Strategic Reasoning: Building Cognitive Models from Logical Formulas

Sujata Ghosh · Ben Meijering · Rineke Verbrugge

Abstract This paper presents an attempt to bridge the gap between logical and cognitive treatments of strategic reasoning in games. There have been extensive formal debates about the merits of the 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-tracking 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. The paper presents a formal language to represent different strategies on a finer-grained level than was possible before. The language and its semantics help to precisely distinguish different cognitive reasoning strategies, that can then be tested on the basis of computational cognitive models and experiments with human subjects. The syntactic framework of the formal system provides a generic way of constructing computational cognitive models of the participants of the Marble Drop game.

Keywords Strategic reasoning · Cognitive modeling · Logical frameworks · Experimental studies

S. Ghosh (✉)
Indian Statistical Institute, SETS Campus, Chennai 600 113, India
e-mail: sujata@isichennai.res.in

B. Meijering · R. Verbrugge
Institute of Artificial Intelligence, University of Groningen, Groningen, Netherlands
e-mail: b.meijering@rug.nl

R. Verbrugge
e-mail: L.C.Verbrugge@rug.nl

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 (Osborne and Rubinstein 1994). 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) (Polya 1945). Nowadays, cognitive scientists construct fine-grained theories about human reasoning strategies (Lovett 2005; Juvina and Taatgen 2007), 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 (Anderson 2007).

1.1 Backward Induction

Every finite extensive form game with perfect information (Osborne and Rubinstein 1994) 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 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 (Jones 1980). We only consider strictly competitive games played between two players.

Consider a finite extensive form game with perfect information G played between two players, say E and A. 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 until 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 $BI(G, i)$.

1.2 Criticism of Backward Induction

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 as 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 (Perea 2010). Extensive form rationalizability is a solution concept for extensive form games [proposed by Pearce (1984)] which incorporates such forward induction reasoning, where players' past behavior is considered to be indicative of the future choices. However, Battigalli (1997) showed that in case of generic perfect information games, which do not have any relevant ties for payoffs, the unique backward induction outcome is identical to the unique extensive form rationalizability outcome, even though the corresponding strategies of players might be quite different [cf. Reny (1992)]. Forward induction reasoning is out of scope for this paper.[1]

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 (1995) and Stalnaker (1996) 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 (Binmore 1996). For more details on these issues see (Bicchieri 1988; Arló-Costa and Bicchieri 2007; Brandenburger 2007; Baltag et al. 2009; Halpern and Pass 2009; Artemov 2009; Fitting 2011).

From the logical point of view, various characterisations of backward induction can be found in modal and temporal logic frameworks (Bonanno 2002; Harrenstein et al. 2003; van der Hoek and Wooldridge 2003; Jamroga and van der Hoek 2004; Baltag et al. 2009). 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 (Ramanujam and Simon 2008).

Critique of a different flavor stems from experimental economics (Camerer 2003; Colman 2003). 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 (1992) have shown that in a traditional centipede

---

[1] In our current research we do investigate forward induction from both experimental and theoretical points of view: we compare the reasoning mechanism underlying backward and forward induction reasoning for human agents as well as for idealised agents and intend to construct cognitive models based on these experimental and theoretical studies, grounded on the work done in this paper.

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 game-theoretic 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. (1982). 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, which means that 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. Perhaps, due to cognitive constraints such as working memory capacity, participants may be unable to perform optimal strategic reasoning, even if in principle they are willing to do so.

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

1.3 Modeling Human Strategic Reasoning

In recent years, a number of questions have been raised regarding the idealization that a formal model undergoes while representing social reasoning methods (for example, see Edmonds 2010; van Benthem 2008; Verbrugge 2009; van Lambalgen and Counihan 2008). Do these formal methods represent human reasoning satisfactorily or should we instead concentrate on the empirical studies and models based on those empirical data? We propose not to force a choice between either logic or the lab at all; rather, we combine empirical studies, formal modeling and cognitive modeling to study human strategic reasoning. Our proposal is the following: rather than thinking about logic and cognitive modeling as completely separate ways of modeling, we consider them to be complementary and investigate how they can aid one another to bring about a more meaningful model of real-life scenarios. Recently there have been other authors, for example Raijmakers et al. (2014), Gierasimczuk et al. (2013) and Szymanik et al. (2013), who also use logic to form their interpretations of human reasoning strategies in games. We believe to be the first, however, who introduce the computational cognitive architecture ACT-R as a very useful *trait-d'union* between logic and the experimental lab.

In this article, an extension of (Ghosh et al. 2010; Ghosh and Meijering 2011), we introduce a formal framework to model human strategic reasoning as exemplified by certain psychological experiments focusing on a dynamic game scenario, namely the Marble Drop game (Meijering et al. 2010). We seek to use the formal framework to give a more realistic reasoning model of the participants. Moreover, we propose to use

$$Experimentation \longrightarrow Formal\ modeling \longrightarrow Cognitive\ modeling$$
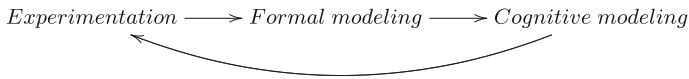
Fig. 1 A schematic diagram of the approach: The experiments discussed in Sect. 2 inform our formal model of reasoning strategies in Sect. 3. This formal model in turn helps to construct computational cognitive models of reasoning strategies in a generic way, presented in Sect. 5. Finally, as described in Sect. 4, simulations with computational cognitive models often lead to new experiments in order to test the models' predictions

cognitive models of participants' strategic reasoning by uniformly translating formulas (strategy specifications) from the formal framework to ACT-R models. As reflected in Fig. 1, this article presents an attempt to bridge the gap between 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.

This paper is meant as a first study of a cognitive model of strategic reasoning that is 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 (Meijering et al. 2010, 2012) and the eye-tracking study reported in (Meijering et al. 2012). 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 computational cognitive models.

2 Empirical Work on a Dynamic Game

We provide here a reminder of the experimental studies on which this work is based, including a description of the Marble Drop game and an analysis of an eye-tracking experiment (Meijering et al. 2010, 2012,?).

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 (see also Verbrugge (2009)).

Meijering et al. (2010, 2012) have investigated higher-order theory of mind in humans by means of experiments with dynamic games. 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. Hedden and Zhang (2002)), they are too crude to go into the details of the actual reasoning steps. To remedy this, Johnson, Camerer, Sen, and Rymon's study employed a novel measure (Johnson et al. 2002). In their sequential bargaining

experiment, participants had to bargain with one other player. The amount to bargain and the participant's role in every round was hidden behind boxes. The participants had to click on these boxes to make elements of this information visible. This allowed Johnson and colleagues to record what the participants select at a particular time within the 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 exerts a higher strain on working memory.

To avoid influencing players' reasoning strategies, Meijering and colleagues chose to employ a less invasive eye-tracking technology to investigate the details of higher-order social reasoning. They conducted an eye-tracking study to investigate the reasoning steps during higher-order social reasoning (Meijering et al. 2012). The findings of this experiment, together with behavioral results, help to determine the cognitive bounds on higher-order social reasoning in a dynamic game. We give a short overview below and refer the reader to the full papers for more details.

