Studying strategies and types of players: Experiments, logics and cognitive models

Sujata Ghosh[1] and Rineke Verbrugge[2]

[1] Indian Statistical Institute, Chennai, India, sujata@isichennai.res.in
[2] Institute of Artificial Intelligence, University of Groningen, The Netherlands, L.C.Verbrugge@rug.nl

Abstract. How do people reason about their opponent in turn-taking games? Often, people do not make the decisions that game theory would prescribe. We present a logic that can play a key role in understanding how people make their decisions, by delineating all plausible reasoning strategies in a systematic manner. This in turn makes it possible to construct a corresponding set of computational models in a cognitive architecture. These models can be run and fitted to the participants' data in terms of decisions, response times, and answers to questions. We validate these claims on the basis of an earlier game-theoretic experiment about the turn-taking game "Marble Drop with Surprising Opponent", in which the opponent often starts with a seemingly irrational move.

We explore two ways of segregating the participants into reasonable "player types". The first way is based on latent class analysis, which divides the players into three classes according to their first decisions in the game: Random players, Learners, and Expected players, who make decisions consistent with forward induction. The second way is based on participants' answers to a question about their opponent, classified according to levels of theory of mind: zero-order, first-order and second-order. It turns out that increasing levels of decisions and theory of mind both correspond to increasing success as measured by monetary awards and increasing decision times.

Next, we use the logical language to express different kinds of strategies that people apply when reasoning about their opponent and making decisions in turn-taking games, as well as the 'reasoning types' reflected in their behavior. Then, we translate the logical formulas into computational cognitive models in the PRIMs architecture. Finally, we run two of the resulting models, corresponding to the strategy of only being interested in one's own payoff and to the myopic strategy, in which one can only look ahead to a limited number of nodes. It turns out that the participant data fit to the own-payoff strategy, not the myopic one. The article closes the circle from experiments via logic and cognitive modelling back to predictions about new experiments.

1 Introduction

Turn-taking games are ubiquitous in our daily life—from debates and deliberations to negotiations, and from competition between firms to coalition formation.

How suitable are idealized formal models of social reasoning processes with respect to the nuances of the real world? In particular, do these formal models represent human strategic reasoning satisfactorily or should we instead concentrate on empirical studies and models based on those empirical data? Such questions have been raised by researchers in game theory, logic and cognitive science (cf. [17,9,60,34,31]).

Game theorists define a *strategy* of a player 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 [46]. 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) [51]. Many cognitive scientists construct theories about human reasoning strategies [35,32], 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 [2].

In [23], together with Meijering, we aimed to bridge the gap between logical and cognitive treatments of strategic reasoning in the turn-taking game "Marble Drop with Rational Opponent". We proposed to combine empirical studies, formal modeling and cognitive modeling to study human strategic reasoning: "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". In the current article, we aim to apply this combination of methods to the questions to what extent people use backward induction or forward induction in a turn-taking game in which the opponent does not always make rational decisions, which we call "Marble Drop with Surprising Opponent", and to what extent they can be differentiated according to reasoning types. Let us give some background first in order to explain our aims more precisely.

Backward and forward induction reasoning In game theory, turn-taking games (or dynamic games) are represented by game trees referred to as extensive-form games. Backward Induction (BI) is the textbook approach for solving extensive-form games with perfect information. In generic games without payoff ties, BI yields the unique subgame perfect equilibrium. The assumptions underpinning BI are that all players commonly believe in everybody's future rationality, no matter how irrational players' past behavior has already proven. Informally, backward induction only considers the opponent's future choices and beliefs, and ignores the opponent's past choices ("let bygones be bygones"). See [46,49] for more details.

In Forward Induction (FI) reasoning, on the other hand, a player takes into account an opponent's past moves and tries to rationalize the past behavior in order to assess that opponent's future moves. Thus, when a player is about to play in a subgame which has been reached due to some strategy of the opponent that is not consistent with common knowledge of rationality for each of the

players, *and* also his past behavior, the player may still rationalize the opponent's past behavior. So how does the player do that? She attributes her opponent a strategy which is optimal against a possible *suboptimal* strategy of hers, or attributes to him a strategy which is optimal against some *rational* strategy of hers, which is only optimal against a suboptimal strategy of *his* and so on. If the player pursues this kind of *rationalizing* reasoning to the highest extent possible [7] and reacts accordingly, she ends up choosing what is called an *Extensive-Form Rationalizable* (EFR) strategy [48] (see also [49,47,22,50]). Thus extensive-form rationalizable strategies are based on forward induction reasoning, and in the following we use the terms extensive-form rationalizable (EFR) and forward induction (FI) synonymously.

Although EFR strategies may be distinct from BI strategies, still, in perfect information games in which both players have a strict ranking among the payoffs at all the game-tree leaves following each of their decision nodes (that is, games without relevant pay-off ties), it has been shown that there is a unique EFR *outcome*, which coincides with the unique BI outcome [8,18,19,49,30]. There have been extensive debates among game theorists and logicians about the merits of backward induction.

Experimental studies on dynamic perfect information games A reason for taking EFR as our predictive concept rather than the more popular BI concept is the fact that experimental economists and psychologists have shown that human subjects do not always follow the backward induction strategy in large *centipede games* [55,17,39,44]. Centipede games, introduced by Rosenthal [55], are two-player turn-taking games of perfect information. The payoffs are arranged in such a way that at each decision point, if a player does not 'go down' to take the first possible exit and the opponent takes the next possible exit, the player receives less than if she had taken the first possible exit; Game 1 in Figure 3 is an example of a relatively small centipede game. Instead of immediately taking the 'down' option, people often show partial cooperation, moving right for several moves before eventually choosing 'down'. Indeed, if a player has reason to believe that the opponent will not exit on the next step, this is a rational decision [55]. For example, Nagel and Tang [44] suggest that people sometimes have reason to believe that their opponent could be an altruist who usually cooperates by moving to the right and McKelvey and Palfrey [39] suggest that players may believe that there is some possibility that their opponent has payoffs different from the ones the experimenter tries to induce by the design of the game. A more recent explanation is that the opponent may have made an error or cannot apply backward induction for the number of steps required [33]; see the paragraph on orders of theory of mind on the next page.

A number of experiments have been done with smaller centipede-like perfect-information games, where the opponent was a rational computer player, and this fact was told to the participants. In some of these experiments, it seemed that people were not able to reason sufficiently deeply about their opponent's strategy [29]. Later, Meijering and colleagues introduced the game "Marble Drop with Rational Opponent", based on a centipede-like game tree with three deci-

4

sion points (first the participant decides, then the computer, then the participant) with a visualization that is intuitive for participants because it resembles a children's toy: a marble drops down a device and its course is influenced by the players' choices of trapdoors to open. Meijering et al. [41,42] showed that both this new visualization as well as several other interventions –namely, stepwise training and questions that prompted participants' reasoning about the opponent– can help the experimental subjects to reason about the rational computer player when they play small centipede-like games. It turned out that with the appropriate interventions, at the end of the experiment after playing more that 40 games, participants made backward induction decisions in more than 90% of games.

Recently, based on an eye-tracking study and complexity considerations, it turned out that even when the participants produced the correct 'backward induction answer' in the "Marble Drop with Rational Opponent" games, they may have used a different internal reasoning strategy to achieve it [43,12].

Theory of mind Theory of mind (ToM) is the ability to attribute beliefs, desires, and intentions to other people, in order to explain, predict and influence their behavior. Even though ToM has been widely studied in the cognitive sciences, relatively little research has concentrated on people's reasoning about their opponents in turn-taking games. We speak of zero-order reasoning in ToM when a person reasons about world facts, as in "Anwesha wrote a novel under pseudonym". In first-order ToM reasoning, a person attributes a simple belief, desire, or intention to someone else, for example in "Khyati knows that Anwesha wrote a novel under pseudonym". Finally, in second-order ToM reasoning, people attribute to other people mental states about mental states, as in "Khyati knows that Soumya thinks that Anwesha did not write a novel under pseudonym".

One way of studying the cognitive basis of theory of mind in a controlled experimental setting is the use of turn-taking games. By investigating the underlying strategies used during these games, one can shed light upon the underlying cognitive processes involved—including ToM reasoning. In recent times, higher-order theory of mind has been the central focus of a lot of research papers that are based on experiments with games (see, for example [17]). Higher-order theory of mind reasoning also became an attractive topic for logical analysis [15].

Typologies of players To the best of our knowledge, studies on the typology of players according to their cognitive strategies in turn-taking games are very scarce. Often it is difficult to gauge from the participants' decisions only, which reasoning patterns (often called 'cognitive strategies') they may actually have been using. Raijmakers et al. [52] have used statistical methods such as latent class analysis to divide children into classes according to the cognitive strategies they may have used in a dynamic game similar to Marble Drop.

In the literature on behavioral game theory, there is a natural tendency to analyze mostly the choices made by players at different turns of the game, thereby ignoring the data on how much time they have taken to make that choice, namely, the *response time* data. Rubinstein [57] does argue for the importance of response times and takes that data into account while discussing a typology of players

in different games. Also, he discusses typologies that are beyond the traditional psychometric typologies originating from 'type theory' and 'trait theory' [6]. Rubinstein views the analysis from a game-theoretic point of view.

In the current article, instead of defining typologies on the basis of game-theoretic approaches, we use latent class analysis [26] as well as an analysis of participants' answers in terms of orders of theory of mind, from zero-order to second-order. Furthermore, we investigate the interplay between the outcomes of the latent class analysis and the theory of mind-based analysis.

The study of such typologies of players may help to explain the differences between people's cognitive attitudes when reasoning strategically and to better understand people's possible behaviors in interactive situations, which in turn may be used for modeling purposes in, for example, economics, artificial intelligence, and linguistics.

Aims of this article Marr [38] has influentially argued that any task computed by a cognitive system must be analyzed at the following three levels of explanation (in order of decreasing abstraction):

the computational level: identification of the goal and of the information-processing task as an input - output function;
the algorithmic and representational level: specification of an algorithm which computes the function;
the implementation level: physical or neural implementation of the algorithm.

In recent years, as part of a revival of interest in Marr's levels in cognitive science, Willems [61] has argued for more attention for the *why* of cognition, "what is the goal for the organism at the present moment". He claims that research in cognitive neuroscience has often been stimulus-driven or capacity-driven, overlooking the organism's goal, which is properly investigated at the computational level. We agree with the importance of the computational level, but are also interested in the *how* of cognition, investigated at the algorithmic level. We think that both logic and computational cognitive modeling can play a fruitful role at both these levels and at the interface between them.

According to Isaac et al. [31], logic can be of use at each of Marr's three levels, but in the history of cognitive science, logic has been especially useful at the computational level. Baggio and colleagues [5] provide some fruitful examples in which computational level theories based on appropriate logics predict and explain behavioral data and even EEG data in the cognitive neuroscience of reasoning and language.

As to computational cognitive modeling, Cooper and Peebles [20] argue that computational cognitive architectures such as ACT-R through their theoretical commitments constrain declarative and procedural learning, thereby constraining both the functions that can be computed (the computational level) and the way that they can be computed (the algorithmic level).

In the current article, our main aim is to construct an appropriate logic to describe participants' possible cognitive reasoning strategies when reasoning

about a surprising opponent in a turn-taking game and then to find a generic method to turn these logical descriptions into computational cognitive models in the recently developed cognitive architecture PRIMS [58].

This aim extends the aim that we had in our paper with Meijering [23]. In the current article, we extend the language that we introduced there to represent strategies by a new belief component, so that we can now describe reasoning about the opponent at a more fine-grained level than was necessary to model participants reasoning in "Marble Drop with Rational Opponent". Figure 1, visually similar to the scheme in [23], presents how the details of our approach are laid out in the current paper.

$$\textit{Experimentation} \longrightarrow \textit{Formal modeling} \longrightarrow \textit{Cognitive modeling}$$
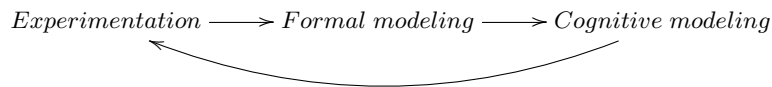
Fig. 1. A schematic diagram of the approach: The experiments discussed in Section 3 inform our logical model of reasoning strategies in "Marble Drop with Surprising Opponent" in Section 2. This logical model in turn helps to construct computational cognitive models of reasoning strategies in the cognitive architecture PRIMs in a generic way, as presented in Section 5; subsequently, two instantiations of the resulting models are validated against the experimental results. Finally, as described in Section 5, simulations with computational cognitive models often lead to new experiments in order to test the models' predictions.

