On combining cognitive and formal modeling: a case study involving strategic reasoning

Sujata Ghosh and Ben Meijering

Department of Artificial Intelligence
University of Groningen.
sujata,b.meijering@ai.rug.nl

Abstract. This paper builds up a bridge between the formal and cognitive modeling of human reasoning aspects. To this end, we focus on empirical studies on playing a certain game, namely marble drop, that involves reasoning about other minds, and build up a formal system that can model the different strategic reasoning methods employed by the participants in the empirical study. Finally, we show how the syntactic framework of the formal system can aid in building up a cognitive model of the participants of the marble drop game.

1 Introduction

In recent years, a lot of questions have been raised regarding the idealization that a formal model undergoes while representing social reasoning methods (e.g. see [3]). Do these formal methods represent human reasoning satisfactorily or should we concentrate more on the empirical studies and models based on those empirical data? Without going into this debate here, we combine empirical studies, formal modeling and cognitive modeling to study human strategic reasoning. Our proposal is the following: rather than thinking about them as separate ways of modeling, we can consider them to be complementary and investigate how they can aid each other to bring about a more meaningful model of the real-life scenarios.

Experimentation ⟶ *Formal modeling* ⟶ *Cognitive modeling*

In [5], a formal framework has been introduced to model human strategic reasoning as exemplified by certain psychological experiments focusing on a dynamic game scenario, namely the Marble Drop game [10]. In continuation of the work done in [5], this paper builds on the formal framework to give a more realistic reasoning model of the participants. Moreover, we propose to use a cognitive model of these participants based on the formal framework.

For the experimental work, the advantage of using dynamic games to study higher-order social reasoning is that they allow for repeated presentation, which yields more observations than is typical in other paradigms such as, for example, false-belief story and/or picture tasks. More observations yield more reliable outcome measures such as accuracy of decisions and decision (or reaction) times (RTs). Examples of dynamic games used in empirical studies are the Centipede game [9], the matrix game [6], the road game [4], and Marble Drop [10, 11]. These examples are all game-theoretically equivalent because they share the same extensive form, namely that of the original Centipede game [17].

Previous empirical studies have shown higher-order social reasoning to be far from optimal, and have argued that higher-order social reasoning is complicated and cognitively demanding (e.g., [19]). However, Meijering et al. [10, 11] demonstrated that performance improved to near ceiling if participants (1) were assigned to stepwise instruction and training, (2) were asked to predict the other player's move, and (3) were presented with concrete and realistic games.

Based on empirical findings that show that the participants do not always follow the backward induction method [13], in this paper a formal framework is presented to model forward, backward as well as combined reasoning attempts of the participants. As discussed in [15], in backward induction reasoning, a player, at every stage of the game, only reasons about the opponents future behavior and beliefs. On the other hand, in forward induction reasoning, a player, at every stage, only considers the past choices of the opponents. Based on the formal framework, a cognitive model is proposed as a better alternative to the model proposed in [8]. The new model can represent these different reasoning methods in the Marble Drop game.

Before proceeding into the main sections of this paper, we should mention here that this paper should be considered as a preliminary report of a cognitive model of strategic reasoning that is being constructed with the aid of a formal framework. We still need to do the important tasks of predicting and testing the strategies that have come to our notice based on the empirical findings in [10, 11] and the eye-tracking study reported in [12]. The formal framework is introduced to capture the findings of the eye-tracking experiment, so that it can provide an easy, mechanical representation of the eye-tracking analyses to be used in the construction of the cognitive computational model.

2 Empirical work

We provide here a short discussion of the experimental studies on which this work is based. The first part gives a description of the Marble Drop game and the second part provides an analysis of the eye-tracking experiment.

2.1 Marble drop game

Figure 1 depicts examples of a zeroth-, first-, and second-order Marble Drop game. A white marble is about to drop, and its path can be manipulated by removing trapdoors (i.e., the diagonal lines). In this example, the participant controls the blue trapdoors and the computer controls the orange ones. Each bin contains a pair of payoffs. The participant's payoffs are the blue marbles