2.2 Dynamic Games

Meijering et al. (2010) presented participants with dynamic games to investigate higher-order social reasoning. In the games they designed, the path of a white marble can be manipulated by removing trapdoors (Fig. 3). Experience with world-physics allows players to see easily how the marble will run through a game, and which player can change the path of the white marble at each decision point. In other words, higher-order social reasoning is embedded in a context that provides an insightful overview of the decisions and the consequences of these decisions. The participants successfully applied second-order social reasoning in a great proportion (i.e., 94 %) of the games (Meijering et al. 2010, 2012, 2013a).

Earlier, Hedden and Zhang (2002) and Flobbe et al. (2008) had also presented participants with dynamic games that are game-theoretically equivalent to Marble Drop, to investigate higher-order social reasoning, but the performance in those games was far from optimal with approximately 60–70 % correct. It seemed to us that the participants could either have had difficulties applying higher-order social reasoning or difficulties understanding the games.

The Matrix game The Matrix games (Fig. 2) presented by Hedden and Zhang (2002) are 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 (Wason and Shapiro 1971); but see (Manktelow and Over 1991; Stenning and van Lambalgen 2001, 2004). More convincingly, subjects have been shown to win the game of tic-tac-toe more easily than its equivalent, Number Scrabble (Michon 1967; Simon 1979; Weitzenfeld 1984). 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' (Gigerenzer and Todd 1999). Meijering and colleagues proposed that

Fig. 2 A schematic overview of a Matrix game (Hedden and Zhang 2002). The *left-side number* in each cell is Player 1's payoff, the *right-side number* is Player 2's payoff. The goal is to attain the highest possible payoff. A participant, always assigned to the role of Player 1, first has to predict what the other player would do in cell B before making a decision what to do in cell A. In this example, Player 1 would have to predict that Player 2 will stop in cell B, because Player 1 will stop in cell C if given a choice between cells C and D, leading to a lower payoff for Player 2, namely 1 instead of 4. Consequently, the rational decision for Player 1 is to move from cell A to B

higher-order social reasoning, which seemed to be very demanding in Matrix games, might also benefit from a context embedding (Meijering et al. 2010).

The Marble Drop game In Marble Drop games (Fig. 3), the payoffs are color-graded marbles instead of the numerical values of the Matrix game. Meijering and colleagues decided for color-graded marbles to minimize the usage of numeric strategies other than first- and second-order reasoning. The color-graded marbles can be ranked according to preference, lighter marbles being less preferred than darker marbles, and this ranking makes it possible to have payoff structures isomorphic to those in Matrix games. In fact, the second-order Marble Drop game in Fig. 3c is game-theoretically equivalent to the Matrix game in Fig. 2. The sets of trapdoors in Marble Drop games correspond to the transitions, from one cell to another, in Matrix games.
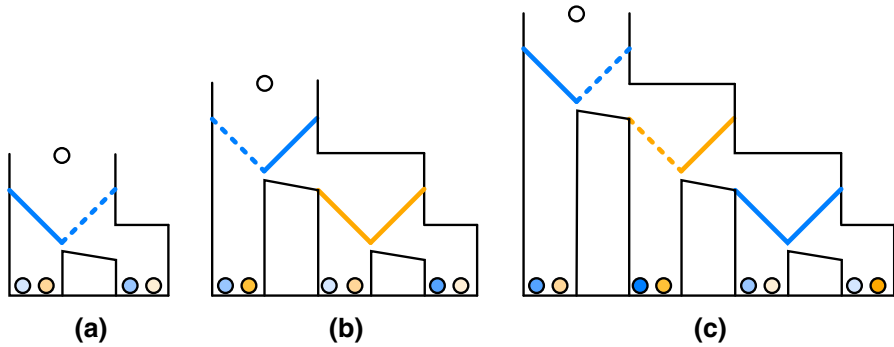
Figure 3 depicts examples of a zero-, first-, and second-order Marble Drop game. A white marble is about to drop, and its path can be manipulated by removing trapdoors (i.e., the diagonal lines). In this example, the participant controls the blue trapdoors and the computer controls the orange ones. Each bin contains a pair of payoffs. The participant's payoffs are the blue marbles and the computer's payoffs are the orange marbles. For each player, the goal is that the white marble drops into the bin that contains the darkest possible color-graded marble of their color.

In the example game in Fig. 3a, participants need to remove the right trapdoor to attain the darkest color-graded marble of their color. The game in Fig. 3a is a zero-order game, because there is no other player to reason about.

In first-order games (Fig. 3b) participants need to reason about another player, the computer. The computer is programmed to let the white 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 are told about their own goal and the other player's goal, and they are also told that the opponent is in fact a computer player. Participants need to reason about the computer, because the computer's decision at the second set of trapdoors affects at which bin a participant can end up.

In the example game in Fig. 3b, if given a choice at the second set of trapdoors, the computer will remove the left trapdoor, because its marble in the second bin is darker

Fig. 3 Examples of a zero, first-, and second-order Marble Drop game. The *blue marbles* are the partici-
pant'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 the player's 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 each player
should remove to attain the darkest possible marble of their color

than its marble in the third bin. Consequently, the participant's darkest marble in the
third bin is unattainable. The participant should therefore remove the left trapdoor of
the first set of trapdoors, because the marble of his target color in the first bin is darker
than the marble of his target color in the second bin.

In a second-order game (Fig. 3c) 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, they need to reason about what the computer, at the second set of
trapdoors, thinks that they, at the third set of trapdoors, think. Player 1 has to decide
whether to remove the left trapdoor (end) or to remove the right trapdoor (continue).
Player 1's marble in bin 2 is darker than in bin 1, but what will Player 2 decide if
Player 1 continues? Player 2 may want to continue the game to the last bin, as Player
2's marble in bin 4 is darker than in bin 2, but what will Player 1 decide at the last set
of trapdoors? Player 1 would stop the game in bin 3, as Player 1's marble in bin 3 is
darker than in bin 4. Thus, Player 2 should stop the game in bin 2, as Player 2's marble
in bin 2 is darker than in bin 3. Consequently, Player 1 should decide to continue the
game from bin 1 to bin 2.[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 (Hedden and Zhang 2002), participants had to reconstruct this from
memory. Meijering et al. (2010) hypothesized that the supporting structure of the
representation of Marble Drop would facilitate higher-order social reasoning, and, in

---

[2] Because the opponent is a perfectly rational player maximising its utility, one could wonder whether
participants are not solving an optimization problem instead of playing a game. However, in earlier experi-
ments with similar games, such as Hedden and Zhang (2002), it turned out that participants actually do not
play these games differently against a computer opponent than against another human being. Also from the
participants' answers to our questions, it was clear that participants did make second-order theory of mind
attributions, such as "he thinks that I plan to go right at the last trapdoor".

fact, participants assigned to Marble Drop games performed better than participants assigned to Matrix games (Meijering et al. 2012).

To investigate what strategies people might use in dynamic games, Meijering et al. (2012) measured participant's eye movements while they were playing Marble Drop games. We briefly explain the experiment and its results below.

2.3 Eye-Tracking Study for the Marble Drop Game

Twenty-three first-year psychology students (14 female) participated in an eye-tracking study in exchange for course credit. Their mean age was 21 years, ranging from 18 to 24 years. All participants had normal or corrected-to-normal visual acuity.

