Errors of programming and ownership of the robot concept made by trainee kindergarten teachers during an induction training

María José Seckel1 · Claudia Vásquez2 · Marjorie Samuel3 · Adriana Breda4

Received: 27 May 2021 / Accepted: 6 August 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
Computational thinking in the educational environment has awaken a rising interest, having been included as part of the curricula from the very beginnings of education. Programmable robots have become a valuable positive resource in order to succeed in the development of computational thinking, demanding proper training from kindergarten teachers and trainees in order to be able to teach robotic programming. This article has the purpose to 1) identify the frequent errors made by trainee kindergarten teachers when solving a series of robotic problems in a computational thinking module, which develops in the course of Didactics of Mathematics and 2) determine the level of comprehension of the robot concept acquired by the trainees when solving robotics problems. The research developed in a qualitative methodology manner; the data used for this article were collected through the solving of five robotic problems and an open-ended question that had to be answered by each of the 25 participants individually. The results showed that, in general, the most frequent error appeared in problems in which movements were set in a Euclidean space of two dimensions. That is to say, the problems’ solving required a selection of different commands: turn, moving forward and/or backwards. Moreover, more than half of the participants who answered the open-ended question succeeded in owning the robot concept once they had solved the problems.

Keywords Trainee kindergarten teachers · Educational robotic · Computational thinking · Errors in programming

* María José Seckel
mseckel@ucsc.cl

Extended author information available on the last page of the article
1 Introduction

The Technological demand in today’s society has increased changes in public policies in several countries (Forsström & Kaufmann, 2018). Within education countries like England, Finland and Japan, among others, are pioneers in introducing computational thinking and programming (CTP) as a topic at school, understanding that today’s students will not only face in the future a society which will use the technology but also they will be involved in its development (Hitschfeld et al., 2015b; Forsström & Kaufmann, 2018). Therefore, it is imminent to step from the usage of technology into the creation of technology, which implies delivering opportunities for the growth of CTP at school levels and from the early ages. This means to develop computational or IT skills that allow students to imagine new ways of executing something or facing the real problems, by acknowledging the input of this educational discipline in the digital transformation as of the development of the twenty-first century’s skills such as: creativity, collaborative work, innovation, critical thinking, problem solving, among others.

Nowadays, in countries where there is no specific topic (in the curricula) for the improvement of CTP, it has been included in others such as science and mathematics (Barker & Ansorge, 2007). Moreover, Mondada et al. (2017) concludes that educational Robotics (ER) is a tool that allows reaching CTP’s development in mathematics classes but due to the high economic cost, CTP usage cannot be increased. This is because buying the technological resources is not enough, it is necessary to train teachers towards the design, management and evaluation of their new teaching practices within the constructivism-learning paradigm (Monsalves, 2011). Now, the investment is justified by the positive effects attributed to the use of Robots in the classroom, for example: the increase in the curiosity of learning mathematics and technology, the reduction of the existing gender gap (Sullivan & Bers, 2019) and the strengthening in solving problems, creativity and collaborative work (Roschelle et al., 2000).

In Chile, for the past twenty-five years, Enlaces (Education and Technology Center, which depends on the Education Ministry) has carried out several initiatives which seek to improve the education; mainly focusing in being supportive of learning by using Information and Communication Technology (ICT). This has ensured undoubtedly students to have access to technology, be connected and, in a lower scale, be nearer to the development of CTP in their last years of elementary level and then at high school level. Accordingly, Hitschfeld et al. (2015a), suggest that Chile has the challenge not only to promote CTP since the beginnings of scholarship in formal or informal scenarios, but also has to decrease the stereotypes linked to technology development. This being so, because there is evidence showing that from an early age this kind of task is associated to male gender (Sullivan & Bers, 2019). Moreover, the Chilean Education Ministry has implemented (in 2019) the recognized national plan of digital languages (Vásquez et al., 2019) which introduces (among other features) the use of the pedagogical robot “Bee-bot” at kindergarten (5–6 years old), giving the chance to learn programming in a linguistic context with intuitive characteristics (visual programming by groups).
Recent public policies, which promotes the development of CTP at school level in Chile, represent a challenge in the training for teachers. In that sense, this article relates to an initial approach to the training process of kindergarten teachers towards the development of the computational thinking in a course of Didactics of Mathematics. More specifically, this article’s purpose is to answer the following questions: which are the frequent errors in programming made by the trainees when solving robotics problems by using Bee-Bot Robot? what level of comprehension do the trainees reach when answering the series of robotics problems by using Bee-Bot Robot? In order to answer those questions, the following objectives were defined: 1) identify the frequent errors made by the participants (trainee kindergarten teachers) when solving a series of robotics problems and 2) determine the level of comprehension of the robot concept reached by the participants when solving a series of robotics problems.

