Logic meets cognition: empirical reasoning in games

Sujata Ghosh, Ben Meijering and Rineke Verbrugge
Department of Artificial Intelligence
University of Groningen
e-mail: {sujata,b.meijering,rineke}@ai.rug.nl

Abstract

This paper presents a first attempt to bridge the gap between logical and cognitive treatments of strategic reasoning in games. The focus of the paper is backward induction, a principle which is purported to follow from common knowledge of rationality by Zermelo's theorem. There have been extensive formal debates about the merits of principle of backward induction among game theorists and logicians. Experimental economists and psychologists have shown that human subjects, perhaps due to their bounded resources, do not always follow the backward induction strategy, leading to unexpected outcomes. Recently, based on an eye-tracker study, it has turned out that even human subjects who produce the outwardly correct 'backward induction answer' use a different internal reasoning strategy to achieve it. This paper presents a formal language to represent different strategies on a finer-grained level than was possible before. The language and its semantics may lead to precisely distinguishing different cognitive reasoning strategies, that can then be tested on the basis of computational cognitive models and experiments with human subjects.

1 Introduction

Strategic reasoning in games concerns the plans or strategies that information-processing agents have for achieving certain goals. *Strategy* is one of the basic ingredients of multi-agent interaction. It is basically the plan of action an agent (or a group of agents) considers for its interactions, that can be modelled as games. From the game-theoretic viewpoint, a strategy of a player can be defined as a partial function from the set of histories (sequences of events) at each stage of the game to the set of actions of the player when it is supposed to make a move [OR94]. Agents devise their strategies so as to force maximal gain in the game.

In cognitive science, the term 'strategy' is used much more broadly than in game theory. A well-known example is formed by George Polya's problem solving strategies (understanding the problem, developing a plan for a solution, carrying out the plan, and looking back to see what can be learned) [Pol45]. Nowadays, cognitive scientists construct fine-grained theories about human reasoning strategies [Lov05, JT07], based on which they construct computational cognitive models. These models can be validated by comparing the model's predicted outcomes to results from experiments with human subjects [And07].

Every finite extensive form game with perfect information [OR94] played by rational players has a sub-game perfect equilibrium and *backward induction* is a popular technique to compute such equilibria. The backward induction strategy, which employs iterated elimination of weakly dominated strategies to obtain sub-game perfect equilibria, is a common strategy followed by rational players with common knowledge (belief) of rationality. We provide below an explicit description of the backward induction algorithm in extensive form game trees [Jon80]. We are only considering strictly competitive games played between two players.

Consider a finite extensive form game with perfect information G played between two players E and A, say. In game G, each player i is associated with a utility function u_i which maps each leaf node of the tree to the set $\{0,1\}$. The backward induction procedure $BI(G, i)$ takes as input such a game G and a player i. It decides whether player i has a winning strategy in G and if so, computes the winning strategy. The procedure proceeds as follows. Initially all nodes are unlabelled.

Step 1: All leaf nodes l are labelled with $u_i(l)$.

Step 2: Repeat the following steps till the root node r is labelled: Choose a non-leaf node t which is not labelled, but all of whose successors are labelled.

a) If it is i's turn at t and there exists a successor t' of t which is labelled 1, then label t with 1 and mark the edge (t, t') which gives the best response at that stage.

b) If it is the opponent's turn at t and *every* successor t' is labelled 1, then label t with 1.

Player i will have a winning strategy in the game G if and only if the root node r is labelled with 1 by the backward induction procedure $BI(G, i)$.

One important critique of this backward induction procedure is that it ignores information, and such ignorance is hardly consistent with a broad definition of rationality. Under backward induction, the fact that a player ends up in one particular subgame rather than another subgame is never considered information for the player. The past moves and reasoning of the players are not taken into consideration. Only what follows is reasoned about. That is, the backward induction solution ignores any forward induction reasoning [Per10].

Before proceeding further we should mention here that there have been numerous debates surrounding the backward induction strategy from various angles. The paradigm discussion concerns the epistemic conditions of backward induction. Here, Aumann [Aum95] and Stalnaker [Sta96] have taken conflicting positions regarding the question whether *common knowledge of rationality* in a game of perfect information entails the backward induction solution. Researchers such as Binmore have argued for the need for richer models of players, incorporating irrational as well as rational behavior [Bin96]. For more details on these issues see [Bic88, ACB07, Bra07, BSZ09, HP09, Art09].

From the logical point of view, various characterisations of backward induction can be found in modal and temporal logic frameworks [Bon02, HvdHMW03, vW03, JvdH04, BSZ09]. There are also critical voices around backward induction arising from logical investigations of strategies. While discussing large (perfect information) games played by resource-bounded players, Ramanujam and Simon emphasize that strategizing follows the flow of time in a *top-down* manner rather the *bottom-up* one advocated by the backward induction algorithm [RS08] .

Critique of a different flavor stems from experimental economics [Cam03]. As sketched above, the 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 [MP92] have shown that in a traditional centipede game participants do not behave according to the Nash equilibrium reached by backward induction. 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. However, in McKelvey and Palfrey's experiment, participants stayed in the game for some rounds before ending the game: in fact, only 37 out of 662 games ended with the backward induction solution. McKelvey and Palfrey's explanation of their results is based on reputation and incomplete information. They compare the complete information dynamic centipede game to an incomplete information game, the iterated prisoner's dilemma as investigated by Kreps et al. [KMRW82]. McKelvey's and Palfrey's main idea is that players may believe that there is some possibility that their opponent has payoffs different from the 'official ones': for example, they might be altruists, i.e., they give weight to the opponent's payoff. Another interpretation of this result is that the game-theoretic perspective fails to take into account the reasoning abilities of the participants. That is, perhaps, due to cognitive constraints such as working memory capacity, participants are unable to perform optimal strategic reasoning, even if in principle they are willing to do so.

In conclusion, we find two very different bodies of work on players' strategies in centipede-like games: on the one hand we find idealized logical studies on games and strategies modelling interactive systems, and on the other there are experimental studies on players' strategies and cognitive modelling of their reasoning processes. Both streams of research have been rather disconnected so far.

To the best of the knowledge of the authors, this article presents a first attempt to bridge the gap between the experimental studies, cognitive modeling, and logical studies of strategic reasoning. In particular, we investigate the question whether a logical model can be used to construct better computational cognitive models of strategic reasoning in games. In the next section we discuss some experimental studies on second-order reasoning of players and cognitive models of such reasoning. Section 3 builds up a logical framework to describe empirical reasoning of players. Finally, the last section provides a discussion with some pointers towards future work.

2 Marble Drop: experiments and cognitive model

This section presents a brief overview of the experimental work on Marble Drop games described in [MMRV10], and of the computational cognitive model (in the cognitive architecture ACT-R [And07]), which was developed in [MV10] to provide a model for second-order social reasoning in predicting the opponent's moves in Marble Drop games.

