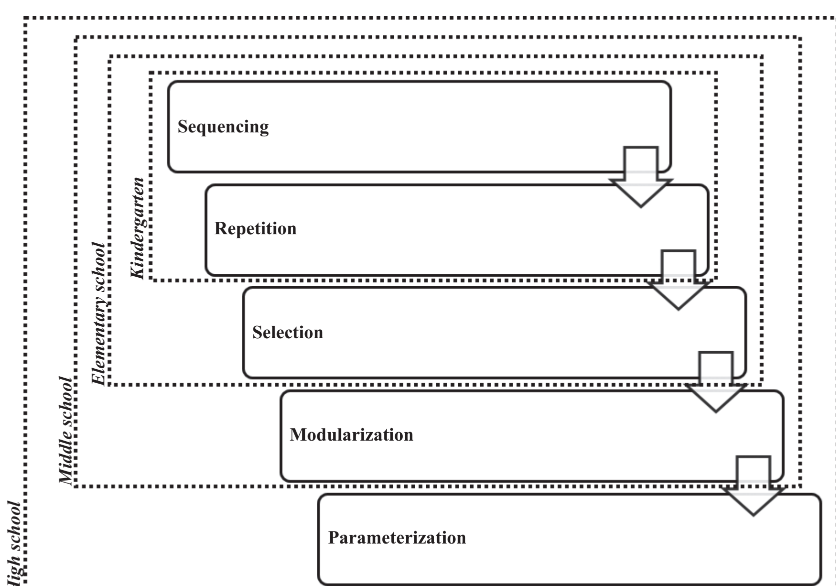
The conclusions of the present chapter can be depicted through figures 6.9–6.11. Figure 6.9 shows what CT cognitive subprocesses should be mainly developed and assessed at each educational stage. Figure 6.10 shows what algorithmic design structures can be understood, properly applied, and assessed at each educational stage. Figure 6.11 shows what basic cognitive abilities support CT development across each educational stage.

It is very important to highlight that the previous three figures are not definitive but rather attempts that must be further investigated and contrasted. In any case, we consider that our present contribution has several relevant implications. On the one hand, it provides a theoretical framework to design longitudinal studies within the CT research field. Although they are more expensive due to experimental and sample mortality, longitudinal studies usually lead to more valid results than cross-sectional studies, since the former take into account the same subjects along time while the latter simultaneously utilize different cohorts of individuals (which implies some threats to the validity of their results). On the other hand, if a CT developmental theory is empirically confirmed, that will contribute to reinforce the construct validity of CT. There is still a long way to go.

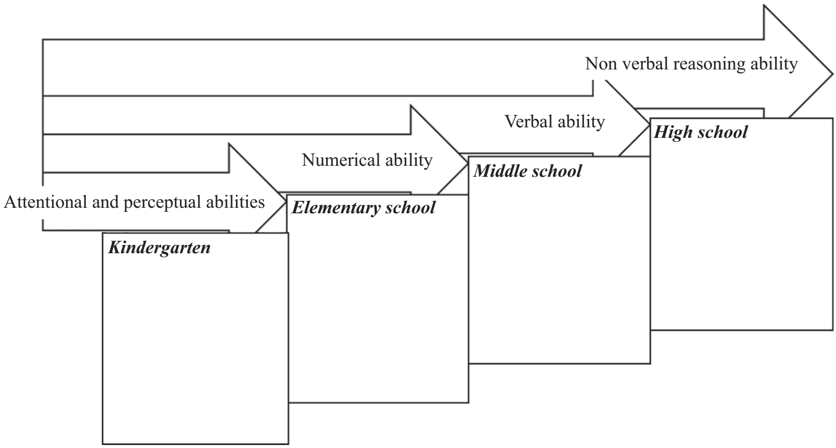
Finally, we must express some limitations to our proposal that coincide with the limitations that have been pointed out to Piagetian theory



6.9 CT cognitive subprocesses at each educational stage.



6.10 Algorithmic design structures at each educational stage.



6.11 Basic cognitive abilities that support CT development at each educational stage.

itself. The main one is that Piagetian stages are just general and rough approximations to the very complex human development, which admit a wide variety of modulations between different subjects, across different cultures, and across different domains of knowledge and expertise. In this vein and in relation to the CT domain, we have found and studied cases of Spanish "computational talents" in middle school who are capable of performing as expected in later stages (Román-González, Pérez-González, Moreno-León, and Robles 2018b).

Overall, we expect that the developmental approach to CT assessment that has been exposed during this chapter will be relevant and useful for policymakers, practitioners, and researchers alike when designing CT evaluations and selecting CT assessment instruments for the different K–12 grades. Nevertheless, they all should keep in mind that no developmental model should serve spuriously to hide the rich diversity and variety among our children nor to limit the talent of our students. Otherwise, we hope that our developmental proposal will serve as guide and encouragement so that each and every child can unfold their full potential.