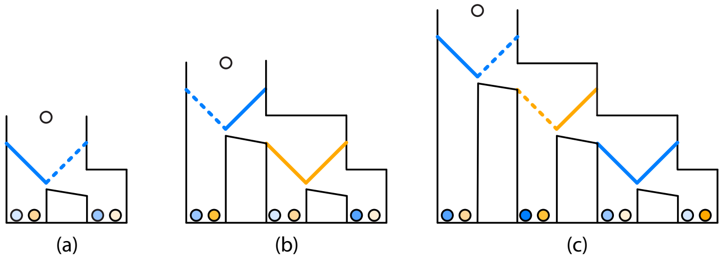

Fig. 1. Examples of a zeroth, first-, and second-order Marble Drop game. The blue marbles are the participant's payoffs and the orange marbles are the computer's payoffs. The marbles can be ranked from light to dark, light less preferred than dark. For each player, the goal is that the white marble drops into the bin with the darkest possible marble of their color. The participant controls the blue trapdoors (i.e., blue diagonal lines) and the computer the orange ones. The dashed lines represent the trapdoors that both players should remove to attain the darkest possible marble of their color.

and the computer's payoffs are the orange marbles. The marbles can be ranked from light to dark, light marbles being less preferred than dark. For each player, the goal is that the white marble ends up in the bin that contains the darkest possible color-graded marble of their color.

For example, at the start of the game in Figure 1c, Player I has to decide whether to remove the left trapdoor (end) or to remove the right trapdoor (continue). Player I's marble in bin 2 is darker than in bin 1, but what will Player II decide if Player I continues? Player II may want to continue the game to the last bin, as Player II's marble in bin 4 is darker than in bin 2, but what will Player I decide at the last set of trapdoors? Player I would stop the game in bin 3, as Player I's marble in bin 3 is darker than in bin 4. Thus, Player II should stop the game in bin 2, as Player II's marble in bin 2 is darker than in bin 3. Consequently, Player I should decide to continue the game from bin 1 to bin 2.

Marble Drop games provide visual cues as to which payoff belongs to whom, who decides where, what consequences decisions have, and how a game concludes. In matrix games [6], participants had to reconstruct this from memory. Meijering et al. [10] hypothesized that the supporting structure of the representation of Marble Drop would facilitate higher-order social reasoning, and, in fact, participants assigned to Marble Drop games performed better than participants assigned to matrix games [11].

2.2 Eye-tracking study

Behavioral measures such as responses and reaction times shed some light on higher-order social reasoning. However, they show the end result of higher-order social reasoning, not the online process. The online process (i.e., the strategies that participants use) may prove valuable in the study of higher-order social reasoning, because strategies determine to a great extent what cognitive resources are employed. For example, an algorithmic strategy such as backward induction puts a lesser strain on working memory than a strategy that explicitly models mental states. Johnson, Camerer, Sen, and Rymon [7] used a novel approach to measure online higher-order social reasoning. In their sequential bargaining games, information displayed on a computer screen was masked with boxes and participants could uncover parts of that information by clicking on the boxes with the mouse. This approach allowed Johnson et al. to investigate the sequence in which participants uncovered information during reasoning.

A concern with this approach is that participants may have felt disinclined to click on the information repeatedly and would rather adopt an artificial strategy that involves fewer mouse clicks but puts a higher strain on working memory. To avoid that, Meijering, Van Rijn, Taatgen, and Verbrugge [12] conducted a study in which they used eye-tracking, which is not as obtrusive as Johnson et al.'s method of masking information. Participants' eye movements were recorded while they were playing Marble Drop games. The eye movement data yielded insight into the comparisons that participants made and the sequence of those comparisons during each game.

The proportions of fixations at bins 1 to 4 are depicted in Figure 2. The proportions were averaged over games, and plotted against position in the total fixation sequence. In other words, Figure 2 shows the general increase or decrease of fixations at a particular bin over time spent in a game. The results showed that participants did not seem to use backward induction, at least, initially. Figure 2(a) shows that, on the first position, the proportion of fixations at bins 1 and 2 was higher than at bins 3 and 4. In contrast, backward induction would yield a higher proportion of first fixations at bins 3 and 4, as backward reasoning starts with a comparison of the payoffs in bins 3 and 4. However, as of position 4 in Figure 2(a), the fixation patterns seem to correspond with backward induction: the proportion of fixations at bins 3 and 4 was higher than the proportion of fixations at bins 1 and 2, and the way that the proportions change over time (i.e., they decrease for bins 3 and 4, and increase for bins 1 and 2) correspond with eye movements that go from right to left.

The patterns are less obvious in Figure 2(b), because the figure shows fixations that were averaged over another set of games. Where Figure 2(a) (bottom panel) depicts mean proportions for games in which a rational participant should end the game because the computer would continue, Figure 2(b) (bottom panel) depicts mean proportions for games in which a rational participant should continue because the computer would continue. Differential fixation sequences imply that participants did not use pure backward induction, because backward induction works independently of the payoff values.