2.1 Higher-order social cognition

One of the pinnacles of intelligent interaction is higher-order theory of mind, an agent's ability to model recursively mental states of other agents, including the other's model of the first agent's mental state, and so forth. More precisely, zero-order theory of mind concerns world facts, whereas $k+1$-order reasoning models k-order reasoning of the other agent or oneself. For example, "Bob knows that Alice knows that he wrote a novel under pseudonym" ($K_{Bob}K_{Alice}p$) is a second-order attribution. Orders roughly correspond to the modal depth of a formula [1]

The authors in [MMRV10, MvRV10] have investigated higher-order social cognition in humans by means of two experiments. They conducted a behavioral experiment to investigate how well humans are able to apply first- and second-order reasoning. Even though behavioral measures can shed some light on the usage of strategies (see e.g. [HZ02]), they are too crude to go into the details of the actual reasoning steps. Johnson, Camerer, Sen, and Rymon's study employed a novel measure to capture those details [JCSR02]. In their sequential bargaining experiment, participants had to bargain with one other player. The amount to bargain and the participant's role in every

[1] We use the term 'higher-order social cognition' instead of 'higher-order theory of mind'. The reason to do this is that in the philosophical controversy between 'theory-theory' and 'simulation-theory', the term 'theory of mind' carries the unwanted connotation that 'theory-theory' is preferred.

round was hidden behind boxes. The participants had to click on a box to make elements of this information visible. This allowed Johnson et al. to record the information regarding what the participants select at a particular time within the bargaining (i.e., reasoning) process. A potential problem of this measure is that participants might feel disinclined to repeatedly check sets of information elements and rather develop an artificial strategy that involves fewer mouse clicks but puts a higher strain on working memory.

To avoid this problem, Meijering et al. choose to employ eye-tracking technology to investigate the details of higher-order social reasoning. They are currently conducting an eye-tracking study to investigate the reasoning steps during higher-order social reasoning [MvRV10]. The findings of this experiment, together with the behavioral results, help to determine the cognitive bounds on higher-order social reasoning. We give a short overview below and refer the reader to the full papers for more details.

2.1.1 Marble Drop games

Meijering et al. [MMRV10] presented participants with strategic games to investigate higher-order social reasoning. In these games, the path of a white marble that was about to drop could be manipulated by removing trapdoors (Figure 2). Experience with world-physics allowed players to see easily how the marble would run through a game, and which player could change the path of the white marble at each decision point in the game. In other words, higher-order social reasoning was embedded in a context that provided an insightful overview of the decisions and the consequences of these decisions.

Earlier, Hedden and Zhang [HZ02] and Flobbe et al. [FVHK08] had also presented participants with strategic games to investigate higher-order social reasoning, but the performance in those games was far from optimal with approximately 60% - 70% correct. The participants could either have had difficulties applying higher-order social reasoning or difficulties understanding the games.

The matrix games (Figure 1) presented in [HZ02] were very abstract, which could have made the games difficult to understand. Embedding the games in a context could have alleviated this problem. Some studies have shown that non-social reasoning can be facilitated by embedding it in a context. For example, for Wason's selection task it has been stated that a social rule-breaking context helps [WS71]; but see [MO91, SvL04]. More convincingly, subjects have been shown to win the game of tic-tac-toe more easily than its equivalent, Number Scrabble [Mic67, Sim79, Wei84]). The role of context and ecological validity in decision making also plays an important role in the work on 'simple heuristics that make us smart' [GT99]. Higher-order social reasoning, which seemed to be very demanding in matrix games, might also benefit from a context embedding.

Flobbe et al. [FVHK08] found some indirect evidence of a facilitative effect of embedding higher-order social reasoning in a context. However, the performance in their experiment was only slightly better than that in Hedden and Zhang's experiments. Flobbe et al. embedded Hedden and Zhang's matrix games in the context of a road game. The road games were played on a computer. The participants were presented with roads that had three junctions, which corresponded with the cell-transitions in the matrix games. At each junction either the participant or the computer decided to move ahead (i.e., continue the game) if there was a higher payoff to attain further in the game, or to turn right (i.e., end the game) if there was no higher payoff to attain further in the game. The participant and the computer alternately took seat in the driver's position; the one in the driver's seat made the decision.

The performance in the road games might not have been optimal because of the unnatural characteristic of the context in these games. The participants (and the computer) alternately changed

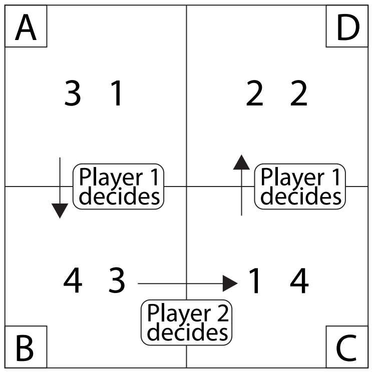

Figure 1: A schematic overview of a matrix game [HZ02]. The first number in each cell is Player 1's payoff, the second Player 2's payoff. The goal is to attain the highest possible payoff. Participants first had to predict what the other player would do at cell B before making a decision what to do at cell A: to stay and stop the game, or to move to cell B. In this example, Player 1 would have to predict that Player 2 will stay, because Player 1 will move to cell D if given a choice at cell C, leading to a lower payoff for Player 2, namely 2 instead of 4. Consequently, the rational decision for Player 1 is to move to cell B in the first move.

driver's seat, which is not common practice in everyday life. Because this unnatural characteristic necessitates some (additional) explanation, the context in the road games was not insightful at first glance.

Meijering et al. [MMRV10] expected that the performance could be improved much further by embedding higher-order social reasoning in their context of a marble game, which was more intuitive and required less explanation. Importantly, these so-called Marble Drop games were game-theoretically equivalent to the matrix games of [HZ02] and the road games of [FVHK08]. All game types have the same extensive form, namely that of the Centipede game [Ros81] (see http://www.ai.rug.nl/ leendert/Equivalence.pdf). Consequently, an improvement in performance can be attributed to a context effect.

In Marble Drop games, the payoffs are color-graded marbles, instead of payoff numbers. Meijering et al. presented color-graded marbles instead of numbers to minimize the usage of numeric strategies other than first- and second-order reasoning. The color-graded marbles can easily be ranked according to preference, lighter marbles being less preferred than darker marbles. The ranking makes it possible to have payoff structures similar to those in matrix and road games. The sets of trapdoors in Marble Drop games correspond to the transitions, from one cell to another, in matrix games [HZ02], and to the junctions in road games [FVHK08].

Figure 2 depicts example games of Marble Drop. The goal is to let the white marble end up in the bin with the darkest color-graded marble of one's own target color (orange or blue). Note that, for illustrative purposes, the color-graded marbles are replaced with codes: a1 - a4 represent the participant's color-graded marbles and b1 - b4 represent the computer's color-graded marbles (which are of another color); 1 - 4 being light to dark grades. (See `http://www.ai.rug.nl/~meijering/MarbleDrop.html` for the original Marble Drop games.) The diagonal lines represent the trapdoors.

