Logical and Psychological Analysis of Deductive Mastermind

Nina Gierasimczuk[1], Han van der Maas[2], and Maartje Raijmakers[2]

[1] Institute for Logic, Language and Computation, University of Amsterdam,
Nina.Gierasimczuk@gmail.com,
[2] Department of Psychology, University of Amsterdam

Abstract. The paper proposes a way to analyze logical reasoning in a deductive version of the Mastermind game implemented within the Math Garden educational system. Our main goal is to derive predictions about the cognitive difficulty of game-plays, e.g., the number of steps needed for solving the logical tasks or the working memory load. Our model is based on the analytic tableaux method, known from proof theory. We associate the difficulty of the Deductive Mastermind game-items with the size of the corresponding logical tree derived by the tableau method. We discuss possible empirical hypotheses based on this model, and preliminary results that prove the relevance of our theory.

Keywords: Deductive Mastermind, Mastermind game, deductive reasoning, analytic tableaux, Math Garden, educational tools

1 Introduction and Background

Computational and logical analysis has already proven useful for the investigations into the cognitive difficulty of linguistic and communicative tasks (see [1–5]). We follow this line of research by adapting formal logical tools to directly analyze the difficulty of non-linguistic logical reasoning. Our object of study is a deductive version of the *Mastermind* game. Although the game has been used to investigate the acquisition of complex skills and strategies in the domain of reasoning about others [6], as far as we know, it has not yet been studied for the difficulty of the accompanying deductive reasoning. Our model of reasoning is based on the proof-theoretical method of analytic tableaux for propositional logic, and it gives predictions about the empirical difficulty of game-items. In the remaining part of this section we give the background of our work: we explain the classical and static versions of the Mastermind game, and describe the main principles of the online Math Garden system. In Section 2 we introduce Deductive Mastermind as implemented within Math Garden. Section 3 gives a logical analysis of Deductive Mastermind game-items using the tableau method. Finally, Section 4 discusses some hypotheses drawn on the basis of our model, and gives preliminary results. In the end we briefly discuss the directions for further work.

1.1 Mastermind Game

Mastermind is a code-breaking game for two players. The modern game with pegs was invented in 1970 by Mordecai Meirowitz, but the game resembles an earlier pen and paper game called Bulls and Cows. The Mastermind game, as known today, consists of a decoding board, code pegs of k colors, and feedback pegs of black and white. There are two players, the code-maker, who chooses a secret pattern of ℓ code pegs (color duplicates are allowed), and the code-breaker, who guesses the pattern, in a given n rounds. Each round consists of code-breaker making a guess by placing a row of ℓ code pegs, and of code-maker providing the feedback of zero to ℓ key pegs: a black key for each code peg of correct color and position, and a white key for each peg of correct color but wrong position. After that another guess is made. Guesses and feedbacks continue to alternate until either the code-breaker guesses correctly, or n incorrect guesses have been made. The code-breaker wins if he obtains the solution within n rounds; the code-maker wins otherwise. Mastermind is an *inductive inquiry* game that involves *trials of experimentation and evaluation.* As such it leads to the interesting question of the underlying logical reasoning and its difficulty. Existing mathematical results on Mastermind do not provide any answer to this problem—most focus has been directed at finding a strategy that allows winning the game in the smallest number of rounds (see [7–10]).

Static Mastermind [11] is a version of the Mastermind game in which the goal is to find the minimum number of guesses the code-breaker can make all at once at the beginning of the game (without waiting for the individual feedbacks), and upon receiving them all at once completely determine the code in the next guess. In the case of this game some strategy analysis has been conducted [12], but, more importantly, Static Mastermind has been given a computational complexity analysis [13]. The corresponding Static Mastermind Satisfiability Decision Problem has been defined in the following way.

Definition 1 (Mastermind Satisfiability Decision Problem).

Input *A set of guesses G and their corresponding scores.*
Question *Is there at least one valid solution?*

Theorem 1. *Mastermind Satisfiability Decision Problem is NP-complete.*

This result gives an objective computational measure of the difficulty of the task. NP-complete problems are believed to be cognitively hard [14–16], hence this result has been claimed to indicate why Mastermind is an engaging and popular game. It does not however give much insight into the difficulty of reasoning that might take place while playing the game, and constructing the solution.