Instead of backward induction, participants may have applied forward reasoning, or a mix of backward and forward reasoning. Figure 2(a) hints at the latter possibility as participants fixated from left to right during the first four fixations, and from right to left during later fixations. To test what strategies participants may have used, we construct cognitive computational models (cf. Section 4) that implement various strategies, and use these models to predict eye movements that we can test against the observed eye movements. To aid in

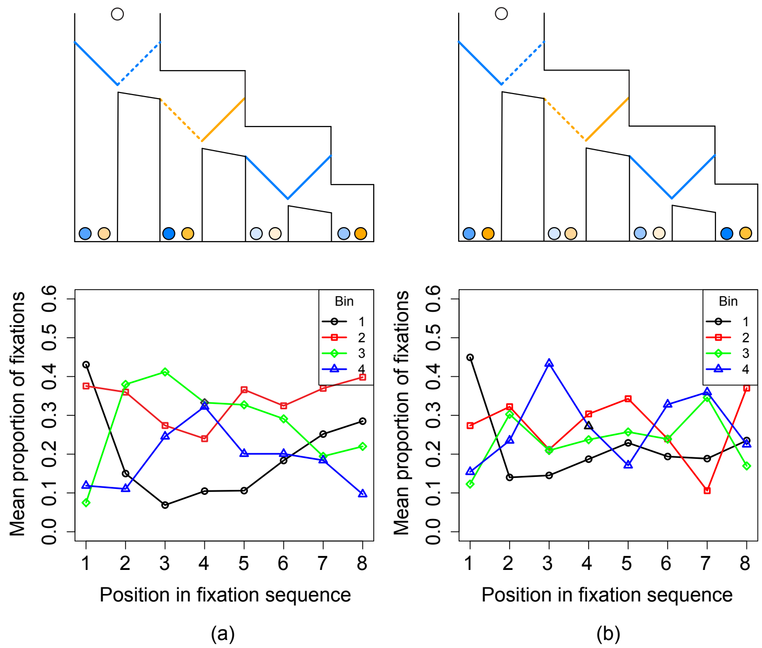

Fig. 2. The bottom panel depicts mean proportions of fixations at bins 1, 2, 3, and 4, calculated separately for position in the total fixation sequence. In (a), Player I should end the game in bin 1, because given the chance, Player II would continue, and the game would end with a lesser payoff for Player I . In (b), Player I should continue the game, because given the chance, Player II would continue, and the game would end with a better payoff for Player I . The games in the top panel are examples of the former and latter type of games. We did not depict standard errors, because we fitted (non-)linear models instead of traditional ANOVAs, which typically include contrasts between (successive) positions of fixations.

the construction we build up a formal framework (cf. Section 3) and show how the formal and cognitive modeling can interplay to provide a better model for strategic reasoning (cf. Section 4.2). As mentioned in the introduction, we are presently at the phase of building up this cognitive model and predicting and testing strategies are our next steps.

3 A formal framework

In this section, we present a formal system to represent the different ways of strategic reasoning that the participants of the Marble Drop game (cf. Section 2.1) undertake, suggested by the eye-tracking study described in Section 2.2. We extend the system developed in [5] by adding special propositional variables representing players' payoffs and comparison of such payoffs, inspired by [2].

3.1 Strategy specifications

Following the lines of work in [16, 14], a syntax for specifying partial strategies and their compositions in a structural manner involving simultaneous recursion has been proposed in [5]. The main case specifies, for a player, what conditions she tests for before making a move. The pre-condition for the move depends on observables that hold at the current game position as well as some simple finite past-time conditions and some finite look-ahead that each player can perform in terms of the structure of the game tree. Both the past-time and future conditions may involve some strategies that were or could be enforced by the players. These pre-conditions are given by the following syntax.

Below, for any countable set X, let $BPF(X)$ (the boolean, past and future combinations of the members of X) be sets of formulas given by the following syntax:

$$BPF(X) := x \in X \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \langle a^+ \rangle \psi \mid \langle a^- \rangle \psi.$$

where $a \in \Sigma$, a finite set of actions.

Formulas in $BPF(X)$ can be read as usual in a dynamic logic framework and are interpreted at game positions. The formula $\langle a^+ \rangle \psi$ (respectively, $\langle a^- \rangle \psi$) talks about one step in the future (respectively, past). It asserts the existence of an a edge after (respectively, before) which ψ holds. Note that future (past) time assertions up to any bounded depth can be coded by iteration of the corresponding constructs. The "time free" fragment of $BPF(X)$ is formed by the boolean formulas over X. We denote this fragment by $Bool(X)$.

