From Stories to Concurrency: How Children Can Play with Formal Methods*

Antonio Cerone

Department of Computer Science, Nazarbayev University, Nur-Sultan, Kazakhstan
`antonio.cerone@nu.edu.kz`
`https://cs-sst.github.io/faculty/cerone`

Abstract. This position paper presents an unplugged, problem-solving-based approach for teaching computer science to children. Our approach is based on story telling, where each story consists of parallel parts, and aims at developing children's observation and reasoning skills. The aim is to understand the global plot by identifying the interaction occurring among different characters in terms synchronisation, collaboration and information sharing. In this sense we focus on concurrency, a very challenging computer science area, to show that children aged 7-14 can be exposed to real-life instantiations of a number of computer science concepts, understand them and even apply them in modelling and analysis contexts.

Keywords: Children Education; Problem-solving; Computer Science; Concurrency; Finite State Machines.

1 Introduction

Computer science lectures tend to point out that their first year students have poor mathematical skills. It is important to stress that mathematics should not be confused with elementary arithmetics and mathematical skills should not be confused with the ability to quickly perform complex calculations mentally [7]. This ability was portrayed by the idiot-savant protagonist of the famous film "The Rain Man", who was an autistic person rather than a genius of mathematics.

In fact, when lecturers describe their students' poor mathematical skills, they do not refer to mere calculation skills, but to the large amount of reasoning skills that enable us to solve general problems, within and outside the domain of mathematics [6]. An essential, though non exhaustive list of such skills is:

- be able to abstract away from irrelevant details (*abstraction*) and model the reality in a symbolic/visual way (*modelling*), not necessarily on paper but even just mentally;

---

* Work partly funded by Seed Funding Grant, Project SFG 1447 "Formal Analysis and Verification of Accidents", University of Geneve, Switzerland.

- understand the difference between a visual description (a sort of *formal syntax*) and its *semantics*;
- find similarities between a problem with a known solution and a new, unsolved problem (*analogy*);
- reduce a complex problem to a smaller, easier one (*divide et impera*, leading to the concept of *recursion*);
- reason top-down from a model to a more articulated, efficient solution (*refinement*);
- reason bottom-up from observed specific cases and pattens to a general law (*induction* or *generalisation*);
- compose components, possibly modifying them to achieve *compositionality*;
- move from causes to consequences (*deduction*);
- distinguish between efficient and inefficient solutions (*complexity*);
- understand the difference between a solution of a problem and the *proof* that such a solution is *correct*.

A number of these skills will be extensively addressed in this papers.

Given that computers have been heavily introduced in schools with the expectation of having a positive impact on the students' computer science skills, the fact that this expectation has not been met at all, as observed by many university lecturers, sounds like a paradox [6]. In many schools computer science has even been introduced as a new, stand-alone subject. However, there are two fundamental problems in this innovative process:

1. computer science is often seen as a "service subject", namely to provide tools that facilitate the students in carrying out their homework and class projects and are supposed to enhance their learning;
2. computer science is normally seen as intrinsically tied to the use of computers.

We do not consider here the most extreme situation in which computer science is taught, either as part of another subject or as a stand-alone subject, by unwilling teachers who did not undergo a proper training.

As a consequence of Problem 1, the teaching of computer science tends to focus on using office-oriented tools to write documents, prepare presentations, organise data in spreadsheets. There is no need here to mention the names of the most taught tools.

In this sense the relation of the subject "computer science" with other subjects is only in one direction. It is, in fact, the other direction, the one normally neglected, that should be taken. Computer science should build on other school subjects, which, in the world of the school pupil, represent the most natural reality to be modelled formally. Obviously, mathematics should be the first subject to provide materials to manipulate in a computer science fashion. However, all other subjects also have plenty of materials on which students may carry out modelling and analysis.

Problem 2 created the misconception of a computer scientist as a programmer. This attitude also contributed to create the belief that computer science does not require mathematical skills. As a consequence, high school students who are not skilled in mathematics and are aware of this, are still confident to pursue a computer science degree.

In this paper we adopt an "unplugged approach" to teaching computer science [1, 2], presenting activities that foster children reasoning and do not require the use of a computer. In fact, we show how, starting from what at first sight appears as a purely literary and linguistic exercise, namely story telling, children carry out observations and perform reasoning leading to the acquisition of important aspects of concurrency, develop a formal model of the story, which allows them to solve the puzzle embedded in the story, and are even enabled to describe the reasoning process that led to the solution as a formal proof. We consider children between 7 and 14 years old and propose slightly different approaches for the two age groups, 7–10, approximately corresponding to primary school, and 11–14, approximately corresponding to middle school.

It is pointless to just provide children with the definitions of new notions, concepts and processes, such as algorithms, and hope they understand them, remember them and are then able to apply them to practical situations. Children learn best if they are actively involved in the process through problem-solving [14]. This is the main idea of *constructivism*, which suggests that humans construct knowledge and meaning from their experiences [4, 5]. And to engage children, experiences have to be fun and challenging. Learning from experience means that children have to discover an algorithm, starting from a game, and progressively perfect their discovery through further experiences in a fashion similar to *iterative refinement*.

