Computational and algorithmic models of strategies in turn-based games

Gerben Bergwerff
Institute of Artificial Intelligence, University of Groningen

Ben Meijering (ben@meijering.com)
Institute of Artificial Intelligence, University of Groningen

Jakub Szymanik (jakub.szymanik@gmail.com)
Institute for Logic, Language and Computation, University of Amsterdam

Rineke Verbrugge (L.C.Verbrugge@rug.nl)
Institute of Artificial Intelligence, University of Groningen

Stefan M. Wierda (wierda.stefan@gmail.com)
Institute of Artificial Intelligence, University of Groningen

Abstract

We study two different models of a turn-based game called the Marble Drop Game, which is an experimental paradigm designed to investigate higher-order social reasoning. Our first model is a computational-level description of the game, associating cognitive difficulty of a game trial with its structural properties. Our second model is an algorithmic-level model postulating a forward reasoning plus backtracking strategy for solving the game, rather than backward induction as prescribed by game theory. Our experiment shows that the algorithmic-level model is more predictive for the participants' reaction times. This research illustrates how various methods of logic and computer science may be used for building computational cognitive models.

Keywords: cognitive difficulty; strategic games; higher-order social reasoning; theory of mind

Introduction

Theory of mind (ToM; Premack & Woodruff, 1978) is the ability to attribute beliefs, desires, and intentions to others. It is a widely studied phenomenon in the fields of psychology, neurosciences, philosophy, and logics. Despite the wide interest in ToM, relatively little research has concentrated on the complexity of the underlying cognitive strategies (Apperly, 2011).

We speak of first-order reasoning in ToM when a person attributes a simple belief, desire, or intention to someone else. For example, imagine Ingrid and Rob interacting. If Rob thinks "Ingrid *knows* that it is snowing", he makes a first-order ToM attribution. However, if the situation becomes more complex, first-order ToM reasoning is not sufficient. When Ingrid thinks "Rob *knows* that I *know* that it is snowing", she makes a second-order attribution.

One way of studying the cognitive basis of ToM in a controlled experimental setting is the use of competitive and collaborative games. By investigating the underlying strategies used during these games, one can shed light upon the underlying cognitive processes involved in this game—including ToM reasoning. In these games, the experimenter can control the order of reasoning required to play the game successfully by selecting the instances of the game.

It has turned out in recent years that logic and the computational sciences can help to delineate the complexity of cognitive tasks, which in turn helps to explain human cognition in general, and human cognitive strategies in particular. Predictions based on computational analyses can be fruitfully compared with empirical evidence. For an overview of this emerging field of research, including several examples of research on theory of mind combining computational and empirical methods, see (Isaac, Szymanik, & Verbrugge, 2014). It appears that turn-taking games form an especially successful application area in which to bring logic and computation to the lab (Szymanik, 2013).

When analyzing cognition from a computational point of view, it is useful to distinguish the levels of analysis of the cognitive task at hand, as proposed by Marr (1983). In this paper, we investigate the cognitive task of making a decision in a particular turn-taking game. We will propose an analysis on Marr's *computational* level, which concerns the problem solved or the function computed, as well as an analysis on the *algorithmic* level, which concerns the particular way of achieving a solution to the problem.

In the current paper, we will focus on the marble drop game—a two-player game in which the players have to take into account the actions, beliefs and goals of the other.

The marble drop game

The marble drop game is a strategic two-player game that has been used to study theory of mind (Meijering, Van Rijn, Taatgen, & Verbrugge, 2011; Meijering, van Rijn, Taatgen, & Verbrugge, 2012; Raijmakers, Mandell, van Es, & Counihan, 2014). Just like well-known games such as poker and bridge, marble drop is turn-based. However, marble drop is a perfect information game, in contrast with poker and bridge, in which players cannot see the others' cards. In the game, each player is assigned a color (orange or blue). Then, a marble is dropped onto trapdoors that are controlled by one of the players: If the trapdoor is blue, the blue player controls the trapdoor; if the trapdoor is orange, the orange player controls the trapdoor. A trapdoor leads either to another trapdoor or to a bin containing marbles. Each bin contains a number of blue and orange marbles[1]. The number of marbles of the player's own color determines his pay-off. In the marble drop games discussed in this paper, there are always four bins and four possible pay-offs, i.e., {1, 2, 3, 4}. The goal of this game for a player is to obtain as many points as possible, irrespective of the score of the other player.