This extension to the logic was needed to make reasonable models of participants' reasoning in the more complex turn-taking game "Marble Drop with Surprising Opponent". Together with Heifetz, we conducted a game-theoretic experiment that involves a participant's expectations about the opponent's reasoning strategies, that may in turn depend on expectations about the participant's reasoning. The resulting article [22] deals with the following question: In the dynamic game of perfect information "Marble Drop with Surprising Opponent", are people generally inclined to do forward induction reasoning (i.e. show EFR behavior)? The main new elements of this article with respect to [23,22] are as follows:

- In comparison to the logical language introduced in [23], we have now included the possibility to represent agents' beliefs about their opponents' moves and beliefs. We conjecture that the new language is more succinct than the one proposed in [23] in describing strategic reasoning (see Section 4.1 for a discussion), which in turn may lead to more efficient computational cognitive modelling, for example, if there is a straightforward generic translation from the logical syntax to the computational representations. An initial presentation of the language was given in our LORI paper [21], which is now extended with worked-out examples of formalized reasoning strategies.
- Instead of the generic trends in participants' choices ("do they generally show EFR behavior or not?") studied in [22], we now turn our attention to

differences between players: can they be characterized in meaningful ways? We introduce two typologies, one based on latent-class analysis and one based on orders of explicit theory of mind in participants' verbal comments regarding the reasoning about the opponent which they applied to make their decisions. An initial analysis of such typologies was given in the conference contribution [28], which is now extended with a comparison between the outcomes of the two analyses.

– In comparison to the computational cognitive models of [23,21] which were based on the cognitive architecture ACT-R, we now base our generic translations from strategic logic formulas to computational cognitive models on the new architecture PRIMs [58].

– Unlike in any of our previous work, we have now implemented two PRIMs models resulting from two logical formulations of possible reasoning strategies in "Marble Drop with Surprising Opponent", and have made predictions based on the simulations about the data of our previous experiment, and then compared the simulations to the experimental results with respect to decisions and reaction times. Thus, this article closes the circle from experiments via logic and cognitive modelling back to predictions about the current and new experiments.

The rest of this article is structured as follows. In Section 2, we extend the language introduced in [23] to describe players' reasoning strategies and types of players, adding a belief operator to reflect players' expectations. In Section 3, we briefly recall Ghosh and colleagues' recent experiment on forward induction [22] and suggest two typologies of players, based on strategic and cognitive analysis, respectively. In Section 4, the reasoning strategies and the reasoning types discussed in Section 3 are described with the logical syntax proposed in Section 2. Finally, in Section 5, we sketch how strategy and belief formulas in this extended language can be turned into production rules of computational cognitive models that help to distinguish what is going on in people's minds when they play dynamic games of perfect information. Finally, we validate two of the resulting models by running them and comparing results with respect to decision and reaction time to the participants' data.

2   A language for types and strategies

The focus of [23] has been to use a logical framework as a bridge between experimental findings and computational cognitive modelling of strategic reasoning in a simpler Marble Drop setting, in which the computer opponent always made rational choices: "Marble Drop with Rational Opponent". Taking off from the work of [23], we now propose a logical language specifying strategies as well as *reasoning types* of players. As mentioned above, our motivation for introducing this logical framework is to build a pathway from empirical to cognitive modelling studies.

This framework uses empirical studies to provide insights into cognitive models of human strategic reasoning as performed during the experiment discussed

in Section 3. The main idea is to use the logical syntax to express the different reasoning procedures as performed and conveyed by the participants and use these formulas to systematically build up reasoning rules of computational cognitive models of strategic reasoning.

A novel part of the proposed language is that we add an explicit notion of belief to the language proposed in [23] in order to describe participants' expectations regarding future moves of the computer. This belief operator is parametrized by both players and nodes of the game tree so that the possible expectations of players at each of their nodes can be expressed within the language itself. The whole point is to explicate the human reasoning process, therefore the participants' beliefs and expectations need to come to the fore. Such expectations form an essential part of the experimental study discussed in the next section.

In addition to describing strategic reasoning, we also describe different typologies of players based on the various factors that might influence human strategic reasoners, as discussed in the previous section. We will use the same syntax to describe such types. Before moving on any further, we first define the concepts necessary for describing the strategies and typologies.

2.1 Describing game trees and strategies in logic

In this subsection, we give reminders of the definitions of extensive form games, game trees and strategies, following [23]. On the basis of these concepts, we present our new logical contribution in Section 2.2, where we formalize reasoning strategies and typologies.

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 $N = \{C, P\}$. We often use the notation i and $\bar{\imath}$ to denote the players, where $\overline{C} = P$ and $\overline{P} = C$. 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 $\vec{s} = \{s' \in S \mid s \overset{a}{\Rightarrow} s' \text{ for some } a \in \Sigma\}$. A node s is called a leaf node (or terminal node) if $\vec{s} = \varnothing$.

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 $\tau^{\bar{i}}$ to denote strategies of player $\bar{i}$. 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 $\bar{i}$. 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 $\bar{i}$, 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 $s \overset{a}{\Rightarrow}_\mu s'$.
- $\widehat{\lambda}_\mu = \widehat{\lambda} \upharpoonright_{S_\mu}$.

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 $j \geqslant 0$ 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 $\pi^{\bar{i}}$ to denote partial strategies of player $\bar{i}$. 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 $\bar{i}$, 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 $s' \in \overrightarrow{s}$, $s' \in S$ 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.

Syntax for extensive form game trees Let us now build a syntax for game trees (cf. [54,24]). We use this syntax to parametrize the belief operators given below so as to distinguish between belief operators for players at each node of a finite extensive form game. Let N denote a finite set of players and let Σ denote a finite set of actions. We use i to range over the set N. As earlier, we restrict our attention to two player games, and we take $N = \{C, P\}$. We use the notation i and $\bar{\imath}$ to denote the players, where $\overline{C} = P$ and $\overline{P} = C$. Let Σ be a finite set of action symbols representing moves of players; we let a, b range over Σ. Let *Nodes* be a finite set. The syntax for specifying finite extensive form game trees is given by:

$$\mathbb{G}(Nodes) ::= (i, x) \mid \Sigma_{a_m \in J}((i, x), a_m, t_{a_m})$$

where $i \in N$, $x \in Nodes$, $J(\text{finite}) \subseteq \Sigma$, and $t_{a_m} \in \mathbb{G}(Nodes)$.

Given $h \in \mathbb{G}(Nodes)$, we define the tree T_h generated by h inductively as follows (see Figure 2.1 for an example):

- $h = (i, x)$: $T_h = (S_h, \Rightarrow_h, \widehat{\lambda}_h, s_x)$ where $S_h = \{s_x\}$, $\widehat{\lambda}_h(s_x) = i$.
- $h = ((i, x), a_1, t_{a_1}) + \cdots + ((i, x), a_k, t_{a_k})$: Inductively we have trees $T_1, \ldots T_k$ where for $j : 1 \leqslant j \leqslant k$, $T_j = (S_j, \Rightarrow_j, \widehat{\lambda}_j, s_{j,0})$.
 Define $T_h = (S_h, \Rightarrow_h, \widehat{\lambda}_h, s_x)$ where
  - $S_h = \{s_x\} \cup S_{T_1} \cup \ldots \cup S_{T_k}$;
  - $\widehat{\lambda}_h(s_x) = i$ and for all j, for all $s \in S_{T_j}$, $\widehat{\lambda}_h(s) = \widehat{\lambda}_j(s)$;
  - $\Rightarrow_h = \bigcup_{j:1 \leqslant j \leqslant k}(\{(s_x, a_j, s_{j,0})\} \cup \Rightarrow_j)$.

Given $h \in \mathbb{G}(Nodes)$, let $Nodes(h)$ denote the set of distinct pairs (i, x) that occur in the expression of h.

2.2 Strategy specifications

We have used the syntax of Section 2.1 in our previous article [23] to describe empirical reasoning of participants involved in a simpler game experiment using "Marble Drop with Rational Opponent" [42,40]. The main case specifies, for a player, which conditions she tests before making a move. In what follows,
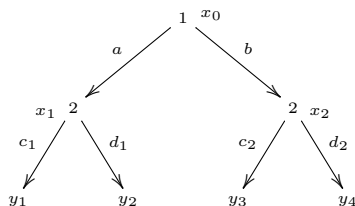
Fig. 2. Extensive form game tree. The nodes are labelled with turns of players and the edges with the actions. The syntactic representation of this tree can be given by:
$h = ((1, x_0), a, t_1) + ((1, x_0), b, t_2)$, where
$t_1 = ((2, x_1), c_1, (2, y_1)) + ((2, x_1), d_1, (2, y_2))$;
$t_2 = ((2, x_2), c_2, (2, y_3)) + ((2, x_2), d_2, (2, y_4))$.

the pre-condition for a move depends on observables that hold at the current game position, some belief conditions, 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 syntax defined below.

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

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

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

Formulas in $BPF(X)$ can be read as usual in a dynamic logic framework and are interpreted at game positions. The formula $\langle a^+ \rangle \psi$ (respectively, $\langle a^- \rangle \psi$) refers to 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 $Bool(X)$.

For each $h \in \mathbb{G}(Nodes)$ and $(i, x) \in Nodes(h)$, we now add a new operator $\mathbb{B}_h^{(i,x)}$ to the syntax of $BPF(X)$ to form the set of formulas $BPF_b(X)$. The formula $\mathbb{B}_h^{(i,x)}\psi$ can be read as "in the game tree h, player i *believes* at node x that ψ holds". One might feel that it is not elegant that the belief operator is parametrized by the nodes of the tree. However, our main aim is not to propose a logic for the sake of its nice properties, but to have a logical language that can be used suitably for constructing computational cognitive models corresponding to participants' strategic reasoning.

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 kinds of propositional variables $(u_i = q_i)$ to denote 'player i's utility (or payoff) is q_i' and $(r \leqslant q)$ to denote that 'the rational number r is less than or equal to the rational number q'.[3] 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 $\psi \in BPF_b(P^i)$. For a detailed explanation see [23]. The basic idea is to use the above constructs to specify properties of strategies as well as to combine

---

[3] as in [23] and inspired by [13].

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.

Semantics We consider perfect information games with belief structures as models. The idea is very similar to that of temporal belief revision frames presented in [13]. Let $M = (T, \{\longrightarrow_i^x\}, V)$ with $T = (S, \Rightarrow, s_0, \widehat{\lambda}, \mathcal{U})$, where $(S, \Rightarrow, s_0, \widehat{\lambda})$ is an extensive form game tree, $\mathcal{U} : frontier(T) \times N \to \mathbb{Q}$ is a utility function. Here, $frontier(T)$ denotes the set of leaf nodes of the tree T. For each $s_x \in S$ with $\widehat{\lambda}(s_x) = i$, we have a binary relation $\longrightarrow_i^x$ over S (cf. the connection between h and T_h presented above). Finally, $V : S \to 2^P$ is a valuation function. The truth value of a formula $\psi \in BPF_b(P)$ 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 $s \xRightarrow{a} s'$ and $M, s' \models \psi$.
- $M, s \models \langle a^- \rangle \psi$ iff there exists an s' such that $s' \xRightarrow{a} s$ and $M, s' \models \psi$.
- $M, s \models \mathbb{B}_h^{(i,x)} \psi$ iff the underlying game tree of T_M is the same as T_h and for all s' such that $s \longrightarrow_i^x s'$, $M, s' \models \psi$.