However, in mathematics, it is essential to find appropriate challenges for the age and cognitive development of the child to avoid loss of interest or even frustration, with the consequent end of the fun. Mathematical puzzles are in general very motivating, but may have the drawback to degenerate in frustration. It is also important to make sure that the higher the effort needed to solve a puzzle, the greater the learning outcome.

Moreover, although some form of competition is necessary for keeping the children involved, the competition must be in the game itself, not in the mathematical skill the game addresses. It is fundamental that the competition does not appear to children as an assessment of their skills. There are plenty of evidences in psychology research that assessments increase anxiety in children and hinder their learning [12]. One way to overcome this problem is to organise competitions among teams rather than between individuals, with a balanced composition of the teams and a frequent mixing of the members, and trying to make the winning objective as much as possible distinct from the winning skills.

1.1 Playing with Concurrency

Concurrency [9–11] is a challenging topic even for postgraduate students. Understanding the global behaviour of two or more synchronising processes may not be intuitive also for simple models. It is therefore very important to develop the skills that allow us to visualise and then understand and model concurrent behaviour.

Children are very much interested in complex stories acted by many characters, who have their own personal stories, but also interact and synchronise on specific situations, collaborate to solve mysteries and fight together to defeat antagonists. The fact that children can follow and enjoy such articulated stories is evidence that they can make sense of the composition of individual stories and understand the resultant global plot. They can, therefore, visualise and understand concurrent behaviour. We aim at exploiting these abilities and prompt children with questions that enable them to reason about the composition of parallel stories, to understand how two or more characters may agree or collaborate, by sharing information and sychronising their decisions and behaviour in specific situations. In this way children can become aware of important aspects of concurrency.

Many children can naturally solve complex problems but, when they are asked about how they achieved the solution, they normally have difficulties in explaining their reasoning process. Formal methods [15] provide rigorous ways of describing problems normally occurring in computer science contexts and they are also very effective in making the reasoning processes that lead to the discovery of the solutions explicit. Among the computer science sub-disciplines, concurrency is probably the one that relies most on formal methods.

In our work we are inspired by the *Choose Your Own Adventure* series of children's gamebooks. The series was based upon a concept created by Edward Packard, who published the first book of the series in 1976 [13]. In Packard's books each story is written from a second-person point of view, with the reader assuming the role of the protagonist and making choices that determine the character's behaviour and the plot outcome. In our approach we consider two parallel stories, whose protagonists are the reader and a friend. We believe that the use of the second-person point of view in both the person's and friend's stories increases the involvement of the child, fosters the expression of personal opinions and the realistic making of informed decisions.

1.2 Structure of the Paper

In Section 2 we start from the example of a story featuring one single character, who can choose among various alternatives. We then add a parallel story, whose single character is a friend of the character of the previous story (Section 2.1), and we make the children reason about the knowledge acquired by the two characters, how it may affect their decisions and how combining what the two characters know allows us to predict the outcome of each decision.

In Section 3 we introduced interactions between the protagonists of the two stories and guide the children to reason about the impact of such interactions on the global plot. We also make important considerations on the teacher's role and the learning outcome of this exercise (Section 3.1).

In Section 4 we investigate how to enable children to visually model stories in a sort of formal way, enabling children, especially the ones in the age group

10–14, to get familiar at an intuitive and visual way with some important aspects of the modelling and analysis of concurrent systems (Sections 4.1–4.3).

In Section 5, we show how children can be guided to use the models they developed to find solutions of a problem and, most important, to prove the correctness of such solutions. Section 6 concludes the paper.

2 Choose Your Own Adventure

Let us consider the following story.

> You are looking for a treasure hidden in an abandoned castle. You enter the castle and you have in front of you a long corridor with many windows on the right side. At the end of the corridor there is a large door guarded by two parrots on their tripods. They both speak but you understand only the one on the left. The other parrot speaks a language unknown to you. The parrot on the left tells you that behind the large door there is a wide room with three small doors of different colours: the green and blue doors are not locked and you can open them and go through; the red door is locked and you do not have the key. The parrot also tells you that one of the two unlocked doors safely leads to the treasure and that if you go through the other unlocked door you will certainly die without finding the treasure.

After the story is presented to the class, children are then asked a number of questions, such as:

1. Which are your possible choices?
2. Which of these choices will certainly lead you to death?
3. Which are your reasonable choices?
4. Which choice would you make?

The questions are put to the entire class through a discussion session that aims at unfolding the logic of the story and understanding which decisions are favourable to the protagonist among the set of possible decisions. New, unplanned questions are likely to be raised during the discussion.

For children between 7 and 10 years old it is important to give a multi-disciplinary flavour to the discussion by considering also literary and linguistic aspect of the story. In fact, the logical analysis of the text also contributes to these aspects.

2.1 Parallel Adventures

We add now the following story in parallel to the one presented in Section 2.

> Your friend is also looking for the treasure. You both start at the same time but following different paths and getting to the castle at different times. Your friend understands only the parrot on the right of the large

> door. The other parrot speaks a language unknown to your friend. The parrot on the right tells your friend that inside the vase next to the last window of the corridor your friend can find the key that opens the red door. In addition, the parrot tells your friend that going through the green unlocked door will certainly lead to death without finding the treasure and that, regarding the other two doors, one will safely lead to the treasure and the other will certainly lead to death. Obviously you do not know what the parrot on the right tells your friend and your friend does not know what the parrot on the left tells you. Moreover, neither you nor your friend are aware that someone else is looking for the treasure.