1.2 Math Garden

This work has been triggered by the idea of introducing a dedicated logical reasoning training in primary schools through the online Math Garden system

(Rekentuin.nl or MathsGarden.com, see [17]). Math Garden is an adaptive training environment in which students can play various educational games especially designed to facilitate the development of their abstract thinking. Currently, it consists of 15 arithmetic games and 2 complex reasoning games. Students play the game-items suited for their level. A difficultly level is appropriate for a student if she is able to solve 70% items of this level correctly. The difficulty of tasks and the level of the students' play are being continuously estimated according to the Elo rating system, which is used for calculating the relative skill levels of players in two-player games such as chess [18]. Here, the relative skill level is computed on the basis of student v. game-item opposition: students are rated by playing, and items are rated by getting played. The rating depends on accuracy and speed of problem solving [19]. The rating scales go from $-$ infinity to $+$ infinity. If a child has the same rating as an item this means that the child can solve the item with probability .5. In general, the absolute values of the ratings have no straightforward meaning. Hence, one result of children playing within Math Garden is a rating of all items, which gives item difficulty parameters. At the same time every child gets a rating that reflects her or his reasoning ability. One of the goals of this project is to understand the empirically established item difficulty parameters by means of a logical analysis of the items. Figure 5 in the Appendix depicts the educational and research context of Math Garden.

2 Deductive Mastermind

Now let us turn to Deductive Mastermind, the game that we designed for the Math Garden system (its corresponding name within Math Garden is *Flowercode*). Figure 1 shows an example item, revealing the basic setting of the game. Each item consists of a decoding board (1), short feedback instruction (2), the domain of flowers to choose from while constructing the solution (3) and the timer in the form of disappearing coins (4). The goal of the game is to guess the right sequence of flowers on the basis of the clues given on the decoding board. Each row of flowers forms a conjecture that is accompanied by a small board on the right side of it. The dots on the board code the feedback about the conjecture: one green dot for each flower of correct color and position, one orange dot for each flower of correct color but wrong position, and one red dot for each flower that does not appear in the secret sequence at all. In order to win, the player is supposed to pick the right flowers from the given domain (3), place them in the right order on the decoding board, right under the clues, and press the OK button. She must do so before the time is up, i.e., before all the coins (4) disappear. If the guess is correct, the number of coins that were left as she made the guess is added to her overall score. If her guess is wrong, the same number is subtracted from it. In this way making fast guesses is punished, but making a fast correct response is encouraged [19].

Unlike the classical version of the Mastermind game, Deductive Mastermind does not require the player to come up with the trial conjectures. Instead, Flowercode gives the clues directly, ensuring that they allow *exactly one correct solution.*

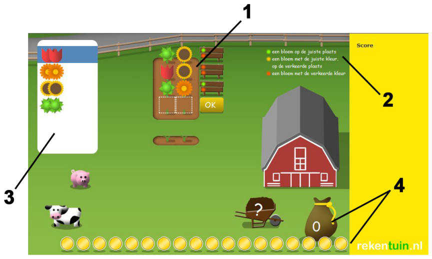

1
2
3
4
Score
een bloem op de juiste plaats
een bloem met de juiste kleur, op de verkeerde plaats
een bloem met de verkeerde kleur
OK
?
0
rekentuin.nl

Fig. 1. Deductive Mastermind in the Flowercode setting

Hence, Deductive Mastermind reduces the complexity of classical Mastermind by changing from an inductive inference game into a very basic logical-reasoning game. On the other hand, when compared to Static Mastermind, Deductive Mastermind differs with respect to the goal. In fact, by guaranteeing the existence of exactly one correct solution, Deductive Mastermind collapses the postulated complexity from Theorem 1, since the question of the Static Mastermind Satisfiability Problem becomes void. It is fair to say that Deductive Mastermind is a combination of the classical Mastermind game (the goal of the game is the same: finding a secret code) and the Static Mastermind game (it does not involve the trial-and-error inductive inference experimentation). Its very basic setting allows access to atomic logical steps of non-linguistic logical reasoning. Moreover, Deductive Mastermind is easily adaptable as a single-player game, and hence suitable for the Math Garden system. The simple setting provides educational justification, as the game trains very basic logical skills.