In the example game in Figure 2a, participants need to remove the right trapdoor to attain the darkest color-graded marble of their color (a2). The game in Figure 2a is a zero-order game, because there is no other player to reason about. In first-order games (Figure 2b) participants need to reason about another player, the computer. The computer is programmed to let the white

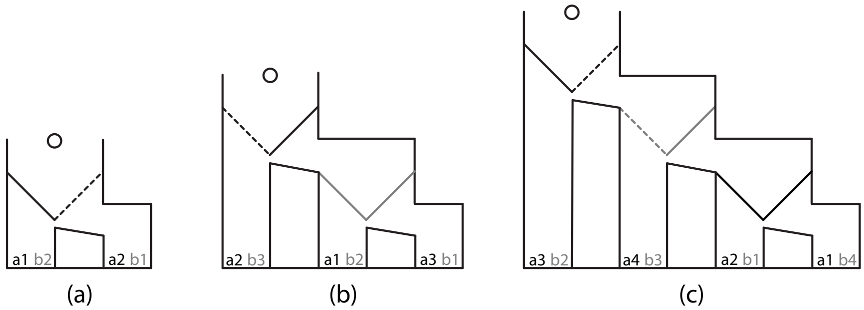

Figure 2: A zero-order (a), first-order (b) and second-order (c) Marble Drop game. The participant's payoffs are represented by a1 - a4, the computer's by b1 - b4, both in increasing order of value. The goal is to let the white marble end up in the bin with the highest attainable payoff. The diagonal lines represent trapdoors. At the first set of trapdoors, the participant decides which of both trapdoors to remove, at the second set the computer decides, and at the third set the participant again decides. The dashed lines represent the trapdoors that both players should remove to attain the highest payoff they can get.

marble end up in the bin with the darkest color-graded marble of its target color, which is different from the participant's target color. Participants need to reason about the computer, because the computer's decision at the second set of trapdoors affects at what bin a participant can end up.

In the example game in Figure 2b, if given a choice at the second set of trapdoors, the computer will remove the left trapdoor, because its marble in the second bin (b2) is darker than its marble in the third bin (b1). Consequently, the participant's darkest marble in the third bin (a3) is unattainable. The participant should therefore remove the left trapdoor (of the first set of trapdoors), because the marble of their target color in the first bin (a2) is darker than the marble of their target color in the second bin (a1).

In a second-order game (Figure 2c) there is a third set of trapdoors at which the participants again decide which trapdoor to remove. They need to apply second-order reasoning, that is, reason about what the computer, at the second set of trapdoors, thinks that they, at the third set of trapdoors, think.

In Marble Drop games, half of the participants first had to predict what the opponent would do (at the second set of trapdoors) before deciding what to do at the first set of trapdoors. The participants were instructed that the opponent was rational, as were the participants in Hedden and Zhang's and Flobbe et al.'s experiments[2].

Similar to the supporting role of scaffolding in the construction of a building, (instructional) scaffolding provides a supporting structure to learn a new concept or skill that is beyond the independent efforts of a student [WBR76]. Asking participants to make a prediction about the opponent before making a decision was thought to support second-order social reasoning, as decisions should be based on predictions, and making a prediction involves thinking about what the opponent thinks that the participant thinks. In matrix and road games, all the participants first had to give their prediction of what the opponent would do before making their own decision. Meijering et al. [MMRV10] set out to investigate whether embedding higher-order social reasoning in a context that allows for an insightful overview of the decisions and the consequences of these decisions might render such scaffolding obsolete.

[2]One group of participants in Hedden and Zhang's experiments [HZ02] knew they were playing against a computer whereas another group did not. Hedden and Zhang found no difference in performance for the two groups.

2.2 Results

Twenty first-year Psychology students participated in exchange for course credit. The mean age of the included participants (12 female) was 21.15 years (SE = 0.36).

Whereas the participants in the matrix games [HZ02], and the road games [FVHK08] had difficulties applying second-order reasoning, in Marble Drop games [MMRV10], participants performed almost at ceiling: in 94% of the games, participants correctly applied second-order reasoning. In the former two game types, the performance ranged between 60 - 70%.

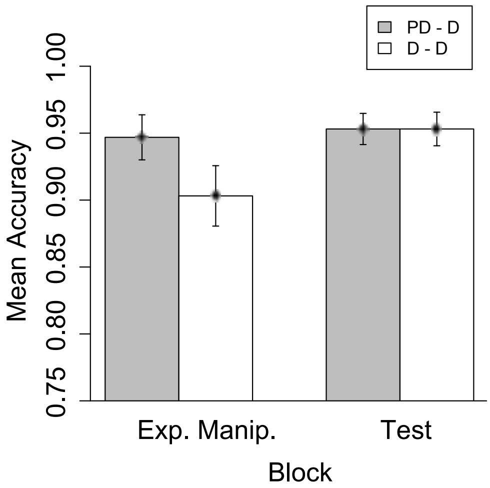

Figure 3: Mean proportion of games in which participants correctly applied second-order reasoning, presented separately for the PD-D and D-D groups in the experimental manipulation block and the test block. The standard errors are depicted above and below the means.

According to the results in [MMRV10], there is no need to scaffold second-order reasoning, given that it is embedded in an ecologically valid context. The performance of the participants that had to make a prediction before making a decision (the PD-D group) was slightly better in the first half (the experimental manipulation block) of the experiment, but in the second half (the test block), the participants that had to make a prediction (the PD-D group) and the participants that did not (the D-D group) performed equally well (Figure 3).

In sum, [MMRV10] demonstrated that adolescents are able to apply second-order reasoning if it is embedded in an ecologically valid context, and that such an embedding renders scaffolding unnecessary. Currently, [MvRV10] are doing an eye-tracking study to elaborate the steps in second-order reasoning during Marble Drop games. We will come back to this discussion in Section 2.5.

2.3 Computational cognitive models of game reasoning

A different perspective, that focuses on cognitive validity in developing formal models, is that of a cognitive architecture [And07]. 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 higher-order strategic reasoning into account. For example, cognitive models of simple games exist in which it is important to know the opponent's behavior [LWW00, WLB06]. These cognitive models demonstrate that declarative memory is important in playing strategically. In [MV10] however, the authors are less interested in how people adapt their strategy to an opposing strategy, but rather in studying the cognitive limitations of explicit second-order reasoning.

To provide a full model of second-order social reasoning, [MV10] implemented their model in the cognitive architecture ACT-R [And07]. 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 at a time. If the current contents of all buffers in the system match 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.