Some children are again asked the questions from Section 2, this time obviously referred to their friend. In discussing and answering the questions, the children should not take into account what they know about the first story. Although this could be effectively achieved with a variant of the game in which the two stories are told to two distinct groups of children, for simplicity we assume that the entire class is told the two stories in sequence.

In a second phase of the discussion the children are urged to combine the information of the two stories. Typical questions during this phase could be:

1. Is the key needed to reach the treasure?
2. Which door leads to the treasure?
3. Can you be sure that you reach the treasure without dying?

However such questions should not be provided by the teacher and the expectation is that they are raised spontaneously (and correctly answered) during the discussion.

Most children, independently of the age group, would find the solution following a sort of deductive approach by considering the two persons' options, extracting a person's knowledge about the negative outcome, and using it to rule out one of the other person's possible options, thus leaving the other option as the globally positive outcome. In our simple story example, the child can exploit the information known by the protagonist's friend that the green door leads to death to rule out such a door in the protagonist's options and leave the blue door as the solution.

It is always important to urge the children to describe the reasoning process they followed to find the solution of a problem. We will illustrate in Section 4 how children can develop a formal model of a problem and in Section 5 how they can use and enrich the model to represent a formal proof.

Some Children aged 11-14, who possibly had been trained to develop tabular representations of problems and their solutions, may perform a systematic analysis of the global plot using tools similar to Table 1. In fact, from the contents of Table 1 it is immediate to deduce that the blue door leads to the treasure from the premises that the green door leads to death and either the green or the blue door leads to the treasure.

<table>
<tr><th>door ⟶</th><th rowspan="2">green door (unlocked)</th><th rowspan="2">blue door (unlocked)</th><th rowspan="2">red door (locked)</th></tr>
<tr><th>↓ person</th></tr>
<tr><td>I (without the key)</td><td colspan="2">one of the two leads to the treasure and the other leads to death</td><td>not accessible</td></tr>
<tr><td>My friend (with the key)</td><td>it leads to death</td><td colspan="2">one of the two leads to the treasure and the other leads to death</td></tr>
</table>

Table 1. Knowledge of the two protagonists

The discussion on parallel adventures should provide answers to questions 1 and 2. Question 3 may or may not be raised during the discussion, but cannot have a positive answer at this stage. In fact, the two friends are not aware of each other looking for the treasure. However, at this point, the discussion may spontaneously identify the possibility of a collaboration between the two friends and investigate how to enable and carry out such a collaboration. One problem is that the two friends arrive at the castle at different times, thus they are unlikely to meet each other unless they deliberately wait for each other. But they would be willing to wait for each other only if they were aware of each other looking for the treasure and they knew that sharing the information they know will enable them to safely get to the treasure. This leads to Section 3.

3 Synchronisation through Collaboration and Agreement

If the children identify the possibility of a collaboration between the two friends, the discussion can be finalised to discover ways to change the story to make this collaboration as a possible decision. Otherwise the teacher will need to explicitly introduce in the story new conditions.

An example of conditions that enable collaborations is:

> The parrot on the left also tells you that your friends knows
> - which between the green and the blue door will certainly lead to death, and
> - that what you know contains the additional information your friend needs in order to be sure to safely reach the treasure.

Children are asked once again to answer the four questions from Section 2, first referred to themselves, then referred to their friend. They will now notice that they will need to wait for their friend or be sure to find their friend waiting for them in order to be able to restrict the number of reasonable choices. However, this would mean to share the treasure with their friend. Therefore it might be reasonable to take the risk to die aiming to own the entire treasure. Some children might be willing to take this risk, others might not, but both the risky and the safe alternatives should be considered reasonable.

Some temporal reasoning can be carried out at this point. The children will have already identified their two possible decisions:

1. wait for their friend, unless their friend is already waiting for them, and use the combined information to choose the right door (which will appear to be the blue one);
2. randomly choose one between the green and blue doors.

At this point the children will be asked how many and which the outcomes of each of such decisions are.

For Decision 1 they would normally identify (1.1) finding the treasure and sharing it with the friend as the unique outcome. However, what if they decide to wait for their friend, but their friend does not? The friend might have already arrived and proceeded through one of the doors alone. Then (1.2) the friend would be waited for forever. This is actually a typical concurrency problem known as *starvation*. The other possibility is (1.3) that the friend arrives later but does not agree on sharing the treasure. Here a lot of potential alternatives are possible for what is going to happen, but discussing them is outside the scope of this paper, although it might be worthy from a didactical point of view.

For Decision 2 the children would normally identify two possible outcomes, (2.1) finding the treasure when going through the blue door, thus opening a number of alternatives for the friend, and (2.2) dying when going through the green door, thus neglecting the fact that if the friend reaches the castle before them, they might choose the blue door that allows them to avoid dying but, and this is alternative (2.3), no longer find the treasure, already taken away by their friend. Finally, symmetric to the previous decision case, there are further alternatives: (2.4) the friend arrives later, which is a case of starvation and (2.5) the friend arrives earlier, with again a lot of further potential alternatives are possible.

