A computational model of second-order social reasoning

van Maanen, L.; Verbrugge, R.

Publication date
2010

Document Version
Final published version

Published in
Proceedings of the 10th International Conference on Cognitive Modeling

A Computational Model of Second-Order Social Reasoning

Leendert van Maanen (leendert@ai.rug.nl) & Rineke Verbrugge (rineke@ai.rug.nl)
Department of Artificial Intelligence, University of Groningen
P.O. Box 407, 9700 AK Groningen, The Netherlands

Abstract

This paper presents the first computational cognitive model of second-order social reasoning. The model uses a decision tree strategy to reason about the opponent's behavior. We hypothesize that a decision tree strategy requires (1) declarative memory, and (2) working memory. Declarative memory is required to retrieve successive reasoning steps, while working memory is required to temporarily store these reasoning steps while the next step is retrieved from memory. The model fit on data from a social reasoning game supports the validity of the model. This initial result leads to an explicit prediction for an experiment in which the reasoning game is combined with another task that requires the same cognitive resources as hypothesized by the model. This work is a first step towards understanding higher-order social reasoning from a cognitive modeling perspective.

Keywords: reasoning; theory of mind**;** cognitive models; ACT-R

Introduction

What is social reasoning?

The ability to successfully interact with others requires knowledge on how your actions are going be interpreted by others. Additionally, successful interaction requires the ability to reason about the actions that other people might take to respond to, or even to anticipate, your own actions. (Verbrugge, 2009). A term that is often used in connection with this ability is *theory of mind* (Premack & Woodruff, 1978). In this paper we will present a computational cognitive model of second-order theory of mind, calling the process *second-order social reasoning*.

Contrary to the case of first-order mental state attributions such as "she *plans* to move her queen", second-order social reasoning requires the ability to attribute mental states *about* mental states to others, as in "she *believes* that I *intend* to sacrifice my horse" (Perner & Wimmer, 1985). In higher-order social reasoning, this ability is recursively applied for successful behavior. The cognitive model presented in this paper will be the first that explicitly addresses higher-order social reasoning. We will present a theory on how people reason in second-order social reasoning games, as well as explicit predictions on how behavior changes if the task is made more complex.

Second-order social reasoning has often been studied by use of simple strategic games in which success is only warranted if the players successfully anticipate each other's moves. A very simple example of such a game is tic-tac-toe (also known as noughts-and-crosses), in which each player has all information available on the playing board, and players have to take into account what the optimal move is for the opponent (that is, games of perfect information, Osborne & Rubinstein, 1994). A more complex example is Cluedo (Van Ditmarsch, 2002) in which not all information is known to each player, and players also have to reason about what information they will provide to their opponents by making a move, in addition to reasoning about optimal moves, for example, "I don't *want* Alice to *know* that I *know* that she has the ace of hearts". In this paper, we will focus on a simpler game called Marble Drop in which all information about the current game state is known. Marble Drop is equivalent to the well-known centipede game (Rosenthal, 1981) and will be discussed in detail in later sections.

What are important questions in social reasoning?

Two issues stand out in studying social reasoning. The first relates to human performance on games such as Marble Drop. Up to this point we have described behavior as "optimal" or "rational", but it turns out that humans perform significantly suboptimally on these games as the complexity increases (Flobbe, Verbrugge, Hendriks, & Krämer, 2008; Hedden & Zhang, 2002). Flobbe et al. for example found that participants in a centipede game only correctly perform 75.5% of second-order games, whereas they are near-perfect on the first-order games (97%).

The second issue relates to the role of memory in reasoning tasks. Taking the perspective of others about your own mental states and then incorporating that knowledge in your own reasoning must require some form of working memory. In this paper, we will present the first computational model that explicitly addresses both issues.

After a brief overview of other models of social reasoning, we will introduce our model. Then we will present the model fit on relevant data and we will discuss how this model can contribute new insights in the understanding of social reasoning.