An example trial of the marble drop game is shown in Figure 1. In this particular trial, the blue (dark grey) player's highest number of marbles is in bin 3; for the orange (light grey) player, the highest number of marbles is in bin 4.

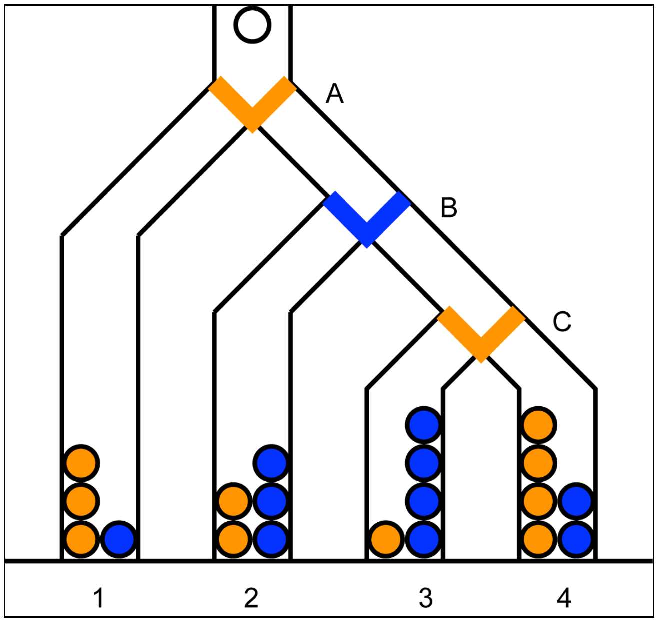

Figure 1: A trial of the marble drop game that requires second-order reasoning: The orange (light grey) player has to reason about which side of trapdoor C the blue (dark grey) opponent *believes* that the orange player himself *intends* to open.

Backward induction, the process of reasoning backwards from the end to determine a sequence of optimal actions, will yield the optimal solution for this game (Aumann, 1995). Although there is a lively debate among game theorists about the question whether common knowledge of rationality in a game of perfect information entails the backward induction solution (see Ghosh et al. 2014 for an overview), game theory textbooks generally propose backward induction as the standard solution (e.g., Osborne & Rubinstein, 1994). For marble drop trials with four bins and three trapdoors that have been used in the experiment, the backward induction solution will always be found in 6 steps, where each step consists of attending to one pay-off. In the example game of Figure 1, the orange player would perform backward induction as follows: check and compare the numbers of orange marbles in bins 3 and 4 (two steps); the number in bin 4 is higher, so now check and compare the numbers of blue marbles in bins 2 and 4 (two steps); the number in bin 2 is higher, so now check and compare the numbers of orange marbles in bins 1 and 2 (two steps); the number in bin 1 is higher, so open the left-hand side of trapdoor A. The total number of steps is 2+2+2=6. It is not hard to see that this is irrespective of pay-off structure.

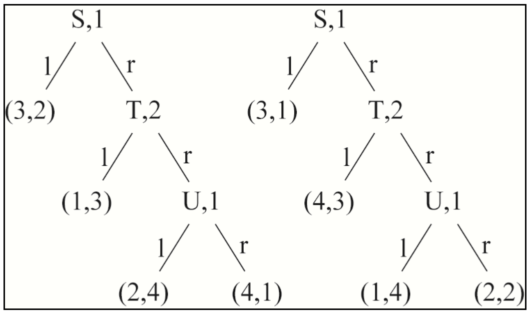

Figure 2: Pay-off structures of two types used in the experiment. The left and right numbers in the leaves correspond to the pay-off of the player and opponent, respectively. The numbers after S, T, and U represent whether it is the player's turn (1) or the opponent's (2).

