J Log Lang Inf (2013) 22:297–314
DOI 10.1007/s10849-013-9177-5

An Analytic Tableaux Model for Deductive Mastermind Empirically Tested with a Massively Used Online Learning System

Nina Gierasimczuk · Han L. J. van der Maas · Maartje E. J. Raijmakers

Published online: 31 July 2013

Abstract The paper is concerned with the psychological relevance of a logical model for deductive reasoning. We propose a new way to analyze logical reasoning in a deductive version of the Mastermind game implemented within a popular Dutch online educational learning system (Math Garden). Our main goal is to derive predictions about the difficulty of Deductive Mastermind tasks. By means of a logical analysis we derive the number of steps needed for solving these tasks (a proxy for working memory load). Our model is based on the analytic tableaux method, known from proof theory. We associate the difficulty of Deductive Mastermind game-items with the size of the corresponding logical trees obtained by the tableaux method. We derive empirical hypotheses from this model. A large group of students (over 37 thousand children, 5–12 years of age) played the Deductive Mastermind game, which gave empirical difficulty ratings of all 321 game-items. The results show that our logical approach predicts these item ratings well, which supports the psychological relevance of our model.

The research of Nina Gierasimczuk is funded by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement no 283963.

The research of Maartje Raijmakers was part of the research project Curious Minds (TalentenKracht, Platform Beta Techniek).

N. Gierasimczuk (✉)
Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam,
The Netherlands
e-mail: nina.gierasimczuk@gmail.com

H. L. J. van der Maas · M. E. J. Raijmakers
Department of Psychology, University of Amsterdam, Amsterdam, The Netherlands

Keywords Deductive Mastermind · Mastermind game · Deductive reasoning · Analytic tableaux · Math Garden (Rekentuin) · Educational tools

1 Introduction and Background

Computational and logical analysis has already proven useful in the explanation of the cognitive difficulty of linguistic and communicative tasks, both theoretically (see Berwick and Weinberg 1984; Cherniak 1986; Barton et al. 1987; Ristad 1993; Szymanik 2009; Van Rooij et al. 2011), and empirically (see Szymanik 2009; Gierasimczuk and Szymanik 2009; Szymanik and Zajenkowski 2010; Zajenkowski et al. 2011). Similarly, we apply formal logical tools to analyze the difficulty of non-linguistic logical reasoning tasks. We study a deductive version of the *Mastermind* game (DMM). The game was implemented in an adaptive, web-based learning system called *Math Garden*, available for children to play at school and at home. We are using the logical approach as a precursor of a computational model—its role is helping to understand the structure of the tasks at hand, and predicting the difficulty of Deductive Mastermind game-items (DMM-items).

In the remaining part of this section we present 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 Sect. 2 we introduce Deductive Mastermind as implemented within Math Garden. Section 3 provides a logical analysis of DMM-items using the analytic tableaux method. In Sect. 4 we draw hypotheses on the basis of our model, and in Sect. 5 we discuss the results. We present a series of statistical analyses on the relation between the empirically established DMM-item difficulties and the predictions according to our logical model. In Sect. 6 we present a preliminary analysis of typical errors made by children on DMM-items. Finally, in Sect. 7 we briefly discuss the directions for future work.

1.1 Mastermind Game

Mastermind is a code-breaking game for two players. It consists of a decoding board, code pegs of k colors, and feedback pegs of red and white (see Fig. 1). 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 the code-breaker making a guess by placing a row of ℓ code pegs, and of the code-maker providing the feedback: a red peg for each code peg of correct color and position, and a white peg for each peg of correct color but wrong position. 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 she obtains the solution within n rounds; the code-maker wins otherwise.

Mastermind is an *inductive inquiry* game that involves *trials of experimentation and evaluation*. The game has been used to investigate the acquisition of complex skills and strategies in the domain of reasoning about others (Verbrugge and Mol 2008). Existing mathematical results on Mastermind focus on finding strategies that allow winning the game in the smallest number of rounds (see Knuth 1977; Irving 1978–79;

Fig. 1 The modern Mastermind Game 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 figure shows the board version of Mastermind game as it is known today (*source* Wikipedia)

Koyama and Lai 1993; Kooi 2005). As far as we know, the deductive reasoning processes involved in Mastermind have not been studied yet.

For the sake of completeness we should also mention a version of the Mastermind game, Static Mastermind (Chvatal 1983), that has been investigated in the context of computational complexity. In Static Mastermind the code-breaker makes a number of initial guesses simultaneously. She then receives feedback on all these guesses at once and has to solve the Mastermind puzzle in the next guess. The goal of Static Mastermind is to determine the minimal number of initial guesses required to solve the Mastermind puzzle in one step. For this game some strategy analysis has been conducted (Greenwell 1999–2000), but, more importantly, the computational complexity of Static Mastermind has been analyzed (Stuckman and Zhang 2006). The corresponding Static Mastermind Decision Problem has been defined in the following way.

Definition 1 *(Mastermind Satisfiability Decision Problem)*

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

Theorem 1 *The 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 (see Ristad 1993; Van Rooij 2008; Szymanik 2010), hence the above theorem might explain why Mastermind is an engaging and popular game. It does not, however, give much insight into the reasoning processes that take place when humans solve Mastermind puzzles.