As a conclusion of this discussion, we can introduce the concept of *assumption*: when the number of possible decisions is too big, and even unclear as in alternatives 1.3 and 2.5 above, we can assume only the most *plausible* alternatives. In our story, we could actually assume that the two friends decide to wait for each other, which, in fact, was probably the implicit assumption of most children. Some care is important here, since we should take into account the children's opinions and not making assumptions that might upset any of them. Note that a majority vote might not be always be the best in this case. Common sense and knowledge of the children's personalities are essential in this situation.

3.1 Teacher's Role and Learning Outcome

All questions considered in Sections 2–3 should be put to the entire class rather than individually and should result in a discussion in which children can freely express their opinions and show their attitudes as risk takers or safe players. Here the teacher needs to play a neutral role, as a moderator who accepts all opinions and attitudes, possibly helping children to provide justification and rationale but without expressing any form of judgement. Furthermore, as we have noted at the end of Section 3 concerning assumptions, not everything can be planned in advance and common sense should be used in choosing the next steps of the game.

The aim of the exercise carried out in Section 3 is for the children to understand that collaboration and agreement are important in solving problems and achieving objectives, although they may require some form of compromise leading, in general, to approximate solutions of the problem or to the partial achievement of the objective (in our story the partial objective is that of getting only half of the treasure). However, the possibility of collaboration does not preclude independent actions, which might lead to better but uncertain results. The uncertainty may be not only due to randomness but also to timing issues. In computer science this is the case of *real-time* and *time-critical* systems.

The broad learning outcome we described goes well beyond mathematics and computer science, but it is definitely worthy that the class discussion covers such general, interdisciplinary aspects, although this may result in lengthy digressions. After all computer science is both a theoretical and practical/applied science with also philosophical and ethical consequences, and it is important to expose children to the practical aspects and consequences of using computer science and, more important, computer science theory and principles.

From a technical point of view, the two parallel stories are actually two concurrent processes, which may evolve independently (green door and red door, respectively) or synchronise (blue door).

As a final note to this section, the number of possible contexts and variations of stories is infinite. The same concepts can be illustrated through completely different story settings, as the result of the teacher's creativity or, even better, produced by the children through class discussions or working groups.

4 Modelling Stories

Throughout the discussion described in Sections 2–3 children should be invited to illustrate the story in a visual form. This should happen with an interdisciplinary approach and may also involve visual arts, especially for the age group 7–10.

We have to note that the stories include a large number of details that are irrelevant for the offered choices. The presence of such details is important to make the stories realistic and engaging. In addition, these "literary" details offer an important context for developping *abstraction* and modelling skills.

Some guidance is needed to enable the children to identify the appropriate representation, namely the appropriate *visual formalism*, with which to create the model. The model in Fig. 1 is a typical representation for children aged 7-14. It may still contain some irrelevant details. For example, the reason why a door is not chosen, whether because it is locked and I do not have the key or because I know that it will lead me to death, is irrelevant, but it is likely that the child would still depict such details. In fact, these semantic details represent the concrete rationale for the choice, which otherwise might lose meaning in the child's mind. That is why some children may still keep such details in their models. However, more irrelevant details, such as the corridor, the vase where

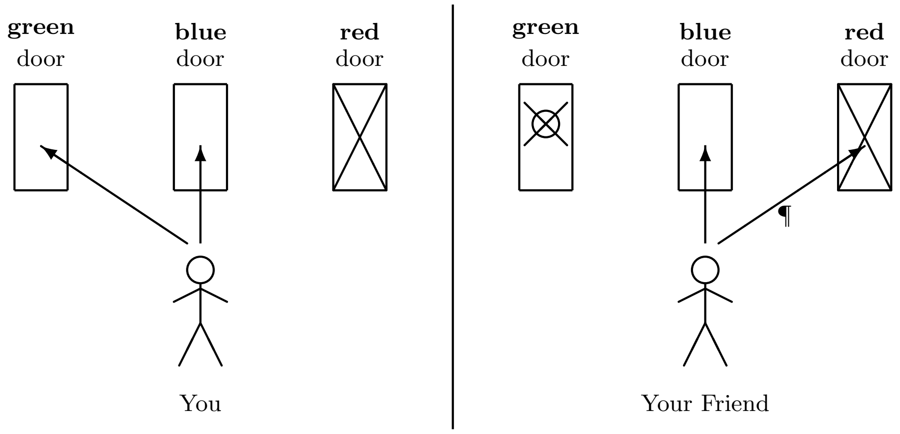


Fig. 1. Story models of reasonable choices from your perspective and your friend's perspective.

the key is normally hidden and even the "enchanting" detail of the two parrots are likely to be abstracted away, though this may require a number of iterations.