Hereunder, the theoretical framework is described, explaining the development of the computational thinking at school, studies on educational robotics for teaching Mathematics during the early ages of the students and the training of teachers for the development of the computational thinking. Moreover, the methodologic choices, results and other considerations are also presented.

2 Theoretical framework

2.1 Development of computational thinking at school

Computational thinking, programming and robotics are concepts found in informatics that have been integrated to the educational context since they have been valuable as transcendental tools for students’ training for the society’s demands (current and future) within the technological field. Said concepts provide students with skills such as problem solving, collaborative work and creativity, among others (Paredes et al., 2015; Forsström & Kaufmann, 2018; Imai & Matsui, 2018). Wing (2006) defines computational thinking as “the solution to problems, the systems’ design and the understanding of human conduct by using the fundamental concepts of informatics” (p.33). Moreover, programming “refers to the algorithm that has been decoded into something that can be executed by a machine, therefore, it’s about producing or creating a program” (Sáez-López, 2017, p.9) which purpose is the solving of informatics problems. Furthermore, the two said concepts are operational within the framework of the called educational robotics, considered a learning complement by which writing programs and testing them is possible. In other words, the robot concept refers to “a programmable physic mechanism capable of acting in order to solve problems by interacting autonomously with its environment” (Arlegui & Pina, 2016, p.20), being the main feature of the mechanism the fact that is controlled by a program which is controlled by the user.

Brennan and Resnick (2012) distinguish three different dimensions in the computational thinking. The first dimension relates to computational concepts such as sequences, cycles, parallelisms, events, conditionals, operators and data. The second dimension relates to computational practices that develops while designers (students
at educational levels) get involved with the concepts. These practices are i.e.: incremental changes (programming as an adaptive process), tests and depuration (solving problems by testing and correcting them), reuse and persevere (finding ideas and codes in order to build solutions out of them), abstracting and modularity (build something bigger from a collection of smaller pieces). Lastly, the third dimension refers to the viewpoints of computational thinking, these being: to express (programming as a language that can be communicated through new media), connect (create with others) and question (raise questions about and with technology).

In the past few years, it has been observed that introducing computational thinking during the early years of education has become a trend in several countries. Being robotics, digital environment and unplugged activities, the core tools in order to achieve the development of computational thinking at early ages. (Bers et al., 2019; Kanaki & Kalogiannakis, 2018; Zapata-Ros, 2019). The studies in this line of investigation determine that through programming activities children improve their skills regarding computational thinking, i.e. planning, logic, algorithm sequences and depuration, requiring different levels of guidance from adults (Critten et al., 2021). Alongside, it has been observed that programming activities reinforce cognitive skills such as reasoning and work memory (Arfé et al., 2020).

2.2 Research on educational robotics for teaching mathematics at early ages

Papert (1980) was one of the first authors to raise the discussion about the close relation between computational thinking and mathematics, suggesting that robots are a resource that triggers the reasoning to solve challenging problems. In this way of thinking, several researchers have inquired about the potentials that introducing programmable robots have when teaching mathematics at early ages at school, being Bee-Bot one of the most used for children aged 5 to 7 (Forsström & Kaufmann, 2018).

The research carried out though the literature was categorized according to the Didactical Suitability construct (Breda et al., 2018; Seckel & Font, 2020). That is to say, recognizing the following thematic focus: a) epistemic, referring to mathematics; b) cognitive, referring to learning; c) emotional; referring to affective matters; d) mediational, referring to resources; e) interactional, referring to management inside the classroom; and f) environmental, referring to the surroundings. Therefore, research with an epistemic focus were found, some of them suggesting activities in order to deal with different mathematics contents (Ferrada et al., 2019; Sáez-López, 2017), and others that exposes a close relation between computational thinking and specific mathematics’ content (i.e. algebra) (Alsina & Acosta, 2018). Moreover, some research with cognitive focus were also identified, in which evidence has shown the progression of memory in mathematical works in short periods after manipulating the robot (Di lieto et al., 2017). In addition, the features considered by students in the decision-making when solving given problems were characterized as well (Diago et al., 2018a), and the difficulties experienced by children aged five and six were explored when they solved problems with the robot (Pérez & Diago, 2018). Bellas et al. (2019), in turn carried out a study whose focus was emotional,
evidencing a greater motivation in the mathematics classes when tasks were solved with the use of the robot. The fourth focus relates to the mediational theme, Diago et al. (2018b) suggest the use of the robot as an additional resource (cards) with the intention that students can perform their plan in order to give a solution to the challenge. The latter could refute one of the defects stated by some authors about Bee-Bot that is, the impossibility to debug (erase or add specific orders) in order to solve errors (Sáez-López, 2017). From the environmental point of view, Elkin et al. (2014) evidenced the feasibility to use this kind of robot at educational environments along with Montessori methodology.