1.2 Math Garden

Our work has been triggered by the idea of introducing a dedicated logical reasoning training in primary schools through the Math Garden learning system (Rekentuin.nl

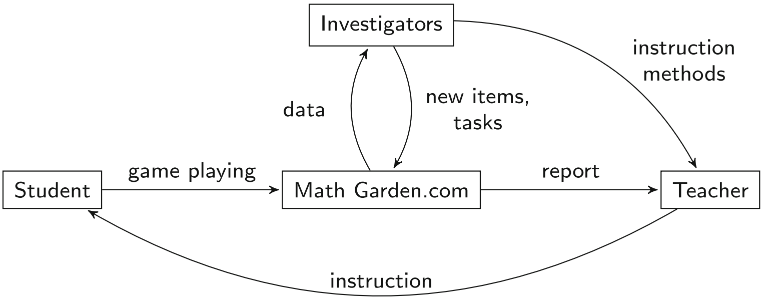


Fig. 2 Math Garden educational and research context. The figure depicts the information flow within the system, in particular the substantial impact of scientific research on the educational process

or MathsGarden.com, see Van der Maas et al. 2010). Math Garden is an adaptive training and monitoring system in which students, mainly primary school children, play various arithmetic games. Currently, it contains 14 arithmetic games and 2 complex reasoning games. The difficulty of game-items is automatically adjusted to the level of the students. A difficultly level is appropriate for a student if she is able to solve 75 % of the game-items of this level correctly. The difficulty of game-items and the level of the students' play are being continuously re-estimated according to a variant of the Elo rating system, which is widely used for calculating the relative skill levels of players in two-player games such as chess (Elo 1978). In Math Garden, the relative skill level is computed on the basis of student versus game-item opposition: students are rated by playing, and items are rated by getting played. The rating depends on accuracy and speed of item responses (Maris and Van der Maas 2012). The rating scales range from $-$ infinity to $+$ infinity. If a student and an item have the same rating then the probability of the student giving the correct answer to the item is expected to be .5. Thus, one result of students playing within Math Garden is an empirical estimation of the difficulty rating of all items. At the same time every student obtains a rating that reflects her or his reasoning ability.[1]

Figure 2 depicts the educational and research context of Math Garden. In four years the number of schools in the Netherlands using Math Garden has grown from 8 to over 700. Currently, over 90,000 students have generated over 200 billion answers to Math Garden items.

2 Deductive Mastermind

Let us now turn to Deductive Mastermind, the game that we designed for the Math Garden system (its name within in Math Garden is *Flowercode*, see Fig. 3).

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*. Hence,

[1] The exact way in which the rating works is described in Klinkenberg et al. (2011).

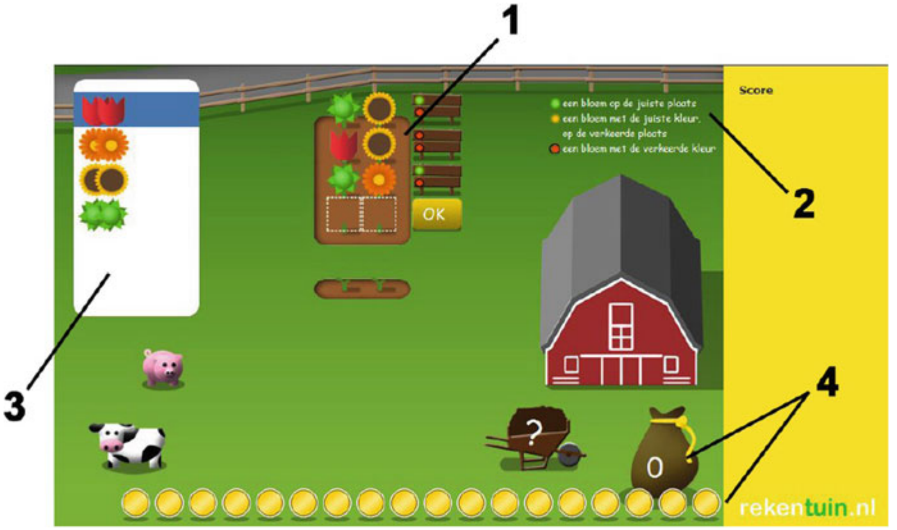

1
2
3
4
Score
een bloem op de juiste plaats
een bloem met de juiste kleur,
op de verkeerde plaats
een bloem met de verkeerde kleur
OK
?
0
rekentuin.nl

Fig. 3 Deductive Mastermind in the Flowercode setting. The figure shows an example of a DMM-item. It 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 correct sequence of flowers on the basis of the clues, consisting of earlier guesses with feedback given on the decoding board. Thus each row of flowers forms a conjecture that is accompanied by a small board on the right side. The dots on this small board code the feedback about the conjecture: one *green dot* for each flower of the correct color and position, one *orange dot* for each flower of correct color but in a wrong position, and one *red dot* for each flower that does not appear in the correct secret sequence at all. In order to win, the player is supposed to pick the right flowers from (*3*) with the mouse, place them in the right order on the decoding board, right under the clues, and click the OK button. She must do so within 60 s i.e., before all the coins (*4*) disappear. If the response is correct, the number of coins that were left as she gave the response is added to her overall score. If her response is wrong, the same number is subtracted. In this way fast incorrect responses are punished, but fast correct responses are encouraged (Klinkenberg et al. 2011; Maris and Van der Maas 2012). (Color figure online)

Deductive Mastermind reduces the complexity of classical Mastermind by changing it from an inductive inference game into a basic logical-reasoning game. 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. Thus 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 phase). Its very basic setting allows access to atomic logical steps of non-linguistic logical reasoning. Moreover, Deductive Mastermind is a single-player game, and hence suitable for the Math Garden system. The simple setting provides educational justification, as the game trains basic logical skills. The game has been running within the Math Garden system since November 2010. It includes 321 DMM-items, with conjectures of various lengths (1–5 flowers; henceforth we will talk about "pins" instead of "flowers") and number of types of