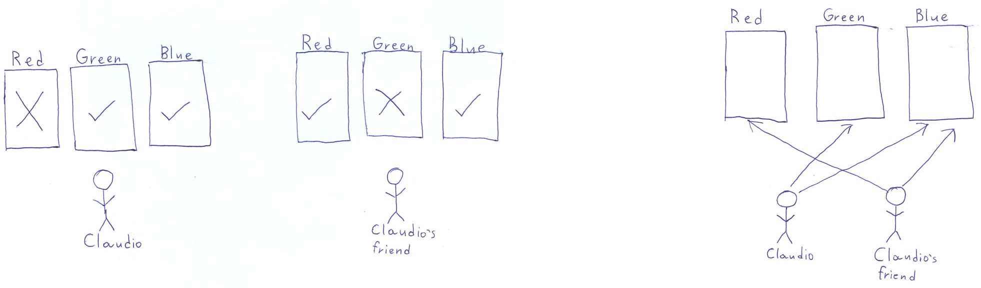
Some children, especially in the age group 7-9, may focus on the solution and provide a concise, abstract model, in which the relation between the person and the door, expressed by arrows in Fig. 1, is not represented. This is the case of Claudio, 9 years old, whose models for himself and his friend are given in Fig. 2(a). Chiara, 13 years old, instead, explicitly includes arrows from her to the doors. Her models for herself and her friend are given in Fig. 3(a).

4.1 Finite State Machines, Composition and Complexity

The age group 11–14 children should be also guided to come out with a more formal model such as the finite state machine [8] in Fig. 4.

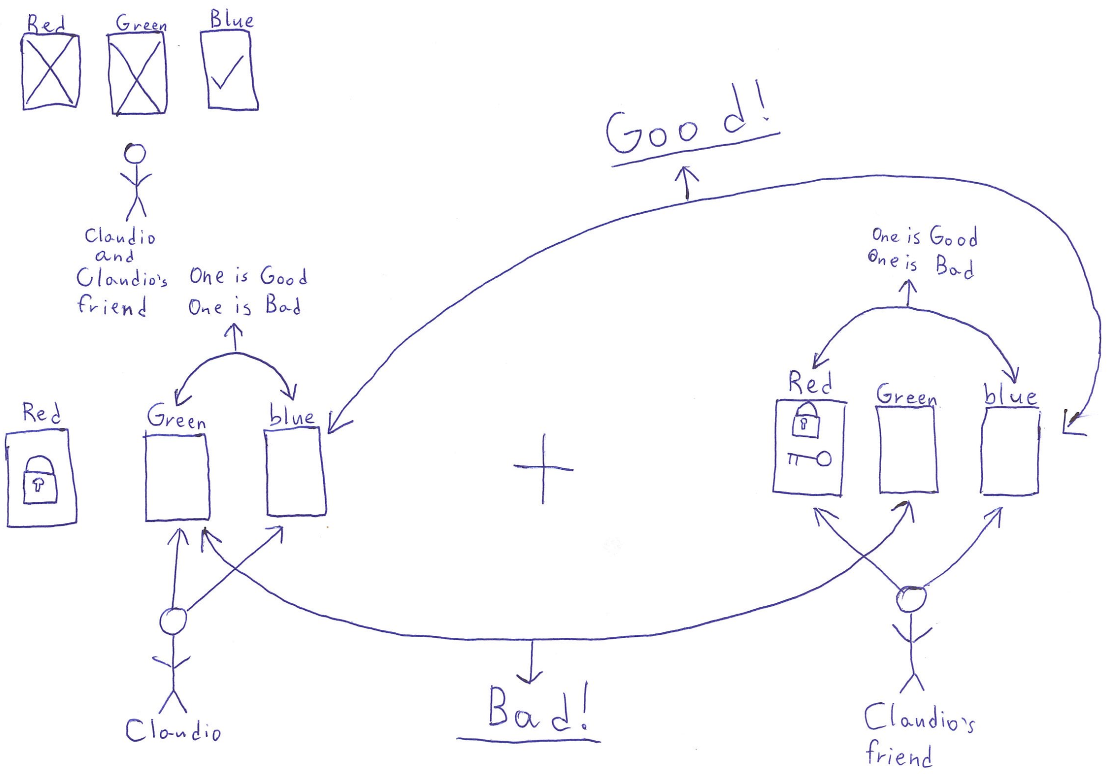
In fact, for this age group, it is also important to enable the child to understand that concurrency may quickly increase the complexity of the modelled system. Children should be guided to combine the two models in Fig. 4, The result should be something like the finite state machine in Fig. 5. Although children of this age should be able to develop this model, they are likely to feel that it is useless, due to the spaghetti-like interwinding of arrows. This is a good chance to show that, even with small systems as the ones in Fig. 4, concurrent composition may lead to a very complex global finite state machine. Depending on the interest expressed by the children, their previous knowledge and their reactions throughout the exercise, the discussion may now more deeply involve the notion of complexity. This may involve *algorithmic complexity* and/or the *state explosion problem.*

However, we must always avoid that the children become frustrated in unsuccessfully trying to develop the global finite state machine. If children are lost in the complexity of arrows, the teacher should help them to reach the solution in Fig. 5.