To study the online process of second-order theory of mind reasoning, it is important that participants apply such reasoning in a large proportion of the games. Fortunately, this was the case. In contrast to earlier studies (Hedden and Zhang 2002; Flobbe et al. 2008), performance was close to ceiling in Marble Drop games (Meijering et al. 2010). Participants successfully applied second-order theory of mind in 94 % of the games.

Participants' eye movements were recorded during each Marble Drop game. The eye movements and fixations on payoffs provide some insight in the strategies that participants used. For example, a general direction of eye movements going from right to left would indicate the use of backward induction: A participant would first consider the last decision point, and reason backwards to the first decision point. An opposite general direction of eye movements would indicate forward reasoning.

Figure 4 (bottom panel) depicts the mean proportions of fixations on each bin, or payoff pair, as a function of fixation position within a game. The proportions are depicted separately for games in which Player I should stop (Fig. 4a), and games in which Player I should continue (Fig. 4b).

Both plots in Fig. 4 (bottom panel) show that in most games participants fixate the payoffs in bin 1 first. At later positions in the fixation sequence, the proportion of fixations on bin 1 decreases. The proportions of fixations on bins 3 and 4 follow an opposite trend. They are small at position 1 and increase from there on, at least until position 4. These trends are most obvious in Fig. 4a (bottom panel).

The fixation patterns correspond best with forward reasoning, which would start with a comparison between payoffs in bins 1 and 2, and a general direction of fixations going from left to right. In contrast, backward induction would yield a higher proportion of first fixations on bins 3 and 4, as backward reasoning starts with comparing payoffs in bins 3 and 4. However, as of position 4 in Fig. 4a, the fixation patterns seem to align with backward induction: the proportion of fixations on bins 3 and 4 is higher than the proportion of fixations on bins 1 and 2. Furthermore, the way that the proportions change from thereon (i.e., they decrease for bins 3 and 4, and increase for bins 1 and 2) corresponds with eye movements going from right to left.

The patterns are less obvious in Fig. 4b, which shows eye movements in another set of games connected to a different type of pay-off structure. Differential fixation sequences imply that participants did not use pure backward induction. More specifically, the first comparisons and thus fixations for backward induction would be the
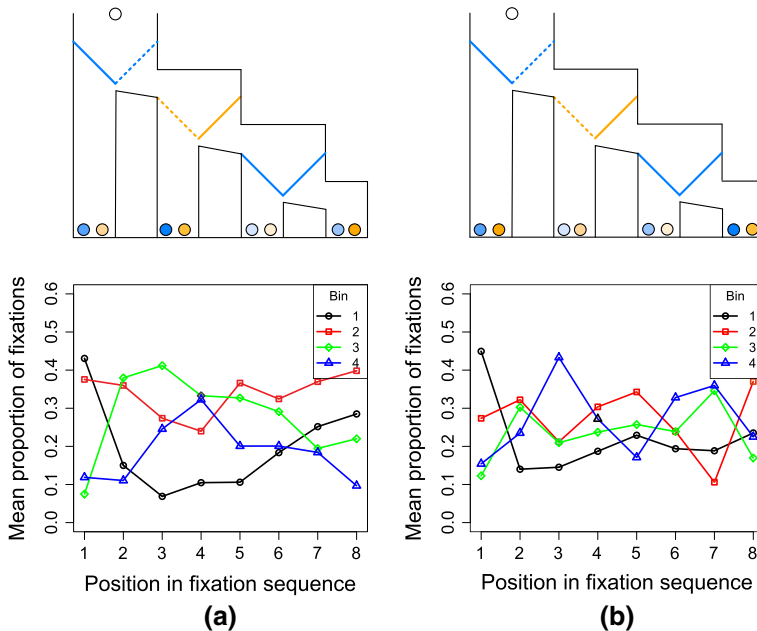
Fig. 4 The *bottom panel* depicts mean proportions of fixations at bins 1, 2, 3, and 4, calculated separately for position in the total fixation sequence. In **a**, Player I should end the game in bin 1, because given the chance, Player II would continue, and the game would end with a lesser payoff for 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 SEs, because we fitted (non-)linear models instead of traditional ANOVAs, which typically include contrasts between (successive) positions of fixations

same in both types of games when using backward induction: first, compare Player I's payoffs in bins 3 and 4, and second, compare Player II's payoffs in bins 2 and 4. Clearly, this is not the case.

In sum, participants' eye movements correspond best with forward reasoning, or forward reasoning mixed with backtracking. Fig. 4a 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. More precisely, in *forward reasoning plus backtracking*, a player could notice, when reasoning from the first decision point onwards towards the last, that he or she had unknowingly skipped the highest attainable outcome at a previous decision point. Thus, the player would need to jump back to ascertain whether that outcome is indeed attainable.[3] Meijering et al. (2012) provide a statistical analysis showing that the strategies actually used by players are significantly closer to forward reasoning with backtracking than to backward induction.

In the current article, we propose a different method to methodically investigate players' strategies. To test what strategies participants may have used, we construct computational cognitive models (cf. Sect. 4) that implement various strategies, and

---

[3] Note that 'forward reasoning' and 'forward reasoning plus backtracking' are very different from the game-theoretical notion of 'forward induction' described in Sect. 1.2.

use these models to predict eye movements that we can test against the observed eye movements. To aid in the construction, we propose a formal framework (cf. Sect. 3) and show how the formal and cognitive modeling can interplay to provide a better model for strategic reasoning (cf. Sect. 5). As mentioned in the introduction, we are presently at the phase of constructing these cognitive models, and predicting and testing strategies are our next steps.

3 A Formal Framework

Inspired by the work of Paul, Ramanujam and Simon on representation of strategies (Ramanujam and Simon 2008; Paul et al. 2009) we now propose a logical language specifying strategies of players. This provides an elegant way to describe the empirical reasoning of the participants of the Marble Drop game (cf. Sect. 2.2), that has been found by the eye-tracking study in (Meijering et al. 2012).

As mentioned in the introduction, our formal framework provides a bridge between empirical studies and cognitive models of human strategic reasoning as performed during the Marble Drop experiment. The formulas aid in systematically constructing reasoning rules in the cognitive model. Hitherto, logic has most often been used to describe idealised agents and more recently, formal models of resource-bounded agents have also been developed (Agotnes and Alechina 2006, 2007). In contrast, here we explore the use of logical language as a pathway from empirical to cognitive modelling studies of the human reasoning process.

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 but perfectly rational 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.

Note that we do not explicitly represent notions like rationality and common knowledge of rationality into our logical model, and even refrain from adding belief and knowledge operators to the language. The simpler language suffices to investigate the uniform description of strategies needed for building a bridge from empirical results to cognitive models. In addition to (Ramanujam and Simon 2008; Paul et al. 2009; Ghosh and Ramanujam 2012; Ghosh 2008), the current literature on strategic reasoning abounds with frameworks that do not include belief and / or knowledge operators, see for example (Hoek et al. 2005; Walther et al. 2007; Chatterjee et al. 2007; Pinchinat 2007; Pauly 2002).

Below, we present a formal system, inspired by (Bonanno 2002), to represent the different ways of strategic reasoning that the participants of the Marble Drop game undertake, suggested by the eye-tracking study described in Sect. 2.3.

3.1 Preliminaries

In this subsection, representations for extensive form games, game trees and strategies are presented, similar to those in Ramanujam and Simon (2008), Paul et al. (2009), and Ghosh and Ramanujam (2012). On the basis of these concepts, reasoning strategies can be formalized in Sect. 3.2.