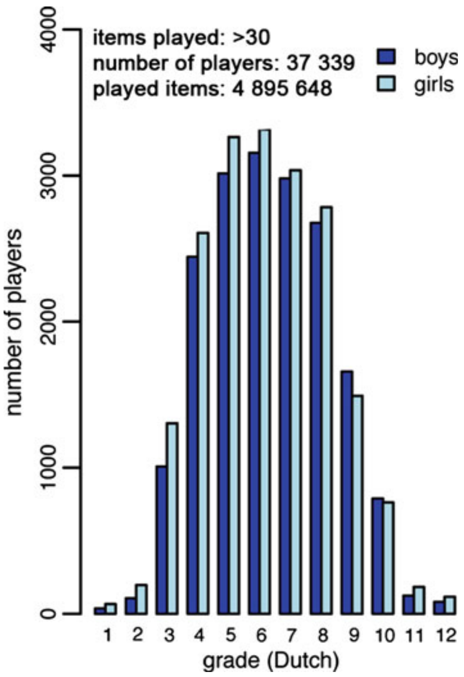


Fig. 4 The figure displays information about the students (grade and sex) playing Deductive Mastermind game. Only students that made at least 30 DMM-items are included. Together they generated almost 5 million item responses

flowers (from 2 to 5; from now on we will call them “colors”). By December 2012, 4,895,648 items had been played by 37,339 primary school students (grades 1–6, age: 6–12 years) in over 700 locations including schools and family homes (see also Fig. 4).

This extensive data-collecting process allows analyzing various aspects of training, e.g., we can access the individual progress of individual students on a single game, or the most frequent mistakes with respect to a DMM-item. Most importantly, due to the student versus item rating system mentioned in Sect. 1.2, we can study the ability ratings of students and difficulty ratings of game-items. Earlier work (Klinkenberg et al. 2011; Maris and Van der Maas 2012) has shown that these ratings are very reliable.

Figure 5 displays the frequency distribution of ratings of students and DMM-items. Multi-modal distributions indicate important qualitative individual differences in responding (Van der Maas and Molenaar 1992). The student distribution is clearly tri-modal. The distribution of 2-pin DMM-item ratings is bi-modally distributed. To further analyze these differences we plotted all DMM-item ratings in Fig. 6. This figure again indicates a gap between easy and difficult items. Interestingly, this gap does occur for different item sets (2–5 pins and different numbers of colors). In the remainder of the paper we will focus on 2-pin Deductive Mastermind game-items. The restriction to the 2-pin items is justified by the large variation of difficulty within this class (see Fig. 6, were 2-pin items are represented in red). For instance, note

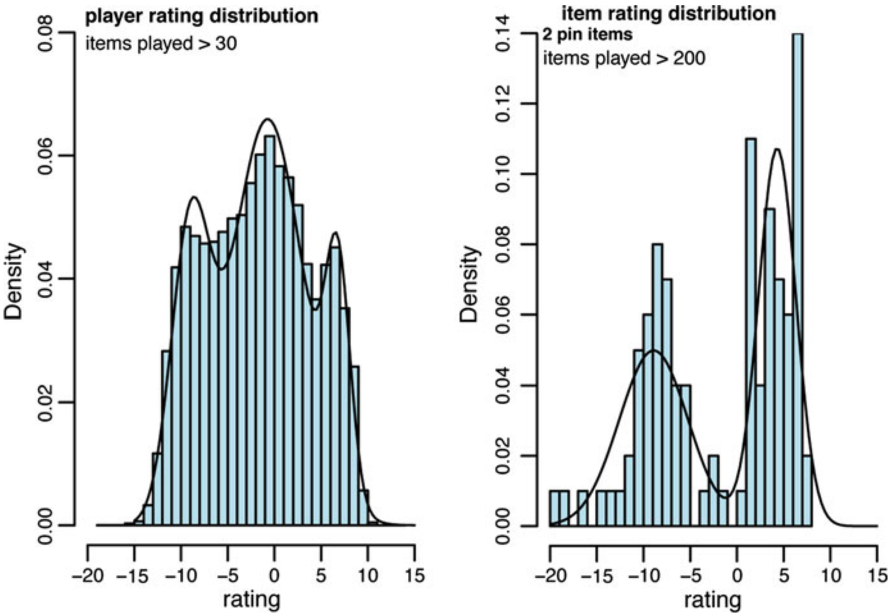


Fig. 5 The figure on the left-hand side shows the distribution of student ratings, which can be fitted with a mixture of three normal distributions, indicating three strategies. The figure on the right-hand side displays the distribution of item ratings for 2 pin items only. This distribution reveals two strategies

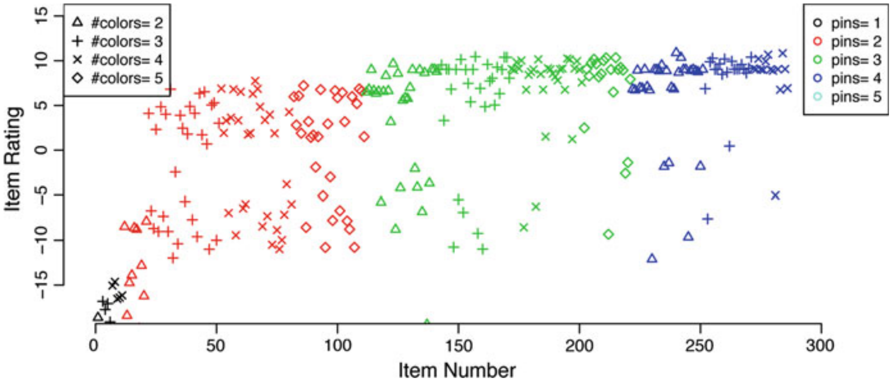


Fig. 6 The distribution of the empirically established difficulty ratings of all DMM-items. Clearly, the domain of items is divided in two groups. The item number (x-axis) is some arbitrary index number of the item. The y-axis shows the item ratings. For example, the item presented in Fig. 9 is number 23 and its rating is 4.2. Note that there are 4 pin items among the easy items, and 2 pin items among the hard items

that 2-pin items with only two colors are always easy whereas some 2-pin items with more than two colors are difficult. To further understand the nature of these qualitative differences in reasoning a logical task analysis is required.