Lastly, from the interactional focus, what were identified as highly predominant (in the above-mentioned research) is the way of working collaboratively with the robot whenever mathematics challenges arises. This gives place to interactive spaces, which suggest the need to go further into the study of research that explore the ability to communicate and produce mathematical arguments.

2.3 Teachers’ training for the development of computational thinking

In order to integrate the computational thinking at schools, it is fundamental to train teachers to assume the challenge. In this sense, Estebanell et al. (2018) suggests a model for training trainees by which he describes four levels of CT’s development: first, user level: intends that teachers raise questions regarding how to use certain computational thinking language when approaching problems with a robot, videogame, app, among others. Then, thoughtful user level, aims teachers to be thoughtful about what they have done when computational problems developed. Next, teacher level, expects teachers to face the challenge of deciding what to teach, what they expect their students will learn from computational thinking and what resources and strategies they will apply. Finally, thoughtful teacher level, means teachers be thoughtful about the teaching and learning process related to computational thinking. Moreover, this model submits the proposition that the training path can be considered when moving through levels, and in order to go further into each level of training.

In other sense, as far as the use of robots for the development of computational thinking at school level is concerned, it is necessary that teachers recognize two basic features when planning tasks (Arlegui & Pina, 2016; Moreira et al., 2018; Gusmão & Font, 2020). The first feature refers to the idea of the robotic problem in which the resolution implies the robot goes from an initial state into a final one, through a planned action sequence (programmable intermediate states). The second feature is about the idea of solving challenging problems (Ponte, 2014 which leads to consider the following criteria in the design of a problem or sequence of problems: 1) be of progressive complexity, 2) to refer to known and unknown features and 3) place the problem within its context (scenario).

Although the revision of the literature reveals certain orientations on the essential aspects for the training of teachers in the field of computational thinking, the empiric research on the teachers’ development in this area is limited. For that reason, Kong et al. (2020) propose the analysis of a program in the development of teachers
referring to the computational thinking, obtaining positive results. In said program, they suggest two modules of training: 1) fundamentals concepts of computational thinking and teaching strategies and 2) advanced concepts of computational thinking with practical experiences and observations on the practices, being the latter the one that obtained the best results. Another reference in the trainees’ training field, is given by Papadakis and Kalogiannakis (2019) they point out the learning based on projects i.e. an adequate strategy as an introduction in programming. Moreover, the authors suggest that the training programs should include programming activities with robots, scratch and unplugged.

3 Methodology

This article was carried out through a qualitative methodology in order to understand, in a global perspective (Ruiz, 2012), the answers obtained from the trainees when solving five robotic problem and an open-ended question given at level 1 of the path of their training module (user level according to Estebanell et al., 2018). During the training process, the trainees worked with the BeeBot robot being it the resource introduced by the public educational system in Chile for the development of the computational thinking during the first years at school.

3.1 Context and participants

This article takes place at a Chilean university were kindergarten teachers attend their induction training. Twenty-five trainees participated and they were attending the course of Didactics of Mathematics at their fifth semester of their curricular path (eight semesters in their training).

The designed training module comprehends 8 sessions in total of 1 h each. In the first two sessions, the participants performed level 1 tasks (user) and level 2 (thoughtful user) accordingly (individually). In the following sessions, they performed level 3 tasks (teacher) and level 4 (thoughtful teacher), for which they teamed up. It is important to mention that due to the COVID-19 context, the sessions were adapted for a virtual implementation in a synchronous and asynchronous learning method (Breda et al., 2020).

Given the purpose of this article and the adaptation made due to the sanitary state of emergency, in the following title the resources for collecting the data and the methodology used are explained in further detail.

3.2 Data collection

The data presented in this article were collected during the first session of work with the participants (user level), prior signing an informed consent authorization. Before the collection of data, during a synchronous work, the trainees were part of an exploration activity that consisted of: 1) watching a brief video in which Bee-Bot robot and its programming functions were presented; 2) answering to a given robotic
problem by planning the robot programming and 3) watching a video of the possible solutions and possible errors. This last video corresponds to one of the adaptations introduced in the training model because the session was virtual the trainees were not able to operate the robot nor programming their planning. In this sense, the exploration activity pretended to introduce the recognition of Bee-Bot programming tools so that afterwards the trainees could solve other robotic problems, which will be described further on.

In Fig. 1 the mat (scenario) used during the exploration activity is presented, also the cards with which the programming should be planned and the robot of both videos.

The exploration activity consisted of answering two questions: 1) how should the robot be programmed so it can move from the firefighter’s headquarters to the house in fire?; and 2) how should the robot be programmed so it can move from the building to the hospital? In order to answer these questions, the participants were given some time to plan the programming (with the cards) and later on watched a video with the possible right answers and they were supposed to recognize errors, the participants were given a moment for discussion of the robot’s programming tools.