Formal models of social cognition

Social reasoning has been formally studied from a number of perspectives. These perspectives differ in the amount of cognitive validity that is considered. One perspective is to study social cognition as an interactive game (Camerer, 2003). This game-theoretic perspective assumes that people are rational agents, optimizing their gain by applying strategic reasoning. However, many experiments have shown that people are not completely rational in this sense. For example, McKelvey and Palfrey (1992) have shown that in a traditional centipede game participants do not behave rationally. In this version of the game, the payoffs are distributed in such a way that the optimal strategy is to always end the game at the first move (i.e., Nash equilibrium, Nash, 1951). However, in McKelvey and Palfrey's experiment participants continued the game for some rounds before ending it. One interpretation of this result is that the game-theoretic perspective fails to take into account the reasoning abilities of participants. That is, due to cognitive constraints such as working memory capacity, participants may be unable to perform optimal strategic reasoning, even if in principle they are willing to do so.

A different perspective, that focuses on cognitive validity in developing formal models, is that of a cognitive architecture (Anderson, 2007; Newell, 1990). Cognitive models developed within this framework aim to explain certain aspects of cognition by assuming only general cognitive principles. However, the current cognitive models that describe social interactions do not take second-order reasoning into account. For example, cognitive models of simple games exist in which it is important to know the opponent's behavior (e.g., Lebiere & West, 1999; West, Lebiere, & Bothell, 2006). These cognitive models demonstrate that declarative memory is important in playing strategically. In the current work however, we are less interested in how people adapt their strategy to an opposing strategy, but rather we are studying the cognitive limitations of explicit second-order reasoning. Related to this, Hendriks and colleagues (e.g., Hendriks, Van Rijn, & Valkenier, 2007; Van Rij, Van Rijn, & Hendriks, in press) have studied the development of first-order theory-of-mind in language using computational cognitive modeling.

An ACT-R model of social reasoning

To provide a full model of second-order social reasoning, we implemented our model in the cognitive architecture ACT-R (Anderson, 2007). ACT-R aspires to explain all of cognition using one theoretical framework. To achieve this, the heart of ACT-R consists of a procedural memory system, which contains condition-action pairs known as production rules. Besides the procedural module, ACT-R has designated modules for specific types of information. For example, the visual module processes visual information, whereas the declarative memory module processes declarative or factual information. Each module has a buffer that may contain one unit of information (a chunk). If the current contents of all buffers in the system matches the conditions of a particular production rule, that rule fires and its actions are executed. Each action may refer to an operation in one of the modules.

This general layout of the cognitive system enables the development of models in which different kinds of information can be processed at the same time, while each module can only process one unit of information at a time. Based on this feature, ACT-R predicts specific interference effects if different aspects of a task require the same cognitive resource at the same time (e.g., Borst, Taatgen, & Van Rijn, 2010; Van Maanen & Van Rijn, 2010; Van Maanen, Van Rijn, & Borst, 2009). In the discussion section of the current paper we will use this feature of the architecture to make explicit predictions for a particular social reasoning task.

Two modules of ACT-R deserve extra attention in the light of our model of second-order social reasoning: the declarative memory module and the problem state module. The declarative memory module retrieves information from long-term memory, called chunks. Each chunk in memory is represented by an activation value that represents the likelihood that that item can be retrieved. If the activation value drops below a certain minimal value (the retrieval threshold), the related information is no longer accessible. In that case, the system will report a retrieval failure after a constant time factor. If the activation value is above the retrieval threshold, the information is accessible. However, the time needed to retrieve it from memory depends on how active the item actually is. The more active, the faster the retrieval will be. Connected to the declarative memory module is a retrieval buffer, which may contain one (retrieved) item at a time. If another item is retrieved, it is stored in the retrieval buffer, with the previous item being pushed back to long-term memory.