Two modules of ACT-R deserve extra attention in the light of the model of second-order social reasoning in [MV10]: the *declarative memory module* and the *problem state module*. The *declarative memory module* represents long-term memory and stores information encoded in so-called *chunks* (i.e., knowledge structures). 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. Anderson [And07] provided a formalization of the mechanism that produces the relationship between the probability and speed of retrieval. 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. Then, the higher the activation value, the faster the retrieval will be. As soon as a chunk is retrieved, it is put into the retrieval buffer. 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 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 Section 2.4 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.

2.4 A computational cognitive model of the Marble drop game

The ACT-R model proposed by [MV10] follows a backward induction strategy to predict the opponent's moves further on in the game. Hedden and Zhang [HZ02] provide a decision tree analysis of this process for their matrix version of the game. 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.

Because Van Maanen et al. implemented the backward induction strategy, the model starts a second-order game by comparing its own payoff values in bins 3 and 4. First, the model retrieves

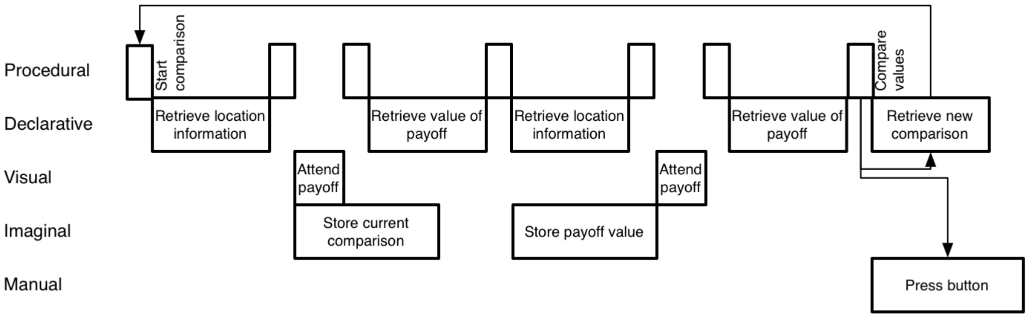

Figure 4: Flow-chart of the ACT-R model for solving Marble Drop game (figure from [MV10])

from declarative memory where the first of two payoffs is located on the screen (i.e., bin 4). If the model retrieves the location of bin 4, it attends bin 4 and tries to retrieve the value of the observed payoff. At the same time, the model frees the retrieval buffer for the upcoming payoff value and stores a chunk in the problem state buffer to represent the comparison that is currently made. Next, the model retrieves the location information for the other of the two payoffs that are currently compared. Again, the model frees the retrieval buffer (for the upcoming payoff value) and the payoff value of the first of two payoffs is stored in the problem state buffer. The problem state buffer can hold only one chunk at the same time and consequently the chunk that represents the current comparison is cleared from the problem state buffer. The location of the other payoff (i.e. bin 3) is attended and the corresponding value is retrieved from memory. Finally, the two payoff values are compared. The model tries to retrieve from declarative memory a chunk with the ordinal representation of both payoff values. Based on the outcome of this retrieval the model retrieves a next comparison.

For example, if the model's payoff value in bin 4 is greater than the model's payoff value in bin 3, the model will next compare the opponent's payoffs in bins 4 and 2. If the model's payoff value in bin 4 was less than the model's payoff value in bin 3, the model will next compare the opponent's payoffs in bins 3 and 2. The model continues to compare payoffs following the decision tree [HZ02] until it reaches the root of this tree. There, it decides its action based on the final comparison.

The model was tested against data from a Marble Drop task described by [MMRV10]. 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 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.

As illustrated by figure 5, it turned out that the fit on the response time is very good (R2 = 1.0; RMSE = 0.42 s). The fit on the accuracy data is slightly less (RMSE = 0.067, R2 = 0.2), but this may be attributed to lack of data, making the estimated means less reliable. The model shows a very high proportion of correct responses, similar to the data of [MMRV10] (but contrary to [FVHK08]).

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

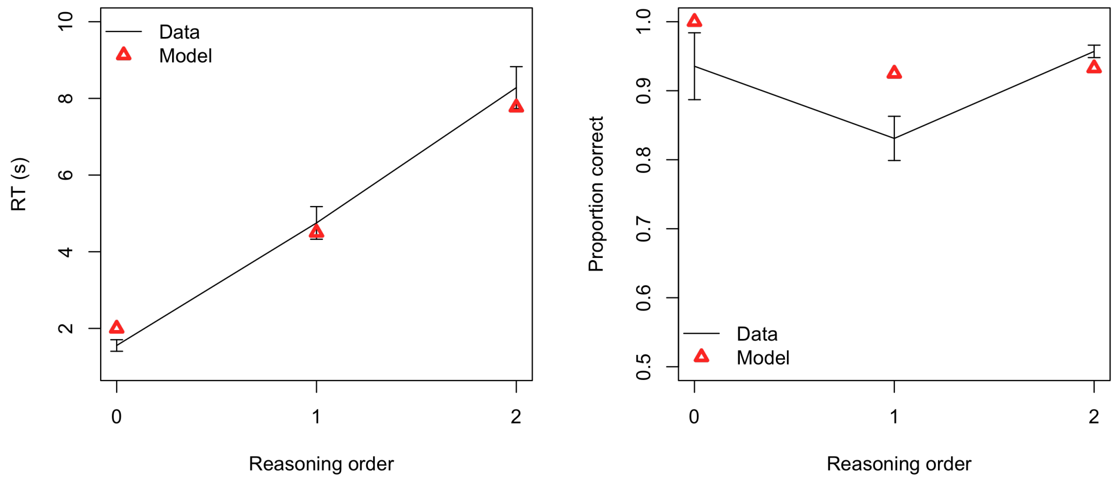

Figure 5: Model fit to data from [MMRV10]. Left: response time. Right: Accuracy (figure from [MV10]

task follow the same reasoning steps as the model does. That is, the hypothesis is supported that participants in a social reasoning game follow a decision tree to make the correct decision.

Another good validation of the model is to compare the sequence of spatial locations it attends with actual eye movements of participants during second-order social reasoning. Sequences of eye-movements (and more generally actions) help to determine the cognitive strategies involved in a task, and the similarity between the predicted and observed eye movements are indicative of the model's validity [SA01].

2.5 Eye movements during second-order reasoning

Game-theoretically, participants are expected to use backward induction [OR94, Ros81]. In Marble Drop games, this would yield eye movements from right to left. However, the preliminary results from the eye-tracking study of [MvRV10] show opposite patterns: initially, participants seem to reason from left to right; they fixate at bins 1, 2, 3, and 4, in that order. Also, Figure 6 clearly shows that at position 1, the mean proportion of fixations is higher in bins 1 and 2, which correspond with the areas of interest (AOIs) A and B rather than in bins 3 and 4, which correspond with the AOIs C and D.