By the end of the synchronous work of the first session the activities which would allow the collection of the data for this article were introduced. The activities referred to five robotic problems that had to be solved individually during the asynchronous worktime. Said solving consisted of planning the robot’s programming with the cards presented in Fig. 1, take pictures of their answers and send them via a virtual platform. Moreover, once the participants gave their solutions to the problems, they had to answer the following open-ended question: what is the computational language of Bee-Bot robot and how did you use it?

It is important to mention that the five presented robotic problems, suggested by Di Lieto et al. (2017), take into account a level of progressive complexity that is determined from a linguistic programming context which allows to give an answer to the problem (Arlegui & Pina, 2016). On the one hand, the use of such programming language is considered the unknown aspect of the problem; while on the other hand, the knowledge of the position and movement in the space as well as the quantification are considered as the known aspects. Furthermore, the resolution of
the problems was subject to two conditions: 1) the robot initiates its path from the “start” box, in the position observed in each mat and 2) in problems 1, 2 and 3, the robot cannot be lifted from the mat and be relocated in another position in order to program the second statement.

Hereunder is Table 1, which presents each problem with the possible correct answers (each answer with its code respectively). These answers are a reference to analyze the data collected taking into account that each participant had to send one reply to each problem.

Table 1 shows the answers that allow solving the problems presented in the correct way. Taking into account this information, if the participant’s answers are different from the ones included in the above table such answers are wrong in the general analysis. Afterwards, the errors are categorized according to a type of error. The data analysis is explained in detail later on.

| Table 1 Robotic problems and possible correct answers |
|------------------------------------------------------|
| **Problem 1**                                        |
| 1. a. Program the robot so it can reach the blue dot. |
| 1. b. Program the robot so it can move from the blue dot to the starting point. |
| **Expected answers**                                  |
| Code: P1-1                                          | Code: P1-2                                      | Code: P1-3                                      |
| ![Image](image1.png)                                  | ![Image](image2.png)                            | ![Image](image3.png)                            |

| **Problem 2**                                        |
| 2.a. Program the robot so it can reach the blue dot. |
| 2.b. Program the robot so it can move from the blue dot to the starting point. |
| **Expected answers**                                  |
| Code: P2-1                                          | Code: P2-2                                      | Code: P2-3                                      |
| ![Image](image4.png)                                  | ![Image](image5.png)                            | ![Image](image6.png)                            |

Code: P2-4                                                                 Code: P2-5
![Image](image7.png)                                                ![Image](image8.png)
Table 1 (continued)

### Problem 3

3. a. Program the robot so it can reach the blue dot.
3. b. Program the robot so it can move from the blue dot to the starting point.

### Expected answers

| Code: P3-1 | Code: P3-2 | Code: P3-3 |
|------------|------------|------------|
| ![Image](code_p3_1.png) | ![Image](code_p3_2.png) | ![Image](code_p3_3.png) |

| Code: P3-4 | Code: P3-5 |
|------------|------------|
| ![Image](code_p3_4.png) | ![Image](code_p3_5.png) |

### Problem 4

Program the robot so it can reach the blue dot, making a 2 seconds pause in the yellow dot.

### Expected answers

| Code: P4 |
|---------|
| ![Image](code_p4.png) |

### Problem 5

Program the robot so it can reach the blue dot without stepping into the triangles.

### Expected answers

| Code: P5-1 | Code: P5-2 |
|------------|------------|
| ![Image](code_p5_1.png) | ![Image](code_p5_2.png) |

| Code: P5-3 | Code: P5-5 |
|------------|------------|
| ![Image](code_p5_3.png) | ![Image](code_p5_5.png) |

| Code: P5-4 | Code: P5-6 |
|------------|------------|
| ![Image](code_p5_4.png) | ![Image](code_p5_6.png) |

Source: Based on suggestions of Di Lieto et al. (2017)
Problems 1 and 4 are problems that require the movement programming in a Euclidean space of one dimension, while problems 2, 3 and 5 require the movement programming in two dimensions. In this sense, problem 4 was more difficult than the other problems; since its programming required the addition of a command that was not tested during the exploration activity because its function was only explained through the robot’s introduction video.

### 3.3 Data analysis

The analysis was carried out in three stages. The first stage consisted in analyzing the answer of each participant considering the following three categories:

- **Correct answer**: planning the programming allows the robot to reach the final stage as expected.
- **Answer partially correct**: the planning of the programming is close enough to the expected one but presents minimum errors related to the wrong use of the planning cards, i.e. inverting the card that stands for the command “forward” in order to try to go “backward”.
- **Wrong answer**: corresponds to the answers that represent an error in planning the programming, and which do not allow the robot to reach the final stage as expected (Saéz-López, 2019).