Syntax Let $P^i = \{p_0^i, p_1^i, \ldots\}$ be a countable set of observables for $i \in N$ and $P = \bigcup_{i \in N} P^i$. To this set of observables we add two new kinds of propositional variables $(u_i = q_i)$ to denote 'player i's utility (or payoff) is q_i' and $(r \leqslant q)$ to denote that 'the rational number r is less than or equal to the rational number q'. The syntax of strategy specifications is given by:

$$Strat^i(P^i) := [\psi \mapsto a]^i \mid \eta_1 + \eta_2 \mid \eta_1 \cdot \eta_2,$$

where $\psi \in BPF(P^i)$. For a detailed explanation see [5]. The basic idea is to use the above constructs to specify properties of strategies as well as to combine them to describe a play of the game. For instance the interpretation of a player i's specification $[p \mapsto a]^i$ where $p \in P^i$, is to choose move "a" at every game position belonging to player i where p holds. At positions where p does not hold, the strategy is allowed to choose any enabled move. The strategy specification $\eta_1 + \eta_2$ says that the strategy of player i conforms to the specification η_1 or η_2. The construct $\eta_1 \cdot \eta_2$ says that the strategy conforms to specifications η_1 and η_2.

Let $\Sigma = \{a_1, \ldots, a_m\}$, we also make use of the following abbreviation.

- $null^i = [\top \mapsto a_1] + \cdots + [\top \mapsto a_m]$.

It will be clear from the semantics (which is defined shortly) that any strategy of player i conforms to $null^i$, or in other words this is an empty specification. The empty specification is particularly useful for assertions of the form "there exists a strategy" where the property of the strategy is not of any relevance.

Semantics We consider perfect information games as models. Let $M = (T, V)$ with $T = (S, \Rightarrow, s_0, \widehat{\lambda}, \mathcal{U})$, where $(S, \Rightarrow, s_0, \widehat{\lambda})$ is an extensive form game tree, $\mathcal{U} : frontier(T) \times N \to \mathbb{Q}$ is a utility function. Here, $frontier(T)$ denotes the leaf nodes of the tree T. Finally, $V : S \to 2^P$ is a valuation function. The truth of a formula $\psi \in BPF(P)$ at the state s, denoted $M, s \models \psi$, is defined as follows:

- $M, s \models p$ iff $p \in V(s)$.
- $M, s \models \neg\psi$ iff $M, s \not\models \psi$.
- $M, s \models \psi_1 \vee \psi_2$ iff $M, s \models \psi_1$ or $M, s \models \psi_2$.
- $M, s \models \langle a^+ \rangle \psi$ iff there exists an s' such that $s \overset{a}{\Rightarrow} s'$ and $M, s' \models \psi$.
- $M, s \models \langle a^- \rangle \psi$ iff there exists an s' such that $s' \overset{a}{\Rightarrow} s$ and $M, s' \models \psi$.

The truth definition for the new propositions are as follows:

- $M, s \models (u_i = q_i)$ iff $\mathcal{U}(s, i) = q_i$.
- $M, s \models (r \leqslant q)$ iff $r \leqslant q$, where r, q are rational numbers.

Strategy specifications are interpreted on strategy trees of T. We also assume the presence of two special propositions $\mathbf{turn}_1$ and $\mathbf{turn}_2$ that specify which player's turn it is to move, i.e. the valuation function satisfies the property

- for all $i \in N$, $\mathbf{turn}_i \in V(s)$ iff $\widehat{\lambda}(s) = i$.

One more special proposition **root** is assumed to indicate the root of the game tree, that is the starting node of the game. The valuation function satisfies the property

- $\mathbf{root} \in V(s)$ iff $s = s_0$.

A partial strategy σ, say of player i, can be viewed as a set of total strategies of the player [14] and each such strategy is a subtree of T.

The semantics of the strategy specifications are given as follows. Given the game $T = (S, \Rightarrow, s_0, \widehat{\lambda}, \mathcal{U})$ and a partial strategy specification $\eta \in Strat^i(P^i)$, we define a semantic function $[\![\cdot]\!]_T : Strat^i(P^i) \to 2^{\Omega^i(T)}$, where each partial strategy specification is associated with a set of total strategy trees.