3 A Logical Analysis

Each DMM-item consists of a sequence of conjectures.

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

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 Sect. 2, will be represented by letters g, o, and 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

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

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

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

3.1 The Informational Content of the Feedback

An important question is 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 $goal$ (for any set X, $card(X)$ stands for the number of elements of X):

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

Since the conjecture length, ℓ, is fixed for any DMM-item, it is 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 $i, j \in \{1, \ldots \ell\}$. They can be viewed as propositional variables $p_{i,j}$, for $i, j \in \{1, \ldots \ell\}$. With respect to sets G, O, and R that induce a partition of $\{1, \ldots \ell\}$, we define $\varphi^g_G, \varphi^o_{G,O}, \varphi^r_{G,O}$, the propositional formulae that correspond to different parts of the feedback, in the following way:

- $\varphi^g_G := \bigwedge_{i \in G} h(i) = goal(i) \wedge \bigwedge_{j \in \{1,\ldots,\ell\} \setminus G} h(j) \neq goal(j)$,
- $\varphi^o_{G,O} := \bigwedge_{i \in O}(\bigvee_{j \in \{1,\ldots,\ell\} \setminus G, i \neq j} h(i) = goal(j))$,

– $\varphi^{r}_{G,O} := \bigwedge_{i\in\{1,\dots\ell\}\setminus(G\cup O), j\in\{1,\dots\ell\}\setminus G, i\neq j} h(i) \neq goal(j)$.

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 $\mathbb{G} := \{G | G \subseteq \{1, \dots, \ell\} \wedge card(G) = a\}$, and, if $G \subseteq \{1, \dots, \ell\}$, then $\mathbb{O}^{G} = \{O | O \subseteq \{1, \dots, \ell\} \setminus G \wedge card(O) = b\}$. Finally, we can set $Bt(h, f)$, the Boolean translation of (h, f), to be given by:

$$Bt(h, f) := \bigvee_{G\in\mathbb{G}} \left(\varphi^{g}_{G} \wedge \bigvee_{O\in\mathbb{O}^{G}} \left(\varphi^{o}_{G,O} \wedge \varphi^{r}_{G,O}\right) \right).$$

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:

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

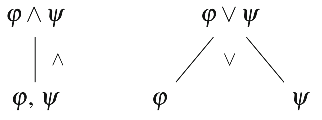
Each DMM-item consists of a sequence of conjectures together with their respective feedbacks. Let us define this formally.

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

Hence, each DMM-item can be viewed as a set of Boolean formulae. Moreover, by the construction of the items we ensure 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 (Beth 1955; Smullyan 1968; Van Benthem 1974). The procedure allows determining 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 by 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 analysis are as follows.[2]



[2] We do not need the rule for negation because in our formulae only propositional atoms are negated.

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	$(goal(1) \neq c_i \wedge goal(2) \neq c_j) \wedge$ $((goal(1) = c_j \wedge goal(2) \neq c_i) \vee (goal(2) = c_i \wedge goal(1) \neq c_j))$

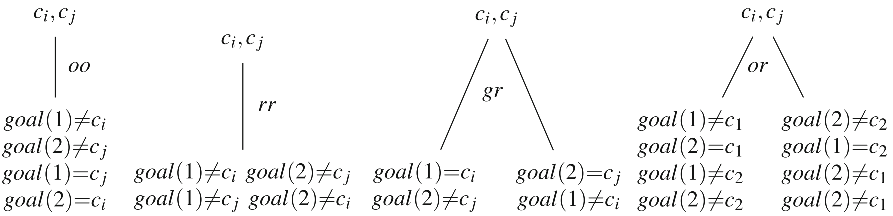


Fig. 7 Formulae and their trees for 2-pin Deductive Mastermind feedbacks

By our construction from Sect. 3.1 we can now conclude that applying the analytic tableaux method to the Boolean translation of a DMM-item will give the unique missing assignment $goal$. In the following section we will restrict our attention to 2-pin DMM-items (where $\ell = 2$), in particular, we will explain the tableau method in more detail for these simple examples. As we argued before, the restriction is justified by the large variation of difficulty within the class of 2-pin items.

2-pin Deductive Mastermind Game-items Since the possible feedbacks consist of three possible values: g (green), o (orange), and r (red), in principle for the 2-pin DMM-items we obtain six possible feedbacks: gg, oo, rr, go, gr, and or. From those gg is excluded as non-applicable and 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 Fig. 7. We can compare the complexity of those feedbacks just by looking at their tree representations created from the boolean translations via the tableau method. As the feedbacks oo and rr are conjunctions, they do not require branching (the other two include disjunctions that 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. Those representations clearly indicate that the order of the tree-difficulty for the four possible feedbacks is: $oo{<}rr{<}gr{<}or$.