During the second stage, the data collected were analyzed in order to identify the frequent errors that appeared in each of the problems. These errors arise from the data collected:

- **Error due to the lack of a programming command (E1)**: not programming one or more commands in order to give an answer to a statement, or previous programming is not deleted in order to solve a second statement.
- **Error due to excessive quantification in the programming of a command (E2)**: the user does not understands the relation between the mat characteristic and the robot’s movements. That is to say, the user sets the robot from an estimation without considering the mat’s measures and the 15 cm the robot moves (the equivalent of a square in the mat).
- **Error due to misunderstanding a command (E3)**: the user does not understands that the robot’s turns is of 90° or else thinks the robot turns and moves forward at the same time. Moreover, the user does not considers that the programming of each pause last one second.
- **Error when applying previous knowledge (E4)**: the user presents difficulties when applying its knowledge of spatial orientation, not using the cards correctly.

From the above-mentioned definitions, it is important to comment that during the process of analyzing the results, it was possible to identify one or more types of errors in each problem.
Finally, during the third stage and with the intention to determine if solving robotics problems leads to the ownership of the concept of robot, we applied the technique of content analysis (Cáceres, 2003). Therefore, the definition of robot suggested by Arlegui and Pina (2016) was established for content segment individualization (analysis units), phrases or paragraphs that can be recognized as signs of ownership of the concept of robot. In this manner, from the data analysis the following three categories appeared:

- **Advanced comprehension of the concept of robot:** describes in detail the programming language used and recognizes the autonomous character of the robot and the user’s role as the programming controller.
- **Intermediate comprehension of the concept of robot:** describes in detail the programming language used but not recognizes the autonomous character of the robot and the user’s role as programming controller.
- **Low comprehension of the concept of robot:** the programming language description is general and not indicates recognition of the autonomous character of the robot nor the user’s role as programming controller.

### 4 Results

In the following section, the results obtained are presented in the first place from a global perspective, showing the percentages of correct, partially correct and incorrect answers to each problem. In addition, the results of the participants in the resolution of the five problems are also presented. Secondly, frequent errors identified in each problem given are as well described. Finally, the results obtained in relation to the ownership of the robot concept according to the analysis of the answers given by the participants to the open-ended question at the end of the asynchronous activity are mentioned.

All of the 25 participants answered the problems presented, but 22 participants answered the open-ended question.

#### 4.1 General data analysis

The data from Table 2 shows that there is a high percentage of correct answers to each problem, recognizing the programming planning as one of the most used one by the participants. The following programming by code were identified: in problem 1 code P1–1; in problem 2 codes P2–1 and P2–5, in problem 3 codes P3–1 and P3–5, and in problem 5 code P5–1.

Moreover, it is observed that the higher percentage of partially correct or incorrect answers are found in problems 3 and 5, whose resolution implies the movement programming in a Euclidean space of two dimensions. In particular, problem 5 implies as well the decision-making regarding the movement path so that the robot can reach the final stage, avoiding obstacles found in the programming scenario.
On the other hand, as it can be observed in Fig. 2, only a 24% of the participants resolved in a correct way all the 5 given problems, meaning six participants. In this sense, it can be observed that a 76% had some kind of error at least in one of the problems.

Furthermore, the kind of errors identified in each problem are presented, being A1, A2, A3… A25 the answers given by the participants.

### 4.2 Frequent errors identification

#### 4.2.1 Problem 1

As it was mentioned before, problem 1 is presented by means of a simple scenario that requires the movement planning in a Euclidean space of one dimension, therefore the answer to the problem implies the selection of commands “forward”, “backwards”, “go” and “delete”.

From the analysis of the data collected, we detected three types of errors: E1, E2 and E4. E1 was the most frequent one, being identified in the programming planning of five of the participants (nine participants made a mistake when solving this problem). In Fig. 3, it can be observed the answer of participant A13 who recognizes the absence of the “delete” function (E1) to program the second statement.

![Fig. 2 Correct problems obtained by the participants](image)
The error observed in the figure, with the absence of the “delete” function, repeats in five participants.

4.2.2 Problem 2

In the same way as the previous problem, problem 2 requires a movement planning in a Euclidean space, but this time in two dimensions. Therefore, the answer to the problem consists of selecting commands “forward”, “backwards”, “turn”, “go” and “delete”, containing the complex movements plus turns in both problem statements.

In this problem, three types of errors were identified: E1, E3 and E4. Out of the three, the most frequent error was E4, appearing in 5 out of 10 participants. In Fig. 4, it can be observed the planning programming of two of the participants, (A25 and A8) were it is clear the difficulty in the use of the command “backwards” when planning the programming (E4).
As can be observed in the figure, this error appears from the resolution of the second statement, which repeats in all the five participants who made the same error. In the first case, A25, it can be inferred that the participant uses the card “forward” in order to go backwards. In the second case, A8, two interpretations can be made: 1) difficulty in planning the command “forward” misunderstanding it with the command “backwards”; 2) difficulty in the planning of the function “turn right” confusing it with function “turn left”.

In the other cases, it repeats the planning programming observed in A8.