The game has been running within the Math Garden system since November 2010. It includes 321 game-items, with conjectures of various lengths (1-5 flowers) and number of colors (from 2 to 5). By January 2012, 2,187,354 items had been played by 28,247 primary school students (grades 1-6, age: 6-12 years) in over 400 locations (schools and family homes). This extensive data-collecting process allows analyzing various aspects of training, e.g., we can access the individual progress of individual players on a single game, or the most frequent mistakes with respect to a game-item. Most importantly, due to the student-item rating system mentioned in Section 1.2, we can infer the relative difficulty of game-items. Within this hierarchy we observed that the present game-item domain contains certain "gaps in difficulty"—it turns out that our initial difficulty estimation in terms of non-logical aspects (e.g., number of flowers, number of colors, number of lines, the rate of the hypotheses elimination, etc.) is not precise, and hence the domain of items that we generated does not cover the whole difficulty space. Providing a logical apparatus that predicts and explains the difficulty of Deductive Mastermind game-items can help solving this problem and hence also facilitate the training effect of Flowercode (see Appendix, Figure 7).

3 A Logical Analysis

Each Deductive Mastermind game-item consists of a sequence of conjectures.

Definition 2. *A* conjecture *of length l over k colors is any sequence given by a total assignment, $h : \{1, \dots, \ell\} \rightarrow \{c_1, \dots, c_k\}$. The* goal sequence *is a distinguished conjecture, $goal : \{1, \dots, \ell\} \rightarrow \{c_1, \dots, c_k\}$.*

Every non-goal conjecture is accompanied by a feedback that indicates how similar h is to the given goal assignment. The three feedback colors: green, orange, and red, described in Section 2, will be represented by letters g, o, r.

Definition 3. *Let h be a conjecture and let goal be the goal sequence, both of length l over k colors. The* feedback f for h with respect to *goal is a sequence*

$$\overbrace{g \dots g}^{a}\ \overbrace{o \dots o}^{b}\ \overbrace{r \dots r}^{c} = g^a o^b r^c,$$

where $a, b, c \in \{0, 1, 2, 3, \dots\}$ and $a + b + c = \ell$. The feedback consists of:

- *exactly one g for each $i \in G$, where $G = \{i \in \{1, \dots \ell\} \mid h(i) = goal(i)\}$.*
- *exactly one o for every $i \in O$, where $O = \{i \in \{1, \dots, \ell\} \backslash G \mid$ there is a $j \in \{1, \dots, \ell\} \backslash G,$ such that $i \neq j$and $h(i) = goal(j)\}$.*
- *exactly one r for every $i \in \{1, \dots, \ell\} \backslash (G \cup O)$.*

3.1 The informational content of the feedback

How to logically express the information carried by each pair (h, f)? To shape the intuitions let us first give a second-order logic formula that encodes any feedback sequence $g^a o^b r^c$ for any h with respect to any *goal*:

$$\begin{aligned}\exists G{\subseteq}\{1, \dots \ell\}(card(G){=}a \wedge \forall i{\in}G\ h(i){=}goal(i) \wedge \forall i{\notin}G\ h(i){\neq}goal(i)\\ \wedge\ \exists O{\subseteq}\{1, \dots \ell\}\backslash G\ (card(O){=}b \wedge \forall i{\in}O\ \exists j{\in}\{1, \dots \ell\}\backslash G(j{\neq}i \wedge h(i){=}goal(j))\\ \wedge\ \forall i{\in}\{1, \dots \ell\}\backslash(G{\cup}O)\ \forall j{\in}\{1, \dots \ell\}\backslash G\ h(i){\neq}goal(j))).\end{aligned}$$

Since the conjecture length, ℓ, is fixed for any game-item, it seems sensible to give a general method of providing a less engaging, propositional formula for any instance of (h, f). As literals of our Boolean formulae we take $h(i) = goal(j)$, where $i, j \in \{1, \dots \ell\}$ (they might be viewed as propositional variables $p_{i,j}$, for $i, j \in \{1, \dots \ell\}$). With respect to sets G, O, and R that induce a partition of $\{1, \dots \ell\}$, we define $\varphi^g_G, \varphi^o_{G,O}, \varphi^r_{G,O}$, the propositional formulae that correspond to different parts of feedback, in the following way:

- $\varphi^g_G := \bigwedge_{i \in G} h(i){=}goal(i) \wedge \bigwedge_{j \in \{1, \dots, \ell\} \backslash G} h(j){\neq}goal(j)$,
- $\varphi^o_{G,O} := \bigwedge_{i \in O} (\bigvee_{j \in \{1, \dots, \ell\} \backslash G, i \neq j} h(i) = goal(j))$,
- $\varphi^r_{G,O} := \bigwedge_{i \in \{1, \dots \ell\} \backslash (G \cup O), j \in \{1, \dots \ell\} \backslash G, i \neq j} h(i) \neq goal(j)$.