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, and we take $N = \{1, 2\}$. We often use the notation i and $\bar{\imath}$ to denote the players, where $\bar{1} = 2$ and $\bar{2} = 1$. 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 $\rho = x_1 x_2 \ldots x_m \in X^*$, let $last(\rho) = x_m$ denote the last element in this sequence.

Game Trees Let $\mathbb{T} = (S, \Rightarrow, s_0)$ be a tree rooted at s_0 on the set of vertices S and let $\Rightarrow : (S \times \Sigma) \to S$ 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 $\vec{s} = \{s' \in S \mid s \overset{a}{\Rightarrow} s'$ for some $a \in \Sigma\}$. A node s is called a leaf node (or terminal node) if $\vec{s} = \emptyset$.

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 $j > 0$, $s_{j-1} \overset{a_{j-1}}{\Rightarrow} s_j$. Let $Plays(T)$ denote the set of all plays in the game tree T.

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 player $\bar{\imath}$. 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 $s \overset{a}{\Rightarrow}_\mu s'$, where $\mu(s) = a$ and $s \overset{a}{\Rightarrow} s'$.
    - if $\widehat{\lambda}(s) \neq i$ then for all s' such that $s \overset{a}{\Rightarrow} s'$, we have $s \overset{a}{\Rightarrow}_\mu s'$.
- $\widehat{\lambda}_\mu = \widehat{\lambda} \restriction_{S_\mu}$.

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 $j \geq 0$ we have that $s_j \in S^i$ implies $\mu(s_j) = a_j$. A strategy profile (μ, τ) consists of a pair of strategies, one for each player.

Partial strategies A partial strategy for player i is a partial function σ^i which specifies a move at some (but not necessarily 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.

A partial strategy can be viewed as a set of total strategies. Given a partial strategy tree $T_\sigma = (S_\sigma, \Rightarrow_\sigma, s_0, \widehat{\lambda}_\sigma)$ 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 $s \in S$ then for all $s' \in \vec{s}$, $s' \in S$ implies $s' \in S_\sigma$
- if $\widehat{\lambda}(s) = i$ then there exists a unique $s' \in S$ and action a such that $s \overset{a}{\Rightarrow} s'$.

Note that $\widehat{T_\sigma}$ is the set of all total strategy trees for player i that are subtrees of the partial strategy tree T_σ for i. Any total strategy can also be viewed as a partial strategy, where the corresponding set of total strategies becomes a singleton set.

3.2 Strategy Specifications

Following to the lines of work in (Ramanujam and Simon 2008; Paul et al. 2009), we present 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 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 Backus-Naur form:

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

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

In the following, X is usually a set of propositional variables. Other Boolean operators can be interpreted in the usual way. 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 $\langle a^+ \rangle$ and $\langle a^- \rangle$ 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 $(r \leq q)$ 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 $\psi \in BPF(P^i)$. 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 properties of the strategy are 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, and where $\mathcal{U} : frontier(T) \times N \to \mathbb{Q}$ is a utility function. As mentioned earlier, $frontier(T)$ 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 propositional variables is as follows:

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

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

– for all $i \in N$, **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

– for all $s \in S$, **root** $\in V(s)$ iff $s = s_0$.

As described in Sect. 3.1, a partial strategy σ, say of player i, can be viewed as a set of total strategies of the player (Paul et al. 2009) and each such strategy is a subtree of T.

The semantics of the strategy specifications are given as follows. Given the game $T = (S, \Rightarrow, s_0, \widehat{\lambda}, \mathcal{U})$, 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.

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, then player i responds following some strategy σ, say. We also want to model situations where 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 $\bar{\imath}?\zeta$ 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:
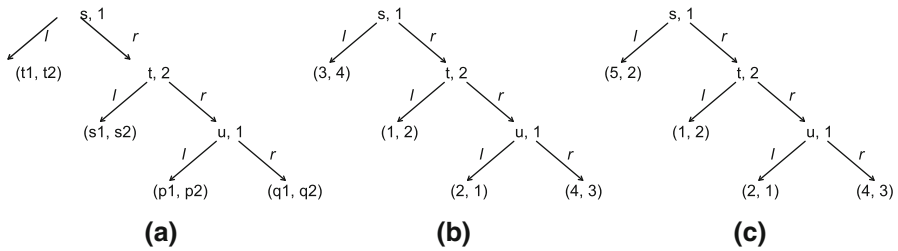
Fig. 5 Example trees for Marble Drop

– $M, s \models \bar{\imath}?\zeta$ iff $\exists T'$ such that $T' \in \llbracket \zeta \rrbracket_T$ 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 (2008) has a simpler version of $BPF(P^i)$, where only past formulas are considered, but they introduce an additional construct in the syntax of strategy specifications, namely $\pi \Rightarrow \sigma$, 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 $\bar{\imath}$ conform 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 Ramanujam and Simon (2008) for future work. For this paper, we concentrate on the translation of these formulas to relevant rules that will aid in the development of a computational cognitive model for strategic reasoning in Marble Drop games.

3.3 Marble Drop Game: A Test Case for Strategy Specification

Unfamiliar with the task setting of Marble Drop games, participants might have used simple rules that correspond, for example, with causal reasoning: "If I open the left-side trapdoor I get a payoff of 2", and "If I open the right-side trapdoor and Player 2 subsequently opens the left-side trapdoor, I get a payoff of 3", and so on. This procedure resembles forward reasoning, as payoffs are compared in a forward succession.

The procedure followed in backward reasoning is less intuitive, and less common in everyday task settings. Still, participants could have discovered backward reasoning after playing many Marble Drop games. Meijering et al. (2012) found some evidence in favor of backward reasoning, but forward reasoning with backtracking was the preferred strategy among participants.

We now express the empirical strategic reasoning performed by the participants of the Marble drop game described in Sect. 2.2. The game form is structurally equivalent to the Centipede game tree. Figure 5a gives the corresponding tree structure, and Fig. 5b, c represent example cases.

Using the strategy specification language introduced in Sect. 3.2, we express the different reasoning methods of participants that have been validated by the experiments described in Sect. 2.3. 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 strategy specification for player 1 describing her backward reasoning giving the correct answer corresponding to the game tree given in Fig. 5b is:

$\eta_1 : [(2?[(1?[\varphi_1^0 \mapsto r]^1 \wedge \varphi_1^1) \mapsto r]^2 \wedge \varphi_1^2) \mapsto r]^1$, where:

– $\varphi_1^0 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (2 \leq 4) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (2 \leq 3) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (3 \leq 4)$

– $\varphi_1^1 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (2 \leq 4) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (2 \leq 3)$

– $\varphi_1^2 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (2 \leq 4)$

In words, η_1 says:

'If the utilities and the turns of players at the respective nodes are as in Fig. 5b, then if player 1 would play r at the root node, and player 2 would continue playing r at his node, player 1 will finish off by playing r.'

Another strategy specification for player 1 describing forward reasoning giving a wrong answer corresponding to the game tree given in Fig. 5b is:

$\eta_2 : [\varphi_2 \mapsto l]^1$, where:

– $\varphi_2 : \mathbf{turn}_1 \wedge \delta \wedge \langle r \rangle \mathbf{turn}_2 \wedge \gamma \wedge (1 \leq 3)$.