For any $\eta \in Strat^i(P^i)$, the semantic function $[\![\eta]\!]_T$ is defined inductively as follows:

- $[\![[\psi \mapsto a]^i]\!]_T = \Upsilon \in 2^{\Omega^i(T)}$ satisfying: $\mu \in \Upsilon$ iff μ satisfies the condition that, if $s \in S_\mu$ is a player i node then $M, s \models \psi$ implies $out_\mu(s) = a$.
- $[\![\eta_1 + \eta_2]\!]_T = [\![\eta_1]\!]_T \cup [\![\eta_2]\!]_T$
- $[\![\eta_1 \cdot \eta_2]\!]_T = [\![\eta_1]\!]_T \cap [\![\eta_2]\!]_T$

Above, $out_\mu(s)$ is the unique outgoing edge in μ at s. Recall that s is a player i node and therefore by definition of a strategy for player i there is a unique outgoing edge at s.

To model players' responses, we introduce the formula $\bar{\imath}?\zeta$ in the syntax of $BPF(P^i)$, where $\bar{\imath}$ denotes the opponent of i. The intuitive reading of the formula is "player $\bar{\imath}$ is playing according to a partial strategy conforming to the specification ζ at the current stage of the game", and the semantics is given by,

- $M, s \models \bar{\imath}?\zeta$ iff $\exists T'$ such that $T' \in [\![\zeta]\!]_T$ and $s \in T'$.

3.2 Marble Drop game: a test case

We now express the empirical strategic reasoning performed by the participants of the Marble drop game described in Section 2.1. The game form is structurally equivalent to the Centipede game tree. Figure 3a gives the corresponding tree structure, and figures 3b and 3c correspond to example cases.

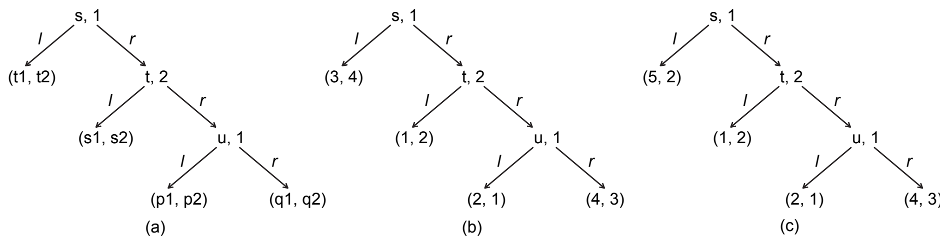

Fig. 3. Example trees.

Using the strategy specification language introduced in Section 3.1, we express the different reasoning methods of participants that have been validated by the experiments described in Section 2. The reasoning is carried out by an outside agent (participant) regarding the question:

> How would the players 1 and 2 play in the game, under the assumptions that both players are rational (thus will try to maximize their utility), and that there is common knowledge of rationality among the players.

We abbreviate some formulas which describe the payoff structure of the game.

$\langle r\rangle\langle r\rangle\langle l\rangle((u_1 = p_1) \wedge (u_2 = p_2)) = \alpha$ (two r moves and one l move lead to (p_1, p_2))
$\langle r\rangle\langle r\rangle\langle r\rangle((u_1 = q_1) \wedge (u_2 = q_2)) = \beta$ (three r moves lead to (q_1, q_2))
$\langle r\rangle\langle l\rangle((u_1 = s_1) \wedge (u_2 = s_2)) = \gamma$ (one r move and one l move lead to (s_1, s_2))
$\langle l\rangle((u_1 = t_1) \wedge (u_2 = t_2)) = \delta$ (one l move leads to (t_1, t_2))

A formula describing backward reasoning giving the correct answer corresponding to the game tree given in Figure 3b is:

$$\varphi_1 : \big([\alpha \wedge \beta \wedge \langle r\rangle\langle r\rangle\mathbf{turn}_1 \wedge (2 \leqslant 4) \wedge \gamma \wedge \langle r\rangle\mathbf{turn}_2 \wedge (2 \leqslant 3) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (3 \leqslant 4) \mapsto r]^1, [\alpha \wedge \beta \wedge \langle r\rangle\langle r\rangle\mathbf{turn}_1 \wedge (2 \leqslant 4) \wedge \gamma \wedge \langle r\rangle\mathbf{turn}_2 \wedge (2 \leqslant 3) \mapsto r]^2, [\alpha \wedge \beta \wedge \langle r\rangle\langle r\rangle\mathbf{turn}_1 \wedge (2 \leqslant 4) \mapsto r]^1\big)$$

> 'If the utilities and the turns of players at the respective nodes are as in Figure 3b, then player 1 would play r at the root node, player 2 would continue playing r at his node, after which player 1 can finish off by playing r.'