The truth definitions for the new propositions are as follows:

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

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

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

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

We recall that a strategy for player i is a function μ^i which specifies a move at every game position of the player, i.e. $\mu^i : S^i \to \Sigma$. A strategy μ can also be viewed as a subtree of T where for each node belonging to the opponent player i, there is a unique outgoing edge and for nodes belonging to player $\bar{i}$, every enabled move is included. 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. $\sigma^i : S^i \to \Sigma$. A partial strategy can be viewed as a set of total strategies of the player [23].

The semantics of the strategy specifications are given as follows. Given a model M and a partial strategy specification $\eta \in Strat^i(P^i)$, we define a semantic function $[\![\cdot]\!]_M : Strat^i(P^i) \to 2^{\Omega^i(T_M)}$, where each partial strategy specification is associated with a set of total strategy trees and $\Omega^i(T)$ denotes the set of all player i strategies in the game tree T.

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

- $[\![[\psi \mapsto a]^i]\!]_M = \Upsilon \in 2^{\Omega^i(T_M)}$ 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 $out_\mu(s) = a$.
- $[\![\eta_1 + \eta_2]\!]_M = [\![\eta_1]\!]_M \cup [\![\eta_2]\!]_M$
- $[\![\eta_1 \cdot \eta_2]\!]_M = [\![\eta_1]\!]_M \cap [\![\eta_2]\!]_M$

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.

Before describing specific strategies found in the empirical study, we would like to focus on the new operator of belief, $\mathbb{B}_h^{(i,x)}$ proposed above. Note that this operator is considered for each node in each game. The idea is that the same player might have different beliefs at different nodes of the game. We had to introduce the syntax of the extensive form game trees to make this definition sound, otherwise we would have had to restrict our discussion to single game trees. The semantics given to the belief operator is entangled in both the syntax and semantics, which might create problems in finding an appropriate axiom system. A possible solution would be to introduce some generic classes of games similar to the idea of generic game boards [10], using the notion of enabled game trees [24]. This is left for future work, as well as a comparison of the expressiveness of the current language with those of existing logics of belief and strategies.

3 Experimental study: do people use forward induction?

We now move on to the *empirical* part of the work. The experiment on which we previously reported in [22] was designed to tackle the question whether people are inclined to use forward induction (*FI/EFR*) reasoning when they play dynamic perfect information games. The main interest was to examine participants' behavior following a deviation from backward induction (*BI*) behavior by their opponent, the computer, right at the beginning of the game. The computer was programmed in such a way that in each game it played according to a strategy that is the best response with respect to some strategy of the human participant, and sometimes this meant a deviation from a *BI* strategy. When the participant was about to play next, the question was whether they would take the computer's previous moves under consideration in assessing its future move and play accordingly, thereby applying extensive form rationalizability, or they would just play as if they were playing a new game starting at their present node without considering the previous move(s), by backward induction reasoning; for details, see [22].
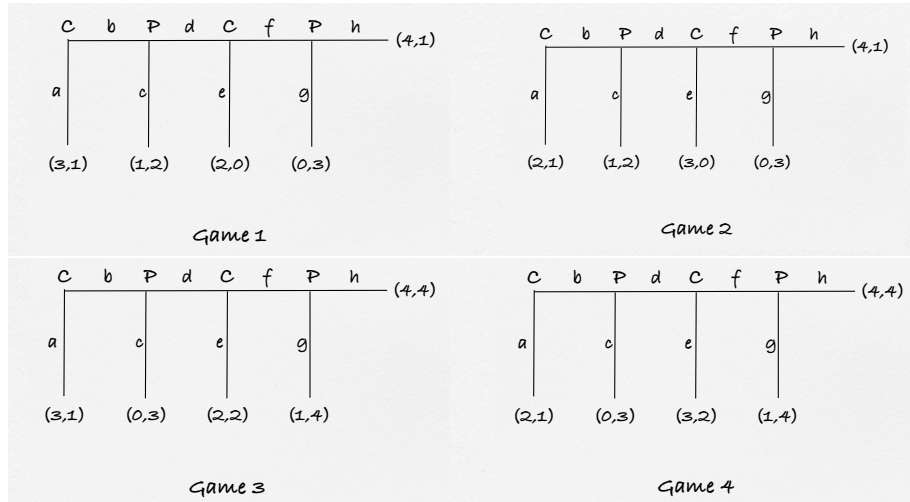
Fig. 3. Collection of the main games used in the experiment presented as extensive form game trees. Vertices represent decision points and are labeled by the player whose turn it is, where C stands for the computer and P for the participant. Edges are labeled by the names of actions; thus a stands for the computer going down, thereby ending the game, while b stands for the computer going to the right and continuing the game. The ordered pairs at the leaves represent pay-offs for the computer (C) and the participant (P), respectively; for example, the $(3, 1)$ at the leftmost leaf of game 1 means that if the game ends there, the computer gains 3 marbles, while the participant gains 1 marble. In games 1–4, the computer plays first. Because of the typical tree structure of these games, they are often called "centipede games" in the literature.

As a reminder, the games that were used in the experiment of [22] are given in Figures 3 and 4. In these two-player games, the players play alternately, therefore they are called turn-taking (or dynamic) games. Let C denote the computer and P the participant. In the first four games (Figure 3), the computer plays first, followed by the participant. The players control two decision nodes each. In the last two games (Figure 4), which are truncated versions of two of the games of Figure 3, the participant moves first.

To explain the difference between BI and EFR behavior consider game 1, one of the experimental games (cf. Figure 3). Here, the unique Backward Induction (BI) strategies for player C and player P are $a; e$ and $c; g$, respectively, which indicate that the game will end at the first node, going down.

In contrast, for forward induction reasoning, the question is how the participant would play if her first decision node was reached; in game 1, reaching the first P-node would already indicate that the opponent C had not opted for its rational decision, namely to go down immediately. Would the participant's (P's) decision depend on her opponent's previous choice? Here, she would have to choose between continuing the game (by moving to the right, action d) and opting out (by moving down, action c).
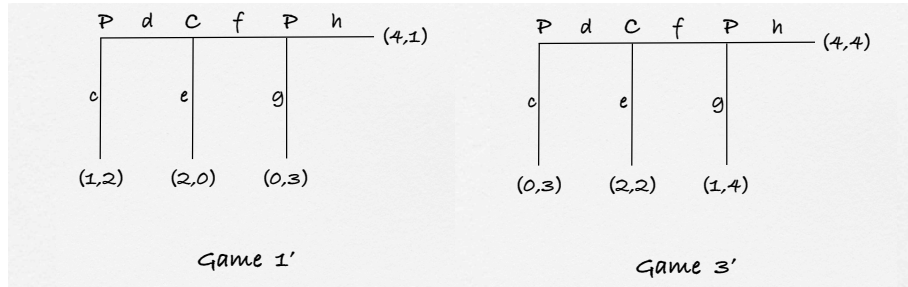
Fig. 4. Truncated versions of Game 1 and Game 3. The ordered pairs at the leaves represent pay-offs for C and P, respectively. The participant (P) plays first.

EFR would proceed as follows, starting from the first decision node of P. Among the two strategies of player C that are compatible with this event, namely $b; e$ and $b; f$, only the latter is rational for player C. This is because of the fact that $b; e$ is dominated by $a; e$, while $b; f$ is optimal for player C if it believes that player P will play $d; h$ with a high enough probability. Attributing to player C the strategy $b; f$ is thus player P's best way to rationalize player C's choice of b, and in reply, $d; g$ is player P's best response to $b; f$. Thus, the unique Extensive-Form Rationalizable (EFR, [48]) strategy (an *FI* strategy) of player P is $d; g$, which is distinct from her BI strategy $c; g$. For a detailed discussion on BI and EFR strategies in games $2, 3, 4, 1', 3'$, see [22]. As a reminder, we repeat the table of BI and EFR strategies here, with permission.

Games — Strategies	BI strategy	EFR strategy
Game 1	C: $a; e$	C: $a; e$
	P: $c; g$	P: $d; g$
Game 2	C: $a; e$	C: $a; e$
	P: $c; g$	P: $c; g$
Game 3	C: $a; e, b; e, a; f, b; f$	C: $a; e, a; f, b; f$
	P: $c; g, d; g, c; h, d; h$	P: $d; g, d; h$
Game 4	C: $a; e, b; e, a; f, b; f$	C: $a; e, b; e, a; f, b; f$
	P: $c; g, d; g, c; h, d; h$	P: $c; g, d; g, c; h, d; h$
Game $1'$	C: e	C: e
	P: $c; g$	P: $c; g$
Game $3'$	C: e, f	C: e, f
	P: $c; g, d; g, c; h, d; h$	P: $c; g, d; g, c; h, d; h$

Table 1. BI and EFR (FI) strategies for the 6 experimental games in Figures 3 and 4. Notice that for Games 1 and 2, having no pay-off ties, the general result implies that there is just one unique EFR outcome, coinciding with the BI outcome, namely C chooses a, exiting the game immediately. Even if there are relevant pay-off ties, the EFR outcomes constitute a subset of the BI outcomes [18,19,49], but the inclusion may possibly be strict. This is illustrated by Game 3, which was first described by Chen and Micali [19]; here, one possible BI outcome is given by C choosing b followed by P choosing c, which cannot be achieved by EFR.

3.1 Materials, methods and aggregated results

The experiment of [22] was conducted at the Institute of Artificial Intelligence at the University of Groningen, the Netherlands. A group of 50 Bachelor's and Master's students from different disciplines took part. They had little or no knowledge of game theory, so as to ensure that neither backward induction nor forward induction was already known to them.[4] The participants played finite perfect-information games that were game-theoretically equivalent to the games depicted in Figures 3 and 4. However, the presentation was made such that participants were able to understand the games quickly, see an example of the graphical interface on the computer screen (cf. Figure 5).

Materials In each game, a marble was about to drop. Both the participant and the computer determined its path by controlling the trapdoors: The participant controlled the orange trapdoors, and the computer the blue ones. The participant's goal was that the marble should drop into the bin with as many orange marbles as possible. The computer's goal was that the marble should drop into the bin with as many blue marbles as possible. In Figure 5, a practice game that did not correspond to any of the six games in Figures 3 and 4, if the computer is rational and uses backward induction, it opens the top right blue trapdoor, leading to 3 blue marbles (its rational choice for this game).

In the experiment, however, the computer often makes an apparently irrational first choice, operationalized as follows. For each game item, the computer opponent had been programmed to play according to plans that were best responses to some plan of the participant. This was told to the participants in advance. We dub this game "Marble Drop with Surprising Opponent".

Procedure Each participant first played 14 practice games so that the participants were familiar with the games before the start of the experiment proper. In the actual experiment, each participant played 48 games divided into 8 rounds, each comprised of the 6 different game structures corresponding to Games 1, 2, 3, 4, 1′ and 3′ that were described above (see Figures 3 and 4). Different graphical representations of the same game were used in different rounds so as to prevent recognition. We were especially interested in the decision at the participant's first decision node if that node was reached: did the participant end the game by choosing c or continue by choosing d?

At some points during the experimental phase, the participants were asked a multiple-choice question: "When you made your initial choice, what did you think the computer was about to do next?" (possibilities: most likely e, most likely f, or neither).