In words, η_2 says:

'If the utilities at the first two leaf-nodes of the game are as Fig. 5b, 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 strategy specification for player 1 describes forward reasoning, giving a correct answer corresponding to the game tree given in Fig. 5c:

$\eta_3: [\varphi_3 \mapsto l]^1$, where:

– $\varphi_3 : \mathbf{turn}_1 \wedge \delta \wedge \langle r \rangle \mathbf{turn}_2 \wedge \gamma \wedge (1 \leq 5)$

In words, η_3 says:

'If the utilities at the first two leaf-nodes of the game are as Fig. 5c, 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 indeed model several possible ways of reasoning that can be performed by human reasoners in the Marble Drop game. A list of possible reasoning strategies aids in developing the cognitive models of the reasoners, as we shall see in Sect. 5.[4]

4 Cognitive Modeling

On the basis of our empirical data about participants' eye momevents when playing the dynamic Marble Drop game, we would like to draw conclusions about which ways they reason. For example, do participants actually use one of the strategies formally presented in Sect. 3.3? Unfortunately, it is very complex to directly compare eye movement data to formally presented reasoning strategies.

Analyses of eye movements are challenging because one has to deal with great variability typically found in eye movement data. The fact that fixations are not always systematic contributes to this variability. Salvucci and Anderson (2001) used cognitive computational models to predict eye movements, which they compared with observed eye movements. Their method helped to disentangle systematic and strategic eye movements from unsystematic, random eye movements.

Taking off from their methods, we will present our ideas about a generic cognitive model that implements backward and forward reasoning as well as possible mixtures of the two in Sect. 5. It is our aim that for each particular strategy, its corresponding model can subsequently simulate experiments, so that it can finally be compared to behavioral and eye-tracking data. Before going into the specific details of our construction of computational cognitive models, we first provide a general description of the cognitive architecture in which we will develop our models, namely ACT-R.

4.1 ACT-R Modeling

ACT-R is an integrated theory of cognition as well as a cognitive architecture that many cognitive scientists use to model human cognition (Anderson 2007). ACT-R consists of modules that link with cognitive functions, for example, vision, motor processing, and declarative processing. Each module maps on to a specific brain region. Furthermore, 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 it models cognitive resources as being bounded. This is reflected in the fact that 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 the pieces of information 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 (Borst et al. 2010).

---

[4] A complete list for backward reasoning specifications for players 1 and 2 is given in the Appendix.

The declarative memory module represents long-term memory and stores information encoded in so-called *chunks*, representing knowledge structures. For example, a chunk can be represented as some formal expression with a defined meaning. 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 (time delay) of retrieval decreases. For example, a chunk that represents a comparison between two payoffs will have a higher probability of retrieval, and will be retrieved faster, if the comparison has been made *recently*, as opposed to an older comparison in some previous game, or if the comparison has been made *frequently* in the past (Anderson and Schooler 1991).

Anderson (2007) 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 placed into the declarative module's 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 of which the cognitive architecture is aware.

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 200 ms).

The cognitive model that we present in Sect. 5 relies on the declarative module and the problem state module. 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 matches if the last two payoff values have been stored in the problem state, and the first is greater than the second. In that case it requests the manual (or motor) module to respond 'stop':

IF

> *Goal* is to to compare the last two payoff values,
> *Problem State* stores the associated payoff values
> First payoff value is greater than second payoff value

THEN

> Decide to stop the game.

Similar to the activation levels of chunks in declarative memory, production rules have so-called utility values that represent their usefulness. If a set of production rules yields

a correct response, the model receives a reward. Similarly, if the response turns out to be incorrect, the model is punished. Both reward and punishment propagate back to previously fired production rules, and the utility values of these production rules are adjusted accordingly. Utility increases in case of reward, and decreases in case of punishment. This process is called *utility learning* (Anderson et al. 2004; Anderson 2007). If two or more production rules match a particular game state, the production rule with the highest utility will be selected.

4.2 Related Work: A Computational Cognitive Model of Marble Drop

van Maanen and Verbrugge (2010) propose an ACT-R model that follows a backward reasoning strategy to predict the opponent's moves further on in the game, inspired by Hedden and Zhang's (2002) decision tree analysis of this process for their matrix version of the game. The ACT-R model has knowledge on how to solve Marble Drop games for all possible distributions of payoffs over the bins. That is, the model stores chunks containing information on which payoffs to compare at each step.

The model provides a nice fit to responses and associated reaction times in Marble Drop games (van Maanen and Verbrugge 2010). However, it implements just one strategy, backward reasoning, and the results in Meijering et al. (2012) show that participants did not use pure backward reasoning. Instead, most participants seemed to prefer forward reasoning plus backtracking. Therefore, we need a more generic model that is able to fit a broader spectrum of possible strategies. This is what we set out to construct in the following section.[5]

5 Modeling Marble Drop in ACT-R by Generic Strategies

The computational cognitive model that we propose here has been inspired by Van Maanen and Verbrugge's model (van Maanen and Verbrugge 2010). However, our new model is much more generic because it is not based on a fixed strategy. Instead, we consider a class of models, where each model is based on a set of strategy specifications that are selected from a list provided by the logical framework presented in Sect. 3.3 (see also Appendix). The specifications can either represent backward reasoning, forward reasoning, or a mix of both; see the discussion of example specifications η_1, η_2, η_3 in Sect. 3.3.

To investigate which strategies or rules participants applied in the Marble Drop games, we propose a computational model that is based on the strategy specifications provided in Sect. 3.3. We acknowledge that some specifications may at first sight seem more plausible than others, but a first step in modelling Marble Drop games is to expose strategy preference. Our main goal for future simulation work is to implement various sets of specifications in separate models, and compare the simulated eye movements,

---

[5] Recently, an alternative ACT-R model of Marble Drop has been proposed (Meijering et al. 2013b), building on the idea that players use negative feedback in order to move from an overly simple reasoning strategy that does not sufficiently take into account the opponent's perspective, to a more complicated strategy only if it is really needed.

responses, and response times of each model with the human data. It will also be interesting to simulate repeated game play in order to model how backward reasoning could originate from simpler strategies such as, for example, forward reasoning.

5.1 From Strategy Specifications to ACT-R Production Rules

Each of the specifications defined in Sect. 3.3 comprises at least one comparison between two payoffs, and for each comparison a model has a set of production rules that specify what the model should do. Consider a simple specification that describes forward reasoning:

$$\eta : [(\textbf{root} \wedge \textbf{turn}_1 \wedge \delta \wedge \langle r \rangle \textbf{turn}_2 \wedge \gamma \wedge (t1 > s1)) \mapsto l]^1$$

In this specification, the first (and only) comparison is between Player 1's payoffs t1 and s1. It specifies that the model should go left and stop the game if t1 is greater than s1. So, how does this specification translate to production rules?

At the start of a game, all module buffers are empty because the model has not attended any payoffs yet, nor has it made any comparisons. The model selects a production rule that matches this condition in which all buffers are empty, and specifies what to do next. For example,

IF
   Goal buffer is empty   If the model does not yet have a goal,
   Problem State buffer is empty and does not yet have a representation of
                                   game state,
   Visual buffer is empty   and has not yet attended any payoffs,
THEN
   Goal is to compare Player 1's   then set the goal to compare Player 1's
   payoffs at locations δ and γ payoffs at δ and γ.