The problem state module (sometimes referred to as the imaginal module) contains a buffer in which information can be temporarily stored. Typically, this information contains a subsolution to the problem at hand. In the case of a social reasoning task, this may be the outcome of a reasoning step that will be relevant in subsequent reasoning. Storing information in the problem state buffer is associated with a time cost (typically 200ms). The model that we present in this paper relies on the combination of the declarative module and the problem state buffer. That is, the model retrieves relevant information from memory and moves that information to the problem state buffer if new information is retrieved from memory that needs to be stored in the retrieval buffer.

Marble Drop game

To study the reasoning processes that are involved in social reasoning, we developed a cognitive model of a reasoning game in which in order to play optimally the players have to anticipate each other's moves. The particular game that was analyzed and modeled is a variant of the centipede game called Marble Drop (Meijering, Van Maanen, Van Rijn, & Verbrugge, 2010).

Marble Drop is a marble run game containing trapdoors (Figure 1). Players take turns in deciding whether to open one trapdoor or the other. In each turn, opening one trapdoor leads to the end of the game, whereas opening the other trapdoor means that the game continues to the next bin on the right and the opponent may choose which trapdoor to open. If a player decides to end the game, both players receive the credits that are associated with that stage of the game. If a player decides to continue the game, the players traverse to a new stage with which new credits are associated. Because all credits are known in advance, both players can reason about their opponent's possible moves further on in the game. The players can do this by applying

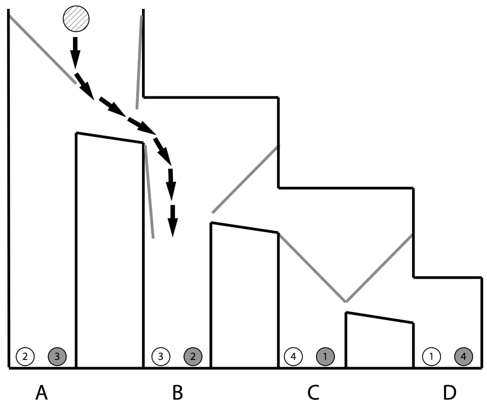

Figure 1. The interface of a second-order Marble Drop game. Color shades of the marbles in the experiment are represented by numbers.

backward induction (Van der Hoek & Verbrugge, 2002; Verbrugge & Mol, 2008). For example, a player can reason that his opponent wants the highest payoff in bins C and D. As a result the player knows the maximal payoff that he can get from bins C and D, and can then compare that information to his own payoff in bin B. If it is possible in a particular game for a player to behave optimally by directly predicting its opponent's actions, we refer to this game as being *first-order*. In a *second-order* game it is necessary to predict the opponent's predictions of ones own actions in order to behave optimally. In principle, Marble Drop games could be developed for third-order or even higher-order games.

The Model

The model follows a backward induction strategy to predict the opponent's moves further on in the game. Hedden and Zhang (2002) provide a decision tree analysis of this process for their matrix version of the game.[1] The model has knowledge on how to solve Marble Drop games for all possible distributions of payoffs over the bins of the marble run game. That is, the model stores chunks containing information on which payoffs to compare at each step. In addition, chunks representing the magnitudes of the payoff shades are stored in declarative memory, as well as chunks representing the location of the payoffs on the screen. Finally, chunks representing ordinal information are stored in declarative memory. This means that the model contains knowledge on the relative magnitudes of each combination of payoff values.

A model run starts with the initial comparison of two payoff values (Figure 2). For second-order games, that initial comparison is always a comparison between the player's own payoffs in Bins C and D. First, it retrieves from declarative memory where the first payoff is located on the screen (Bin D in Figure 1). If it retrieves that knowledge, the model attends Bin D and tries to retrieve the magnitude of the observed payoff. At the same time, the model stores the current comparison in the problem state buffer, to free the retrieval buffer for the upcoming payoff information.