However, backward induction is not the only possible reasoning strategy. Meijering et al. (2012) investigated whether participants used this *backward induction* strategy, which is indeed in general the most optimal way to play the marble drop game, or if participants rather used the so-dubbed *forward reasoning plus backtracking* strategy. See the next section for an explanation how that strategy works for the example games represented in Figure 2.

Forward reasoning plus backtracking appears at first sight to be a suitable candidate because of the prevalence of a forward causal or temporal direction in human reasoning. Moreover, for most pay-off structures that can occur in theory, the solution can be found with forward reasoning plus backtracking in a smaller number of steps than when backward induction were used (see Szymanik, Meijering, and Verbrugge, 2013 for a simulation showing this). For example, if bin 3 had contained 4 blue and 4 orange marbles, the forward reasoning plus backtracking algorithm would only have taken four steps: "attend the number of marbles in bin 1, 2, 3 consecutively (three steps) and conclude that bin 3 contains the maximum possible number of 4 blue marbles; then check that it also contains the maximum number, 4, of orange marbles (fourth step), so choose the right-hand side of trapdoor A".

In their study, Meijering et al. (2012) recorded eye-fixations while participants were playing the marble drop game. Next, they analyzed the fixation patterns and compared the found patterns to predicted patterns by either the backward induction strategy or the forward reasoning

[1] Note that in the original game, the hue of the color of the marbles in the bin determined its payoff. However, for the convenience of the reader, we will use the number of marbles.

plus backtracking strategy. The eye-fixation data suggested that participants were using the forward reasoning plus backtracking strategy more than backward induction.

Forward reasoning plus backtracking

As suggested by the name of the strategy, forward reasoning plus backtracking is a combination of forward reasoning and backward reasoning. In principle, forward reasoning alone can yield a very fast solution if the highest value for the blue player is in bin 1—there is no need for ToM reasoning in that case. However, all items used in the experiment of Meijering et al. (2012) were carefully picked such that they all required second-order ToM reasoning in order to obtain the highest possible pay-off. In these items, backtracking is required to predict the succeeding action of the opponent in order to determine whether the highest possible pay-off (4) is accessible. A player who employs this strategy starts at the top trapdoor and tries to find out which trapdoor to open to obtain the highest pay-off, and then uses backward reasoning to find out whether that bin is reachable.

Szymanik et al. (2013) investigated the use of the forward reasoning plus backtracking strategy by looking at the reaction times obtained by Meijering et al. (2012). Szymanik et al. used an *ad hoc* forward reasoning plus backtracking algorithm that had been used by Meijering et al. to create fixation-patterns for their eye-tracking analyses of the 16 item types used in their experiments. The algorithm was then used to predict the number of decision steps necessary for each type of trial. Szymanik et al. found that the number of steps as calculated by the algorithm indeed predicted the reaction times. In the current paper, we present a more general forward reasoning plus backtracking algorithm that can be applied to any binary turn-taking (extensive form) game tree (see Algorithm 1).

Structural Complexity of Game Trees

Inspired by the work of Szymanik (2013) on the computational complexity of solving finite extensive-form turn-taking games, Szymanik et al. (2013) investigated possible computational-level explanations of the marble drop task. They introduced a method to quantify the difficulty of a marble drop trial that was constructed such that it is independent from particular algorithmic-level implementations. In their study, they proposed to look at the structure of marble drop game trials. The main idea is to quantify the complexity of the corresponding game trees with respect to the number of alternations between two players. The intuition is that every alternation potentially corresponds to the next level of higher-order ToM reasoning. Therefore, the difficulty of the game should increase with the number of alternations. Additionally, the pay-off distribution must be taken into account, because many alternations may be simply ignored by the players if reasoning about them clearly does not lead to better pay-offs. Let us give a reminder of the definitions.