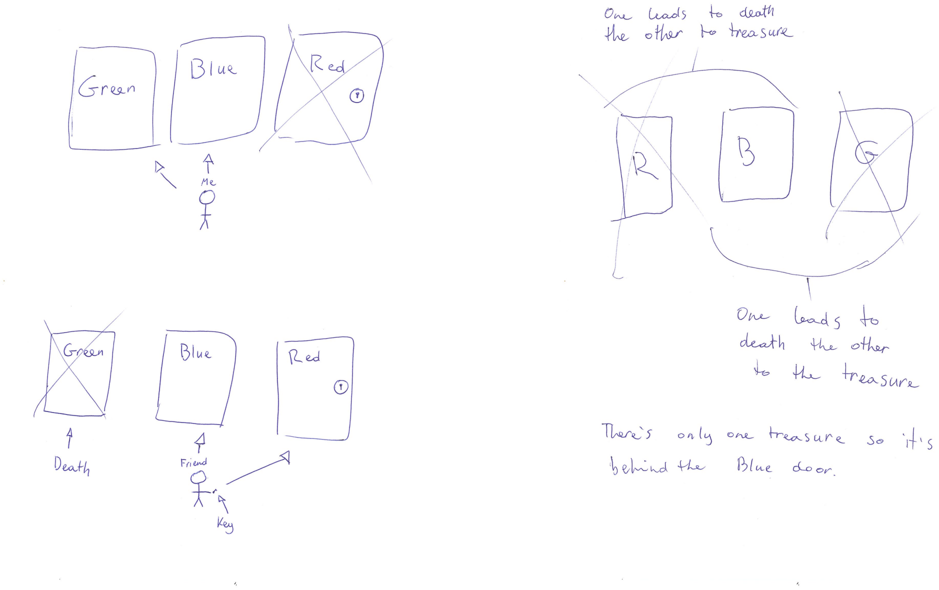
(a) Models of Claudio's and his friend's perspectives.

(b) Combined model of the two pespectives.



(c) Claudio's proof.

Fig. 2. Claudio's drawings.



(a) Models of Chiara's and her friend's perspectives.

(b) Chiara's proof.

Fig. 3. Chiara's drawings.

4.2 Temporal Reasoning using Finite State Machines

A number of observations and exercises may be carried out on the complex visual representation depicted in Fig. 5:

1. The model illustrates all possible temporal orderings in which the two friends arrive at the castle and choose the door.
2. Some children might note the fact that the model does not capture the case that the two friends arrive at the castle at the same time, with further questions arised
   (a) how to modify the finite state machine to cover this case?
   (b) how much the complexity would increase?
   (c) is such a change necessary? if so, why?
   and the chance to introduce and discuss the difference between *true concurrency* and *interleaving*.
3. The model can be enriched with further information, for example by colouring the states (the circles) in which a certain property is true. (Examples of properties are: you find the treasure, your friend finds the treasure, you die, your friend dies, or a combination of some of them using logical connectives 'or' or 'and'.)
4. Perform a temporal analysis on the model coloured as in Item 3 to find out whether, starting from the initial state, a property is true [3]

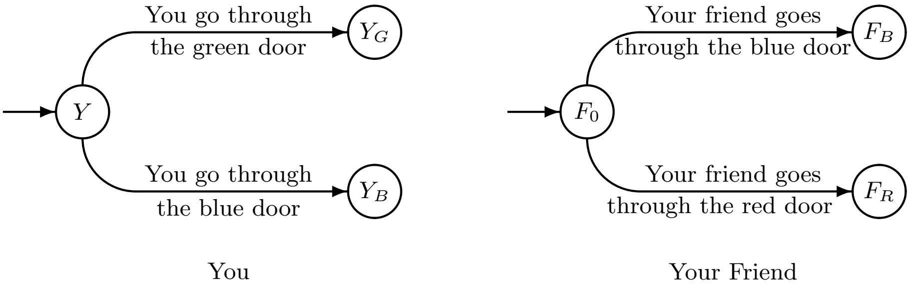


Fig. 4. Age group 11-14: Story models of reasonable choices from your perspective and your friend's perspective using finite state machines.

(a) for some state (temporal modality: $\exists\Diamond$);
(b) for all states (temporal modality: $\forall\Box$);
(c) for all states along some path (temporal modality: $\exists\Box$);
(d) for some states along each path (temporal modality: $\forall\Diamond$).

If the discussion covers the difference between true concurrency and interleaving, it may be worthy to note that the model in Fig. 7 is based on true concurrency and the model in Fig. 6 is based on interleaving.

Furthermore, as a result of the temporal reasoning carried out in Sect. 3, we can observe that the models in Fig. 4 do not carry any information about the story outcomes in terms of finding the treasure or die.

4.3 Refinement and Formal Verification

A next step for the children is to add final states to the models in Fig. 4 to describe the problem possible outcomes: you find the treasure (Y_T), you do not find the treasure (Y_N), you die (Y_D), your friend finds the treasure (D_T), your friend does not find the treasure (D_N) and your friend dies (F_D). This is clearly a form of *model refinement*.

Here the issue is whether states Y_N and F_N are needed. After all, in our story, the parrot on the left side tells you that one of the two unlocked doors safely leads to the treasure and the other leads to death. Can we avoid death but not find the treasure? We have seen in Section 3 that alternatives 2.1 and 2.3 allow for this situation. However, the point to be made here is that this situation was observed only when we tried to compose the two stories. There is a double moral here.