Another formula describing forward reasoning giving a wrong answer corresponding to the game tree given in Figure 3b is:

$$\varphi_2 : \big([\mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge \langle r\rangle \mathbf{turn}_2 \wedge \gamma \wedge (1 \leqslant 3) \mapsto l]^1\big)$$

'If the utilities at the first two leaf-nodes of the game are as Figure 3b, and players 1 and 2 move respectively in the first two non-terminal nodes, then player 1 would play l at the root node finishing it off.'

The last formula describes forward reasoning giving a correct answer corresponding to the game tree given in Figure 3c is:

$$\varphi_3 : \big([\mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge \langle r\rangle \mathbf{turn}_2 \wedge \gamma \wedge (1 \leqslant 5) \mapsto l]^1\big)$$

'If the utilities at the first two leaf-nodes of the game are as Figure 3c, and players 1 and 2 move respectively in the first two non-terminal nodes, then player 1 would play l at the root node finishing it off.'

These are just some examples to show that one can actually list possible ways of reasoning that can be performed by human reasoners in the Marble Drop game. Such a list aids in developing the cognitive models of the reasoners, as we shall see in the next section.

4 Cognitive modeling

Analyses of eye movements are challenging because they have to deal with great variability typically found in eye-movement data. Salvucci and Anderson [18] suggested using a cognitive computational model to predict eye movements, which can be compared with observed eye movements. This method helps to disentangle explained (i.e., hypothesized) variance from unexplained variance (due to e.g. measurement errors).

Van Maanen and Verbrugge [8] suggested a cognitive model that implemented backward induction. However, the eye-tracking study conducted by Meijering et al. [12] suggests that participants did not use pure backward induction. Thus, in this paper we present preliminary ideas about a more generic cognitive model that implements backward and forward reasoning as well as possible mixtures of the two. Before going into the specific details of our construction of the cognitive computational model, we first provide a general description of the model that we are going to develop.

4.1 ACT-R modeling

The model that we propose has been implemented in ACT-R, which is an integrated theory of cognition as well as a cognitive architecture that many cognitive scientists use to model human cognition [1]. ACT-R consists of modules that link with cognitive functions (e.g., vision, motor processing, and declarative processing) and map with specific brain regions. Each module has a buffer associated with it, and the modules communicate among themselves via these buffers.

A very important property of ACT-R is that cognitive resources are bounded, because each buffer can store just one piece of information at a time. Consequently, if a model has to keep track of more than one piece of information, it has to move it back and forth between two important modules: declarative memory and the problem state. Moving information back and forth comes with a time cost, and could cause a so-called cognitive bottleneck.

The declarative memory module represents long-term memory and stores information encoded in so-called chunks (i.e., knowledge structures). For example, a chunk can be represented as some expression with a defined meaning (e.g. formal expressions). Each chunk in declarative memory has an activation value that determines the speed and success of its retrieval. Whenever a chunk is used, the activation value of that chunk increases. As the activation value increases, the probability of retrieval increases and the latency of retrieval decreases. For example, whenever the chunk of a successful formula is used, its activation value increases. As the activation value of a successful formula increases, its probability (and speed) of retrieval increases.

Anderson [1] provided a formalization of the mechanism that produces the relationship between the probability and speed of retrieval. As soon as a chunk is retrieved from declarative memory, it is put into the module buffer. As mentioned earlier, each ACT-R module has a buffer that may contain one chunk at a time. On a functional level of description, the chunks that are stored in the various buffers are the knowledge structures the cognitive architecture is aware of.

The problem state module (sometimes referred to as 'imaginal') slightly alleviates bounds on cognitive resources, as it also contains a buffer that can hold one chunk. Typically, the problem state stores a sub-solution 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 cognitive model that we present relies on the declarative and problem state modules. More specifically, it retrieves relevant information from declarative memory and moves that information to the problem state buffer whenever it requests the declarative module to retrieve new information, which the declarative module stores in its buffer.

A central procedural system recognizes patterns in the information stored in the buffers, and responds by sending requests to the modules, for example, 'retrieve a fact from declarative memory'. This condition-action mechanism is implemented in production rules. For example, the following production rule represents comparing the last two payoff values in order to decide whether to end or continue a Marble Drop game:

IF the goal is to compare the last two payoff values,
AND the first is greater than the second,
THEN respond end the game.