Definition 1 *Let us assume that players {1,2} strictly alternate in the game; Let player i* ∈ {1,2}. Then:

- In a Λ^{i}_{1} tree, all nodes are controlled by Player *i*.
- *A* Λ^{i}_{k+1} *tree, a tree of k-alternations for some k ≥ 0, starts with a player I node.*

Note that all 16 game trees corresponding to item types used in the experiments of Meijering et al. (2012) are Λ^{1}_{3} trees.

Definition 2 *A game T is generic, if for each player, distinct end nodes have different pay-offs.*

Note that all 16 item types in the experiments of Meijering et al. (2012) are generic games.

Definition 3 *Suppose i* ∈ {1,2}. *If T is a generic game tree with the root node controlled by Player 1 and n is the highest possible pay-off for Player 1, then* T^{-} *is the minimal sub-tree of T containing the root node and the node with pay-off n for Player i.*

For example, consider both Λ^{1}_{3} trees from Fig. 2. Taking the minimal sub-trees containing the root node and the node with pay-off 4 for Player 1 yield a Λ^{1}_{3} sub-tree for the item 1 and a Λ^{1}_{2} sub-tree for item 3 (also see Szymanik et al., 2013 for more explanations).

The levels of lambda-difficulty of reduced trees T^{-} (later "lambda-difficulty") was indirectly tested by comparing trials in which the highest pay-off was accessible to trials in which the highest pay-off was not accessible (Szymanik et al., 2013). The rationale behind this test was that non-accessible trials would generally include more alternations and would therefore be more difficult. Indeed, it turned out that the non-accessible trials took more time to complete than the accessible trials. However, Szymanik et al. did not investigate the direct relation between the structural difficulty of the reduced trees and the reaction times.

The current study builds on the work of Szymanik et al. (2013). Now for the first time we directly explore the use of the lambda-difficulty of the reduced trees. In addition, we introduce an algorithmic-level explanation, namely the forward reasoning plus backtracking algorithm. The predictive power of both the structural lambda-difficulty and forward reasoning plus backtracking strategy are investigated. Thus, two hypotheses can be formulated.

H_1: Is lambda-difficulty of reduced game trees predictive for the reaction time of the marble drop game?

H_2: Is the forward reasoning plus backtracking strategy predictive for the reaction time of the marble drop game?

Implementation

Forward reasoning + backtracking algorithm

Algorithm 1 shows the implementation of the forward reasoning plus backtracking strategy as used in the current study. The algorithm computes the number of *attentional* *steps* (henceforth referred to as steps). The steps are computed by counting the number of times a value gets attended. For example, comparing two values in bins of the marble drop game would be counted as two steps, because both values need to be attended for the comparison.

Algorithm 1. The following algorithm computes the number of forward reasoning plus backtracking steps, where m is the number of nodes, P_n is the pay-off for the player at node n, and O_n is the pay-off for the opponent at node n.

```
Require: $P_n \in \{1 : m\}$ and $O_n \in \{1 : m\}$
Ensure: all $P_n$ are unique and all $O_n$ are unique
n ← 1 {start with forward reasoning at the first node}
Steps ← 1
while not max $P_n$ do
    n ← n + 1 and Steps ← Steps + 1 {While the highest pay-off is not found continue with the next node}
    if max $P_n$ and max $O_n$ then
        Steps ← Steps + 1 {Do not backtrack if the highest pay-off of both players is in this node}
        return Steps
    end if
end while
High ← n {Remember the node with the highest pay-off}
Back ← m
n ← m - 1 {Start backtracking at the last two nodes}
while Back ≠ High and n > 0 do
    if trapdoor(n) = player then
        if $P_{Back} > P_n$ then
            Back ← Back {Node Back has the highest pay-off for the player, therefore the nodes can be substituted by node Back}
        else if $P_{Back} < P_n$ then
            Back ← n {Node n has the highest pay-off for the player, therefore the nodes can be substituted by node n}
        end if
    else if trapdoor(n) = opponent then
        if $O_{Back} > O_n$ then
            Back ← Back {Node Back has the highest score for the opponent, therefore the nodes can be substituted by node Back}
        else if $O_{Back} < O_n$ then
            Back ← n {Node n has the highest pay-off for the opponent, therefore the nodes can be substituted by node n}
        end if
    end if
    n ← n - 1
    Steps ← Steps + 2 {There are two pay-offs being compared, hence this takes 2 steps}
end while
return Steps {Return the number of Steps for forward reasoning plus backtracking}
```

As an example, we will walk through two items that were actually presented in the game experiment (see Figure 2).