Observe that there will be as many substitutions of each of the above schemes of formulae, as there are ways to choose the corresponding sets G and O. To fix the domain of those possibilities we set $\mathbb{G} := \{G | G \subseteq \{1, \ldots, \ell\} \wedge card(G){=}a\}$, and, if $G \subseteq \{1, \ldots, \ell\}$, then $\mathbb{O}^G = \{O | O \subseteq \{1, \ldots, \ell\} \backslash G \wedge card(O){=}b\}$. Finally, we can set $Bt(h, f)$, the Boolean translation of (h, f), to be given by:

$$Bt(h, f) := \bigvee_{G \in \mathbb{G}} (\varphi^g_G \wedge \bigvee_{O \in \mathbb{O}^G} (\varphi^o_{G,O} \wedge \varphi^r_{G,O})).$$

Example 1. Let us take $\ell = 2$ and (h, f) such that: $h(1){:=}c_1$, $h(2){:=}c_2$; $f{:=}or$. Then $\mathbb{G}{=}\{\emptyset\}$, $\mathbb{O}^{\{\emptyset\}}{=}\{\{1\}, \{2\}\}$. The corresponding formula, $Bt(h, f)$, is:

$$(goal(1){\neq}c_1 \wedge goal(2){\neq}c_2) \wedge ((goal(1){=}c_2 \wedge goal(2){\neq}c_1) \vee (goal(2){=}c_1 \wedge goal(1){\neq}c_2))$$

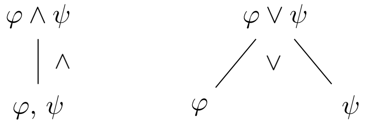
Each Deductive Mastermind game-item consists of a sequence of conjectures together with their respective feedbacks. Let us define it formally.

Definition 4. A Deductive Mastermind game-item *over ℓ positions, k colors and n lines, $DM(l, k, n)$, is a set $\{(h_1, f_1), \ldots, (h_n, f_n)\}$ of pairs, each consisting of a single conjecture together with its corresponding feedback. Respectively, $Bt(DM(l, k, n)) = Bt(\{(h_1, f_1), \ldots, (h_n, f_n)\}) = \{Bt(h_1, f_1), \ldots, Bt(h_n, f_n)\}$.*

Hence, each Deductive Mastermind game-item can be viewed as a set of Boolean formulae. Moreover, by the construction of the game-items we have that this set is satisfiable, and that there is a unique valuation that satisfies it. Now let us focus on a method of finding this valuation.

3.2 Analytic Tableaux for Deductive Mastermind

In proof theory, the analytic tableau is a decision procedure for propositional logic [20–22]. The tableau method can determine the satisfiability of finite sets of formulas of propositional logic by giving an adequate valuation. The method builds a formulae-labeled tree rooted at the given set of formulae and unfolding breaks these formulae into smaller formulae until contradictory pairs of literals are produced or no further reduction is possible. The rules of analytic tableaux for propositional logic, that are relevant for our considerations are as follows.[3]



By our considerations from Section 3 we can now conclude that applying the analytic tableaux method to the Boolean translation of a Deductive Mastermind game-item will give the unique missing assignment *goal*. In the rest of the paper we will focus on 2-pin Deductive Mastermind game-items (where $\ell = 2$), in particular, we will explain the tableau method in more detail on those simple examples.

[3] We do not need the rule for negation because in our formulae only literals are negated.

2-pin Deductive Mastermind Game-items Since the possible feedbacks consist of letters g (green), o (orange), and r (red), in principle for the 2-pin Deductive Mastermind game-items we get six possible feedbacks: gg, oo, rr, go, gr, or. From those: gg is excluded as non-applicable; go is excluded because there are only two positions. Let us take a pair (h, f), where $h(1){=}c_i$, $h(2){=}c_j$, then, depending on the feedback, the corresponding boolean formulae are given in Figure 2. We can compare the complexity of those feedbacks just by looking at their tree representations created from the boolean translations via the tableau method. Those representations clearly indicate that the order of the tree-difficulty for the four possible feedbacks is: $oo{<}rr{<}gr{<}or$. As the feedbacks oo, rr are conjunctions, they do not require branching (the other two include disjunctions, and as such demand reasoning by cases). Unlike rr, the feedback oo in fact gives the solution immediately. Within the two remaining rules, gr requires less memory to store the information in each branch.