At the end of the experiment, each participant was asked the following question: "When you made your choices in these games, what did you think about the ways the computer would move when it was about to play next?" The participants were asked to describe in their own words which plan they thought was

---

[4] The candidate participants were asked about their educational details. Two students who had followed a course on game theory were excluded.
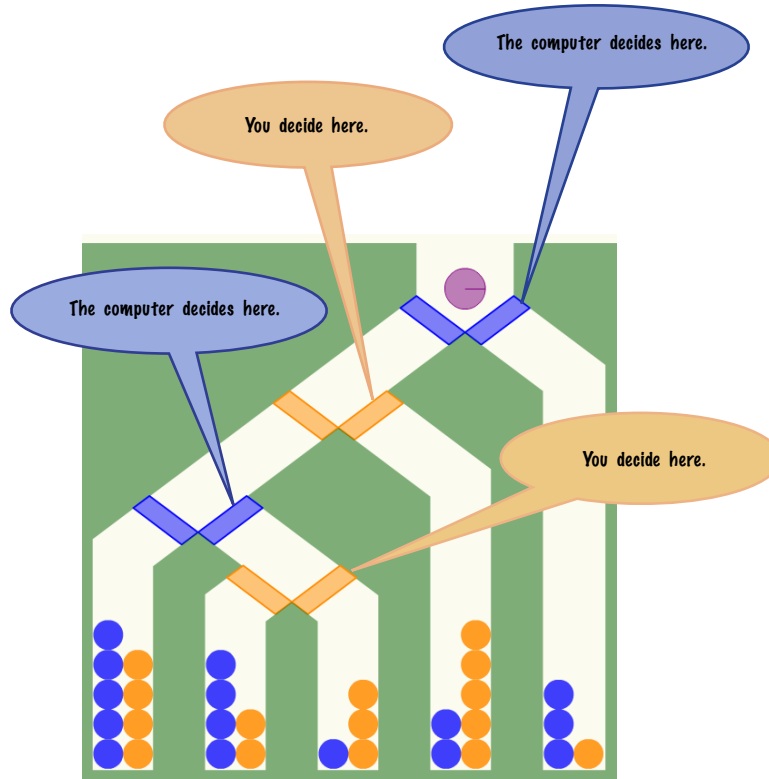
Fig. 5. Graphical interface for the participants. The computer controls the blue trap-doors and acquires blue marbles (represented as dark grey in a black and white print) as pay-offs, while the participant controls the orange trapdoors and acquires orange marbles (light grey in a black and white print) as pay-offs.

followed by the computer on its next move after the participant's initial choice. We used these answers to classify various strategic reasoning processes applied by the participants while playing the experimental games. Participants earned 10-15 euros for participation, depending on points earned.

The forward induction hypothesis In [22], to analyse whether participants P played *FI* strategies in the games described in Figures 3 and 4, we formulated the following *forward induction hypothesis* (cf. Table 3) concerning the participant's choice in his first decision node (if reached in games 1, 2, 3, 4, and in all rounds of games $1'$ and $3'$):

> *Action d will be played more often in game 1 than in game 2 or $1'$, and more often in game 3 than in game 4 or $3'$.*

18

Note that game 2 is similar to game 1 except for the pay-offs for *C* after the moves *a* and *e*, which are interchanged, and game 4 is similar to game 3 except for the pay-offs for *C* after the moves *a* and *e*, which are interchanged. Games 1′ and 3′ are truncated versions of games 1 and 3, respectively. In games 1 and 3, *d* is the only *EFR* move; in games 1′ and 2, *d* is neither a *BI* nor an *EFR* move; and in games 3′ and 4, both *c* and *d* are *EFR* moves.

General results on strategic reasoning in the game It turned out that in the aggregate, participants were indeed more likely to make decisions in accordance with their best-rationalization *EFR* conjecture, i.e., consistent with *FI* reasoning [22]. However, there exist alternative explanations for the choices of most participants, and such alternative explanations also emerge from several of the participants' free-text verbal descriptions of their considerations as solicited from them at the end of the experiment. One likely alternative explanation had to do with the extent of risk aversion that some participants at their first decision nodes (which was reached because the computer played *b*, instead of the outside option *a*) attributed to the computer in the remainder of the game, rather than reasoning about the sunk outside option that the computer had already foregone at the beginning of the game. For a detailed study and a discussion of some alternative explanations of the results, see [22].

In the next subsections, we explore several ways of segregating the participants into groups to see whether and how they can be divided into reasonable "player types". We started with the most obvious ways to divide the participants: We segregated the participants in terms of gender and discipline (topic of study) and went on to test the *Forward induction hypothesis* over the different groups formed by segregation.[5] The statistical analyses based on gender and discipline suggest that the results mentioned above about participants' behavior at their first decision node are robust. We only found minor variations corresponding to certain groups (see [21] for a report). Because the results on the hypothesis turn out to be rather robust, we considered more subtle typologies that emerge out of the experimental findings, in two ways: (i) by latent class analysis of the participants based on their choices, *c* or *d*, at the first decision node in the game items corresponding to games 1, 2, 3 and 4 of Figure 3; and (ii) by theory of mind analysis, as exhibited by the participants in their free-text verbal descriptions of their considerations about the computer's moves.

3.2 Latent Class Analysis

Latent class analysis (LCA) is a statistical method that can be applied to classify binary, discrete or continuous data in a manner that does not assign subjects to classes absolutely, but with a certain probability of membership for each class [26]. Latent class analysis can be used to explore how participants can best be distinguished according to reasoning strategies, in cases where no fixed

---

[5] Because of little variance among participants, we did not segregate by age.

19
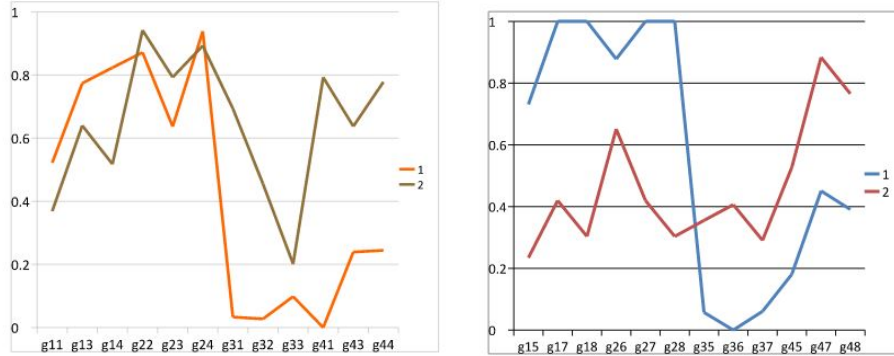
Fig. 6. Graphical representations of latent class analysis for the set containing the first three rounds for each game (left) and the set containing the last three rounds for each game (right). The horizontal axes correspond to the different instantiations of the games at the rounds of the game, where `gij` stands for the jth round of game i of Figures 3 and 4, while the vertical axes correspond to the probability of playing c.

set of reasoning strategies has been defined in advance. Raijmakers and colleagues [52] have profitably applied latent class analysis to analyze children's reasoning strategies in turn-taking games.

As mentioned above, for the current experiment, the participants were categorized into certain classes based on their choices, c or d, at the first decision node in the game items corresponding to games 1, 2, 3 and 4 of Figure 3. Note that each participant played 8 rounds of each game, in 2 rounds of which the computer, playing first, immediately ended the game playing a. So, the participant only had to reason in 6 rounds of each of the games 1, 2, 3 and 4.

The latent class analysis was performed using the statistical software R, with 25 estimated parameters and 25 residual degrees of freedom. Since each participant played in 6 rounds of 4 games, we had 24 data points in total for each participant. So even if we had wanted to divide the participants into two classes, we did not have enough parameters to work with, as the total number of participants was 50. Consequently, we divided the available data points into two sets of 12 and subsequently performed the analysis. The data for 50 participants were separated into two sets: the set containing the first three rounds for each game in which they had to make a decision at the first decision point and the set containing the last three rounds for each game in which they had to make a decision at the first decision point.The participants were classified into two groups based on their behavior in each set of three rounds. Figure 6 shows the graphs depicting the fraction of their choices of c in each of the relevant rounds in each of the games: on the left for rounds 1-4 and on the right for rounds 5-8 (g_{ij} denotes behavior at the jth round of the ith game).

20

The different predicted groups are denoted by different colors in Figure 6. Group 1 behaved in an expected fashion (akin to EFR behavior) in both cases, compared to the more random behavior of the other group. Considering group 1 for both sets of rounds, 24 common participants were noted down, who were predicted to behave in an expected fashion in all the rounds. The available data on the behavior of these 24 participants at their first decision node in the six games were considered and hypothesis testing was done for these 24 participants exclusively,[6] for the games 1, 2, 3 and 4 of Figure 3. The result for the forward induction hypothesis was as follows:

− d was played more often in game 3 than in game 4 and more often in game 1 than in game 2.

For the individual games, the tests revealed the following behaviour. The null hypothesis was that c and d were chosen equally often at the first decision node, whereas the alternatives were chosen accordingly:

− Game 1: c was chosen more often than d.
− Game 2: c was chosen more often than d.
− Game 3: d was chosen more often than c.
− Game 4: d was chosen more often than c.

Further groups that resulted from the latent class analysis are as follows:

Group 1: These participants played in an expected fashion in both the initial three rounds and the later three rounds; there were 24 such players.
Group 2: These participants did not play in an expected fashion in the initial three rounds but played in an expected fashion in the later three rounds; there were 9 such players.
Group 3: These participants played in an expected fashion in the initial three rounds but did not play in an expected fashion in the later three rounds; there were 7 such players.
Group 4: These participants did not play in an expected fashion in either the earlier or the later set of three rounds; there were 10 such players.

Statistical Typology On the basis of the above analysis, we propose the following statistically developed typology of players:

Expected: the 24 players who belong to group 1 above;
Learner: the 9 players from group 2 above;
Random: the 17 players from groups 3 and 4 combined.

Interestingly, this classification corresponds neatly with the amount of money that participants gained in the game by earning points corresponding to the marbles gained in each game (€10 fixed reward plus €0.04 for each marble achieved). While overall the total rewards for the 50 participants ranged between €14.10 and €14.85, the Expected players earned an average of €14.64, which is

---

[6] The results are based on one sample and two sample proportion tests.

quite a bit more than the Learners' average earnings of €14.46, which in turn surpasses the Random players' average earnings of €14.42.

For further statistical validations of the proposed typology, we tested a number of hypotheses using standard statistical methods. One such hypothesis is to check whether the answering time is more in case of expected players than random players. The intuition behind this hypothesis is that a person who is playing in an expected fashion or learning to do so would pay greater attention in choosing a correct option than a person who is playing less sensibly (randomly), cf. [56,57]. This hypothesis was tested twice using two sample t-test for difference of means, firstly Expected versus Random and secondly Expected+Learner versus Random. In both cases, our null hypothesis of equality of means was rejected at 5% level of significance (p-values 0.02 and 0.04, respectively). Hence, we may regard that the Expected and Learner players took more time in answering than the players termed as Random. As a conclusion of the above analysis, we can regard that the three statistically developed types proposed above are robust at 5% level of significance.

3.3 Theory of Mind Study

At the completion of the game-theoretic experiment, each participant was asked to answer the following final question:

> "When you made your choices in these games, what did you think about the ways the computer would move when it was about to play next?"

The participant needed to describe in his or her own words, the plan he or she thought was followed by the computer on its next move after the participant?s initial choice. Based on their answer, 48 players were classified into three types according to the order of theory of mind exhibited in their answer to the final question.[7]. These were the types:

Zero-order players, who did not mention mental states in their answer; there were 5 such players.
First-order players, who presented first-order theory of mind in their answer; there were 27 such players;
Second-order players, who presented second-order theory of mind in their answer; there were 16 such players.

This classification, as mentioned above, was done by manual scrutiny of each answer. If an answer referred to behavior only but not to mental states, we classified it as zero-order. If mental state verbs such as think, decide, expect, plan, know, believe, intend, and take a risk were attributed to the computer, we classified the answer as (at least) level 1. If similar mental state verbs about the participant were embedded into mental state clauses referring to the computer,

---

[7] The two participants who answered "I don't know" have been excluded from classification.

as in "He thinks that I plan to choose to go left", we classified the answer as second-order. We did not find any deeper embeddings, corresponding to third- or higher-order answers. The set of all participants' answers will be made available at `http://www.ai.rug.nl/SocialCognition/experiments/`. Typical answers from each group are as follows:

Zero-order answers: "It would repeat its former choice in the same situation."

First-order answers: "I thought the computer took the option with the highest *expected* value. So if on one side you had a 4 blue + 1 blue marble and on the other side 2 blue marbles he would take the option 4+1=2.5."

Second-order answers: "...I thought the computer *anticipated* that I (his opponent) would *go for* the bin with the most orange marbles in his *decision* to open doors. This could lead to him getting less marbles than '*expected*' because I would choose a safe option (3 marbles) over a chance between 4 marbles or 1 (depending on the computer's doors)."