Item 1 At first, the player attends all leaves until she finds her highest pay-off. The highest pay-off is in the fourth leaf (i.e., the right leaf of U), hence it takes 4 steps. Next, the player needs to compare the pay-off of the opponent in this leaf with the pay-off of the opponent in the left leaf of T. Since there are two nodes to compare, this will take 2 steps. Because the highest pay-off for the opponent is in the left leaf of node T, the opponent will never let the first player reach her highest pay-off. Therefore, the highest pay-off is not accessible. Finally, the player has to compare her pay-off in the left leaf of T with the pay-off of the left leaf of the first node (i.e., node S). This comparison also requires attending to two nodes and thus takes 2 steps. This left leaf has the highest possible pay-off. In total, the algorithm finds the highest possible solution in 8 steps.

Item 3 Again, the player attends all leaves until she finds her highest pay-off. In this case, the highest pay-off is in the second leaf (the left leaf of node T), and thus it takes 2 steps to find her highest pay-off. Next, the algorithm computes the number of steps needed to find out whether the pay-off is accessible. To that end, the pay-off in both leaves of node U are compared—this also takes 2 steps. Then, the opponent's pay-off in the left leaf of node T is compared to the opponent's pay-off in the right leaf of node U. This comparison also takes 2 steps. A rational opponent would choose the left leaf at node T, because that is the highest possible pay-off for the opponent. Thus, for the player, the highest pay-off is accessible. For this item, the algorithm computes for a total of 6 steps.

Output

The two different proposed methods were used to describe the difficulty of the game items as used in the data obtained by Meijering et al. (2012). The descriptions of the 16 items that were used in the current dataset are shown in Table 1.

Experimental results

Similarly to Szymanik et al (2013), the experimental data of Meijering et al. (2012) was used[2]. To recall, 23 psychology students participated in the experiment. They were asked to play a marble drop game, as depicted in Figure 1, in which they only had to make the first decision (either stop and take the pay-off or continue to the next trapdoor).

Both the lambda-difficulty (abbreviated as Lambda) and the number of steps as calculated by the forward reasoning plus backtracking algorithm (abbreviated as Steps) were calculated for each trial that the participants had received during the experiment.

Next, linear mixed-effects models were used to investigate the predictive power of both the lambda-difficulty and the forward reasoning plus backtracking strategy.

[2] Following Meijering et al. (2012), only reaction times from the second block were analyzed.

Table 1: Number of steps when using forward reasoning plus backtracking (Steps) and the levels of lambda-difficulty (Lambda) for all 16 items of the marble drop game in the analyzed dataset. “Attainable” represents whether the player’s highest possible pay-off 4 is in fact attainable.

Item[3]	Steps	Lambda (Λ_i^1)	Attainable
1	8	3	No
2	8	2	No
3	6	2	Yes
4	8	2	No
5	8	2	No
6	8	3	No
7	5	3	Yes
8	6	2	Yes
9	6	2	Yes
10	8	3	No
11	6	3	Yes
12	5	3	Yes
13	8	3	No
14	6	3	Yes
15	6	2	Yes
16	8	2	No

Mixed-effect models

The data was analyzed with linear mixed-effect models using the LME4 package (Bates, Maechler, Bolker, & Walker, 2013) available in R Project (Team R Core, 2013). To find the best model, we formulated a full model based on theoretical assumptions. We then dredged the model by systematically leaving out different fixed factors and interactions (Bartón, 2013). This dredge process resulted in a subset of all possible models that the full model allowed for. Next, the Akaike information criterion (AIC; Akaike, 1974) was calculated for each model, and the model with the lowest AIC (i.e., the best model) was selected for further analyses. The AIC is suitable for this particular procedure, because it takes into account the trade-off between the complexity of a model and its fit. Thus, we were able to select the best model out of our subset of models.