Feedback	**Boolean translation**
oo	$goal(1) \neq c_i \wedge goal(2) \neq c_j \wedge goal(1) = c_j \wedge goal(2) = c_i$
rr	$goal(1) \neq c_i \wedge goal(2) \neq c_j \wedge goal(1) \neq c_j \wedge goal(2) \neq c_i$
gr	$(goal(1) = c_i \wedge goal(2) \neq c_j) \vee (goal(2) = c_j \wedge goal(1) \neq c_i)$
or	$(goal(1) \neq c_i \wedge goal(2) \neq c_j) \wedge$ $((goal(1) = c_j \wedge goal(2) \neq c_i) \vee (goal(2) = c_i \wedge goal(1) \neq c_j))$

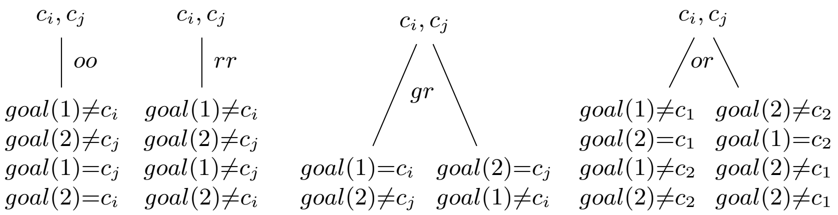


Fig. 2. Formulae and their trees for 2-pin Deductive Mastermind feedbacks

We will now briefly discuss the tableau method on an example. Let us consider the following Deductive Mastermind game-item (Figure 3). The tree on the left stands for the reasoning which corresponds to analyzing the conjectures as given, from top to bottom. The first branching gives the result of applying the gr feedback. In the next level of the tree we apply the oo feedback to the second conjecture. We must first do so assuming the left branch of the first conjecture to be true. This leads to a contradiction—on this branch we get that $goal(2){=}c_1$ and $goal(2){\neq}c_1$. Then we move to the right branch of the first conjecture. This assumption leads to discovering the right assignment, $goal(1){=}c_2$ and $goal(2){=}c_1$, there is no contradiction on this branch. This tableau procedure is required to build the whole tree for the game-item. That is not always necessary. The right-most part of Figure 3 shows what happens if you chose to start the analysis from the second conjecture. We first apply the feedback oo to the second conjecture. We immediately get: $goal(1){=}c_2$ $goal(2){=}c_1$. The full

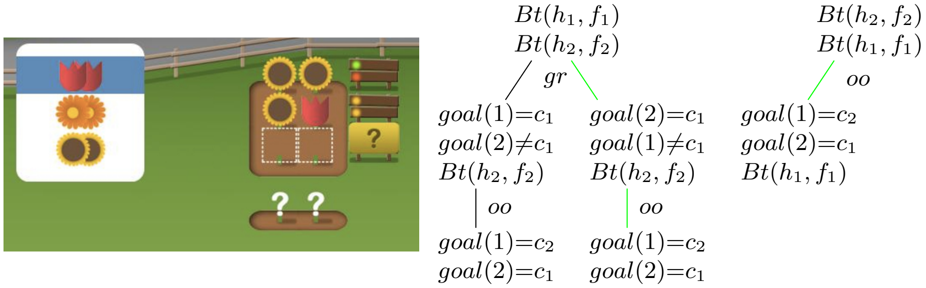


Fig. 3. Comparison of two different trees for one item. The formalization is as follows: c_1 stands for sunflower, c_2 for tulip; $h_1(1){=}c_1, h_1(2){=}c_1, f_1{=}gr$, etc. Green branch gives the right valuation. The tree on the right hand-side analyzes feedback oo first.

unique assignment with no contradiction. We can stop the computation at this point—if the other conjecture contradicted this assignment, then it would mean that the two conjectures must be inconsistent and hence not satisfiable. This contradicts the setting of our game.

The tree might not always give us the complete valuation explicitly. In some game-items it is required to use a flower that did not appear in the conjectures. This is so in the example in Figure 4; the right-most branch does not give a contradiction, it does assign color 1 to the second position of the goal conjecture. In such case we draw the remaining color, c_3 (a tulip in the picture), as the missing value of the first position of the goal conjecture, i.e., $goal(1){=}c_3$.