As an example, let us consider the DMM-item shown in Fig. 8. The tree on the left stands for the reasoning that 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 obtain that $goal(2){=}c_1$ and $goal(2){\neq}c_1$. Thus we move to the right branch of the first conjecture. This assumption leads to discovering the correct assignment, $goal(1){=}c_2$ and $goal(2){=}c_1$ (note that there is no contradiction in this branch). This tableau procedure requires building the complete tree for the DMM-item. However, this is not always necessary. The right-most part of Fig. 8 shows what

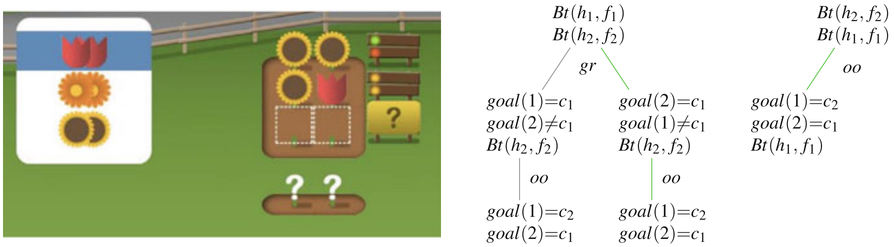


Fig. 8 Comparison of two different trees for one DMM-item. The formalization is as follows: c_1 stands for sunflower, c_2 for tulip; $h_1(1)=c_1$, $h_1(2)=c_1$, $f_1=gr$, etc. *Green* branch gives the correct valuation. The tree on the right hand-side analyzes feedback *oo* first. (Color figure online)

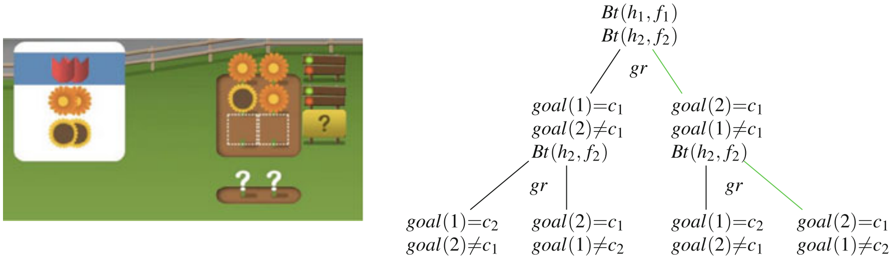


Fig. 9 A tableau for a DMM-item: conjectures processed from top to bottom. Here, c_1 stands for marguerite, c_2 for sunflower, c_3 for tulip. The difficulty rating of this item is 4.2 (see Fig. 6). Compare the size and shape of the tree with Fig. 8

would happen if one chose to start the analysis from the second conjecture. We first apply the feedback *oo* to the second conjecture. We immediately get: $goal(1)=c_2$ and $goal(2)=c_1$: the complete unique assignment with no contradiction. We can stop the computation at this point—if the other conjecture contradicted this assignment, it would mean that the two conjectures must be inconsistent and hence not satisfiable. This in turn would contradict the setting of our game where there is always one correct solution.

The tree might not always give us the complete valuation (flower assignment) explicitly. In some DMM-items it is required to use a flower that did not appear in the conjectures at all. This is the case in the example in Fig. 9; the right-most branch does not give a contradiction, it does assign color 1 to the second position of the goal conjecture. In such a case we draw the remaining color, c_3 (a tulip in the picture), as the missing value at the first position of the goal conjecture, i.e., $goal(1)=c_3$.

4 Predictors of DMM-Item Difficulty

Normatively speaking, the full tree generated by the tableau method for the set of formulae corresponding to a DMM-item, represents an adequate reasoning scheme. Therefore the size of the tree (e.g., the number of nodes) can be thought of as an abstract complexity measure. Obviously, the shape and the size of the tree for each DMM-item depends to some extent on the order in which the lines of the item are processed (see Fig. 8).

The empirical data resulting from students playing the Deductive Mastermind game in Math Garden includes item ratings. In the analysis we aimed at predicting the item ratings with the parameters of the trees. To this end we defined a computational method based on the analytic tableaux method. The computational method makes two assumptions:

1. The formulae are not processed from top to bottom, but instead the order depends on the complexity of rules that are 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.
2. The computational method halts once a consistent solution is found, making use of the fact that there exists one unique solution.

Based on these principles and on the analytic tableaux method, we programmed a recursive algorithm for calculating the type and number of steps taken until the solution is reached.[3] The algorithm results in 4 measures of DMM-item difficulty. They are the required number of oo, rr, gr, and or application steps. Below we test how well those measures predict the difficulty ratings of 2-pin DMM-items. In addition, the importance of the two above-listed assumptions is tested.

5 Results

For the analyses we used the data of January 2012. Participants were 28,247 students (see Fig. 4). Together, they played 2,187,354 DMM-items between November 2010 and January 2012. The main results were replicated with the data of December 2012. From the total of 321 items in DMM, 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. The distribution of difficulty ratings of these items is displayed in Fig. 5.

The regression model is based on four measures obtained from the tableau method model: the required number of oo, rr, gr, and or steps (in the tables denoted as oo, rr, gr, and or). The simplest regression model of item difficulty only includes basic item characteristics. These are the number of colors (col), the number of rows (rows), and whether or not all colors are included in at least one of the conjectures (allcolinitem). In Table 1 this model is included as Model 0. All three predictors contribute significantly, explaining 34 percent of the variance in difficulty ratings. Model 1 extends Model 0 with the measures that follow from the analytic tableaux method. The regression weights of all these measures (except gr) have statistically significant values. The explained variance is much higher for this model ($R^2 = .75$). Note that the number of colors was an important predictor in Model 0, but not in Model 1, in which the tableau-derived measures are used. The significance of the number of colors factor in Model 0 was caused by the fact that 2-pin items with

[3] The recursive algorithm predicts difficulty of an item that corresponds to the minimal tableau build according to the following rules: each line is read from left to right and feedbacks are analyzed in the order oo, rr, gr, and or. The abstract complexity measure then corresponds to the number of nodes in such a tableau.