Here, the first line refers to the goal buffer, the second line to the problem state buffer, and the third line to a manual action. With this brief introduction to ACT-R modeling, we now move on to the specific model construction.

4.2 A cognitive computational model of Marble Drop

The cognitive model that we propose here is based on the model presented previously by [8], but it is more generic because it is not based on a fixed strategy. Instead, the model is based on formulas (cf. Section 3.2) that are selected from a list provided by the logical framework. The formulas can either represent backward reasoning, forward reasoning, or a mix of both (see examples φ_1, φ_2, φ_3 in Section 3.2).

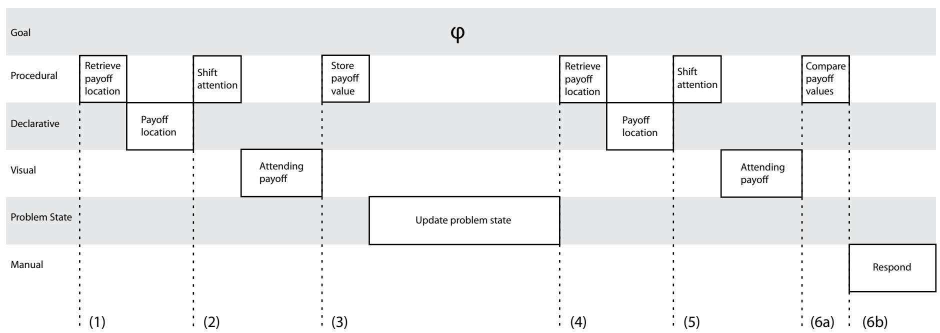

Fig. 4. Flowchart of the ACT-R model.

The flowchart of the model is depicted in Figure 4. Throughout an entire game, the goal buffer stores a chunk that represents which formula (represented by φ) is used. For each pair of payoffs that are compared in the formula, the model iterates through the following steps: The model retrieves the location of the first payoff from declarative memory (1). That location is represented in a chunk, and the more often a location chunk is retrieved the faster that retrieval will be, as the activation value of a chunk increases with each retrieval. As soon as the location of the first payoff is retrieved from declarative memory, the model shifts attention to that location (2). More specifically, the model requests the visual module to shift attention to the location it has just retrieved. After attending the payoff, the model stores the payoff value in the problem state buffer (3). If the model does not store the payoff value in the problem state buffer, it will be lost (i.e., replaced) when the model retrieves a new piece of information. Whenever the payoff value is moved from declarative memory to the problem state, the model retrieves the location of the second payoff from declarative memory (4). After retrieving that location, the model shifts attention to it (5). Now, the model has attended both payoffs, and it compares the payoff value stored in the problem state with the payoff value stored in the visual buffer (6a). After comparing the last pair of payoffs in the formula, the model produces a response (6b).

At the start of each game, a new formula chunk is retrieved from declarative memory. The model tags a formula chunk according to its success, that is, whether the model's response was correct or incorrect, which is indicated by the task feedback presented after each game. The model learns to play Marble Drop games better and faster, as it requests the declarative module to retrieve successful formulas, and the more often those are retrieved and tagged, the higher their activation value becomes. Higher activation value, in turn, increases the probability and speed of retrieving a formula.

The model produces responses and associated reaction times, which we can analyze and compare with the behavioral data. In addition, the model also produces fixations, which we can compare with the human eye movement data. By comparing the model's fixation sequences with the observed fixation sequences in Marble Drop games (Meijering et al. [12]), we can determine what formulas provide a good description of human higher-order social reasoning.

5 Conclusion

The eye-tracking study of Meijering et al. [12] has shown that participants did not use a pure backward induction strategy in the Marble Drop game. We, therefore, constructed a logical model to describe the game, and possible strategies. We use the logical model as a basis for a cognitive computational model, implemented in the cognitive architecture ACT-R.

We want to emphasize that the cognitive model can be considered as a virtual human being. It can do the very same task presented to the participants in Meijering et al.'s [10, 11] studies, and it produces responses and associated response times. The cognitive computational model is useful for a better understanding of higher-order social reasoning, because we can analyze the model output and see which formulas are successful and how quickly the model learns to apply one (set of) formula(s) instead of other formulas.

An advantage of having cognitive models, besides having statistical models, is that cognitive models can be broken down into mechanisms. Our ACT-R model comprises cognitive functions (e.g., a declarative memory and a problem state representation), and we can determine to what extent each cognitive function contributes to the model's behavior (i.e., the responses and response times) in Marble Drop games.