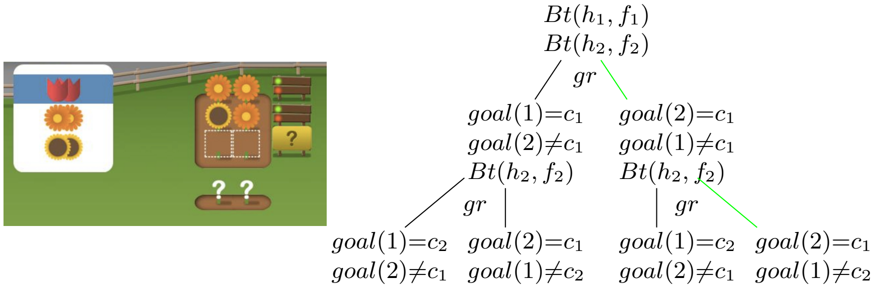


Fig. 4. Comparison of the trees of two different items, while processing the conjectures from top to bottom. Here, c_1 stands for marguerite, c_2 for sunflower, c_3 for tulip. This item rating is 4.2 (see Figure 6).

4 Hypotheses and Preliminary Results

Normatively speaking, the full tree generated by the tableau method for the set of formulae corresponding to a Deductive Mastermind game-item gives its ideal reasoning scheme and thus the size of the tree can be thought of as an abstract complexity measure. Obviously, the shape and the size of the tree for each Deductive Mastermind item depends to some extent on the order in which the formulae are analyzed (see Figure 3). Hence, using the tableau method it is even possible to analyze whether and in what way the students apply reasoning strategies, i.e., how they manipulate with the elements of the task in order to optimize the size of the reasoning tree (i.e., the length of the computation). In this way, items' logical difficulty can be expressed via the size of their minimal trees. The empirical data, resulting from children playing the Deductive Mastermind game in Math Garden, includes item ratings. In the first analysis of this data we aimed at relating the item ratings to the parameters of the trees. To this end we define a computational method based on the tableau method. The computational method makes two assumptions. First, the formulae are not processed from top to bottom, but instead the order depends on the length of the rule that is associated with the feedback. That is, feedback is processed in the following order: oo, rr, gr, or. Ties are solved by processing the top formula first. Second, the computational method is stopped once a consistent solution is found, assuming that there exists at least one solution. Based on these principles and the tableau method we programmed a recursive algorithm for calculating the type and number of steps until solution for each item is reached. We define 4 measures of theoretically derived item difficulties; the required number of oo, rr, gr, and rr steps, which together might predict item difficulty. Below we will describe the empirically derived item ratings of the 2-pin items and we will show how these relate to the theoretically derived measures of item difficulties.

Method Participants were 28,247 students from grades 1-6, of age: 6-12 years. Together, they played 2,187,354 items between November 2010 and January 2012. From the total of 321 items in the Master Mind game, 100 items have two pins. From these 100 items, 10 items involved 2 colors, 30 items involved 3 colors, 30 items involved 4 colors and 30 items involved 5 colors.

Results To test the relation between empirically derived item ratings and theoretically derived measures of item difficulty we did a regression analysis that includes the basic features of the items (number of colors and number of guesses) and the required number of *oo*, *rr*, *gr*, and *or* analysis steps as predicted by the tableau-based computational algorithm ($F(6,93)$=33, p < .0001, $R2$=.66). All these factors but one (i.e., number of *gr* evaluations) were significant in predicting item difficulties: number of colours (β=1.07, p=.02), number of hypotheses (β=1.75, p<.01), number of *oo* feedbacks ($\beta = -5.1$, p<.001), number of *rr* feedbacks ($\beta = -3.19, p$<.0001), number of *gr* evaluations (ns), number of *or* evaluations (β=1.6, p<.0001). Note that among the rules, only the required number of or steps increases item difficulty. A second aspect of the observed item ratings to explain is the shape of its distribution. The distribution of the item difficulties shows a remarkable property for the 2-pin items (see Appendix, Figure 6). The distribution is bimodal, meaning that there is a cluster of more difficult items (item ratings >0) and a cluster of easier items (item ratings <0). In the cluster of easy items, there are items with 2, 3, 4, and 5 colors. The cluster of difficult items consists of items with 3, 4 and 5 colors. The two clusters could be expressed fully in terms of model-relevant features. It appeared that items are easy in the following two cases: (1) no *or* feedback and no *gr* feedback; (2) no *or* feedback, at least one *gr* feedback, and all colors are included in at least one of the conjectures. Items are difficult otherwise. This shows that or steps make the item relatively difficult, but only if or is required to solve the item. A second aspect that makes an item difficult is the exclusion of one of the colors in the solution from the hypotheses rows.