Table 1 Parameter estimates of the tableau regression models

	Model 0	Model 1	Model 2	Model 3	Model 1'
(Intercept)	−9.21***	−7.55***	−8.64***	−8.14***	−6.34***
	(2.65)	(1.72)	(1.89)	(1.72)	(1.44)
col	1.73**	0.29	0.57	0.19	0.65
	(0.72)	(0.50)	(0.55)	(0.52)	(0.41)
rows	1.52*	3.16***	2.47***	8.30***	2.33***
	(0.81)	(0.69)	(0.84)	(1.06)	(0.58)
allcolinitem	−5.88***	−5.83***	−6.00***	−6.43***	−3.89***
	(1.22)	(0.90)	(0.97)	(0.91)	(0.75)
oo		−2.47*			−2.95**
		(1.44)			(1.20)
rr		−3.56***			−3.14***
		(0.69)			(0.57)
gr		0.12			0.04
		(0.25)			(0.21)
or		2.23***			1.89***
		(0.35)			(0.29)
$ooUO$			−0.41		
			(1.40)		
$rrUO$			−1.94***		
			(0.62)		
$grUO$			0.45		
			(0.29)		
$orUO$			2.47***		
			(0.38)		
$ooFB$				−5.09***	
				(1.13)	
$rrFB$				−8.03***	
				(0.71)	
$grFB$				−4.71***	
				(0.71)	
R^2	0.34	0.75	0.70	0.75	0.75
Adj. R^2	0.36	0.73	0.68	0.73	0.73
Num. obs.	100	100	100	100	100

Model 0: Basic model with simple item characteristics; Model 1: Tableau Method Model; Model 2: Tableau model based on unordered measures (UO); Model 3: Tableau model based on number of feedbacks (FB); Model 1': Model 1 fitted on data of 11 months later

*** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$

only 2 colors are always easy (see Fig. 6). The tableau measures explain this fact in another way, a way in which the number of colors is not a required factor anymore (see below).

Table 2 Correlations between measures of the three Tableau Method Models

	ooFB	*rrFB*	*grFB*	*orFB*	*ooUO*	*rrUO*	*grUO*	*orUO*
oo	1.00	−0.13	−0.35	−0.30	1.00	−0.16	−0.29	−0.28
rr	−0.33	0.94	−0.08	−0.26	−0.33	0.91	−0.04	−0.24
gr	−0.36	−0.17	0.79	−0.21	−0.36	−0.06	0.96	−0.10
or	−0.28	−0.27	−0.35	0.90	−0.28	−0.23	−0.26	0.96

Note the high correlations between predictors (measures) of the different models

To test the importance of the two assumptions of the tableau-generating algorithm, we fitted two additional regression models. First, we tested whether the ordering of the formulae before solving the item is an important aspect of the algorithm. To this end, we applied the recurrent algorithm without first ordering the formulae. This leads to new measures (denoted as *ooUO, rrUO, grUO*, and *orUO*). The regression model (Model 2) with these measures also explains the data well, although less good than Model 1 ($R^2 = .70$). Second, we tested the importance of the assumption to stop the algorithm once a consistent solution is found (i.e., we calculated the frequency of each feedback type for each item). The measures of this regression model (Model 3) are denoted as *ooFB, rrFB, grFB*, and *orFB*.[4] This model fits equally well as Model 1 ($R^2 = .75$). The measure $orFB$ can be directly computed from $ooFB$, $rrFB$, $grFB$ and the number of rows, and is therefore not estimated.

A further statistical comparison of models 1, 2, and 3 is hampered by the high correlations between the measures of models 1, 2, and 3. A model with all these measures suffers from multicollinearity (see Table 2). This means that a better test of the assumptions requires a new set of items, created such that the correlations of the measures between these three tableaux-based models are decreased. The last regression model (Model 1') is a replication of Model 1 on the item ratings of the same items, 11 months later in Math Garden. The results of Model 1 and Model 1' are very similar, indicating stable results. Overall, the increase in fit from Model 0 to 1 is impressive and demonstrates the usefulness of the analytic tableaux method.

The bimodal item rating distribution in Fig. 5 indicated two types of items: easy and hard. It turned out 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 an item relatively difficult, but only if analyzing the *or* feedback is necessary to solve the item. The second aspect that makes an item

[4] In general, the tableau-generating algorithm could be halted either after computing the full tree or immediately after finding a consistent solution. In the first case all conjectures are evaluated and hence $ooFB$, $rrFB$, and $grFB$ feedbacks would be the best predictors. In the second case only the non-redundant conjectures are evaluated, whose number can be less than the total number of conjectures. The second case would allow solving an item with many conjectures much faster, resulting in a lower item rating. In this case oo, rr, gr, and or would be better predictors.

difficult is the exclusion of one of the colors in the solution from the hypotheses rows. Figure 6 shows that 2-pin items with only 2 colors are always easy, which is not the case for ($>$2)-pin items. This follows directly from the criterion of easiness, 2-pin items with only two colors, for logical reasons, always fall under one of the two easy cases.[5]

The trimodal distribution of student ratings is related to this classification in easy and difficult DMM-items. The first component consists of students that apparently do understand the task. If we select only students that play a minimum number of items (for instance 30), this component disappears from the distribution. The two remaining components represent students that can only solve the easy items and students that can solve both easy and difficult items.