Accordingly, second-order reasoning seems to be context-driven. There is an evident left to right orientation in the Marble Drop context, as the white marble will run from left to right through a game for as long as both players continue the game, and the eye movements seem to follow that orientation.

Another finding that corroborates the idea of context-driven reasoning is that the proportion trends differ for some of the payoff structures (Figure 3). If participants had used backward induction, the eye movements would not vary, and occur independent of payoff structure. Differential proportion trends do fit context-driven reasoning, as it is susceptible to a *bottom-up* influence of orientation (left to right) and probably also features such as color-saliency (i.e., payoff value).

After an initial exploration of the context, participants will have to make some additional, backward comparisons if the intermediate reasoning steps have not yielded an optimal outcome (i.e., there is a darker, attainable marble in another bin than the bin that is the outcome of the last decision). Consider the payoff structure 3:1 (bin 1), 4:3 (bin 2), 1:2 (bin 3), and 2:4 (bin 4), with

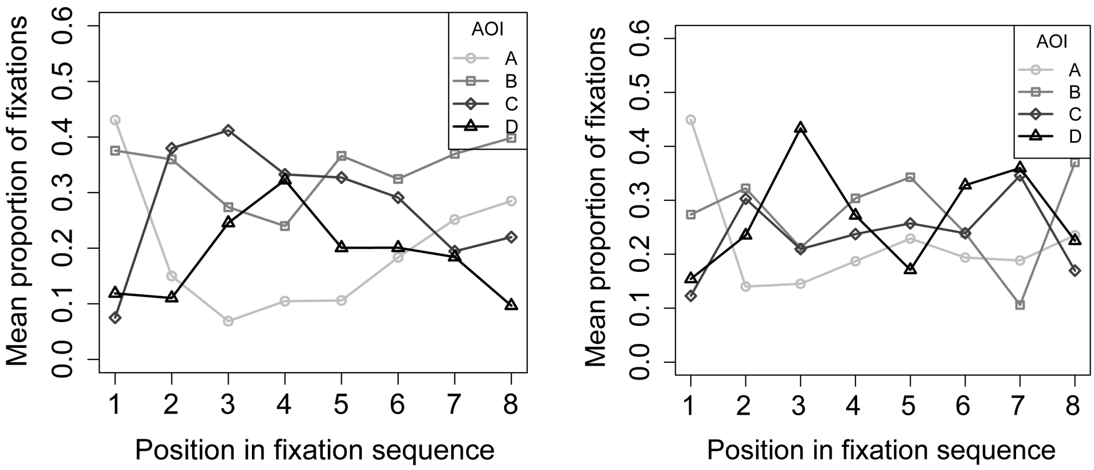

Figure 6: Mean proportion of fixations in bins 1 - 4, which correspond with the areas of interest (AOIs) A - D, as a function of position in the fixation sequence. The proportion trends differ for games in which a rational opponent would end the game and a rational participant would continue the game (a) and games in which both players would continue the game (b).

the first payoff value that of the participant and the second that of the opponent. A participant could reason forwardly: "I would go right (i.e., remove the right trapdoor) if the opponent ends the game at bin 2 (i.e., removes the left trapdoor)" and then "the opponent would end the game if I do not to continue the game from bin 3 to bin 4". However, as the reasoning continues with "I would go to the right from bin 3 to bin 4", a participant would have to reason backwards: "thus the opponent will not stop the game at bin 2" and therefore "if the value of my first payoff (at bin 1) is higher than that of my last payoff (at bin 4) I will stop the game, otherwise I will continue the game".

3 A logical study

Following the lines of work in [RS08, PRS09], we now propose a language specifying strategies of players. This provides an elegant way to describe the empirical reasoning of the participants of the Marble Drop game (cf. Section 2.1.1), that has been found by the eye-tracking study in [MvRV10].

The basic ingredient that is needed for a logical system to model empirical reasoning of human agents, is to forego the usual assumption of *idealised* agents, but rather consider agents with limited computational and reasoning abilities. Though players with limited rationality are much more realistic to consider, for the time being, we only focus on *perfectly rational* players, whose only goal is to win the game. To model strategic reasoning of such resource-bounded players, we should note that these players are in general forced to strategize *locally* by selecting what part of the past history they choose to carry in their memory, and to what extent they can look ahead in their analysis. We consider the notion of *partial strategies* (formalised below) as a way to model such resource-bounded strategic reasoning.

3.1 Preliminaries

In this subsection, representations for extensive form games, game trees and strategies are presented, similar to those in [RS08, PRS09]. On the basis of these notations, reasoning strategies can be formalized in Subsection 3.2.

3.1.1 Extensive form games

Extensive form games are a natural model for representing finite games in an explicit manner. In this model, the game is represented as a finite tree where the nodes of the tree correspond to the game positions and edges correspond to moves of players. For this logical study, we will focus on game forms, and not on the games themselves, which come equipped with players' payoffs at the leaf nodes of the games. We present the formal definition below.

Let N denote the set of players; we use i to range over this set. For the time being, we restrict our attention to two player games, i.e. we take $N = \{1, 2\}$. We often use the notation i and $\bar{\imath}$ to denote the players where $\bar{\imath} = 2$ when $i = 1$ and $\bar{\imath} = 1$ when $i = 2$. Let Σ be a finite set of action symbols representing moves of players, we let a, b range over Σ. For a set X and a finite sequence $\rho = x_1 x_2 \dots x_m \in X^*$, let $last(\rho) = x_m$ denote the last element in this sequence.

3.1.2 Game trees

Let $\mathbb{T} = (S, \Rightarrow, s_0)$ be a tree rooted at s_0 on the set of vertices S and $\Rightarrow : (S \times \Sigma) \to S$ be a *partial* function specifying the edges of the tree. The tree $\mathbb{T}$ is said to be finite if S is a finite set. For a node $s \in S$, let $\overrightarrow{s} = \{s' \in S \mid s \overset{a}{\Rightarrow} s' \text{ for some } a \in \Sigma\}$. A node s is called a leaf node (or terminal node) if $\overrightarrow{s} = \emptyset$.

An extensive form game tree is a pair $T = (\mathbb{T}, \widehat{\lambda})$ where $\mathbb{T} = (S, \Rightarrow, s_0)$ is a tree. The set S denotes the set of game positions with s_0 being the initial game position. The edge function $\Rightarrow$ specifies the moves enabled at a game position and the turn function $\widehat{\lambda} : S \to N$ associates each game position with a player. Technically, we need player labelling only at the non-leaf nodes. However, for the sake of uniform presentation, we do not distinguish between leaf nodes and non-leaf nodes as far as player labelling is concerned. An extensive form game tree $T = (\mathbb{T}, \widehat{\lambda})$ is said to be finite if $\mathbb{T}$ is finite. For $i \in N$, let $S^i = \{s \mid \widehat{\lambda}(s) = i\}$ and let $frontier(\mathbb{T})$ denote the set of all leaf nodes of T.