4.2.3 Problem 3

Problem 3 likewise the previous problem, involves two dimension movements and its expected answers includes the selection of commands “forward”, “backwards”, “turn”, “go” and “delete”. In this scenario, the errors identified are type E1, E2, E3 and E4; the last two were the most frequent ones. These errors are identified together in the same planning programming (in three cases) or separately (three cases with E3 and two cases with E4). In Fig. 5, it can be observed examples of these identified errors in participants A15, A1 and A25.

In planning the programming, participant A15 makes error E3 in the first and second statement. In the first statement, it can be inferred that the participant does not understands that the turning does not implies movement, since the participant does not consider in the programming sequence the command “forward” after command “turn right”. The same interpretation is seen in the second statement, when in the sequence does not considers the command “backwards” before “turn left”.

In the case of participant A1 and A25, the errors appeared when solving the second statement. Participant A1 makes error E4, since it was evident a difficulty in the

---

**Fig. 5** Examples of frequent errors identified in problem 3
4.2.4 Problem 4

Problem 4, as well as problem 1, consists of one-dimension movements and its expected answer requires the selection of the following commands “forward”, “pause” and “go. In this scenario, the identified problems are type E1, E2, and E3. Error 3 is the most frequent one, appearing in three cases. In Fig. 6 it is shown the example of this programming common error made by participant A9.

In the example it can be seen the representation of the planning proposed by all the participants who made this error. In the case of interpreting error E3 made by participant A9, it can be observed that the sequence of programming only used just one time the command “pause”, which means that the robot stops for a second instead of two as it is requested in the problem’s statement.

4.2.5 Problem 5

Problem 5, once more, implies movements in two dimensions and unlike the other problems, the answer implies taking a decision regarding the movement path that the robot will follow in order to reach the final state. In this scenario, the identified errors are type E2, E3 and E4; being E4 the most frequent one (see Fig. 7) because it appeared in six out of nine participants who wrongly planned the programming.

The example presented in Fig. 7, refers to the error E4 made by the participant since it can be interpreted the difficulty in the use of the command “turn”. In this case (A8), the observed error is found in the seventh card of the sequence (left to right), where the participant selects the command “turn left” instead of
“turn right”. This error is also present in other cases, i.e. wrongly selecting “turn right” or “turn left” according to the planned path when answering the problem.

4.3 Ownership of the robot concept

The data collected indicated that three of the twenty-five participants did not answer the open-ended question at the end of the asynchronous activity. Along with this fact, it was evidenced that twelve participants have reached an advanced comprehension of the robot concept (55% of the participants who answered the question), four reached an intermediate comprehension (18%) and six showed a low comprehension of the concept. Furthermore, some evidence of the participants’ answers in relation to each category will be described.

Advanced comprehension of the concept of robot:

These instructions, which are spatial concepts, I codified them in order to make a programming that changed accordingly to where I intended to reach with my robot in the mat. (A3).
The bee-bot contains directional keys that are used to enter several commands that will control Bee-bot’s movements. (A7).
The computational language of bee-bot is made of buttons that must be pushed in order to obtain an answer from the robot. (A12).
One has to program the robot in order to make it move accordingly to the programming made. (A14).

Intermediate comprehension of the concept of robot:

Is a language that uses the concepts of forward, backwards, left, right, go (start), pause (stop) and delete. (A9).
Is a language with spatial orientations that have different options to use, for example: forward, backwards, turn left and right, among others. (A19).
Is a language through cards in which there is a bee with different movements (forward, backwards, turn left and right) beside other cards, which indicates: pause and delete, and go card. (A24).

Low comprehension of the concept of robot:

I personally used it taking as reference the arrows in order to guide me in the process. (A2).
Its language is easy when solving a children's problem. I used the computational thinking taking as reference the cards and videos showed in class refereeing to the explanation of it. (A11).
I consider it is easy to use and adaptable to do different movement activities; I used as references the plans handed out in the PPT in order to localize and programming basing in bee-bot language. (A21).
5 Final considerations

The results obtained in this article revealed that the participants have reached a high percentage of correct answers in each problem (more than 50%). Nevertheless, it was also evidenced a high percentage of participants making mistakes (76%) at least in one of the problems solved. In this sense, in problem 1 the identified errors were type E1, E2 and E4, being E1 the most frequent one (not programming one or more commands in order to provide an answer to the statement; or not deleting the programming to solve the second statement). In problem 2, the identified errors were type E1, E3 and E4; being E4 the most frequent, (the user presents difficulties to apply the knowledge on spatial orientation, doing a wrong use of the cards). In the case of problem 3, the identified errors were type E1, E2, E3 and E4. E3 was one of the most frequent (the user does not understand the robot’s 90° turn, or thinks that it turns and moves at the same time; also, does not take into account that the programming of each pause corresponds to one second) as well as E4. As for the types of errors found in problem 4, there were recognized E1, E2 and E3, being the most frequent E3. Finally, in problem 5, errors type E2, E3 and E4 appeared, being the most frequent E4. Having said this, it can be perceived that errors E1 and E4 are the most repeated ones, because E1 is present in four out of five problems (1, 2, 3 and 4) while E4 is present in three out of five problems (2, 3 and 5).