5 Conclusions and Future Work

In this paper we proposed a way to use a proof-theoretical concept to analyze the cognitive difficulty of logical reasoning. The first hypotheses that we have drawn from the model gave a reasonably good prediction of item-difficulties, but the fit is not perfect. However, it must be noted that several non-logical factors may play a role in the item difficulty as well. For example, the motor requirements to give the answer also introduce some variation in item difficulty, which depends not only on accuracy but also on speed. The minimal number of clicks required to give an answer varies between items. We did not take these aspects into account so far. In particular, the two difficulty clusters observed in the empirical study can be explained with the use of our method.

Our further work can be split into two main parts. We will continue this study on the theoretical level by analyzing various complexity measures that can be obtained on the basis of the tableau system. We also plan to compare the fit with empirical data of the tableau-derived measures and other possible logical formalizations of level difficulty (e.g., a resolution-based model). On the empirical side we first would like to extend our analysis to game-items that consist of conjectures of higher length. This will allow comparing difficulty of tasks of different size and measure the trade-off between the size and the structure of the tasks. We will also study this model in a broader context of other Math Garden games. This would allow comparing individual abilities within Deductive Mastermind and the abilities within other games that have been correlated for instance with the working memory load. Finally, it would be interesting to see whether the children are really using the proposed difficulty order of feedbacks in finding an optimal tableau—this could be checked in an eye-tracking experiment (see [23, 24]).

References

1. Szymanik, J.: Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language. PhD thesis, Universiteit Van Amsterdam (2009)
2. Gierasimczuk, N., Szymanik, J.: Branching quantification v. two-way quantification. Journal of Semantics **26**(4) (2009) 367–392
3. Szymanik, J., Zajenkowski, M.: Comprehension of simple quantifiers. Empirical evaluation of a computational model. Cognitive Science **34**(3) (2010) 521–532
4. Zajenkowski, M., Styła, R., Szymanik, J.: A computational approach to quantifiers as an explanation for some language impairments in schizophrenia. Journal of Communication Disorders **44**(6) (2011) 595–600

5. Van Rooij, I., Kwisthout, J., Blokpoel, M., Szymanik, J., Wareham, T., Toni, I.: Intentional communication: Computationally easy or difficult? Frontiers in Human Neuroscience **5** (2011) 1–18
6. Verbrugge, R., Mol, L.: Learning to apply theory of mind. Journal of Logic, Language and Information **17**(4) (2008) 489–511
7. Knuth, D.E.: The computer as master mind. Journal of Recreational Mathematics **9**(1) (1977) 1–6
8. Irving, R.W.: Towards an optimum Mastermind strategy. Journal of Recreational Mathematics **11** (1978-79) 81–87
9. Koyama, M., Lai, T.: An optimal Mastermind strategy. Journal of Recreational Mathematics **25** (1993) 251—256
10. Kooi, B.: Yet another Mastermind strategy. ICGA Journal **28**(1) (2005) 13–20
11. Chvatal, V.: Mastermind. Combinatorica **325–329** (1983)
12. Greenwell, D.L.: Mastermind. Journal of Recreational Mathematics **30** (1999-2000) 191–192
13. Stuckman, J., Zhang, G.: Mastermind is NP-complete. INFOCOMP Journal of Computer Science **5** (2006) 25–28
14. Van Rooij, I.: The tractable cognition thesis. Cognitive Science **32** (2008) 939–984
15. Szymanik, J.: Computational complexity of polyadic lifts of generalized quantifiers in natural language. Linguistics and Philosophy **33**(3) (2010) 215–250
16. Ristad, E.S.: The Language Complexity Game. The MIT Press (1993)
17. Van der Maas, H.L.J., Klinkenberg, S., Straatemeier, M.: Rekentuin.nl: combinatie Van oefenen en toetsen. Examens (2010) 10–14
18. Elo, A.: The Rating of Chessplayers, Past and Present. Arco (1978)
19. Klinkenberg, S., Straatemeier, M., Van der Maas, H.L.J.: Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. Computers in Education **57** (2011) 1813–1824
20. Beth, E.W.: Semantic entailment and formal derivability. Mededelingen Van de Koninklijke Nederlandse Akademie Van Wetenschappen. Afdeling Letterkunde **18(13)** (1955) 309–342
21. Smullyan, R.: First-order logic. Springer-Verlag, Berlin (1968)
22. Van Benthem, J.: Semantic tableaus. Nieuw Archief voor Wiskunde **22** (1974) 44–59
23. Ghosh, S., Meijering, B., Verbrugge, R.: Logic meets cognition: Empirical reasoning in games. In Boissier, O., Fallah-Seghrouchni, A.E., Hassas, S., Maudet, N., eds.: MALLOW, CUER Workshop Proceedings (2010)
24. Ghosh, S., Meijering, B.: On combining cognitive and formal modeling: a case study involving strategic reasoning. In Van Eijck, J., Verbrugge, R., eds.: Proceedings of the Workshop on Reasoning About Other Minds: Logical and Cognitive Perspectives (RAOM-2011), Groningen, CEUR Workshop Proceedings (2011)