Because in the experiment the payoffs are represented by shaded marbles, the model has to retrieve the value corresponding to the observed shade. Next, the model retrieves the location information for the other payoff value that is part of the current comparison. Again to free the retrieval buffer, the payoff value of the first payoff is stored in the problem state buffer. The payoff is attended and the corresponding value is retrieved from memory. Finally, the two values are compared by trying to retrieve a chunk with ordinal information from memory. Based on the outcome of this retrieval the model now retrieves a new payoff comparison. For example (Figure 1), if the value in bin D was smaller than the value in bin B, the model attends the payoff in bin B, and compares that with the payoff in bin A. If the value in bin D was larger than the value in bin B, then the model attends the opponent's payoff in bin D, and compares that with the opponent's payoff in bin C. The model continues to compare payoffs following the decision tree (Hedden & Zhang, 2002) until it reaches the bottom of the tree. There, it decides its action based on the final comparison.

Model fit The model was tested against data from a Marble Drop task (Meijering et al., 2010). In the experiment the participants were asked to solve zero-order, first-order, and second-order Marble Drop problems. In all these conditions, participants were instructed to indicate the optimal *first*

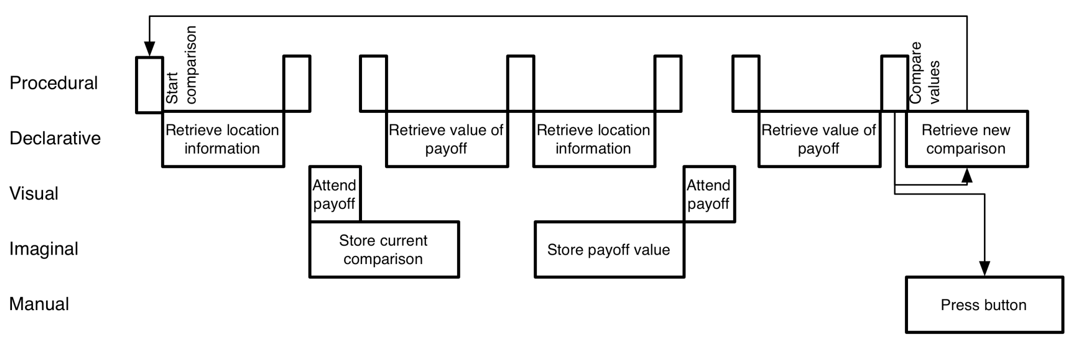

Figure 2. Flow chart of the model activity in ACT-R modules. The width of each box denotes the duration of each stage. Arrows indicate possible next actions.

[1] an analysis that shows the logical equivalence of these games can be found at http://www.ai.rug.nl/~leendert/Equivalence.pdf

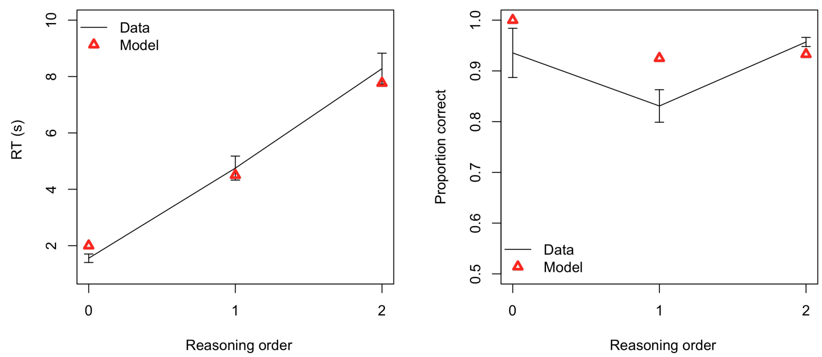

Figure 3. Model fit to data from Meijering et al. (2010). Left: Response time, Right: Accuracy.

move as quickly as possible. That is, even in second-order games participants had to make only one choice. However, because the opponent always played rationally (and the participants were informed of this), there was always only one optimal choice.