After firing this production rule, the model will set a goal. More specifically, the model will store a chunk in the goal buffer, representing that the goal is to compare Player 1's payoffs at locations δ and γ. This step is depicted in the first, uppermost, box of the flowchart in Fig. 6. The flowchart represents the general process that the model follows while applying specification η (and also while applying formula $\eta*$ discussed on page 24).

To compare Player 1's payoffs at δ and γ, the model first has to find, attend, and encode them in the problem state buffer. Figure 6 shows that for each subsequent payoff, the model performs the following procedure:

– request the visual module to find the payoffs' visual locations [cf. Nyamsuren and Taatgen (2013)];
– direct visual attention to that location; and
– update the problem state (buffer).

The specification η specifies what the model should do after encoding both payoffs in the problem state: decide to go left if the payoff at location1 (i.e., t1) is greater than the payoff at location2 (i.e., s1). The corresponding production rule would be:
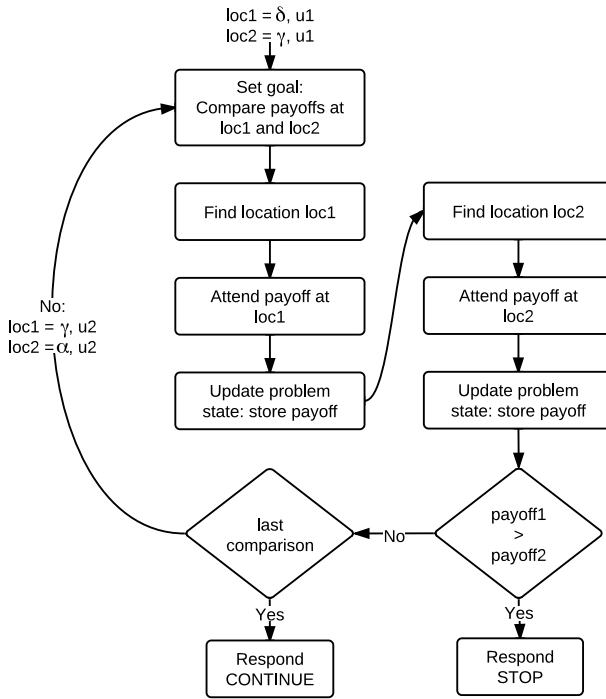
loc1 = δ, u1
loc2 = γ, u1

Set goal:
Compare payoffs at
loc1 and loc2

Find location loc1   Find location loc2

No:
loc1 = γ, u2
loc2 = α, u2

Attend payoff at   Attend payoff at
loc1   loc2

Update problem   Update problem
state: store payoff   state: store payoff

last   ←No—   payoff1
comparison   >
                              payoff2

Yes   Yes

Respond   Respond
CONTINUE   STOP

Fig. 6 Flow-chart of a forward reasoning process

IF
   Goal is to compare Player 1's   If the current goal is to compare Player 1's

   payoffs at δ and γ payoffs at δ and γ,

   Problem State represents   and the problem state has stored those payoffs,

   Player 1's payoffs at δ and γ

   are t1 and s1

   t1 > s1   and payoff1 > payoff2
THEN
   Decision is go-left   then request the manual (or motor) module
                                to produce a key press (i.e., go left).

The model will select and fire this production rule to generate a response. As mentioned in Sect. 4.1, the model has a motor module that produces actual key presses. These key presses and associated reaction times can be compared to the human data, as a means to evaluate the model.

There is another specification, $\eta*$, that fits a scenario in which Player 1's payoff at δ happens to be smaller than Player 1's payoff at γ. According to this specification, which also describes forward reasoning, the next comparison should be between Player 2's payoffs at γ and α:

$$\eta* : [(\mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge \langle r \rangle \mathbf{turn}_2 \wedge \gamma \wedge (t1 < s1) \wedge \alpha \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (s2 > p2)) \mapsto r]^1$$

The corresponding production rule changes the goal (chunk) to instruct the model to continue with a comparison between Player 2's payoffs at γ and α:

IF

> *Goal* is to compare Player 1's payoffs at δ and γ
> *Problem State* represents that Player 1's payoffs at δ and γ are t1 and s1
> t1 < s1

THEN

Goal is to compare Player 2's payoffs at γ and α

As can be seen in Fig. 6, the model will proceed with this particular comparison and all the visual processing involved. As soon as the model has attended Player 2's payoffs at γ and α, it can produce a response. According to specification $\eta*$, the model should decide to go-right if the payoff at γ is greater than the payoff at α:

IF

> *Goal* is to compare Player 2's payoffs at γ and α
> *Problem State* represents that Player 2's payoffs at γ and α are s2 and p2
> $s2 > p2$

THEN

> *Decision* is go-right

Similarly to the actual experiment with human participants, feedback will indicate whether the model's response was correct, in the sense of providing the maximal possible pay-off. The model will attend the feedback and receive a reward in case of a correct response, and punishment in case of an incorrect response. Accordingly, the model will learn from experience which production rules are useful.

This process of utility learning allows us to explore how backward reasoning might evolve over the course of playing many Marble Drop games. Timing plays an important role in utility learning, as the reward or utility received by a particular production rule is the external reward (given to the model) minus the time between firing that production rule and the model receiving the reward. Thus, the shorter the time between firing a production rule and the model receiving the reward, the greater the reward that production rule receives. If, then, backward reasoning is more efficient than, for example, forward reasoning, the set of production rules that comprise backward reasoning should be rewarded more than the set of rules comprising forward reasoning.

A model based on both forward and backward reasoning specifications could simulate competition between these two strategies. At first, the model might apply forward reasoning more frequently than backward reasoning, for example, if the productions representing forward reasoning have higher utility. However, forward reasoning does not always yield the correct answer. In case of an incorrect answer, the productions rules will receive a penalty, lowering their utility value. After a while, the model will select production rules that represent backward reasoning, and these will be rewarded

each time because backward reasoning always yields the game-theoretically optimal outcome.

6 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 (Marr 1982):

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

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. As we argue in this article, logic has a fruitful role to play in theories of resource-bounded strategic reasoning at the algorithmic level, namely in the construction of computational cognitive models in ACT-R. We presented some steps in this direction in the previous section.

To summarize, the strategy algorithms are represented as specifications, which form the basis of an ACT-R model. The specifications are implemented as production rules, which handle visual processing, problem state updates, and motor processing. ACT-R's utility learning mechanism allows us to explore the evolution of strategies over the course of many Marble Drop games. For example, if backward reasoning is more efficient than forward reasoning, the corresponding production rules will be used increasingly more often. Finally, to further our understanding of strategy preference in games such as Marble Drop, we can compare the human data, including responses, response times, and fixations, against the 'behavior' of various models.