Appendix

The appendix contains three figures. The first one illustrates the educational and research context of the Math Garden system (Figure 5); the second shows the distribution of the empirical difficulty of all items in Deductive Mastermind (Figure 6); and the third gives the frequency of players and game-items with respect to the overall rating (Figure 7). We refer to those figures in the proper text of the article.

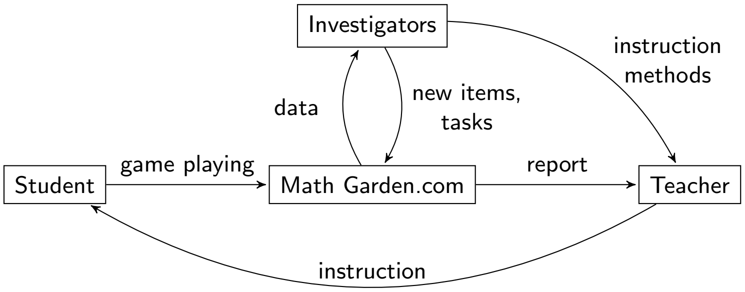


Fig. 5. Math Garden educational and research context

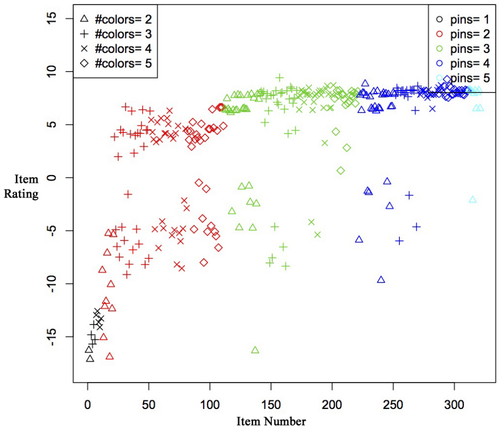


Fig. 6. The distribution of the empirically established game-item difficulties for all 321 game-items in the Deductive Mastermind game. The item number (x-axis) is some arbitrary number of the game-item. The y-axis shows the item ratings. For example, the item presented in Figure 4 is number 23 and its rating is 4.2.

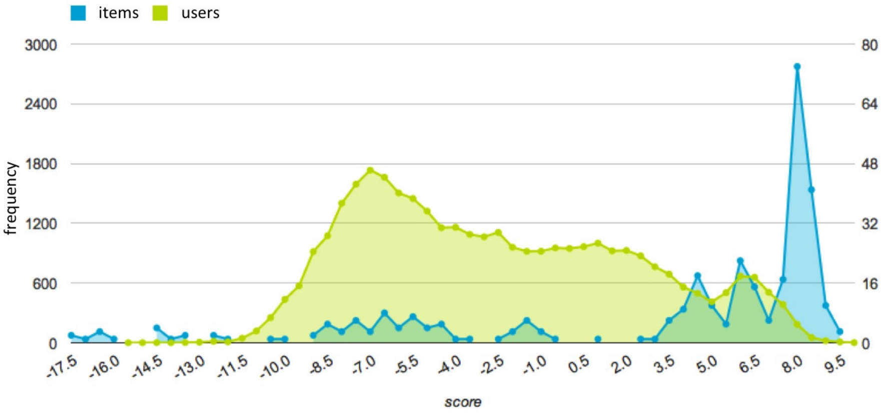


Fig. 7. The frequency of players (green, y-axis on the left) and game-items (blue, y-axis on the left) with respect to the overall rating (x-axis).