Figure 3 presents the model fit on both response times and accuracy of the first moves. The fit on the response times is very good ($R^2 = 1.0$; RMSE = 0.42 s). The fit on the accuracy data is slightly less (RMSE = 0.067, $R^2 = 0.2$), but this may be attributed to lack of data, making the estimated means less reliable.[2]

As the order of the Marble Drop reasoning problems increases, the model requires more time to respond. This is because more comparisons have to be made, and therefore more information has to be retrieved from declarative memory and stored in the problem state buffer. These steps take time, increasing the response time for higher-order reasoning problems. Because of the similar behavioral patterns between model and data, this study supports the view that participants in this task follow the same reasoning steps as the model does. That is, participants in a social reasoning game follow a decision tree to make the correct decision.

Discussion & Predictions

First model of second-order social reasoning

The ACT-R model of second-order social reasoning described in this paper is the first cognitive model to account for second-order social reasoning. Other cognitive models in the field of social reasoning have either not explicitly addressed orders of reasoning (e.g., Lebiere & West, 1999; West et al., 2006), or have focused on first-order reasoning only (e.g., Hendriks et al., 2007; Van Rij et al., in press).

Because the model is based on Hedden and Zhang's (2002) decision tree analysis of behavior in 2x2 matrix games, the model provides support for their theory of second-order social reasoning. The model can be considered as a cognitively plausible implementation of that analysis.

Model predictions

Our model can be used to provide explicit predictions regarding the use of memory in second-order social cognition (Verbrugge, 2009). In particular, the model relies on various declarative memory retrieval steps, in combination with storage of information in a problem state buffer. An explicit prediction would be that second-order theory of mind reasoning would be affected by performing another task at the same time that would require the same resources (Borst et al., 2010). To our knowledge, such an experiment has not been done yet. Therefore, in the remainder of this paper we would like to propose such an experiment, combined with explicit, quantitative predictions provided by the model. By providing the predictions of our model before actually doing the experiment, we counter the criticism that insufficiently constrained cognitive models can be made to fit any dataset (Roberts & Pashler, 2000).

A task that would require the same resources as hypothesized for social reasoning is a tone counting task. Participants are presented with tones of two different pitches and are requested to count the number of tones for each pitch. This task would tap into the same cognitive resources as hypothesized for the Marble Drop reasoning task, as maintaining two counters at the same time can be considered a heavy working memory load. A control condition in this task would be one in which participants would not need to maintain a counter, but rather just say "high" or "low" every time they heard a tone of a particular (higher or lower) pitch. Because the control task does not require maintaining a counter (a problem state), concurrent execution of this task and the social reasoning task does not pose a conflict, and the different stages of the tasks could be interleaved without much loss of time (Anderson, Taatgen, & Byrne, 2005).

A dual-task model of social reasoning A simple model of this task would involve maintaining the current counter in a problem state buffer. In addition, the model would – upon hearing a tone – check whether the pitch of the tone is the same as the pitch of the previous tone. Specifically, the model compares the pitch of the tone with the pitch

[2] As the data presented here are actually the practice block of the experiment performed by Meijering et al. (2010), the number of observations per participant was 4 for zero-order games, and 8 for first and second-order games.

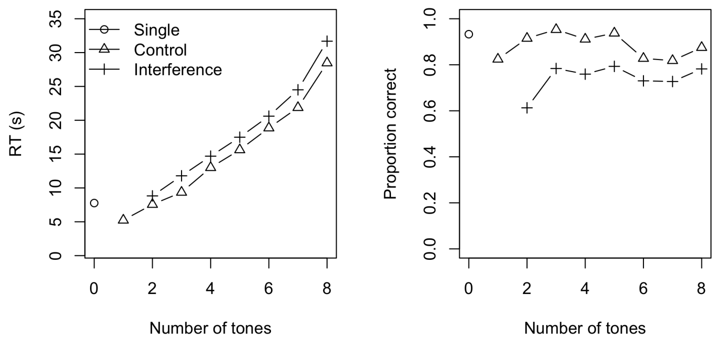

Figure 4. Model predictions for the dual-task social reasoning task. Left: Response time, Right: Accuracy.

associated with the counter in the problem state buffer. If this is the case, the model then retrieves the subsequent number of the stored counter from memory. If this is not the case, the model retrieves the other counter from memory, and based on that retrieves the subsequent number.