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. These theories have been built over the decades on the basis of hundreds of tasks modeled in ACT-R and compared to experimental data: from learning high school algebra (Anderson 2005) and playing the game of SET (Nyamsuren and Taatgen 2013) to driving cars (Gunzelmann et al. 2011). Thus, there is no need to add possibly arbitrary resource bounds in the logical language. The combined strengths of logic, coupled with cognitive modeling in ACT-R and experiments, will hopefully lead to an improved understanding of human resource-bounded reasoning in games.
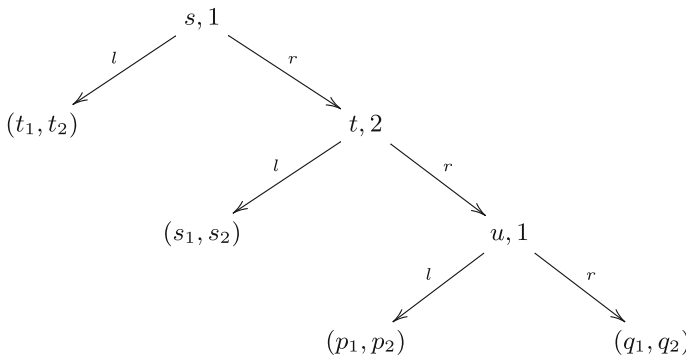
From the logical perspective, providing a sound and complete system for strategic reasoning that models empirical human reasoning will be an essential next step. Furthermore, we would need to take players' preferences into consideration as well as

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.

Appendix: List of Backward Reasoning Specifications

We consider a general extensive form game equivalent to the Marble Drop game and list below all possible specifications describing backward reasoning in such a game. For simplification, we consider generic forms of such games, that is, $p_1 \neq q_1 \wedge p_1 \neq s_1 \wedge p_1 \neq t_1 \wedge q_1 \neq s_1 \wedge q_1 \neq t_1 \wedge s_1 \neq t_1$ and $p_2 \neq q_2 \wedge p_2 \neq s_2 \wedge p_2 \neq t_2 \wedge q_2 \neq s_2 \wedge q_2 \neq t_2 \wedge s_2 \neq t_2$. As mentioned in Sect. 3.3, we use the following abbreviations of formulas:

$$\langle r \rangle \langle r \rangle \langle l \rangle ((u_1 = p_1) \wedge (u_2 = p_2)) = \alpha$$
$$\langle r \rangle \langle r \rangle \langle r \rangle ((u_1 = q_1) \wedge (u_2 = q_2)) = \beta$$
$$\langle r \rangle \langle l \rangle ((u_1 = s_1) \wedge (u_2 = s_2)) = \gamma$$
$$\langle l \rangle ((u_1 = t_1) \wedge (u_2 = t_2)) = \delta$$

All possible cases for backward reasoning that are used to formulate the production rules in Sect. 5 are given by:

1. $[\varphi_1 \mapsto l]^1$, where $\varphi_1 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (p_1 \leq q_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (s_2 \leq q_2) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (q_1 \leq t_1)$.
2. $[(2?[(1?[\varphi_2^0 \mapsto r]^1 \wedge \varphi_2^1) \mapsto r]^2 \wedge \varphi_2^2) \mapsto r]^1$, where:

- $\varphi_2^0 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (p_1 \le q_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (s_2 \le q_2) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (t_1 \le q_1)$;
- $\varphi_2^1 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (p_1 \le q_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (s_2 \le q_2)$;
- $\varphi_2^2 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (p_1 \le q_1)$.

3. $[\varphi_3 \mapsto l]^1$, where $\varphi_3 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (p_1 \le q_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (q_2 \le s_2) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (s_1 \le t_1)$.

4. $[(1?[\varphi_4^0 \mapsto r]^1 \wedge \varphi_4^1) \mapsto l]^2$, where:
   - $\varphi_4^0 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (p_1 \le q_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (q_2 \le s_2) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (t_1 \le s_1)$;
   - $\varphi_4^1 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (p_1 \le q_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (q_2 \le s_2)$.

5. $[(2?[(1?[\varphi_5^0 \mapsto r]^1 \wedge \varphi_5^1) \mapsto r]^2 \wedge \varphi_5^2) \mapsto l]^1$, where:
   - $\varphi_5^0 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (q_1 \le p_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (s_2 \le p_2) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (t_1 \le p_1)$;
   - $\varphi_5^1 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (q_1 \le p_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (s_2 \le p_2)$;
   - $\varphi_5^2 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (q_1 \le p_1)$.

6. $[\varphi_6 \mapsto l]^1$, where $\varphi_6 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (q_1 \le p_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (s_2 \le p_2) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (p_1 \le t_1)$.

7. $[(1?[\varphi_7^0 \mapsto r]^1 \wedge \varphi_7^1) \mapsto l]^2$, where:
   - $\varphi_7^0 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (q_1 \le p_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (p_2 \le s_2) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (t_1 \le s_1)$;
   - $\varphi_7^1 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (q_1 \le p_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (p_2 \le s_2)$.

8. $[\varphi_8 \mapsto l]^1$, where $\varphi_8 : \alpha \wedge \beta \wedge \langle r \rangle \langle r \rangle \mathbf{turn}_1 \wedge (q_1 \le p_1) \wedge \gamma \wedge \langle r \rangle \mathbf{turn}_2 \wedge (p_2 \le s_2) \wedge \mathbf{root} \wedge \mathbf{turn}_1 \wedge \delta \wedge (s_1 \le t_1)$.

References

Agotnes, T., & Alechina, N. (2006). (Eds.), *Proceedings of the workshop on resource-bounded agents*. European Summer School on Logic, Language and Information, Malaga.

Agotnes, T., & Alechina, N. (2007). The dynamics of syntactic knowledge. *Journal of Logic and Computation*, *17*(1), 83–116.

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

Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science*, *29*, 313–341.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*(6), 396–408.

Arló-Costa, H., & Bicchieri, C. (2007). Knowing and supposing in games of perfect information. *Studia logica*, *86*, 353–373.

Artemov, S. (2009). Knowledge-based rational decisions. *Technical Report TR-2009006*, The CUNY Graduate Center, New York, NY.

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

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

Battigalli, P. (1997). On rationalizabiliry in extensive games. *Journal of Economic Theory*, *74*, 40–61.

Bicchieri, C. (1988). Common knowledge and backward induction: A solution to the paradox. In M. Vardi (Ed.), *TARK '88: Proceedings of the 2nd conference on theoretical aspects of reasoning about knowledge* (pp. 381–393), San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Binmore, K. (1996). A note on backward induction. *Games and Economic Behavior*, *17*(1), 135–137.

Bonanno, G. (2002). Modal logic and game theory: Two alternative approaches. *Risk, Decision and Policy*, *7*(03), 309–324.

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

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

Camerer, C. F. (2003). *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton: Princeton University Press.

Chatterjee, K., Henzinger, T., & Piterman, N. (2007). Strategy logic. In *Proceedings of the 18th international conference on concurrency theory (CONCUR 07), LNCS 4703*, (pp. 59–73). Springer.

Colman, A. W. (2003). Cooperation, psychological game theory, and limitations of rationality in social interaction. *Behavioral and Brain Sciences*, *26*, 139–198.

Edmonds, B. (2010). Social intelligence and multi-agent systems. In *Invited talk, MALLOW'10*.

Fitting, M. (2011). Reasoning about games. *Studia Logica*, *99*(1–3), 143–169.

Flobbe, L., Verbrugge, R., Hendriks, P., & Krämer, I. (2008). Children's application of theory of mind in reasoning and language. *Journal of Logic, Language and Information*. In M. Counihan (Ed.), *Special issue on formal models for real people*, (Vol. 17, pp.417–442).

Ghosh, S. (2008). Strategies made explicit in dynamic game logic. In J. v. Benthem & E. Pacuit (Eds.), *Proceedings of the workshop on logic and intelligent interaction, ESSLLI*, (pp. 74–81).