Similar to the case of latent class analysis, the classification by orders of theory of mind also corresponds to the average rewards that participants from each group gained in the game by earning points corresponding to the marbles gained in each game. The Second-order ToM participants earned an average of €14.58, which is more than the First-order ToM participants' average earnings of €14.51, which in turn surpasses the Zero-order participants' average earnings of €14.46.

For statistical validation of the theory of mind classification into zero-order, first-order, and second-order participants, we set up different hypotheses. Intuitively, one can expect that the players adopting second-order theory of mind would take maximum time to make a decision at the first decision node in comparison to players adopting first-order theory of mind and that people adopting zero-order theory of mind would take the least time among all three classes. This fact was validated statistically by performing difference of means test on the response time data of the first decision node for the three classes. We tested the hypotheses at 5% level of significance. Combining the results, we found that $\mu s > \mu f > \mu z$ for first decision time. Here, μs stands for the mean first decision time of second-order players, μf and μz denotes the first decision times for the first-order and zero-order players, respectively. Reviewing the results obtained, we can conclude that the three types of participants based on theory of mind are statistically valid and robust at 5% level of significance.

3.4 Comparing typologies: latent class analysis and theory of mind

To get a sense of whether and how the two typologies which both have three classes that intuitively correspond to growing levels of rationality correspond to each other, we have started from the LCA classes and counted how many participants were in each of the 9 possible intersections according to the theory of mind levels of their answers:

Random players (17 players)
  No answer: 1 participant;

Zero-order players: 2 participants;
First-order players: 7 participants;
Second-order players: 7 participants.
Learners (9 players)
Zero-order players: 1 participant;
First-order players: 7 participant;
Second-order players: 1 participant.
Expected players (24 players)
No answer: 1 participant;
Zero-order players: 2 participants;
First-order players: 13 participants;
Second-order players: 8 participants.

Contrary to intuitive expectations, the levels do not match exactly. There is a clear match at the intermediate levels in the sense that if a player is a Learner according to LCA, than he/she has a much higher chance to give a first-order answer than in the general population (7 out of 9 compared to 27 out of 48), and therefore much lower chances to give a zero-order answer and to give a second-order answer. It seems that these 7 Learners are doing less than perfect reasoning at first, but slowly come to understand the game in a better way, even with their First-order theory of mind reasoning.

Surprisingly, Second-order theory of mind players are divided almost equally over the Expected players (8) and the Random players (7). It appears that a slight majority of the Second-order reasoners understand the game properly and hence play in the Expected way. When looking more closely at the answers of the Second-order players who are classified as Expected players, four of the eight mention *aversion to risk* (that they are, that the opponent is, or that the opponent thinks they are risk-averse) and three of them mention the opponent making *surprising* choices. Among the Second-order Random players, in contrast, the aspect of risk-aversion is only mentioned by one player and the aspect of surprise does not occur at all; instead, two of these Second-order Random players mention *risk-seeking* attitudes of themselves or the opponent, while three others mention the *(non-)competitive or trusting* nature of the opponent.

4   Describing strategies and types of reasoning

We are now ready to describe the reasoning strategies and the reasoning types discussed in Section 3 with the syntax proposed in Section 2.

4.1   Describing specific strategies in the experimental games

Let us now express some actual reasoning processes that participants displayed during the experiment. Some participants described how they reasoned in their answers to the final question. **Example 1** of such reasoning: "If the game reaches my first decision node and if the payoffs are such that I believe that the computer

would not play e if its second decision node is reached, then I play d at my current decision node". This kind of strategic reasoning can be expressed using the following formal notions.

Let us assume that actions are part of the observables, that is, $\Sigma \subseteq P$. The semantics for the actions can be defined appropriately. Let $n_1, \ldots, n_4$ denote the four decision nodes of Game 1 of Figure 3, with C playing at n_1 and n_3, and P playing at the remaining two nodes n_2 and n_4. We have four belief operators for this game, namely two per player. We abbreviate some formulas that describe the payoff structure of the game:

$\alpha := \langle d \rangle \langle f \rangle \langle h \rangle ((u_C = p_C) \wedge (u_P = p_P))$
(from the current node, a d move followed by an f move followed by an h move lead to the payoff (p_C, p_P))

$\beta := \langle d \rangle \langle f \rangle \langle g \rangle ((u_C = q_C) \wedge (u_P = q_P))$
(from the current node, a d move followed by an f move followed by a g move lead to the payoff (q_C, q_P))

$\gamma := \langle d \rangle \langle e \rangle ((u_C = r_C) \wedge (u_P = r_P))$
(from the current node, a d move followed by an e move lead to the payoff (r_C, r_P))

$\delta := \langle c \rangle ((u_C = s_C) \wedge (u_P = s_P))$
(from the current node, a c move leads to the payoff (s_C, s_P))

$\chi := \langle b^- \rangle \langle a \rangle ((u_C = t_C) \wedge (u_P = t_P))$
(the current node can be accessed from another node by a b move from where an a move leads to the payoff (t_C, t_P))

Now we can define the conjunction of these five descriptions:

$$\varphi := \alpha \wedge \beta \wedge \gamma \wedge \delta \wedge \chi$$

Let ψ_i denote the conjunction of all the order relations of the rational payoffs for player i ($\in \{P, C\}$) given in Game 1 of Figure 3.
A strategy specification describing the strategic reasoning of Example 1 above at the node n_2 is:

$$\eta_P^1 : [(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root} \wedge \mathbb{B}_{g1}^{n_2, P} \langle d \rangle \neg e \wedge \mathbb{B}_{g1}^{n_2, P} \langle d \rangle \langle f \rangle g) \mapsto d]^P$$

In words: If the payoffs of players at the respective nodes are given by φ and ψ_P and ψ_C are given accordingly, then if player P is at n_2 and believes at that node that after her move d, e will not be played by C, and believes that after her d move and player C's f move she will play g, then P will play d at the current node.

Backward induction reasoning at the same node n_2 can be formulated as follows:

$$\eta_P^2 : [(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root} \wedge \mathbb{B}_{g1}^{n_2, P} \langle d \rangle e \wedge \mathbb{B}_{g1}^{n_2, P} \langle d \rangle \langle f \rangle g) \mapsto c]^P$$

In words: If the payoffs of players at the respective nodes are given by φ and if ψ_P and ψ_C are given accordingly, then if player P is at n_2 and believes at that node that after her move d, e will be played by C, and believes that after her d move and player C's f move, she will play g, then P will play c at the current node.

For a comparison to the experiment described in Section 3, we should add here that for games 1 and 2, about 84% of the players showed similar strategic behavior to what is depicted by the former strategy formula η_P^1 corresponding to game 1, whereas for games 3 and 4, even 97% of the players showed such behavior.

The examples above show how strategic reasoning of participants can be described by a logical formula, which could then be converted to appropriate reasoning rules to construct computational cognitive models (see Section 5). Note that our representations have become quite succinct using the newly added belief operator, compared to the representations in [23], because expressions for response strategies are not needed anymore. Let us look at an example to have an idea of the relative succinctness of the extended language.

To model players' responses in [23], we introduced 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:

- $M, s \models \bar{\imath}?\zeta$ iff $\exists T$ such that $T \in [\![\zeta]\!]_M$ and $s \in T$.

A strategy specification for player P describing her backward reasoning giving the *rational choice* corresponding to the game tree $1'$ given in Figure 4 is:

$\eta : [(C?[(P?[\varphi^0 \mapsto g]^P \wedge \varphi^1) \mapsto e]^C \wedge \varphi^2) \mapsto c]^P$, where:

- $\varphi^0 : \alpha \wedge \beta \wedge \langle d \rangle \langle f \rangle \mathbf{turn}_P \wedge (1 \leqslant 3) \wedge \gamma \wedge \langle d \rangle \mathbf{turn}_C \wedge (0 \leqslant 2) \wedge \mathbf{root} \wedge \mathbf{turn}_P \wedge \delta \wedge (0 \leqslant 2)$

- $\varphi^1 : \alpha \wedge \beta \wedge \langle d \rangle \langle f \rangle \mathbf{turn}_P \wedge (1 \leqslant 3) \wedge \gamma \wedge \langle d \rangle \mathbf{turn}_C \wedge (0 \leqslant 2)$

- $\varphi^2 : \alpha \wedge \beta \wedge \langle d \rangle \langle f \rangle \mathbf{turn}_P \wedge (1 \leqslant 3)$

In words, η says:

> 'If the utilities and the turns of players at the respective nodes are as in Game $1'$ (cf. Figure 4), then player P would play c at the root node, as player C would have played e at his node (had it been reached), and player P would have played g at her node (had it been reached).'

The same strategy specification can be expressed in the current specification language with beliefs as follows:

$\eta : [(\alpha \wedge \beta \wedge \gamma \wedge \delta \wedge (1 \leqslant 3) \wedge (0 \leqslant 2) \wedge \mathbf{root} \wedge \mathbb{B}_{g1'}^{n_1,P} \langle d \rangle e \wedge \mathbb{B}_{g1'}^{n_1,P} \mathbb{B}_{g1'}^{n_2,C} \langle f \rangle g) \mapsto c]^P$.

Notice that this representation is much more succinct and more easily understandable than the corresponding representation η from [23]. We conjecture that in general, the new language is more succinct than the one proposed in [23] in describing strategic reasoning, but we leave this as an intuitive conjecture for now. A detailed formal study of the current extended framework regarding its expressive power and axiomatics is left for future work.

4.2 Describing specific types in the experimental games

In this subsection, we show how to formalize several types of strategic reasoning, both according to the two typologies used in Sections 3.2 and 3.3 and typologies used in the literature.

Theory of mind types We now show how to express the reasoning of players who apply different orders of theory of mind, with the syntax proposed in Section 2.2. Participants who are not familiar with playing turn-taking games such as Marble Drop, may start playing the games according to some simple strategies (cf. [40]). An example of such a simple strategy is to compare the participant's payoff in case of going down immediately and stopping the game with the maximum of all her possible future payoffs in case the game continues. Such a participant stops if the payoff of going down is larger and continues otherwise. Note that such a participant does not attribute mental states such as beliefs or plans to the other agent or herself but merely acts upon some facts, and hence can be considered to be a *zero-order theory of mind* player.

Next, one can consider a more complex strategy to play one of our experimental games: A participant considers what her opponent might play in the next node in case it is reached and plays according to what she thinks about her opponent's mental states, for example, she believes that the opponent is playing according to the simple zero-order strategy described above. Participants who reason in this way can be considered to be *first-order theory of mind* players.

Finally, at a next level of complexity, a participant could consider at her first decision node n_2 that her opponent would believe at the next decision node n_3 that the participant's strategy at the final decision node n_4 would be the simple zero strategy described above. Then the participant's considerations at n_2 would be an example of applying a *second-order theory of mind* strategy.

Note that the way the participants answered the final question (cf. Section 3.3) in the experiment indicated what kind of reasoners they were with respect to theory of mind. We now express theory of mind types of our Marble Drop experiment in the language proposed in Section 2.2. A similar syntax for expressing player types has been proposed in [53].

We use the abbreviated formulas $\alpha, \beta, \gamma, \delta, \chi$ that describe the payoff structure of the game as given above in Section 4.1.

A zero-order theory of mind participant can be described by the following specification:

$$\tau_P^0 : [(\varphi \wedge \psi_P \wedge \langle b^- \rangle \mathbf{root}) \mapsto c]^P$$

In words: If the payoffs of players at the respective nodes are given by φ and ψ_P is given accordingly, then player P will play c at the current node.

A first-order theory of mind participant can be described by the following specification:

$$\tau_P^1 : [(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root} \wedge \mathbb{B}_{g1}^{n_2,P} \langle d \rangle \neg e) \mapsto d]^P$$

In words: If the payoffs of players at the respective nodes are given by φ and ψ_P and ψ_C are given accordingly, then if player P is at n_2, believes at that node that after her move d, e will not be played by C, then P will play c at the current node.