Such a model would require both the problem state buffer and the retrieval buffer, resulting in interference with performance on the Marble Drop game. For the control task, both the retrieval and the problem state resources are not required. The model of the control task consists of a simple stimulus response mechanism: When a tone of a particular pitch is heard, the model responds with a vocal response (either "high" or "low").

We adapted our model to also perform the tone counting task. The model was extended with a control mechanism that maintained which task was currently given preference (Salvucci & Taatgen, 2008). The model performs the Marble Drop task until a tone is presented. At that point a switch is made to the counting task. If necessary, the model tries to retrieve the current count and restore the problem state of the counting task. Then, it retrieves the subsequent number from declarative memory followed by a vocal response saying the number. After that, the model tries to restore the problem state of the Marble Drop task by retrieving a comparison from memory.

Model predictions We ran the second-order reasoning model in three conditions for a sufficient number of trials to obtain a stable estimate of the predicted response. In the first condition (Single) the model only performed the Marble Drop task. In the Control condition, the model performed the Marble Drop task in combination with the simple response task. The tones were presented with stimulus onset asynchronies (SOAs) of 2s, 5s, 8s, 11s, 14s, 17s, 20s, and 23s. Only those tones were presented that preceded the model response on the reasoning task. In the Interference condition, the model performed the Marble Drop task in combination with the tone counting task. The tones were presented similarly as in the Control condition.

Figure 4 presents the predicted reaction time and accuracy of the dual-task model as a function of the number of tones presented. The left-most data point in each graph (where the number of tones is zero) represents the behavior of the model under single-task conditions. This is the same as the model fit presented in Figure 3. For the Control condition the model predicts an increase in the response time, and no change in accuracy. This is because the single response task used as secondary task in the Control condition does not share any resources with the Marble Drop task. Thus, responding to the tones only adds time to the Marble Drop response, but does not change the difficulty of the task. In contrast, the tone counting task that the model performs in the Interference condition adds considerable time to the response. In addition, the accuracy of the model decreases as well. Moreover, the mean response time in the Interference condition increases dramatically to 27s (Figure 5), whereas the mean response time in the control condition is 8.3s, which is only slightly above the mean response time of the single response task (7.7s). Our interpretation of these results is that the tone counting task and the Marble Drop task share a cognitive resource. In particular, both tasks require a problem state buffer for maintaining intermediate results. Swapping these problem states takes extra time and is prone to errors, explaining the increased reaction times and the decreased accuracy.

Conclusion

This paper presents the first computational cognitive model of second-order social reasoning. The model uses a decision tree strategy to reason about the opponent's behavior in a social reasoning game. We hypothesize that a decision tree strategy requires (1) declarative memory, and (2) working memory. Declarative memory is required to retrieve successive reasoning steps, while working memory is required to temporarily store these reasoning steps while the

Figure 5. Predicted mean response time for the dual-task model. Left: Response time, Right: Accuracy.

next step is retrieved from memory. We implemented working memory as a problem state buffer using the ACT-R cognitive architecture (Borst et al., 2010). The model fit on data from a social reasoning game called Marble Drop (Meijering et al., 2010) supports the validity of the model. This initial result leads to an explicit prediction for an experiment in which the reasoning game is combined with another task that requires the same cognitive resources as hypothesized by the model. In particular, if the other task also requires the problem state resource, the interference of that task is substantial. On the other hand, a secondary task that is equivalent but does not require the problem state resource exhibits minimal interference. This work is a first step towards understanding higher-order social reasoning from a cognitive modeling perspective.

Acknowledgements

This research was supported by NWO Vici grant NWO-277-80-001 awarded to Rineke Verbrugge.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford UP.