Ghosh, S., & Meijering, B. (2011). On combining cognitive and formal modeling: A case study involving strategic reasoning. In J. van Eijck & R. Verbrugge (Eds.), *Proceedings of the workshop on reasoning about other minds (RAOM 2011), CEUR workshop proceedings* (Vol. 751, pp. 79–92).

Ghosh, S., Meijering, B., & Verbrugge, R. (2010). 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*, (Vol. 627, pp. 15–34).

Ghosh, S., & Ramanujam, R. (2012). Strategies in games: A logic-automata study. In N. Bezhanishvili & V. Goranko (Eds.), *Lectures on Logic and Computation: ESSLLI 2010 Copenhagen* (pp. 110–159)., Denmark, August 2010, ESSLLI 2011, Ljubljana, Slovenia, August 2011, Selected Lecture Notes, volume 7388 of Lecture Notes in Computer Science Berlin: Springer.

Gierasimczuk, N., Maas, H. v. d., & Raijmakers, M. (2013). An analytic tableau model for deductive mastermind empirically tested with a massively used online learning system. *Journal of Logic, Language and Information*. In J. Szymanik & R. Verbrugge (Eds.), *Special issue on logic and cognition*, (Vol. 22, No. 3, pp. 297–314).

Gigerenzer, G., & Todd, P. (1999). *Simple heuristics that make us smart*. New York: Oxford University Press.

Gunzelmann, G., Moore, L. R., Salvucci, D. D., & Gluck, K. A. (2011). Sleep loss and driver performance: Quantitative predictions with zero free parameters. *Cognitive Systems Research*, *12*(2), 154–163.

Halpern, J. Y., & Pass, R. (2009). Iterated regret minimization: A new solution concept. In C. Boutilier (Ed.), *Proceedings of IJCAI-09: 21st international joint conferences on artificial intelligence* (pp. 153–158). USA: Pasadena, CA.

Harrenstein, B., van der Hoek, W., Meyer, J.-J. C., & Witteveen, C. (2003). A modal characterization of Nash equilibrium. *Fundamenta Informaticae*, *57*(2), 281–321.

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

Hoek, W. v. d., Jamroga, W., & Wooldridge, M. (2005). A logic for strategic reasoning. In *Proceedings of the fourth international joint conference on autonomous agents and multi-agent systems (AAMAS 05)*, (pp. 157–164). ACM Inc, New York.

Jamroga, W., & van der Hoek, W. (2004). Agents that know how to play. *Fundamenta Informaticae*, *63*, 185–219.

Johnson, E. J., Camerer, C. F., Sen, S., & Rymon, T. (2002). Detecting failures of backward induction: Monitoring information search in sequential bargaining. *Journal of Economic Theory*, *104*(1), 16–47.

Jones, A. (1980). *Game theory: Mathematical models of conflict*. Chichester: Wiley.

Juvina, I., & Taatgen, N. A. (2007). Modeling control strategies in the N-back task. In *Proceedings of the 8th international conference on cognitive modeling*, (pp. 73–78). New York, NY, Psychology Press.

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

Lovett, M. C. (2005). A strategy-based interpretation of Stroop. *Cognitive Science*, *29*(3), 493–524.

Manktelow, K., & Over, D. (1991). Social roles and utilities in reasoning with deontic conditionals. *Cognition*, *39*(2), 85–105.

Marr, D. (1982). *Vision*. New York: Freeman and Company.

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

Meijering, B., Maanen, L. v., Rijn, H. v., & Verbrugge, R. (2010). The facilitative effect of context on second-order social reasoning. In *Proceedings of the 32nd annual meeting of the cognitive science society*, (pp. 1423–1428). Philadelphia, PA, Cognitive Science Society.

Meijering, B., Rijn, H. v., Taatgen, N., & Verbrugge, R. (2012). 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*, (pp. 2486–2491), Austin, TX. Cognitive Science Society.

Meijering, B., Rijn, H. v., Taatgen, N. A., & Verbrugge, R. (2013a). Reasoning about diamonds, gravity and mental states: The cognitive costs of theory of mind. In *Proceedings of the 35th annual conference of the cognitive science society*, Berlin.

Meijering, B., Taatgen, N. A., Rijn, H. v. & Verbrugge, R. (2013b). Reasoning about mental states in sequential games: As simple as possible, as complex as necessary. In *Proceedings of the 12th international conference on cognitive modeling*, Ottawa.

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

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

Nyamsuren, E., & Taatgen, N. (2013). Set as instance of a real-world visual-cognitive task. *Cognitive Science*, *37*(1), 146–175.

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

Paul, S., Ramanujam, R., & Simon, S. (2009). Stability under strategy switching. In *Proceedings of the 5th conference on computability in Europe (CiE 2009), volume 5635 of LNCS*, (pp. 389–398). Berlin: Springer.

Pauly, M. (2002). A modal logic for coalitional power in games. *Journal of Logic and Computation*, *12*, 149–166.

Pearce, D. (1984). Rationalizable strategic behaviour and the problem of perfection. *Econometrica*, *52*, 1029–1050.

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

Pinchinat, S. (2007). A generic constructive solution for concurrent games with expressive constraints on strategies. In *Proceedings of the 5th international symposium on automated technology for verification and analysis (ATVA'07), LNCS* (Vol. 4762, pp. 253–267). Springer.

Polya, G. (1945). *How to solve It. A new aspect of mathematical method*. Princeton, NJ: Princeton University Press.

Raijmakers, M., Mandell, D., Es, S. v., & Counihan, M. (2014). Children's strategy use when playing strategic games. *Synthese*. In R. Verbrugge (Ed.), *Special issue on reasoning about other minds*.

Ramanujam, R., & Simon, S. (2008). 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*, (pp. 183–208). Amsterdam University Press.

Reny, P. (1992). Backward induction, normal form perfection and explicable equilibria. *Econometrica*, *60*, 627–649.

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

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

Stalnaker, R. (1996). On the evaluation of solution concepts. *Theory and Decision*, *37*, 49–73.

Stenning, K., & van Lambalgen, M. (2001). Semantics as a foundation for psychology: A case study of Wason's selection task. *Journal of Logic, Language and Information*, *10*(3), 273–317.

Stenning, K., & van Lambalgen, M. (2004). 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.

Szymanik, J., Meijering, B., & Verbrugge, R. (2013). Using intrinsic complexity of turn-taking games to predict participants' reaction times. In *Proceedings of the 35th annual conference of the cognitive science society*, Berlin.

van Benthem, J. (2008). Logic and reasoning: Do the facts matter? Studia logica. In H. Leitgeb (Ed.), *Special issue on logic and the new psychologism*, (Vol. 88, pp. 67–84).

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

van Lambalgen M., & Counihan, M. (2008). Formal models for real people. *Journal of Logic, Language and Information*. In M. Counihan (Ed.), *Special issue on formal models for real people*, (Vol. 17, pp. 385–389).

van Maanen, L. v., & Verbrugge, R. (2010). A computational model of second-order social reasoning. In *Proceedings of the 10th international conference on cognitive modeling*, (pp. 259–264).

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

Walther, D., Hoek, W. v. d., & Wooldridge, M. (2007). Alternating-time temporal logic with explicit strategies. In *Proceedings of XIth conference (theoretical aspects of rationality and knowledge)*, (pp. 269–278).

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

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