On the one hand, refinement is not an easy task and it is likely to miss some essential behaviour while refining a model. In fact, there are normally many possible refinements, but in order to get a *correct refinement* we need to choose one of those that allow us to achieve our objective. Furthermore, it is important to choose the *best refinement* among all correct refinements. What is "best" is

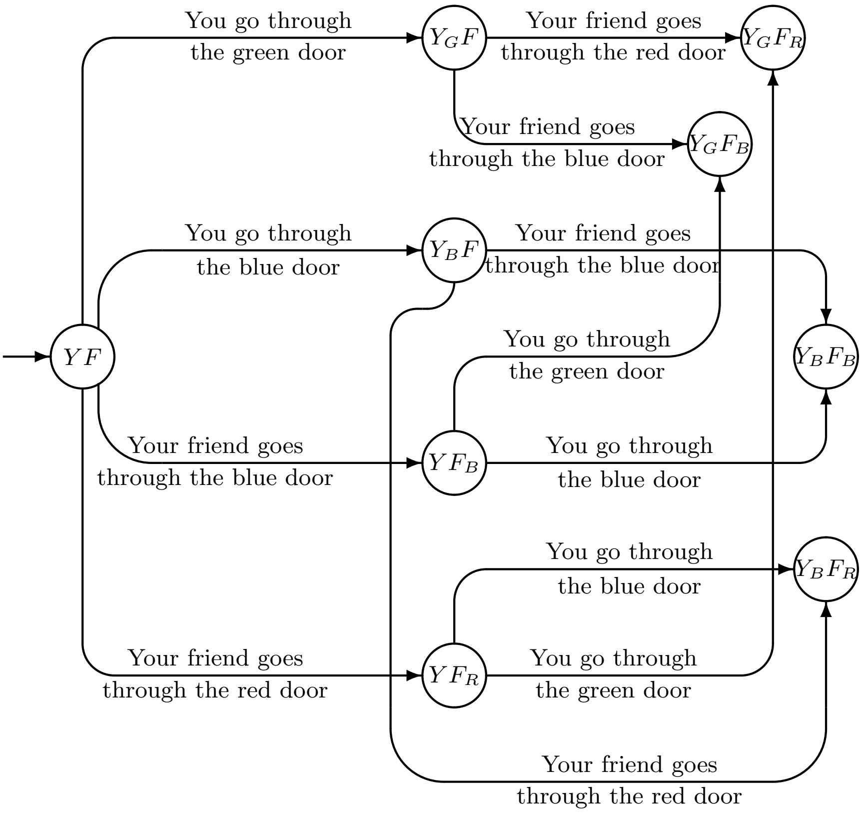


Fig. 5. Age group 11–14: Global model of the two independent perspectives.

then a matter of *efficiency* and other *non-functional* system properties. But we are now going too far.

On the other hand, a missing *requirement* of a component can be identified when analysing the global behaviour through *formal verification*.

After this discussion we ask the children to compose the two new, extended finite state machines into a global one, observing that this can be carried out by just modifying the finite state machine in Fig. 6 through the addition of the appropriate composition of the final states. The spaghetti-like interwinding of arrows makes the model unreadable, but the children would still be able to build it. Trying to compose together the new, refined finite state machine components directly would actually be impossible, whereas adding the composition of the refined parts to the global machine is actually feasible. The moral here is that refinement makes the building of complex systems feasible.

The discussion considered in this section has not taken into account synchronisation yet. Once we reduce the number of reasonable choices with the

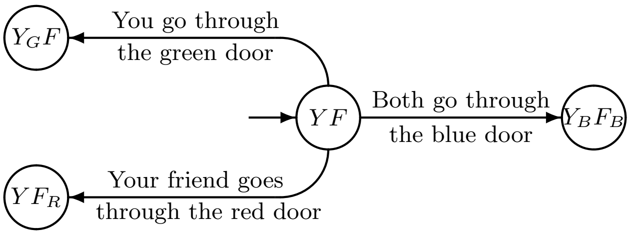


Fig. 6. Age group 11–14: Global model of the two synchronised perspectives.

additional conditions introduced in Section 3 by synchronising on the blue door or going independently through the green and red doors, then the age group 11–14 should come out with something like the finite state machine in Fig. 6.

Both age groups can also work with the component models in Fig. 1 and come out with something like the representation in Fig. 7. It is important to

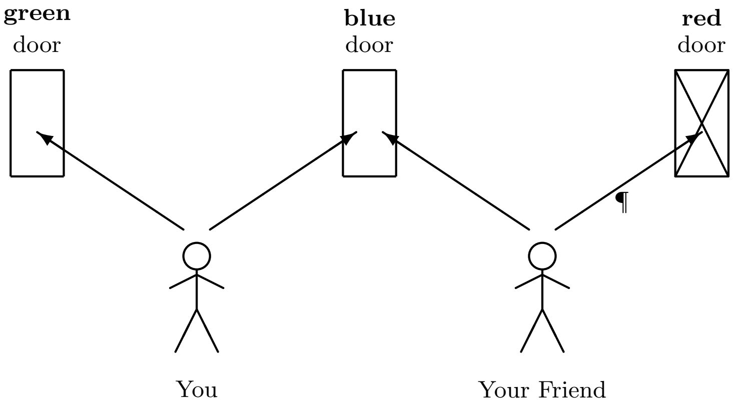


Fig. 7. Age group 7-10: Global model of the two synchronised perspectives.

note that when Claudio was asked to draw the global model, he realised that his models of the two separate perspectives, given in Fig. 2(a), were inadequate to be combined into a global model. In order to make his models *compositional*, he replaced the markers ✓ and × on the doors with arrows between persons and doors, thus getting the global model in Fig. 2(b), which is very similar with the expected model given in Fig. 7.