Anderson, J.R., Taatgen, N.A. & Byrne, M.D. (2005). Learning to Achieve Perfect Time Sharing: Architectural Implications of Hazeltine, Teague, & Ivry (2002). *Journal of Experimental Psychology: Human Perception and Performance, 31*(4), 749-761.

Borst, J. P., Taatgen, N. A., & Van Rijn, H. (2010). The problem state: A cognitive bottleneck in multitasking. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 36*(2), 363-382.

Camerer, C. F. (2003). *Behavioral game theory: Experiments in strategic interaction*. Princeton: Princeton UP.

Flobbe, L., Verbrugge, R., Hendriks, P., & Krämer, I. (2008). Children's application of theory of mind in reasoning and language. *Journal of Logic, Language and Information, 17*(4), 417-442.

Hedden, T., & Zhang, J. (2002). What do you think I think you think?: Strategic reasoning in matrix games. *Cognition, 85*(1), 1-36.

Hendriks, P., Van Rijn, H., & Valkenier, B. (2007). Learning to reason about speaker's alternatives in sentence comprehension: A computational account. *Lingua, 117*(11), 1879-1896.

Lebiere, C., & West, R. L. (1999). A dynamic ACT-R model of simple games, *Proceedings of the Twenty-First Annual Conference of the Cognitive Science Society* (pp. 296-301): Erlbaum.

McKelvey, R. D., & Palfrey, T. R. (1992). An experimental study of the centipede game. *Econometrica, 60*(4), 803-836.

Meijering, B., Van Maanen, L., Van Rijn, H., & Verbrugge, R. (2010). The facilitative effect of context on second-order social reasoning. In R. Catrambone & S. Ohlsson (Eds.), *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Nash, J. (1951). Non-cooperative games. *The Annals of Mathematics, 54*(2), 286-295.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard UP.

Osborne, M., & Rubinstein, A. (1994). *A course in game theory*. Cambridge, MA: MIT Press.

Perner, J. & Wimmer, H. (1985). "John thinks that Mary thinks that...": Attribution of second-order beliefs by 5- to 10-year old children. *Journal of Experimental Child Psychology, 5,* 125-137.

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

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review, 107*(2), 358-367.

Rosenthal, R. (1981). Games of perfect information, predatory pricing, and the chain store. *Journal of Economic Theory, 25*, 92-100.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review, 115*(1), 101-130.

Van der Hoek, W., & Verbrugge, R. (2002). Epistemic logic: A survey. In L. A. Petrosjan & V. V. Mazalov (Eds.), *Game theory and applications* (Vol. 8, pp. 53-94). New York: Nova Science.

Van Ditmarsch, H. P. (2002). The description of game actions in cluedo. In L. A. Petrosian & V. V. Mazalov (Eds.), *Game theory and applications* (Vol. 8, pp. 1-28). Hauppage, NY: Nova Science Publishers.

Van Maanen, L., & Van Rijn, H. (2010). The locus of the Gratton effect in picture-word interference. *Topics in Cognitive Science, 2*(1), 168-180.

Van Maanen, L., Van Rijn, H., & Borst, J. P. (2009). Stroop and picture-word interference are two sides of the same coin. *Psychonomic Bulletin & Review, 16*(6), 987-999.

Van Rij, J., Van Rijn, H., & Hendriks, P. (in press). Cognitive architectures and language acquisition: A case study in pronoun comprehension. *Journal of Child Language*.

Verbrugge, R. (2009). Logic and social cognition: The facts matter, and so do computational models. *Journal of Philosophical Logic, 38*(6), 649-680.

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

West, R. L., Lebiere, C., & Bothell, D. (2006). Cognitive architectures, game playing, and human evolution. In R. Sun (Ed.), *Cognition and multi-agent interaction: From cognitive modeling to social simulation* (pp. 103-123). New York, NY: Cambridge UP.

Wickelgren, W. A. (1977). Speed-accuracy tradeoff and information-processing dynamics. *Acta Psychologica, 41*(1), 67-85.