NOTES

1. Recently in Spain, *Articoding* (https://github.com/WeArePawns/Articoding) was developed. *Articoding* is a serious game aimed at high schoolers that specifically uses and relies on "variables" as an anchor to teach all the rest of computational concepts (Faouaz, García, and Poyatos 2021).

2. The initial pool of items under validation is available at: https://www.surveymonkey.com/r/DN9V7YW.

REFERENCES

Aho, Alfred V. 2012. "Computation and Computational Thinking." *The Computer Journal* 55, no. 7: 832–835. https://doi.org/10.1093/comjnl/bxs074.

Angeli, Charoula, and Nicos Valanides. 2020. "Developing Young Children's Computational Thinking with Educational Robotics: An Interaction Effect Between Gender and Scaffolding Strategy." *Computers in Human Behavior* 105: 105954. https://doi.org/10.1016/j.chb.2019.03.018.

Brennan, Karen, and Mitchel Resnick. 2012. "New Frameworks for Studying and Assessing the Development of Computational Thinking." In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA '2012)*, 1–25. http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf.

Critten, Valerie, Hannah Hagon, and David Messer. 2021. "Can Pre-school Children Learn Programming and Coding Through Guided Play Activities? A Case Study in Computational Thinking." *Early Childhood Education Journal*. https://doi.org/10.1007/s10643-021-01236-8.

Diago, Pascual D., David Arnau, and José Antonio González-Calero. 2018. "La resolución de problemas matemáticos en primeras edades escolares con Bee-Bot." *Matemáticas, Educación y Sociedad* 1, no. 2: 36–50. https://www.uco.es/ucopress/ojs/index.php/mes/article/view/12835.

Faouaz, Dany, Arturo García, and Álvaro Poyatos. 2021. *Serious Games to Promote Computational Thinking and Coding* [Bachelor Thesis]. Madrid: UCM. https://eprints.ucm.es/id/eprint/66940/.

Georgiou, Kyriakoula, and Charoula Angeli. 2019. "Developing Preschool Children's Computational Thinking with Educational Robotics: The Role of Cognitive Differences and Scaffolding." In *Proceedings of the 2019 International Conference on Cognition and Exploratory Learning in the Digital Age (CELDA '2019)*, 101–108. http://dx.doi.org/10.33965/celda2019_201911L013.

Ginsburg, Herbert P., and Sylvia Opper. 1988. *Piaget's Theory of Intellectual Development (3rd Ed.)*. New Jersey: Prentice-Hall.

Grover, Shuchi, and Roy Pea. 2013. "Computational Thinking in K–12: A Review of the State of the Field." *Educational Researcher* 42, no. 1: 38–43. https://doi.org/10.3102/0013189X12463051.

Howland, Kate, and Judith Good. 2015. "Learning to Communicate Computationally with Flip: A Bi-Modal Programming Language for Game Creation." *Computers & Education* 80: 224–240. https://doi.org/10.1016/j.compedu.2014.08.014.

Kazakoff, Elizabeth R., Amanda Sullivan, and Marina U. Bers. 2013. "The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing

Ability in Early Childhood." *Early Childhood Education Journal* 41: 245–255. https://doi.org/10.1007/s10643-012-0554-5.

Kotsopoulos, Donna, Lisa Floyd, Brandon A. Dickson, Vivian Nelson, and Samantha Makosz. 2021. "Noticing and Naming Computational Thinking during Play." *Early Childhood Education Journal*. https://doi.org/10.1007/s10643-021-01188-z.

Lafuente Martínez, , M., Olivier Lévêque, Isabel Benítez, Cécile Hardebolle, and Jessica Dehler Zufferey. 2022. "Assessing Computational Thinking: Development and Validation of the Algorithmic Thinking Test for Adults." *Journal of Educational Computing Research* 60, no. 6: 1436–1463. https://doi.org/10.1177/07356331211057819.

Marinus, Eva, Zoe Powell, Rosalind Thornton, Genevieve McArthur, and Stephen Crain. 2018. "Unravelling the Cognition of Coding in 3-to-6-Year Olds: The Development of an Assessment Tool and the Relation between Coding Ability and Cognitive Compiling of Syntax in Natural Language." In *Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18)*, 133–141. https://doi.org/10.1145/3230977.3230984.

Moreno-León, Jesús, Gregorio Robles, Marcos Román-González, and Juan David Rodríguez García. 2019. "Not the Same: A Text Network Analysis on Computational Thinking Definitions to Study Its Relationship with Computer Programming." *Revista Interuniversitaria de Investigación en Tecnología Educativa* 7: 26–35. https://doi.org/10.6018/riite.397151.