A play in the game T starts by placing a token on s_0 and proceeds as follows: at any stage, if the token is at a position s and $\widehat{\lambda}(s) = i$ then player i picks an action which is enabled for her at s, and the token is moved to s' where $s \overset{a}{\Rightarrow} s'$. Formally a play in T is simply a path $\rho : s_0 a_0 s_1 \cdots$ in $\mathbb{T}$ such that for all $j > 0$, $s_{j-1} \overset{a_j}{\Rightarrow} s_j$. Let $Plays(T)$ denote the set of all plays in the game tree T.

3.1.3 Strategies

A strategy for player i is a function μ^i which specifies a move at every game position of the player, i.e. $\mu^i : S^i \to \Sigma$. For $i \in N$, we use the notation μ^i to denote strategies of player i and $\tau^{\bar{\imath}}$ to denote strategies of player $\bar{\imath}$. By abuse of notation, we will drop the superscripts when the context is clear and follow the convention that μ represents strategies of player i and τ represents strategies of $\bar{\imath}$. A strategy μ can also be viewed as a subtree of T where for each node belonging to player i, there is a unique outgoing edge and for nodes belonging to player $\bar{\imath}$, every enabled move is included. Formally we define the strategy tree as follows: For $i \in N$ and a player i's strategy $\mu : S^i \to \Sigma$, the strategy tree $T_\mu = (S_\mu, \Rightarrow_\mu, s_0, \widehat{\lambda}_\mu)$ associated with μ is the least subtree of T satisfying the following property:

- $s_0 \in S_\mu$
- For any node $s \in S_\mu$,
  - if $\widehat{\lambda}(s) = i$ then there exists a unique $s' \in S_\mu$ and action a such that $s \overset{a}{\Rightarrow}_\mu s'$.
  - if $\widehat{\lambda}(s) \neq i$ then for all s' such that $s \overset{a}{\Rightarrow} s'$, we have $s \overset{a}{\Rightarrow}_\mu s'$.

Let $\Omega^i(T)$ denote the set of all strategies for player i in the extensive form game tree T. A play $\rho : s_0 a_0 s_1 \cdots$ is said to be consistent with μ if for all $j \geqslant 0$ we have $s_j \in S^i$ implies $\mu(s_j) = a_j$. A strategy profile (μ, τ) consists of a pair of strategies, one for each player.

3.1.4 Partial strategies

A partial strategy for player i is a partial function σ^i which specifies a move at some (and, not all) game positions of the player, i.e. $\sigma^i : S^i \rightharpoonup \Sigma$. Let $\mathfrak{D}_{\sigma^i}$ denote the domain of the partial function σ^i. For $i \in N$, we use the notation σ^i to denote partial strategies of player i and $\pi^{\bar{\imath}}$ to denote partial strategies of player $\bar{\imath}$. When the context is clear, we refrain from using the superscripts. A partial strategy σ can also be viewed as a subtree of T where for some nodes belonging to player i, there is a unique outgoing edge and for other nodes belonging to player i as well as nodes belonging to player $\bar{\imath}$, every enabled move is included. Formally we define a partial strategy tree as follows: For $i \in N$ and a player i (partial) strategy $\sigma : S^i \rightharpoonup \Sigma$ the strategy tree $T_\sigma = (S_\sigma, \Rightarrow_\sigma, s_0, \widehat{\lambda}_\sigma)$ associated with σ is the least subtree of T satisfying the following property:

- $s_0 \in S_\mu$
- For any node $s \in S_\mu$,
  - if $\widehat{\lambda}(s) = i$ and $s \in \mathfrak{D}_\sigma$ then there exists a unique $s' \in S_\mu$ and action a such that $s \overset{a}{\Rightarrow}_\mu s'$.
  - if $(\widehat{\lambda}(s) = i$ and $s \notin \mathfrak{D}_\sigma)$ or $\widehat{\lambda}(s) \neq i$ then for all s' such that $s \overset{a}{\Rightarrow} s'$, we have $s \overset{a}{\Rightarrow}_\mu s'$.

A partial strategy can be viewed as a set of total strategies. Given a partial strategy tree T_σ for a partial strategy σ for player i, a set of trees $\widehat{T_\sigma}$ of total strategies can be defined as follows. A tree $T = (S, \Rightarrow, s_0, \widehat{\lambda}) \in \widehat{T_\sigma}$ if and only if

- if $s \in T$ then for all $s' \in \overrightarrow{s}$, $s' \in T$ implies $s' \in T_\sigma$
- if $\widehat{\lambda}(s) = i$ then there exists a unique $s' \in S_\sigma$ and action a such that $s \overset{a}{\Rightarrow}_\sigma s'$.

Note that any total strategy is also viewed as a partial strategy, where the corresponding set of total strategies becomes a singleton set.

3.2 Strategy specifications

We now propose a syntax for specifying partial strategies and their compositions in a structural manner involving simultaneous recursion. 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.

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 $a \in \Sigma$.

Formulas in $BPF(X)$ 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 $Bool(X)$.

3.2.1 Syntax

Let $P^i = \{p_0^i, p_1^i, \ldots\}$ be a countable set of observables for $i \in N$ and $P = \bigcup_{i \in N} P^i$. 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 $\psi \in BPF(P^i)$.

The 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.

3.2.2 Semantics

Let $M = (T, V)$ where $T = (S, \Rightarrow, s_0, \widehat{\lambda})$ is an extensive form game tree and $V : S \to 2^P$ 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 a s' such that $s \overset{a}{\Rightarrow} s'$ and $M, s' \models \psi$.
- $M, s \models \langle a^- \rangle \psi$ iff there exists a s' such that $s' \overset{a}{\Rightarrow} s$ and $M, s' \models \psi$.

Strategy specifications are interpreted on strategy trees of T. We 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 $\lambda(s) = i$.

One more special proposition $\mathbf{root}$ 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 $s = s_0$.

Recall that a partial strategy σ of player i can be viewed as a set of total strategies of the player and each such strategy is a subtree of T.

The semantics of the strategy specifications are given as follows. Given the game tree $T = (S, \Rightarrow, s_0, \widehat{\lambda})$ and a partial strategy specification $\eta \in Strat^i(P^i)$, we define a 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 $\eta \in Strat^i(P^i)$, $[\![\cdot]\!]_T$ 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 there is a unique outgoing edge at s.

3.2.3 Response and future planning of players

Modelling a player's response to the opponent's play is one of the basic notions that we need to deal with while describing reasoning in games. To this end, we introduce one more construct in our language of pre-conditions $BPF(X)$. The idea is to model the phenomenon that if player $\bar{\imath}$ has played according to π in the history of the game, player i responds following some strategy σ, say. We may also describe that since player $\bar{\imath}$ may play according to π at a certain future point of the game (if it so happens that the game reaches that point), in anticipation to which player i can now play according to σ.