Finally, a second-order theory of mind player can be described by:

$$\tau_P^2 : [(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root} \wedge \mathbb{B}_{g1}^{n_2,P} \langle d \rangle \neg e \wedge \mathbb{B}_{g1}^{n_2,P} \mathbb{B}_{g1}^{n_3,C} \langle f \rangle h) \mapsto d]^P$$

In words: If the payoffs of players at the respective nodes are given by φ and ψ_P and ψ_C are given accordingly, then if player P is at n_2, believes at that node that after her move d, e will not be played by C, and believes that player C believes that after the f move player P will play h, then P will play d at the current node.

Note the subtle differences in the belief expressions between these three theory of mind formulas and the formulas provided in the previous section: The formula τ_P^1 only considers P's belief about C's move at the next node and nothing beyond that (describing a first-order theory of mind participant), whereas the formulas η_P^1 and η_P^2 do consider beliefs about all possible future plays, the way a game theorist would go about strategic reasoning.

Expected, learner and random types We now provide a brief discussion regarding how the type categories found by the latent class analysis in Section 3.2 can be described in a similar way using appropriate temporal representations of the specification formulas. To this end, we introduce a finite set of time-points *Time*, say, and parametrize the specification formulas η with respect to those time-points $t \in Time$, denoted by η_t. The semantic function $[\![\eta_t]\!]_M$ is given by:

$$[\![\eta_t]\!]_M = [\![\eta]\!]_{M_t},$$

where $M_t = (T_t, \{\longrightarrow_{i,t}^x\}, V_t)$ is almost the same as $M = (T, \{\longrightarrow_i^x\}, V)$ with $T_t = T$, and $V_t = V$, the only possible change happening in the set of relations $\{\longrightarrow_{i,t}^x\}$. So, for any given set of time-points *Time* and a model M, we first define the M_t's for $t \in Time$ and then we can interpret the strategy specifications corresponding to those time-points.

As a simple exemplification consider the set of time-points $Time = \{t_1, t_2\}$, Game 1 (cf. Figure 3) where the expected move at the first decision node for player P is d, then the expected, learner and random types can be differentiated by the following pairs of formulas, respectively:

$$\zeta_P^E : ([(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root}) \mapsto d]_{t_1}^P, [(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root}) \mapsto d]_{t_2}^P)$$

In words: If the payoffs of players at the respective nodes are given by φ and ψ_P and ψ_C are given accordingly, then if player P is at n_2, she will play d at that node at both time points t_1 and t_2.

$$\zeta_P^L : ([[(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root}) \mapsto c]_{t_1}^P, [(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root}) \mapsto d]_{t_2}^P)$$

In words: If the payoffs of players at the respective nodes are given by φ and ψ_P and ψ_C are given accordingly, then if player P is at n_2, she will play c at that node at time-point t_1, and d at time-point t_2.

$$\zeta_P^R : ([[(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root}) \mapsto d]_{t_1}^P, [(\varphi \wedge \psi_P \wedge \psi_C \wedge \langle b^- \rangle \mathbf{root}) \mapsto c]_{t_2}^P)$$

In words: If the payoffs of players at the respective nodes are given by φ and ψ_P and ψ_C are given accordingly, then if player P is at n_2, she will play d at that node at time-point t_1, and c at time-point t_2.

Note that for separating these classes of participants we had to take the individual rounds of the games under consideration, and hence we had tuples of specification formulas indicating the different time-points where the strategies are played. This formalization suffices to list out the possibilities of typologies which could be used as a controlling factor in the build-up of computational cognitive models.

In the experiment described in Section 3, the participants had to make their decisions at 6 rounds of each of the games 1, 2, 3 and 4, and hence to model the strategies we need to consider the set *Time* with 6 different time points, and we could describe Expected players as those playing d in the last 5 time-points, Learners as those playing d in the last 3 time-points, and the others as Random players.

Formalizing other player types from the literature We end this section showcasing some other simple player types which describe players with different kinds of restrained reasoning capabilities. Note that such limited reasoning is ubiquitous in our daily life (see e.g. [29,40]). A *myopic* (or near-sighted) player can be considered as one who only considers her current node and the next one to compare her payoffs and act rationally depending on those payoffs without being able to look further into the game (cf. [29]). Such a player-strategy can be described for games $1'$ and $3'$ as follows:

$$\kappa_P^{1'} : [(\delta_{1'} \wedge \gamma_{1'} \wedge (0 \leqslant 2) \wedge \mathbf{root}) \mapsto c]^P$$

$$\kappa_P^{3'} : [(\delta_{3'} \wedge \gamma_{3'} \wedge (2 \leqslant 3) \wedge \mathbf{root}) \mapsto c]^P$$

One can also consider players who are only capable or interested to look at their own payoffs and do not consider the opponent's payoffs at all and move wherever they get more payoff (cf. [52]). Their strategies in games $1'$ and $3'$ can be described as follows:

$$\chi_P^{1'} : [(\alpha_{1'} \wedge \beta_{1'} \wedge \delta_{1'} \wedge \gamma_{1'} \wedge (0 \leqslant 2) \wedge (2 \leqslant 3) \wedge (1 \leqslant 2) \wedge \mathbf{root}) \mapsto d]^P$$

$$\chi_P^{3'} : [(\alpha_{3'} \wedge \beta_{3'} \wedge \delta_{3'} \wedge \gamma_{3'} \wedge (2 \leqslant 3) \wedge (3 \leqslant 4) \wedge \mathbf{root}) \mapsto d]^P$$

Note that in the above set of formulas, we only consider the relevant pay-offs, e.g. δ and γ in case of the κ formulas, and α, β, δ, and γ for the χ formulas. In fact, one could ignore the payoffs for C for the χ formulas. We will come back to these strategies in the next section when we validate the model predictions with the experimental results.

5 Modelling strategic reasoning processes in a cognitive architecture

Our aim in this section is to sketch a way how some of the strategy descriptions that we formulated in the logical strategic language in Subsections 4 can be translated in a straightforward way into computational cognitive models in the state-of-the art cognitive architecture PRIM, which is based on ACT-R. The upshot of coupling our strategy logic to PRIM is that PRIM, through its association with ACT-R, 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 [2] and playing the game of SET [45] to driving cars [27]. Thus, there is no need to add possibly arbitrary resource bounds in the logical language.

We start with providing a brief description of the cognitive architectures at the basis of our computational cognitive model, ACT-R and PRIMs and of previous computational cognitive models of Marble Drop based on ACT-R. Then in Subsection 5.4 we translate a number of the strategies that were represented by strategy formulas in the previous section into PRIMs models – both strategies well-known from game theory such as backward induction, followed by reasoning formulas corresponding to the different players' typologies, such as the one based on theory of mind. Finally, we come full circle and we compare the simulation results of two PRIMs models with actual participant data, to show that participants probably do *not* apply the reasoning strategy that Hedden and Zhang [29] called "myopic" (near-sighted).

5.1 ACT-R

ACT-R is an integrated theory of cognition as well as a cognitive architecture that many cognitive scientists use [2]. It consists of modules that link with cognitive functions, for example, vision, motor processing, and declarative processing. Each module is associated with a buffer and the modules communicate via these buffers. Importantly, cognitive resources are bounded in ACT-R models: Each buffer can store just one piece of information at a time.

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 a 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. Therefore, a chunk representing a comparison between two payoffs will have a higher probability of retrieval, and will be retrieved faster, if the comparison has been made *recently*, or *frequently* in the past [1,2]. As soon as a chunk is retrieved from declarative memory, it is placed into the declarative module's buffer.

The *problem state* module also contains a buffer that can hold one chunk. Typically, the problem state stores a sub-solution to the problem at hand. In the case of a social reasoning task, this may be the outcome of a reasoning step that will be relevant in subsequent reasoning. Storing information in the problem state buffer is associated with a time cost (typically 200ms).

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. Production rules have so-called utility values. The model receives reward or punishment depending on the correctness of its response. Both reward and punishment propagate back to previously fired production rules, and the utility values of these production rules are increased in case of reward and decreased in case of punishment by a process called *utility learning* [2]. If two or more production rules match a particular game state, the production rule with the highest utility is selected.

5.2 PRIMs

PRIM, the *primitive elements theory*, is a recent cognitive theory developed by Taatgen, who implemented it in the computational cognitive architecture PRIMs [58]. It builds on ACT-R, using ACT-R modules, buffers and mechanisms such as production compilation. However, in contrast to ACT-R, PRIMs is suited for modeling general reasoning strategies that are not included in the basic cognitive architecture shared by all humans, but that are at the same time more general than *ad hoc* task-specific reasoning rules. Thereby, PRIMs is especially suitable for modeling the nature and transfer of cognitive skills. Because of our need to model participants' beliefs about the opponent's beliefs, we decided to use PRIMs as cognitive architecture to model more sophisticated reasoning strategies rather than ACT-R, which we used in [23].

More specifically, PRIM breaks down the complex production rules typically used in ACT-R models into the smallest possible elements (PRIMs) that move, compare or copy information between modules (cf. Figure 7). There is a fixed number of PRIMs in the architecture. When PRIMs are used often over time, production compilation combines them to form more complex production rules. While those PRIMs may have some task-specific elements, PRIMs also have task-general elements that can be used by other tasks. Taatgen [58,59] showed the predictive power of PRIMs by modeling a variety of transfer experiments such
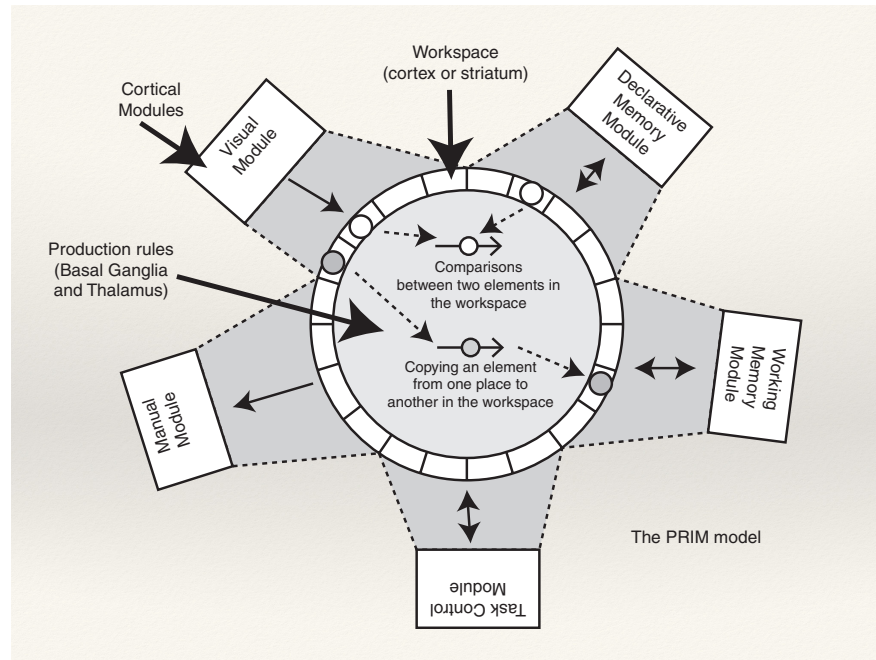
**Fig. 7.** Schema of a PRIM model as represented in [58]

as text editing, arithmetic, and cognitive control. The architecture has been used to model children's development of theory of mind [4], transfer between the 'take the best' heuristic and the balance beam task [25], and children's mistakes in arithmetic [16]. PRIMs models can be run to predict the estimated time to complete certain tasks, which we will use in Section 5.4 to fit the predictions of our PRIMs models of reasoning strategies in "Marble Drop with Surprising Opponent".

Like ACT-R, PRIMs models cognitive resources as being bounded: 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*, representing long-term memory, and the *problem state*, in which a small chunk of information can be stored for a short time. Moving information back and forth comes with a time cost, in some cases causing a cognitive bottleneck [14].