5 Representing Proofs

When Claudio was asked to show the solution of the problem he came up with the three marked doors in the top left corner of Fig. 2(c). A discussion followed to understand how he obtained such a solution. When asked to show the way he reached the solution he was initially puzzled. Then following the suggestion to use his previous models (drawings), combining them in some way and showing on them his reasoning, Claudio worked out the proof illustration in Fig. 2(c). It is interesting to note that some of the information abstracted in the models in Fig. 2(a) and 2(b), namely the padlock and the key, reappears in this representation of the proof.

A final note is that proofs developped by children of the age group 7-10 normally have purely visual representations, whereas children of the age group 11-14 can already articulate reasoning in a textual form. Their proof will be in a hybrid visual and textual form as the one in Fig. 3(b), which was developed by Chiara.

6 Conclusion and Future Work

In this paper we have adopted an "unplugged approach" in teaching computer science to children [1,2] and taken inspiration from the *Choose Your Own Adventure* series of children's gamebooks, in which the reader may experience various alternatives about the characters' actions. We used the example of a story consisting of two parallel parts and made the children reason about the knowledge acquired by the two protagonists of the two parts and then, on the one hand, combine what the two protagonists know in order to predict the outcome of each decision and, on the other hand, to explore how the plot would evolve in the absence or in the presence of collaboration and information sharing between the two protagonists.

In this exploratory process, children have been exposed to real-life instantiations of a number of computer science concepts, expecially from the theory of concurrency. Instances of concepts, such has synchronisation, assumption, starvation, complexity, state explosion, true concurrency, interleaving, refinement, correctness, efficiency, property, formal verification and proof have been observed in the story plot. Such observations have been used to foster discussion and debate among the children, and enable reasoning, modelling as well as awareness and externalisation of their reasoning process throughout some form of written proof.

Three fundamental remarks are:

- The fact that we use a single example to explore a large variety of concepts througt the paper is purely illustrative. In real classroom work a single, sequential or parallel story would probably be used to introduce one concept or a few strictly related concepts in a very targeted way. Obviously stories can also be revisited, expanded and compared at a later stage.

- Although for the benefit of the teacher, who might pursue a deeper understanding of the concepts underlying the observations, we have introduced technical computer science terminology, such a technical jargon should be avoided with the children, unless it is important for future topics or may appear curious or interesting for the children (e.g. the use of the word "starvation").
- The focus on concurrency has been used to show that an unplugged, problem-based approach can successfully work well beyond the most basic mathematical and computer science concepts, and cover one of the most challenging areas of computer science. It is by no means our intention to propose a children's course on "formal methods for concurrency" but, instead, to integrate the approach we presented within a more general unplugged, problem-based approach to be used in a interdiscplinary way.

We have to point out that the author experimented the approach presented in this paper with his own children, Claudio and Chiara, respectively 9 years old and 13 years old at the moment when the paper was written. Some aspects of the approach, especially the development of problem-solving skills, however, have been used with both children since they were 5–6 years old.

As future work we plan to experiment our proposal as part of a general unplugged, problem-based approach in a real classroom context.

References

1. Computer science without a computer. https://www.csunplugged.org/en/.
2. T. Bell. A low-cost high-impact computer science show for family audiences. In *23rd Australian Computer Science Conference*, pages 10–16. ACM, 2000.
3. M. Ben-Ari and Z. M. A. Pnueli. The temporal logic of branching time. In *POPL'81*, pages 164–176. ACM, 1981.
4. C. J. Brainerd. *Piaget's Theory of Intelligence.* Prentice Hall, 1978.
5. J. S. Bruner. *Toward a Theory of Instruction.* Bwelknap Press, 1966.
6. J. P. Gibson. Formal methods: Never too young to start. In *FORMED 2008*, pages 151–160, Budapest, Hungary, March 2008.
7. P. Hilton. The mathematical component of a good education. *Miscellanea Mathematica*, pages 145–154, 1991.
8. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.
9. L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.
10. L. Lamport. Turing lecture: "The computer science of concurrency: The early years". *Communications of the ACM*, 58(6):71–76, 2015.
11. J. Magee and J. Kramer. *Concurrency: State Models and Java Programs.* Wiley, 2006.
12. A. Murphy Paul. Researchers find that frequent tests can boost learning. *Scientific American*, August 1, 2015.
13. E. Packard. *Sugarcane Island.* Vermont Crossroads Press, 1976.
14. A. H. Schoenfeld. *Mathematical Problem solving.* Academic Press, 1985.
15. J. Wang and W. Tepfenhart. *Formal Methods in Computer Science.* Chapman and Hall/CRC, 2019.