Another advantage of a cognitive model is that we can compare the model's output with Meijering et al.'s human data, and acquire a better understanding of individual differences. Higher-order social reasoning probably consists of multiple serial and concurrent cognitive functions, and thus it may be prone to great individual differences. Our cognitive model may help to determine what formulas fit the responses of a particular (subset of) participant(s). This fit not only concerns patterns in responses and response times, but also patterns in eye-movements. The model's execution of a formula yields eye movements, and we can calculate the explanatory power of eye movement patterns in (subsets of) the human data.

Acknowledgements: The authors gratefully thank Rineke Verbrugge for motivating this work, reading the preliminary drafts a number of times, and always suggesting meaningful changes in both the content and the language of the presentation. The authors would also like to thank the anonymous reviewers for their comments which helped to improve this paper. The first author acknowledges the Netherlands Organisation of Scientific Research grant 600.065.120.08N201, and the second author acknowledges the Netherlands Organisation of Scientific Research grant 227-80-001.

References

1. J. Anderson. *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press, New York (NY), 2007.
2. G. Bonanno. Modal logic and game theory: Two alternative approaches. *Risk, Decision and Policy*, 7(03):309–324, December 2002.
3. B. Edmonds. Social intelligence and multi-agent systems. In *Invited talk, MALLOW'10*, 2010.
4. L. Flobbe, R. Verbrugge, P. Hendriks, and I. Krämer. Children's application of theory of mind in reasoning and language. *Journal of Logic, Language and Information*, 17:417–442, 2008. Special issue on formal models for real people, edited by M. Counihan.
5. S. Ghosh, B. Meijering, and R. Verbrugge. Logic meets cognition: empirical reasoning in games. In *Proceedings of the 3rd International Workshop on Logics for Resource Bounded Agents (LRBA 2010), in 3rd Multi-Agent Logics, Languages, and Organisations Federated Workshops, MALLOW'10, CEUR Workshop Proceedings*, volume 627, pages 15–34, 2010.
6. T. Hedden and J. Zhang. What do you think I think you think? Strategic reasoning in matrix games. *Cognition*, 85:1–36, 2002.
7. E. J. Johnson, C. F. Camerer, S. Sen, and T. Rymon. Detecting failures of backward induction: Monitoring information search in sequential bargaining. *Journal of Economic Theory*, 104(1):16–47, 2002.
8. L. v. Maanen and R. Verbrugge. A computational model of second-order social reasoning. In *Proceedings of the 10th International Conference on Cognitive Modeling*, pages 259–264, 2010.
9. R. McKelvey and T. Palfrey. An experimental study of the centipede game. *Econometrica*, 60(4):803–836, 1992.
10. B. Meijering, L. v. Maanen, H. v. Rijn, and R. Verbrugge. The facilitative effect of context on second-order social reasoning. In *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society, Cognitive Science Society*, pages 1423–1428, 2010.
11. B. Meijering, H. v. Rijn, N. Taatgen, and R. Verbrugge. I do know what you think I think: Second-order theory of mind in strategic games is not that difficult. In *Proceedings of the 33nd Annual Meeting of the Cognitive Science Society, Cognitive Science Society*, 2011.
12. B. Meijering, H. van Rijn, N. Taatgen, and R. Verbrugge. Eye movements during higher-order social reasoning. Technical report, University of Groningen, 2011. (in prep).
13. M. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, Cambridge, MA, 1994.

14. S. Paul, R. Ramanujam, and S. Simon. Stability under strategy switching. In *Proceedings of the 5th Conference on Computability in Europe (CiE 2009), LNCS 5635*, pages 389–398. Springer, 2009.
15. A. Perea. Backward induction versus forward induction reasoning. *Games*, 1(3):168–188, 2010.
16. R. Ramanujam and S. Simon. A logical structure for strategies. In *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, volume 3 of *Texts in Logic and Games*, pages 183–208. Amsterdam University Press, 2008.
17. R. Rosenthal. Games of perfect information, predatory pricing and the chain-store paradox. *Journal of Economic theory*, 25(1):92–100, 1981.
18. D. D. Salvucci and J. R. Anderson. Automated eye-movement protocol analysis. *Human-Computer Interaction*, 16(1):39–86, 2001.
19. R. Verbrugge and L. Mol. Learning to apply theory of mind. *Journal of Logic, Language and Information*, 17:489–511, 2008. Special issue on formal models for real people, edited by M. Counihan.