5.3 Earlier models of strategic reasoning in Marble Drop

Van Maanen and Verbrugge [36] proposed an ACT-R model that follows a backward reasoning strategy to predict the opponent's moves further on in a game of "Marble Drop with Rational Opponent" against a computer opponent that

was known to be rational. The drawback of this model is that it implements just one reasoning strategy, while the results in [43,12] show that participants used several reasoning strategies. There have been two follow-up ACT-R models to remedy this problem. Meijering and colleagues [40] have implemented the idea that players use negative feedback in order to move from an overly simple reasoning strategy without theory of mind to a more complicated second-order theory of mind strategy only if it is really needed.
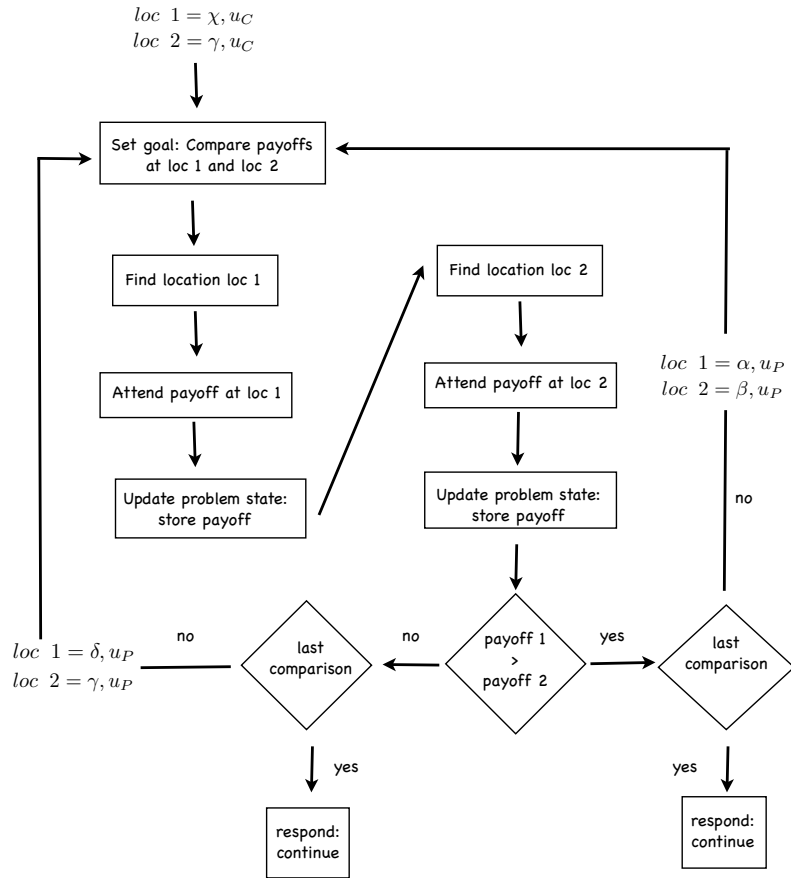
Fig. 8. Flowchart for reasoning processes as described in Example 1, constructed from formula η_P^1 of Section 4.1

Ghosh, Meijering and Verbrugge [23] constructed a more generic model that is able to fit a broader spectrum of possible strategies than [36,40]. It relies on the declarative memory and the problem state, by retrieving relevant infor-

mation from declarative memory and moving that information to the problem state buffer whenever it requests the declarative module to retrieve new information.The PRIMs models that we present in the next subsection are based on similar ideas, but they can also incorporate reasoning about beliefs of opponents.

5.4 Modeling reasoning strategies in PRIMs

We consider a class of models, where each model is based on a set of strategy and type specifications that can be generated using the logical framework we presented in Section 2. As explained in Sections 4.1 and 4.2, both backward induction reasoning and forward induction reasoning (in particular, *EFR* reasoning), as well as other types of reasoning can be represented using logical specifications.

Modeling specific strategies from Section 4.1 in PRIMs Each of the specifications defined in Section 4.1 comprises comparisons between relevant payoffs for both players. For each comparison, a cognitive model has a set of production rules that specify what the model should do. To compare player C's payoffs, say at two leaf nodes, the model first has to find, attend, and encode them in the problem state. For each subsequent payoff, the model performs the following procedure (cf. Figure 8):

- request the visual module to find the visual location of the payoff (cf. [45]);
- direct visual attention to that location; and
- update the problem state (buffer).

The specifications η_P^1 (corresponding to the choices of the vast majority of participants, see Example 1 in Section 4.1) and η_P^2 (corresponding to backward induction, see Section 4.1) specify what the model should do after encoding the payoffs in the problem state. First, the payoffs need to be compared and the comparison needs to stored. Then the belief operators are dealt with as follows (cf. Figure 9):

- attend to the visual location of the node depicted by the belief operator; and
- encode the actions and beliefs at the problem state (buffer).

These beliefs can be taken care of in the PRIMs model in the same way as in [3], namely some n-th order strategy chunk can be created in the declarative memory for an n-th order belief in the strategy / type formulas followed by creating an $(n-1)$-th order chunk for the $(n-1)$-th order belief. This process can be continued until the model creates a zero-order chunk corresponding to a zero-order belief. For each n, the model would keep a reference to the $(n-1)$-th order chunk in the declarative memory, which in turn would have a pointer towards the n-th order chunk. The stored beliefs are retrieved accordingly in the problem state buffer and production rules are fired depending on the retrieval in order to make decisions.

The decisions are made corresponding to the recorded payoffs and the resulting beliefs. An example production rule could be as follows; the model will select and fire this production rule to generate a response:
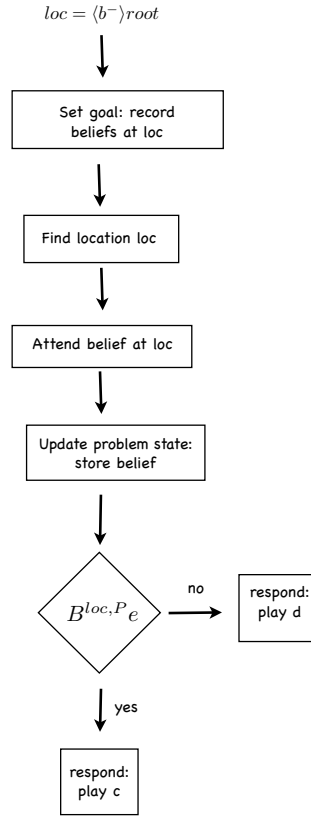
Fig. 9. Flowchart for reasoning processes utilized in backward induction, constructed from formula η_P^2 of Section 4.1

|	
IF	
Goal is to record Player P's belief at node n	If the current goal is to record Player P's beliefs at node n,
Problem State represents Player P's actions at n, c and d	and the problem state has stored the actions,
$\mathbb{B}^{(P,n)} f$	and the belief is that f will be played (by C),
THEN	
Decision is play d	then request the manual (or motor) module to produce a key press (i.e., play d).

Modeling specific player types from Section 4.2 in PRIMs Based on the same syntax as used for the strategies, one can model the player types, for example, according to levels of theory of mind or according to the latent class analysis.
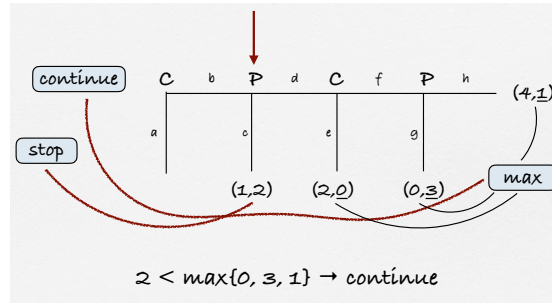
Fig. 10. Representing the simple zero-order theory of mind strategy from the player P's perspective playing at the second node. The model will compare P's payoff in case it stops with her maximum possible payoff in case she continues. This corresponds to zero-order theory of mind reasoning represented by formula τ_P^0 of Section 4.2.

One can add different assumptions to the model with regard to the strategies being used, the roles of players, whether they are considering the roles of their opponents, and also with regard to the beliefs players have regarding opponents' moves and strategies. Figure 10 shows a schematic representations of the reasoning processes of a model performing zero-order theory of mind reasoning from the viewpoint of the participant P. One level of complexity higher, Figure 11 shows a schematic representation of the model attributing zero-order reasoning to player C from the viewpoint of P, who is thereby applying first-order theory of mind. Similarly, one could use different models with regard to different time-points in *Time* based on the different specification formulas as given by the tuples of such formulas in Section 4.2 to deal with the Expected, Learner and Random types of players. As with the strategy formulas, the type formulas
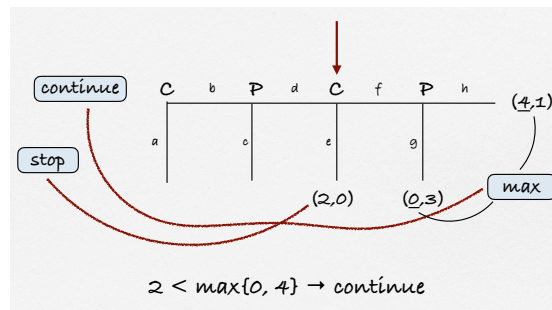
Fig. 11. Representing the simple first-order theory of mind strategy from the player C's perspective playing at the third node. The model will compare C's payoff in case it stops with her maximum possible payoff in case she continues. This corresponds to attributing zero-order theory of mind reasoning to player C by player P, who thereby performs first-order theory of mind reasoning, as represented by formula τ_P^1 of Section 4.2.

can be implemented in production rules in the cognitive architecture PRIMs. Such production rules can determine, for example, what the payoff would be when going down immediately and stopping the game, what the maximum of all P's possible future payoffs could be in case she continues the game, and which beliefs influence which moves. The production rules are generally executed from the perspective of the player who is currently deciding which course of action to follow. Thus, the Learner types that have been captured by tuples of formulas in Section 4.2, can be transformed into production rules of a tuple of models to simulate the behavior of such Learner types of players. Another kind of learning, namely the move from simpler to more complex theory of mind levels, can be reflected in PRIMs models as follows (inspired by, but implemented differently than [40]). The model attributes a player's moves and beliefs from the perspective of the current decision node to the opponent operating at the next decision point, stepping into the opponent's shoes, and while doing this, the model updates its belief levels. Subsequently, the model acts in its heightened order of theory of mind reasoning.

On the whole, the strategy and the type specification formulas can be used to construct various PRIMs models to simulate behaviors of players involved in varying kinds of strategic reasoning, belonging to various type categories. Based on the validation of the predictions of such models with respect to the experimental results (cf. Section 3), one can narrow down the set of reasoning formulas and type formulas that provide apt descriptions of human strategic reasoning and typologies. From another perspective, these specification formulas act as controlling factors for suggesting the production rules in different PRIMs models, providing a formal basis of the algorithms used to construct the models. In other words, rather than having some ad-hoc production rules for the PRIMs model, one could be guided by the logical formulas in formulating rules leading to different PRIMs models that correspond to different strategies and types of players. The models can then be compared with each other in terms of decisions and reaction times with respect to the validations of their predictions, in order to provide better modelling of human strategic reasoning and typologies.

5.5 Validating some reasoning types modeled in PRIMs against experimental results

We have seen in section 4.1 that some fitting of formal strategies to experimental data can be done based directly on the logical strategy formula; for example, it turns out that more than 84% of participants made decisions according to Example 1, formalized as formula η_P^1. However, in order to be able to use more of the participant data, such as their reaction times for making their first decision, the formulas do not suffice but a PRIMS model corresponding to one or more strategy formulas can be constructed and run a number of times, as if the models are virtual participants that perform the game experiment.