Null-model In order to get a reference for the calculated AIC for both the forward reasoning plus backtracking model and the lambda-difficulty model, we calculated a null-model in which we only put the random effect of participant. The AIC of this null-model is 1056.

The p-values of the factors in the selected models were calculated by estimating the degrees of freedom (Kuznetsova, Brockhoff, & Christensen, 2013).

Forward reasoning + backtracking First, Steps was entered in the model, because we hypothesized that Steps is predictive of the reaction times. Secondly, Accuracy was entered in the model. Accuracy was 0 or 1, corresponding to an incorrect or correct response, respectively. Furthermore, an interaction effect between Accuracy and Steps was entered. Rationale behind this interaction is that we cannot know what happens when an incorrect response is given, thus one could expect that Steps is not predictive for incorrect responses. To account for speed-up effects due to learning, the sequence in which trials were presented to the participant was coded as the factor Trial. Thus, the fixed factors of the full model were entered as follows: Steps + Accuracy + Steps × Accuracy + Trial. Participant was entered as random factor. Automatic model selection selects the full model as the best model with an AIC of 954. The AIC of this model is lower than the AIC of the null-model thus the full model is a better model.

The fixed factors of the selected model are listed in Table 2. First, a main effect of Accuracy is found. The negative estimate β suggests that individuals are faster at correct trials. Furthermore, the Trial factor reveals the presence of a learning effect. The more trials an individual does, the faster he/she responds. The interaction effect Steps x Accuracy shows that Steps predicts the reaction times of marble drop games that are correctly solved. The lack of a main effect for Steps suggests that for incorrect trials, the forward reasoning plus backtracking does not predict reaction times. This is due to the algorithm’s incapability to predict errors.

Table 2. The factors of the forward reasoning plus backtracking model and the corresponding estimate (β), t-statistics, and p-values.

Factor	β	*t*	*p*
Intercept	9.512	20.67	>0.001
Steps	-0.060	-0.87	0.387
Accuracy	-1.531	-3.27	0.001
Steps×Accuracy	0.209	3.00	0.003
Trial	-0.007	-4.02	>0.001

Lambda-difficulty Lambda, Accuracy, Trial, and the interaction between Lambda and Accuracy were entered in the full model, following the same rationale as in the before-mentioned analyses. Thus, the fixed factors of the model were entered as follows: Lambda + Accuracy + Lambda × Accuracy + Trial. Participant was entered as random factor. Automatic model selection preferred the model with factors Lambda + Trial to the full model. The simpler model has an AIC of 992, which is higher than the AIC of the forward reasoning plus backtracking model, but lower than the AIC of the null-model.

The effects of the fixed factors for the Lambda + Trial model are shown in Table 3. Both the main effects of Trial and Lambda significantly predict the reaction time on a marble drop game. As with the forward reasoning plus backtracking model, Trial can be interpreted as a learning effect. The effect of Lambda is more difficult to explain. If lambda-difficulty positively predicts reaction times (i.e., the more difficult a trial, the slower the participant), one would expect a positive estimate. However, the estimate of lambda

[3] The corresponding trees can be found on the website http://www.ai.rug.nl/SocialCognition/wp-content/uploads/trees.pdf

is negative, meaning that participants are faster in solving trials that are defined as difficult by the lambda-difficulty.

Finally, when we compare the AIC scores of the lambda-difficulty model to the forward reasoning plus backtracking model, the latter has a lower AIC score and thus best explains the data.

Table 3. The factors of the lambda-difficulty model and the corresponding estimate (β), *t*-statistics, and *p*-values.