When reading the data collected, it must be considered that the path of the resolution of the problems the trainees were exposed to agrees with an experience that was characterized by a first approach to this kind of resource, which allowed them to apply their spatial knowledge from a new language perspective (programming). In other words, the purpose of the problems presented was to build knowledge in the use of the robot and therefore, face in the future new problems were the functioning of the robot would be a known aspect of the problem, while the unknown aspect will be regarding new knowledge i.e. mathematical knowledge (Ferrada et al., 2019).

The usage of cards has been recommended for the robot’s programming planning (Diago et al. (2018b) and the cards answer some critics referring to the depuration as a computational practice of Bee-bot (Sáez-López, 2017). Nevertheless, the detection of error E4 repeatedly in problems 2, 3 and 5 shows the need of a more profound ownership as well as a relation with each of the movement commands suggested by the cards, in such a way that the usage of this resource becomes actually effective.

In this sense, the results of this article suggest the implementation of previous tasks, with lower complexity, for a better ownership of the turn commands. Tasks that can be designed following the distinction between exercise and problem as proposed by Arlegui and Pina (2016). The exercise is a type of assignment which purpose is to show what does a command do, therefore, the exercise is stated and solved by the professor, asking the students what do they expect from the command and then showing what does a command or sequence of commands do (i.e.: turn, turn and move forward or turn and move backwards).

On the other hand, the concept of problem presented in this article is aligned with the one suggested by the above-mentioned authors. However, it is important
to mention the need of working in an environment that consists of two-dimension movements, but suggesting as well “turn problems” (Arlegui & Pina, p.236) and then “movement + turn” (Arlegui & Pina, p. 238) departing from a sequence of problems for each case and contemplating simple statements.

As for the ownership of the concept of robot is concerned, it was observed that a high percentage of the participants (55%) have reached an advanced comprehension by the end of the asynchronous activity. Furthermore, the answers from this group of trainees have become a useful learning resource for the teachers because it allows institutionalizing the knowledge for the use of the robot (programming language) in the feedback processes (Arlegui & Pina, 2016).

The innovative aspect of this research lies on the exploration of the programming planning developed by the trainees, who interacted for the first time with Bee-Bot robot. This type of research has been underexplored because the focus of interest has always been on the knowledge acquired during training programs (Papadakis & Kalogiannakis, 2019) and not on the difficulties detected at the beginning of the programming training. Moreover, the results collected are useful for designing assignments which include the use of Bee-Bot or a similar language programming robot for elementary or kindergarten trainees during their induction trainings. From the data collected it is possible to design reverse tasks that enable the recognition of programming problems, that is to say, deliver a programming planning that pretends to give answers to a problem so that students can recognize and describe the error or errors that prevents the robot from reaching its final state.

In the program researched, the BeeBot robot was selected as a tool for the induction of programming for the kindergarten trainees because it has been specially designed to introduce through programming language by stages the concepts of computational thinking and programming in children’s early years at school. Even though the data collected, are a preview in the environment of induction trainings, further articles could be enlarged and explore the difficulties on programming when solving unplugged problems or in digital environments such as Scratch. Furthermore, in a following study it could be presented the same problem to the same participant who made programming errors with the purpose of verifying if the error persist. In that case, it would be interesting to interview the participant in order to learn the error’s route.

Finally, another research alternative would be to analyze the subsequent stages of the programming induction where the trainees design learning experiences to be put into practice through lessons simulations, combining mathematical and computational thinking. Specifically it would be suggested to identify the programming errors made by their students and how such errors are managed in the lesson development.

**Funding** This article was supported by the Fondecyt research project N°11190547 and PGC2018–098603-B-I00 (MCIU/AEI/FEDER, EU).
Hitschfeld, N., Pérez, J. & Simmonds, J. (2015b). Pensamiento computacional y programación a nivel escolar en Chile: el valor de formar a los innovadores tecnológicos del futuro [Computational thinking and programming at school in Chile: the worth of mentoring future technological innovators]. Revista Bits de Ciencia [Science Bits Magazine], 12, 28–33.