6 Towards the Error Analysis

As the Math Garden system allows access to students' responses, it is possible to analyze the frequencies of particular responses to each item and search for error patterns. As it turns out, the frequencies of typical answers are consistent with the logical analysis of DMM-items. First of all, the most common erroneous responses are structurally the same within the items that have the same tableau representation. This consistency in error types indicates that indeed logical structure of tasks plays a main role in solving the DMM-items. Secondly, within items that are similar in difficulty and in overall structure, different tableaux representations correlate with differences in the most frequent errors (see Fig. 10).

The frequencies of certain error types change with age (again see Fig. 10). The logical analysis sheds light on the strategies students use to obtain certain types of answers. This in turn contributes to a better understanding of the learning process. For instance, we can investigate the intermediate, partially correct strategies students apply before they adopt the correct strategy. Knowledge of these strategies might help to adapt instruction to individual students (Van Bers et al. 2011; Schmittmann et al. 2012).

[5] Note that the DMM-items are difficult if: 1) a line with *or* feedback needs to be analyzed in order to find the solution; or 2) a line with *gr* feedback needs to be analyzed and one color that is in the solution was not included in any line of the DMM-item. We will show that none of those cases is possible within 2-pin DMM-items of 2 colors. First let us consider 1: In the 2-pin DMM-items there are two possibilities, a line consists of either (a) two pins of the same color, or (b) two pins of different colors. If (a) is the case the *or* feedback implies that one of the pins is of the correct color but in a wrong position, which does not make sense, because the other pin is of the same color and in the only remaining position. If (b) is the case the *or* implies means that one pin is not in the solution, which implies that the two pins have to be of the same color (since there are only two colors). But then it means that one of the pins in the line was in fact of the correct color and position to start with, which was not reflected by an adequate *g* feedback. This renders such a case impossible. Let us now consider 2: Here, a line must consist of two flowers of the same color (otherwise there would be no color that is not included in any of the lines, because there are only 2 colors). But then, the *gr* feedback does not give a unique solution. The latter is only possible if another line introduces the missing color. In conclusion, the properties of 2-pin DMM-items of only 2 colors make it impossible for any of them to fall into the "difficult" category.

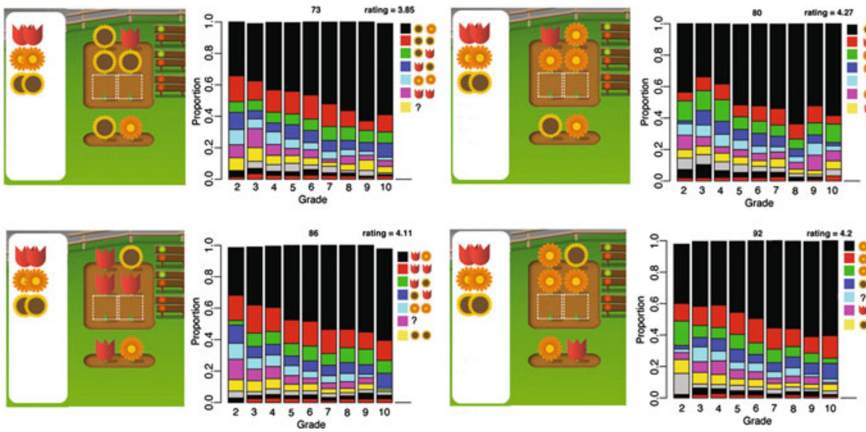


Fig. 10 The figure displays the frequencies of various answers to four similar DMM-items. All items have the same logical structure (the same reasoning trees) and display very similar error patterns, see items 73, 86, and 92. In terms of structure, item 80 (*top-right corner*) is a mirror image of the remaining ones. Although similar, it clearly displays a different error pattern. This can be explained via the respective tableaux. The tableau for item 80 requires more steps and thus a higher memory load

7 Discussion and Future Work

In this paper we proposed a proof-theoretical method to perform a so-called task analysis of logical reasoning items in a deductive reasoning task. The measures that we have drawn from the logical task analysis predicted the empirical difficulty of DMM-item very well. However, it must be noted that several non-logical factors may play a role in the item difficulty as well. For example, the hand-mouse movements required in answering items also introduce some variation in item difficulty, which depends not only on accuracy but also on speed. We did not take these aspects into account so far.

With the analytic tableaux method it might be possible to analyze whether and in what way the students apply reasoning strategies. We learn to understand how students mentally manipulate with the elements of the task in order to optimize the size of the reasoning tree (i.e., the length of the computation). With a strategy assessment methodology (Jansen and Maas 1997) we expect to be able to analyze individual differences in strategy choices.

Our future work will consists of two lines. 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 of the tableau-derived models with other possible logical formalizations of item difficulty. Empirically, we will first extend our analysis to DMM-items that consist of conjectures of greater length (i.e., with more that two pins). This will allow us to compare the difficulty of items of different length and to analyze the trade-off between the length and logical structure of the items. This theoretical and empirical work focusses on the quantitative differences between items. We started however with the observation that the ratings of 2-pin items are bimodally distributed, suggesting that there exist two subgroups of students. We also intend to study whether these subgroups of the students really use the proposed difficulty order of feedbacks in finding an optimal tableau. This can be investigated in an eye-tracking experiment (see Ghosh et al. 2010; Ghosh and Meijering 2011; Meijering et al. 2012). Last but not least, we want to see how our logically grounded analysis of cognitive difficulty extends to other reasoning tasks. In this context we plan to study other logical games, like Sudoku (see Pelánek 2011). We hope that with our logical analysis we can eventually develop a psychologically relevant computational model for logical, step-by-step reasoning.

References

Barton, E., Berwick, R., & Ristad, E. (1987). *Computational complexity and natural language*. Cambridge, MA: The MIT Press.

Berwick, R., & Weinberg, A. (1984). *The grammatical basis of linguistic performance*. Cambridge, MA: The MIT Press.