Mühling, Andreas, Alexander Ruf, and Peter Hubwieser. 2015. "Design and First Results of a Psychometric Test for Measuring Basic Programming Abilities." In *Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE '15)*, 2–10. https://doi.org/10.1145/2818314.2818320.

Relkin, Emily, Laura E. de Ruiter, and Marina U. Bers. 2021. "Learning to Code and The Acquisition of Computational Thinking by Young Children." *Computers & Education* 169: 104222. https://doi.org/10.1016/j.compedu.2021.104222

Repenning, Alexander. 2017. "Moving Beyond Syntax: Lessons from 20 Years of Blocks Programing in AgentSheets." *Journal of Visual Language and Sentient Systems* 3: 68–91. https://dblp.org/rec/journals/jvlc/Repenning17.

Román-González, Marcos. 2016. *Codigoalfabetización y pensamiento computacional en Educación Primaria y Secundaria: validación de un instrumento y evaluación de programas* [doctoral thesis]. Madrid: UNED. http://e-spacio.uned.es/fez/view/tesisuned:Educacion-Mroman.

Román-González, Marcos, Jesús Moreno-León, and Gregorio Robles. 2019. "Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions." In *Computational Thinking Education*, edited by Siu-Cheung Kong and Harold Abelson, 79–98. Singapore: Springer. https://doi.org/10.1007/978-981-13-6528-7_6.

Román-González, Marcos, Juan-Carlos Pérez-González, and Carmen Jiménez-Fernández. 2017. "Which Cognitive Abilities Underlie Computational Thinking?

Criterion Validity of the Computational Thinking Test." *Computers in Human Behavior* 72: 678–691. https://doi.org/10.1016/j.chb.2016.08.047.

Román-González, Marcos, Juan-Carlos Pérez-González, Jesús Moreno-León, and Gregorio Robles. 2018a. "Extending the Nomological Network of Computational Thinking with Non-Cognitive Factors." *Computers in Human Behavior* 80: 441–459. https://doi.org/10.1016/j.chb.2017.09.030.

Román-González, Marcos, Juan-Carlos Pérez-González, Jesús Moreno-León, and Gregorio Robles. 2018b. "Can Computational Talent Be Detected? Predictive Validity of the Computational Thinking Test." *International Journal of Child-Computer Interaction* 18: 47–58. https://doi.org/10.1016/j.ijcci.2018.06.004.

Shute, Valerie J., Chen Sun, and Jodi Asbell-Clarke. 2017. "Demystifying Computational Thinking." *Educational Research Review* 22: 142–158. https://doi.org/10.1016/j.edurev.2017.09.003.

Sinnott, Jan D. 1998. *The Development of Logic in Adulthood. Postformal Thought and Its Applications*. Boston, MA: Springer. https://doi.org/10.1007/978-1-4757-2911-5.

Stephens, Max, and Djordje M. Kadijevich. 2020. "Computational/Algorithmic Thinking." In *Encyclopedia of Mathematics Education*, edited by Stephen Lerman, 117–123. Cham: Springer. https://doi.org/10.1007/978-3-030-15789-0_100044.

Terroba, Marta, Juan Miguel Ribera, and Daniel Lapresa. 2020. "Pensamiento computacional en la resolución de problemas contextualizados en un cuento en Educación Infantil." *Edma 0–6: Educación Matemática en la Infancia* 9, no. 2: 73–92. http://www.edma0-6.es/index.php/edma0-6/article/view/156.

Tsarava, Katerina, Korbinian Moeller, Marcos Román-González, Jessika Golle, Luzia Leifheit, Martin V. Butz, and Manuel Ninaus. 2022. "A Cognitive Definition of Computational Thinking in Primary Education." *Computers & Education* 179: 104425. https://doi.org/10.1016/j.compedu.2021.104425.

Urlings, Corrie C., Karien M. Coppens, and Lex Borghans. 2019. "Measurement of Executive Functioning Using a Playful Robot in Kindergarten." *Computers in the Schools* 36, no. 4: 255–273. https://doi.org/10.1080/07380569.2019.1677436.

Wing, Jeannette M. 2006. "Computational Thinking." *Communications of the ACM* 49, no. 3: 33–35. https://doi.org/10.1145/1118178.1118215.

Zapata-Cáceres, María, Estefanía Martín-Barroso, and Marcos Román-González. 2020. "Computational Thinking Test for Beginners: Design and Content Validation." In *Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON '2020)*, 1905–1914. https://doi.org/10.1109/EDUCON45650.2020.9125368.

Zapata-Cáceres, María, Estefanía Martín-Barroso, and Marcos Román-González. 2021. "Computational Thinking Collaborative Game-Based Learning Environment and Assessment Tool: A Case Study in Primary School." *IEEE Transactions on Learning Technologies* 14, no. 5: 576–589. https://doi.org/10.1109/TLT.2021.3111108.