Factor	β	*t*	*p*
Intercept	9.684	84.38	>0.001
Lambda	-2.263	-7.51	>0.001
Trial	-0.008	-3.89	>0.001

Discussion

Overview

We have investigated two cognitive models of playing a turn-based game called the Marble Drop Game. Our computational-level model is based on the logical description of the game trees in terms of player alternations and the distribution of highest pay-off. Our more specific algorithmic-level model proposes a concrete strategy that can be used by subjects to solve the game trials. The previous experiments (Szymanik et al., 2013) have not been able to distinguish between the two modeling approaches, as both models are consistent with the eye-tracking study of Meijering et al. (2012). In this paper, by generalizing the forward reasoning with backtracking algorithm put forward by Meijering et al. (2012), we have managed to disentangle the predictions of the two models. We have shown that for the experimental items of Meijering et al. (2012) only the forward reasoning plus backtracking model allows to predict subjects' behavior: the number of steps that the algorithm must take for a given marble drop game item predicts the reaction time subjects will need to correctly solve the trial.

Outlook

In the future we plan to continue the reported research in a number of directions. First of all, we would like to better understand why the computational model based on the structural descriptions has failed. Is it because the lambda-hierarchy does not take into account the decision of the other player? And if that is the reason, how could we fix it? Or maybe, the lambda predictions would approximate the cognitive difficulty better for a wider variety of game items? Finally, what is the precise relation between our two models? To answer the last two questions it would be necessary to generalize the forward reasoning plus backtracking algorithm even more, in such a way that it could be applied to any turn-based game.

Acknowledgments

The authors are grateful for the support of Vici Grant NWO-277-80-001 awarded to RV. JS acknowledges NWO Veni Grant 639.021.232.

References

Apperly, I. (2011). *Mindreaders: The Cognitive Basis of "Theory of Mind"*. Hove etc.: Psychology Press.

Aumann, R.J. (1995). Backward induction and common knowledge of rationality. *Games and Economic Behavior*, *8*(1), 6-19.

Bartón, K. (2013). *MuMIn: Multi-model inference.*

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2013). *lme4: Linear mixed-effects models using Eigen and S4*

Ghosh, S., Meijering B., Verbrugge, R. (in press). Strategic reasoning: Building cognitive models from logical formulas. *Journal of Logic, Language and Information*

Isaac, A., Szymanik, J., & Verbrugge, R. (2014). Logic and complexity in cognitive science. In A. Baltag, & S. Smets (Eds.), *Johan van Benthem on Logical and Informational Dynamics* (forthcoming). Trends in Logic book series, Berlin: Springer.

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2013). *lmerTest: Tests for random and fixed effects for linear mixed effect models (lmer objects of lme4 package).*

Marr, D. (1983). *Vision: A Computational Investigation into the Human Representation and Processing Visual Information*. San Francisco, CA: W.H. Freeman.

Meijering, B., Van Rijn, H., Taatgen, N., & Verbrugge, R. (2011). I do know what you think I think: Second-order theory of mind in strategic games is not that difficult. *Proceedings of the 33rd Annual Conference of the Cognitive Sciene Society,* Austin, TX. 2486-2491.

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.

Osborne, M. J., & Rubinstein, A. (1994). *A Course in Game Theory*. Cambridge, MA: MIT Press.

Premack, D., & Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences, 1*(4), 515-526.

Raijmakers, M., Mandell, D., van Es, S., & Counihan, M. (2014). Children's strategy use when playing strategic games. *Synthese, 191*(3), 355-370.

Szymanik, J. (2013). Backward Induction Is PTIME-complete. *Proceedings of the Fourth International Workshop on Logic, Rationality, and Interaction.* D. Grossi, O. Roy, H. Huang (Eds.), Lecture Notes in Computer Science, Vol. 8196, (pp. 352-356). Berlin: Springer.

Szymanik, J., Meijering, B., & Verbrugge, R. (2013). Using intrinsic complexity of turn-taking games to predict participants' reaction times. *Proceedings of the 35th Annual Conference of the Cognitive Science Society,* Austin, TX.

Team R Core. (2013). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.