To model such scenarios we introduce the formula $\bar{\imath}?\zeta$ in the syntax of $BPF(P^i)$. 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 $s \in T'$.

Note that this involves simultaneous recursion in the definitions of $BPF(X)$ and $Strat^i(P^i)$. The framework introduced by Ramanujam and Simon [RS08] has a more simpler version of $BPF(P^i)$, where only past formulas are considered, but they introduce an additional construct in the syntax of strategy specifications, viz. $\pi \Rightarrow \sigma$, which says that at any node player i plays according to the strategy specification σ if on the history of the play, all the moves made by $\bar{\imath}$ conforms to π. The introduction of the formula $\bar{\imath}?\zeta$ in the language of $BPF(P^i)$ empowers us to model notions expressed by the specification $\pi \Rightarrow \sigma$. We leave the detailed technicalities involving our proposal as well as the comparative discussion with the related framework of [RS08] for future work.

3.3 Marble Drop game: a test case

We are now well-equipped to express the empirical reasoning performed by the participants of the Marble drop game described in Section 2.1.1. The game form is structurally equivalent to the Centipede game tree given in Figure 7.

Using the strategy specification language introduced in Section 3.2, we express the different reasoning methods of participants that have been validated by the experiments described in Section 2.5. 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.

In particular, we encode three different types of reasoning. Note that we are not talking about why the players reason in some way, i.e. what considerations lead them to such reasoning, (e.g. in the marble drop game, the darker marble may lie at the first bin itself or may be at the third bin) but rather how they can go about playing the game. Combining the "why" and "how" at this level is something we leave for future work:

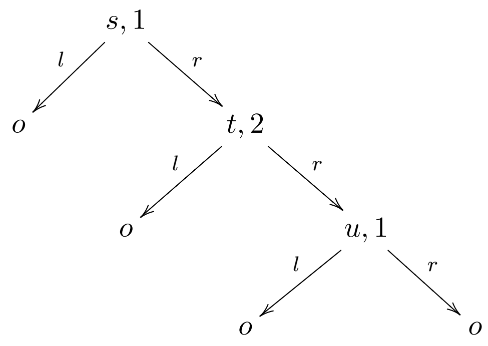

Figure 7: Centipede game tree

- forward reasoning: $\langle [1?[\mathbf{root} \wedge \mathbf{turn}_1 \mapsto r]^1 \wedge \langle r^- \rangle \mathbf{root} \wedge \mathbf{turn}_2 \mapsto r]^2, [2?[\langle r^- \rangle \mathbf{root} \wedge \mathbf{turn}_2 \mapsto r]^2 \wedge \langle r^- \rangle \langle r^- \rangle \mathbf{root} \wedge \mathbf{turn}_1 \mapsto l]^1 \rangle$.

  if player 1 makes the move r at the root node, player 2 will respond with playing r, and if player 2 behaves like that, player 1 would play l.

- backward reasoning: $\langle [1?[\langle r^- \rangle \langle r^- \rangle \mathbf{root} \wedge \mathbf{turn}_1 \mapsto r]^1 \wedge \mathbf{turn}_2 \mapsto r]^2, [2?[\langle r^- \rangle \mathbf{root} \wedge \mathbf{turn}_2 \mapsto r]^2 \wedge \mathbf{root} \wedge \mathbf{turn}_1 \mapsto l]^1 \rangle$.

  if player 1 makes the move r at the u node (if the game reaches there) , player 2 will play r when her turn comes, and if player 2 behaves like that, player 1 would play l at the start node.

- combined reasoning: $\langle [\mathbf{root} \wedge \mathbf{turn}_1 \mapsto r]^1, [1?[\langle r^- \rangle \langle r^- \rangle \mathbf{root} \wedge \mathbf{turn}_1 \mapsto r]^1 \wedge \langle r^- \rangle \mathbf{root} \wedge \mathbf{turn}_2 \mapsto r]^2 \rangle$.

  player 1 would play l at the root node, and then player 2 will play r after that, since if the game reaches the u node, player 1 will play r.

The partial strategy profiles of the two players describe in each case what sort of reasoning players 1 and 2 might perform in the course of the game so as to gain maximum benefits. As apparent from the descriptions of such reasoning, this really captures the arbitrary as well as methodical reasoning that humans can do when confronted with such games, as exemplified (to a certain extent) in Section 2.5. In fact, the combined reasoning suggests that player 1 might make an arbitrary move at the beginning, which can often be the case in reality. As found in the eye-tracking study of [MvRV10], the human participants follow different procedures of reasoning (in particular, first a forward scan of the context, followed if necessary by backward reasoning), rather than the backward induction strategy prescribed by game theorists. These reasoning procedures are adequately described by the (partial) strategy specification language proposed in Section 3.2. This formal representation can provide the building blocks for a better cognitive model based on the ACT-R architecture, in order to construct precise computational cognitive models for the varied thought processes of human agents.

4 Discussion and future work

To put this first attempt at bridge-building between logic and experimental work on games and strategies into perspective, it may be fruitful to keep in mind the three levels of inquiry for cognitive science that David Marr characterized [Mar82]:

- identification of the information-processing agents task as an input - output function: the computational level;
- specification of an algorithm which computes the function: the algorithmic level;
- physical/neural implementation of the algorithm specified: the implementation level.

Researchers aiming to answer the question what logical theories may contribute to the study of resource-bounded strategic reasoning could be disappointed when it turns out that logic is not the best vehicle to describe such reasoning at the implementation level. Still, logic surely makes a contribution at Marr's first computational level by providing a precise specification language for cognitive processes. Quite possibly, logic may also have a fruitful role to play in theories of resource-bounded strategic reasoning at the algorithmic level, in the construction of computational cognitive models in ACT-R. A first step in this direction was made in the previous subsection.

Future work would be to distinguish several possible reasoning strategies for resource-bounded agents in games. For example, Van Maanen et al. propose a new ACT-R model that predicts how humans perform in experiments with a dual task done in parallel to the Marble Drop game [MV10]. That ACT-R model presumes a reasoning strategy following the decision tree analysis of Hedden and Zhang [HZ02]. It would be interesting to also test the predictions of alternative models that correspond to different reasoning strategies, for example, the forward, backward and combined ones introduced in the previous section.

It would also be interesting to define reasoning strategies for games that consist of many more steps and construct corresponding ACT-R models to drive new experimental work.

The great advantage of coupling a strategy logic to ACT-R is that ACT-R already implements very precise, experimentally validated theories about human memory and cognitive bounds on reasoning processes. Thus, there is no need to add (possibly arbitrary) resource bounds in the logical language. The combined strengths of logic, coupled with cognitive modeling and ACT-R, will hopefully lead to an improved understanding of human resource-bounded reasoning in games.