Beth, E. W. (1955). Semantic entailment and formal derivability. *Mededelingen van de Koninklijke Nederlandse Akademie van Wetenschappen Afdeling Letterkunde*, *18*(13), 309–342.

Cherniak, C. (1986). *Minimal rationality*. Cambridge, MA: The MIT Press.

Chvatal, V. (1983). Mastermind. *Combinatorica*, *3*, 325–329.

Elo, A. (1978). *The rating of chessplayers, past and present*. Arco.

Ghosh, S., & Meijering, B. (2011). On combining cognitive and formal modeling: A case study involving strategic reasoning. In J. Van Eijck, & R. Verbrugge (Eds.) *Proceedings of the workshop on reasoning about other minds: Logical and cognitive perspectives (RAOM-2011), Groningen, The Netherlands, July 11th, 2011, CEUR-WS.org. CEUR Workshop Proceedings, vol751* (pp. 79–92).

Ghosh, S., Meijering, B., & Verbrugge, R. (2010). Logic meets cognition: Empirical reasoning in games. In O. Boissier, A. E. Fallah-Seghrouchni, S. Hassas & N. Maudet (Eds.) *Proceedings of the multi-agent logics, languages, and organisations federated workshops (MALLOW 2010), Lyon, France, August 30–September 2, 2010, CEUR-WS.org, CEUR Workshop Proceedings, vol*, *627* (pp. 15–34).

Gierasimczuk, N., & Szymanik, J. (2009). Branching quantification v. two-way quantification. *Journal of Semantics*, *26*(4), 367–392.

Greenwell, D. L. (1999–2000). Mastermind. *Journal of Recreational Mathematics, 30*, 191–192.

Irving, R. W. (1978–79). Towards an optimum Mastermind strategy. *Journal of Recreational Mathematics, 11*, 81–87.

Jansen, B., & Van der Maas, H. (1997). A statistical test of the rule assessment methodology by latent class analysis. *Developmental Review*, *17*, 321–357.

Klinkenberg, S., Straatemeier, M., & Van der Maas, H. L. J. (2011). Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers in Education*, *57*, 1813–1824.

Knuth, D. E. (1977). The computer as master mind. *Journal of Recreational Mathematics*, *9*(1), 1–6.

Kooi, B. (2005). Yet another Mastermind strategy. *ICGA Journal*, *28*(1), 13–20.

Koyama, M., & Lai, T. (1993). An optimal Mastermind strategy. *Journal of Recreational Mathematics*, *25*, 251–256.

Maris, G., & Van der Maas, H. L. J. (2012). Speed-accuracy response models: Scoring rules based on response time and accuracy. *Psychometrika*, *77*(4), 615–633.

Meijering, B., van Rijn, H., Taatgen, N. A., & Verbrugge, R. (2012). What eye movements can tell about theory of mind in a strategic game. *PLoS One, 7*(9), e45961

Pelánek, R. (2011). Difficulty rating of Sudoku puzzles by a computational model. In R. C. Murray & P. M. McCarthy (Eds.), *Proceedings of the twenty-fourth international Florida artificial intelligence research society conference, May 18–20, 2011*. Palm Beach, Florida, USA: AAAI Press.

Ristad, E. (1993). *The language complexity game*. Cambridge, MA: The MIT Press.

Schmittmann, V. D., Van der Maas, H. L. J., & Raijmakers, M. E. J. (2012). Distinct discrimination learning strategies and their relation with spatial memory and attentional control in 4- to 14-year-olds. *Journal of Experimental Child Psychology*, *111*(4), 644–62.

Smullyan, R. (1968). *First-order logic*. Berlin: Springer.

Stuckman, J., & Zhang, G. (2006). Mastermind is NP-complete. *INFOCOMP Journal of Computer Science*, *5*, 25–28.

Szymanik, J. (2009). *Quantifiers in time and space. Computational complexity of generalized quantifiers in natural language*. PhD thesis, Universiteit van Amsterdam.

Szymanik, J. (2010). Computational complexity of polyadic lifts of generalized quantifiers in natural language. *Linguistics and Philosophy*, *33*(3), 215–250.

Szymanik, J., & Zajenkowski, M. (2010). Comprehension of simple quantifiers. Empirical evaluation of a computational model. *Cognitive Science*, *34*(3), 521–532.

Van Benthem, J. (1974). Semantic tableaus. *Nieuw Archief voor Wiskunde*, *22*, 44–59.

Van Bers, B. M. C. W., Visser, I., Van Schijndel, T. J. P., Mandell, D. J., & Raijmakers, M. E. J. (2011). The dynamics of development on the dimensional change card sorting task. *Developmental Science*, *14*(5), 960–71.

Van der Maas, H., & Molenaar, P. (1992). Stagewise cognitive development: An application of catastrophe theory. *Psychological Review*, *99*(3), 395–417.

Van der Maas, H., Klinkenberg, S., & Straatemeier, M. (2010). Rekentuin.nl: Combinatie van oefenen en toetsen. *Examens*, *4*, 10–14.

Van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science*, *32*, 939–984.

Van Rooij, I., Kwisthout, J., Blokpoel, M., Szymanik, J., Wareham, T., & Toni, I. (2011). Intentional communication: Computationally easy or difficult? *Frontiers in Human Neuroscience*, *5*, 1–18.

Verbrugge, R., & Mol, L. (2008). Learning to apply theory of mind. *Journal of Logic, Language and Information*, *17*(4), 489–511.

Zajenkowski, M., Styła, R., & Szymanik, J. (2011). A computational approach to quantifiers as an explanation for some language impairments in schizophrenia. *Journal of Communication Disorders*, *44*(6), 595–600.