Imai, S., & Matsui, H. (2018). Practice of Control Education by Experiment using Robot. Journal of Robotics, Networking and Artificial Life, 5(3), 190–193. https://doi.org/10.2991/jrnal.2018.5.3.10

Kanaki, K., & Kalogiannakis, M. (2018). Introducing fundamental object-oriented programming concepts in preschool education within the context of physical science courses. Education and Information Technologies, 23, 2673–2698. https://doi.org/10.1007/s10639-018-9736-0

Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. Computers & Education, 151, 103,872. https://doi.org/10.1016/j.compedu.2020.103872

Mondada, F., Bonani, M., Riedo, F., Briod, M., Pereyre, L., Retornaz, P., & Magenrat, S. (2017). Bringing Robotics to formal education: The Thymio Open-Source Hardware Robot. IEEE Robotics & Automation Magazine, 24(1), 77–85. https://doi.org/10.1109/mra.2016.2636372

Monsalves, S. (2011). Estudio sobre la utilidad de la robótica educativa desde la perspectiva del docente [Study on the utility of educational robotics from the teacher’s perspective]. Revista de Pedagogía [Educational Magazine], 32(90), 81–117.

Moreira, C. B., Gusmão, T. C. R. S., y Font, V. (2018). Tareas Matemáticas para o Desenvolvimento da Percepção na Educação Infantil: potencialidades e limites. Bolema, 32(60), 231–254.

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. International Journal of Technology Enhanced Learning, 11(3), 100,478. https://doi.org/10.1504/IJTEL.2019.100478

Papert, S. (1980). Mindstorms: Children, computers and powerful ideas. Harvester Press.

Paredes, J., Guíñier, M., & Rubia, B. (2015). La innovación y la tecnología educativa como base de la formación inicial del profesorado para la renovación de la enseñanza [Innovation and technology as the foundations of teacher’s training for the renewal of teaching]. RELATEC. Revista latinoamericana de Tecnología Educativa [Educational Technology Latin American Magazine], 14(1), 101–114.

Pérez, G. y Diago, P. (2018). Estudio exploratorio sobre lenguajes simbólicos de programación en tareas de resolución de problemas con Bee-bot [Exploratory study on programming symbolic languages in solving problems with Bee-bot]. Magister, 30 (1 y 2), 9–20.

Ponte, J. P. (2014). Tarefas no ensino e na aprendizagem da Matemática. En Ponte, J. P. (Org.). Práticas Profissionais dos Professores de Matemática (pp. 13–27). : Instituto de Educação da Universidade de Lisboa.

Roschelle, J. M., Pea, R. D., Hoadley, C. M., Gordin, D. N., & Means, B. M. (2000). Changing how and what children learn in school with computer-based technologies. Future Child, 10(2), 76–101.

Ruiz, J. I. (2012). Metodología de la investigación cualitativa [Qualitative research methodology] (5th ed.). Universidad de Deusto.

Sáez-López, J. (2017). “Robots educativos y programación por bloques en educación infantil y primaria: propuestas con Bee-Bot y mBot” [Educational robots and programming by groups at preschool and primary level proposed by Bee-Bot and mBot]. In R. Cózar, & M. Del Valle de Moya (Eds). Entornos humanos digitalizados: experiencias tic en escenarios educativos [Digital human environments: CTP experiences in educational scenarios] (pp. 35–48). Editorial Síntesis.

Sáez-López, J. (2019). Programación y robótica en educación infantil, primaria y secundaria. España: UNED Seckel, M. J. & Font, V. (2020). Competencia reflexiva en formadores del profesorado de matemática. Magis, Revista Internacional de Investigación en Educación, 12(25), 127–144. https://doi.org/10.11144/Javeriana.m12-25.crpf

Sullivan, A., & Bers, M. U. (2019). Investigating the use of robotics to increase girls’ interest in engineering during early elementary school. International Journal of Technology and Design Education, 29(5), 1033–1051.

Vásquez, E., Bottamedi, J. & Brizuela, ML. 2019. Pensamiento computacional en el aula: el desafío en los sistemas educativos de Latinoamérica [Computational thinking in the classroom: the challenge for the educational systems in Latin America]. Revista Interuniversitaria de Investigación en Tecnología Educativa [Inter University Magazine in Educational Technology Resarch], 7, 36–47.

Wing, J. (2006). Computational Thinking: It represents a universally applicable attitude and skill set everyone’s, not just computer scientists, would be eager to learn and use. Communications of the ACM, 49(3), 33–35.

Zapata-Ros, M. (2019). Computational Thinking Unplugged. Education in the Knowledge Society, 20, 18–29. https://doi.org/10.14201/eks2019_20_a18

Springer
Publisher’s note  Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

María José Seckel⁠¹ • Claudia Vásquez⁠² • Marjorie Samuel⁠³ • Adriana Breda⁠⁴

¹ Departament of Didactics, Universidad Católica de la Santísima Concepción, Concepción, Chile
² Departament of Mathematics, Pontificia Universidad Católica de Chile, Villarrica, Chile
³ Education Faculty, Universidad Autónoma de Chile, Talca, Chile
⁴ Departament of Linguistic and Literary Education, and Teaching and Learning of Experimental Sciences and Mathematics, Universitat de Barcelona, Barcelona, Spain