From the logical perspective, providing a sound and complete system for strategic reasoning that models empirical human reasoning will be the essential next step. We would need to take players' preferences into consideration as well as intentions of other players. Evidently, reasoning about intentions is essential for forward induction and solution concepts like extensive-form rationalizability and refinements of sequential equilibrium.

Acknowledgements:

The authors would like to thank the anonymous referees for their insightful comments which helped in the discussions throughout the paper. They also thank Leendert van Maanen and Hedderick van Rijn for the helpful comments regarding the experimental work and cognitive architecture.The first author acknowledges NWO grant # 600.065.120.08N201 and the second and third authors acknowledge NWO grant # 227-80-001.

References

[ACB07] H. Arló-Costa and C. Bicchieri. Knowing and supposing in games of perfect information. *Studia logica*, 86:353373, 2007.

[And07] J.R. Anderson. *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press, New York (NY), 2007.

[Art09] S. Artemov. Knowledge-based rational decisions. Technical report, The CUNY Graduate Center, 2009.

[Aum95] Robert J. Aumann. Backward induction and common knowledge of rationality. *Games Econ. Behav.*, 8(1):6–19, 1995.

[Bic88] C. Bicchieri. Common knowledge and backward induction: a solution to the paradox. In *TARK '88: Proceedings of the 2nd conference on Theoretical aspects of Reasoning about Knowledge*, pages 381–393, San Francisco, CA, USA, 1988. Morgan Kaufmann Publishers Inc.

[Bin96] Ken Binmore. A note on backward induction. *Games and Economic Behavior*, 17(1):135–137, November 1996.

[Bon02] Giacomo Bonanno. Modal logic and game theory: two alternative approaches. *Risk, Decision and Policy*, 7(03):309–324, December 2002.

[Bra07] A. Brandenburger. The power of paradox: Some recent developments in interactive epistemology. *International Journal of Game Theory*, 35:465–492, 2007.

[BSZ09] A. Baltag, S. Smets, and J. Zvesper. Keep 'hoping' for rationality: A solution to the backward induction paradox. *Synthese*, 169(2):301–333, 2009.

[Cam03] C. F. Camerer, editor. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, Princeton, 2003.

[FVHK08] 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.

[GT99] G. Gigerenzer and P.M. Todd. *Simple Heuristics that make us Smart*. Oxford University Press, New York, 1999.

[HP09] J. Y. Halpern and R. Pass. Iterated regret minimization: A new solution concept. In C. Boutilier, editor, *Proceedings of IJCAI-09: 21st International Joint Conferences on Artificial Intelligence, Pasadena, CA, USA*, pages 153–158, 2009.

[HvdHMW03] B.P. Harrenstein, W. van der Hoek, J.-J. Ch. Meyer, and C. Witteveen. A modal characterization of Nash equilibrium. *Fundamenta Informaticae*, 57(2):281–321, 2003.

[HZ02] T. Hedden and J. Zhang. What do you think I think you think? Strategic reasoning in matrix games. *Cognition*, 85:1–36, 2002.

[JCSR02] 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.

[Jon80] A. Jones. *Game theory: Mathematical Models of Conflict*. John Wiley, 1980.

[JT07] I. Juvina and N. A. Taatgen. Modeling control strategies in the n-back task. In *Proceedings of the 8th International Conference on Cognitive Modeling*, New York, NY, 2007. Psychology Press.

[JvdH04] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63, 2004.

[KMRW82] D. M. Kreps, P. Milgrom, J. Roberts, and R. Wilson. Rational cooperation in the finitely repeated prisoners' dilemma. *Journal of Economic Theory*, 27(2):245–252, 1982.

[Lov05] Marsha C. Lovett. A strategy-based interpretation of Stroop. *Cognitive Science*, 29(3):493–524, 2005.

[LWW00] C. Lebiere, D. Wallach, and R. West. A memory-based account of the prisoner's dilemma and other 2x2 games. In N.A. Taatgen and J. Aasman, editors, *Proceedings of Third International Conference on Cognitive Modeling*, pages 185–193, Veenendaal, 2000. Universal Press.

[Mar82] D. Marr. *Vision*. Freeman and Company, New York, 1982.

[Mic67] J. A. Michon. The game of JAM–an isomorph of Tic-Tac-Toe. *American Journal of Psychology*, 80(1):137–140, 1967.

[MMRV10] B. Meijering, L.. van Maanen, H. van 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*, 2010.

[MO91] K.I. Manktelow and D.E. Over. Social roles and utilities in reasoning with deontic conditionals. *Cognition*, 39(2):85–105, 1991.

[MP92] R.D. McKelvey and T.R. Palfrey. An experimental study of the centipede game. *Econometrica*, 60(4):803–836, 1992.

[MV10] L.. van Maanen and R. Verbrugge. A computational model of second-order social reasoning. In *Proceedings of the 10th International Conference on Cognitive Modeling*, 2010.

[MvRV10] B. Meijering, H. van Rijn, and R. Verbrugge. The facilitative effect of context on second-order social reasoning. Technical report, University of Groningen, 2010. (in prep).

[OR94] M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, 1994.

[Per10] A. Perea. Backward induction versus forward induction reasoning. *Games*, 1(3):168–188, 2010.

[Pol45] G. Polya. *How to Solve It. A New Aspect of Mathematical Method*. Princeton University Press, Princeton, N.J., 1945.

[PRS09] 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.

[Ros81] R.W. Rosenthal. Games of perfect information, predatory pricing and the chain-store paradox. *Journal of Economic theory*, 25(1):92–100, 1981.

[RS08] 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, Amsterdam University Press, 2008.

[SA01] D. D. Salvucci and J. R. Anderson. Automated eye-movement protocol analysis. *Human-Computer Interaction*, 16(1):39–86, 2001.

[Sim79] H.A. Simon. Information processing models of cognition. *Annual Review of Psychology*, 30(1):363–396, 1979.

[Sta96] R. Stalnaker. On the evaluation of solution concepts. *Theory and Decision*, 37:49–73, 1996.

[SvL04] K. Stenning and M. van Lambalgen. A little logic goes a long way: basing experiment on semantic theory in the cognitive science of conditional reasoning. *Cognitive Science*, 28(4):481–529, 2004.

[vW03] W. van der Hoek and M. Wooldridge. Time, knowledge and cooperation: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75 (1):125–157, 2003.

[WBR76] D. Wood, J. S. Bruner, and G. Ross. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100, 1976.

[Wei84] J. S. Weitzenfeld. Valid reasoning by analogy. *Philosophy of Science*, 51(1):137–149, 1984.

[WLB06] R.L. West, C. Lebiere, and D.J. Bothell. Cognitive architectures, game playing, and human evolution. In R. Sun, editor, *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*, pages 103–123. Cambridge University Press, New York (NY), 2006.

[WS71] P.C. Wason and D. Shapiro. Natural and contrived experience in a reasoning problem. *The Quarterly Journal of Experimental Psychology*, 23(1):63–71, 1971.