As a test case, we have constructed PRIMs models based on the specification formulas corresponding to two relatively simple player types inspired by the literature: *myopic* players (κ_P formulas of Section 4.2, inspired by [29]), and *own payoff* players (χ_P formulas of Section 4.2, inspired by [52]). The models were
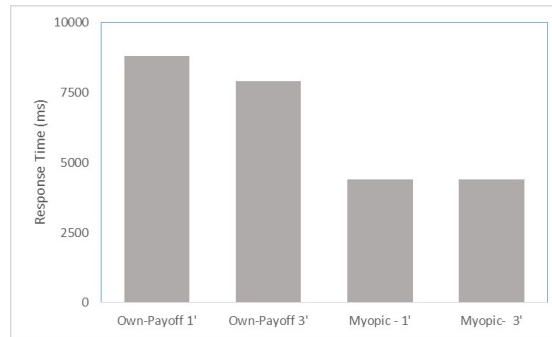
Fig. 12. Reaction time predictions in milliseconds from the PRIMs models for games $1'$ and $3'$, from left to right corresponding to the formulas $\chi_P^{1'}$ and $\chi_P^{3'}$ (*own payoff* strategy) and the formulas $\kappa_P^{1'}$ and $\kappa_P^{3'}$ (*myopic* strategy)
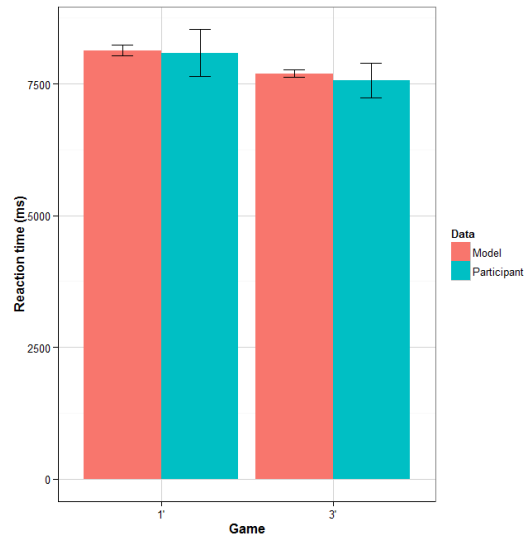
Fig. 13. Comparison of reaction times in milliseconds between the predictions of the *own payoff* strategy model and the participants, for game $1'$ (left two bars) and game $3'$ (right two bars). The red bars represent the model predictions, while the blue bars represent the mean reaction times corresponding to participants' choices that were consistent with the *own payoff* strategy. Error bars represent standard deviations.

constructed following the general translation procedures described in the previous subsection and can be found at `http://www.ai.rug.nl/SocialCognition/experiments/`. In our simulations, both models were run 50 times (corresponding to 50 "virtual participants" each), playing 50 rounds each for the games $1'$ and $3'$ of Figure 4. The reaction time predictions obtained from the models are given in Figure 12.

As can be seen in Figure 12, the "virtual participants" who use the *own payoff* strategy (based on the formulas $\chi_P^{1'}$ and $\chi_P^{3'}$ of Section 4.2), need on average more time for their first decision in game $1'$ (more than 800 milliseconds) than in Game $3'$ (around 7500 milliseconds).

The "virtual participants" who use the *myopic* strategy (based on the formulas $\kappa_P^{1'}$ and $\kappa_P^{3'}$ of Section 4.2), in contrast, need about the same amount of time for their decisions in both games $1'$ and $3'$ (both around 4000 ms), and moreover, this is much less than the mean time needed for the "virtual participant" that uses the *own payoff* strategy.

The reaction times for the PRIMs model corresponding to the *own payoff* types and those of the *myopic* types were fitted against those of the participants in the experiment described in Section 3. It turns out that the participants' reaction times fit well with the *own payoff* model predictions for two reasons. Qualitatively, as Figure 13 shows, participants were slower in their decisions on Game $1'$ than they were on Game $3'$, just like the "virtual participants" that use the *own payoff* strategy.[8] More quantitatively, the reaction times for the real participants in Game $1'$ (more than 8000 milliseconds) and for Game $3'$ (around 7500 milliseconds) are quite similar to those of the virtual ones.

In contrast, the findings from the PRIMs model corresponding to the *myopic* types (based on the formulas $\kappa_P^{1'}$ and $\kappa_P^{3'}$ of Section 4.2) do not fit the reaction time data at all: in general, the real participants use much more time (mean around at least 7000) than the "virtual participant" who uses the myopic strategy does (mean around 4000 ms).

A great advantage of computational cognitive models in an architecture such as PRIMs is that one can also make predictions for future experiments. We will make one such prediction now. Currently, together with Aviad Heifetz and Eric Jansen, we are in the midst of a set of experiments in The Netherlands, India and Israel, based on games that are variations of those of the experiments of [22], with the same centipede-like trees as those in Figures 3 and 4 but different payoff structures. In particular, the new truncated game $1''$ corresponding to game $1'$ of the current paper has new payoffs $(1, 2)$ after c, $(3, 1)$ after e, $(1, 4)$ after g, and $(6, 3)$ after h; and the new truncated game $3''$ corresponding to game $3'$ of this paper has payoffs $(1, 2)$ after c, $(3, 1)$ after e, $(1, 4)$ after g, and $(6, 4)$ after h. We predict that also for these games, participants whose choices fit the *own payoff* strategy as well as the *myopic* one, are still more likely to reason following the *own payoff* strategy, as shown by their reaction times: we predict that they will be slower on Game $1''$ than on Game $3''$.

6 Conclusions and future work

---

[8] Note that to construct Figure 13, we considered those instantiations of the games in which participants' choices were consistent with the choice prescribed in the relevant χ_P. To account for individual differences between participants, only those participants were considered for whom the majority of choices in games $1'$ and $3'$ were consistent with the specification formula $\chi_P^{1'}$ and $\chi_P^{3'}$, respectively.

In this paper we have explored the question "How do people really reason about their opponent in turn-taking games?" for specific turn-taking games of the type "Marble Drop with Surprising Opponent", in which the opponent often started with a seemingly irrational move. We began with presenting a logical language that expresses different kinds of strategies which people can apply when reasoning about their opponent and making decisions in turn-taking games such as "Marble Drop". It can also express different possible 'reasoning types' reflected in participants' behavior. The new logical language extends our earlier strategy language of [23] with (higher-order) beliefs. The extended language results in more user-friendly and more concise formulas than the earlier one; this is an advantage because it makes the formulas more understandable for cognitive modelers who are not logicians.

We then explored the data of our earlier experiment with Heifetz about the games that was presented first in [22]. In the current article, we moved beyond the question whether participants in general use forward induction reasoning and instead first explored two ways of segregating the participants into groups to see whether and how they can be divided into reasonable "player types". The first way to construct a typology was based on latent class analysis, which turned out to divide the players into three classes according to their first decisions in the game: Random players, Learners, and finally Expected players who make decisions consistent with forward induction. This typology appeared to be reasonable, because the three levels correspond with increasing gains in the games and with increasing time spent on decisions. The second way of constructing a typology was based on the participants' answers to a question about their opponent, classified according to levels of theory of mind: the resulting types are Zero-order, First-order and Second-order. This typology was also validated by increasing levels of theory of mind turning out to correspond to increasing monetary awards and increasing decision times. The logical language was then used to describe different reasoning strategies and reasoning types that were displayed by the participants during the experiment, including the types discussed previously.

We mainly aimed for contributions on Marr's computational and algorithmic levels and the interplay between them through this study based on logic, experiment and computational cognitive model. The logical language helped us delineate a number of plausible reasoning strategies in a systematic manner. In general it is possible to translate such logical formulas into computational models in the computational cognitive architecture PRIMs, and this can be done in a generic way, enabling the construction of a corresponding set of computational models in PRIMs. More specifically, the formulas are implemented as production rules, which handle visual processing, problem state updates, and motor processing. We have shown how such translation works for two specific reasoning strategies and we have run the computational models and made predictions from the simulations about the data. It turned out that the predictions of one of the models, corresponding to the *own payoff* strategy, fit the actual participants'

data in terms of their response times remarkably well. We have also formulated a model-based prediction for a future experiment.

All in all, we have shown that logic makes a contribution at Marr's computational level by providing a precise specification language for cognitive processes. Moreover, we have illustrated that 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 PRIMs. The great advantage of using the cognitive architecture PRIMs rather than an *ad hoc* computational model, is that it already implements very precise, experimentally validated theories about human memory and cognitive bounds on reasoning processes. In comparison to ACT-R, PRIMs appears to be easier for logicians to learn.

Future work We aim to implement various sets of specifications of reasoning strategies in separate models, inspired by the 39-model study of [37]. The aim is to simulate repeated game play, both to determine which participants in a new experiment most closely belong to which player types, as well as to study possible learning effects. An advantage of constructing PRIMs models, not only logical formulas, is that quantitative predictions are generated, for example, concerning decision times and locus of attention, which can then be tested in further experiments, for example, using an eye-tracker.

Now that we have models in PRIMs, we can also make specific predictions for training experiments, e.g. training people with second-order false belief tasks or complex working memory tasks and investigating whether that helps them to transfer these skills to "Marble drop with Surprising Opponent".

From the logical perspective, the next step will be to provide a sound and complete axiom system for strategic reasoning that models empirical human reasoning in dynamic games of perfect information, including reasoning about the higher-order beliefs of the opponent.

Acknowledgments

References

1. J. R. Anderson and L. J. Schooler. Reflections of the environment in memory. *Psychological Science*, 2(6):396–408, 1991.
2. J.R. Anderson. *How Can the Human Mind Occur in the Physical Universe?* OUP, 2007.
3. B. Arslan, N. Taatgen, and R. Verbrugge. Modeling developmental transitions in reasoning about false beliefs of others. In *Proceedings of the 12th International Conference on Cognitive Modeling, Ottawa: Carleton University*, pages 77–82, 2013.
4. B. Arslan, S. Wierda, N. Taatgen, and R. Verbrugge. The role of simple and complex working memory strategies in the development of first-order false belief reasoning: A computational model of transfer of skills. In *Proceedings of the 13th International Conference on Cognitive Modeling*, pages 100–105, 2015.
5. G. Baggio, M. van Lambalgen, and P. Hagoort. Logic as Marr's computational level: Four case studies. *Topics in Cognitive Science*, 7(2):287–298, 2015.
6. C. Bateman, R. Lowenhaupt, and L. E. Nacke. Player typology in theory and practice. In *Proceedings of DiGRA*, Utrecht, 2011. Utrecht School of the Arts.
7. P. Battigalli. Strategic rationality orderings and the best rationalizability principle. *Games and Economic Behavior*, 13:178–200, 1996.
8. P. Battigalli. On rationalizability in extensive games. *Journal of Economic Theory*, 74:40–61, 1997.
9. J. van Benthem. Logic and reasoning: Do the facts matter? *Studia Logica*, 88:67–84, 2008. Special issue on logic and the new psychologism, edited by H.Leitgeb.
10. J. van Benthem, S. Ghosh, and F. Liu. Modelling simultaneous games with dynamic logic. *Knowledge, Rationality and Action*, 165:247–268, 2008.
11. J. van Benthem, S. Ghosh, and R. Verbrugge, editors. *Models of Strategic Reasoning: Logics, Games and Communities*, volume 8972 of *LNCS-FoLLI*. Springer, New York, NY, 2015.
12. G. Bergwerff, B. Meijering, J. Szymanik, R. Verbrugge, and S. Wierda. Computational and algorithmic models of strategies in turn-based games. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pages 1778–1783, 2014.
13. G. Bonanno. Axiomatic characterization of the AGM theory of belief revision in a temporal logic. *Artificial Intelligence*, 171(2-3):144–160, 2007.
14. J. Borst, N. Taatgen, and H. van Rijn. The problem state: A cognitive bottleneck in multitasking. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 36(2):363–382, 2010.
15. T. Braüner, P. R. Blackburn, and I. Polyanskaya. Second-order false-belief tasks: Analysis and formalization. In J. Väänänen, Å. Hirvonen, and R. de Queiroz, editors, *Logic, Language, Information, and Computation: Proceedings of WOLLIC 2016*. Springer, 2016.
16. T.A. Buwalda, J.P. Borst, H. van der Maas, and N.A. Taatgen. Explaining mistakes in single digit multiplication: A cognitive model. In *Proceedings of the 14th International Conference on Cognitive Modeling*, pages 131–136, 2016.
17. C. Camerer. *Behavioral Game Theory*. Princeton University Press, 2003.
18. J. Chen and S. Micali. The robustness of extensive-form rationalizability. Working paper, 2011.
19. J. Chen and S. Micali. The order independence of iterated dominance in extensive games. *Theoretical Economics*, 8:125–163, 2013.

20. R.P. Cooper and D. Peebles. Beyond single-level accounts: The role of cognitive architectures in cognitive scientific explanation. *Topics in Cognitive Science*, 7(2):243–258, 2015.

21. S. Ghosh, T. Halder, K. Sharma, and R. Verbrugge. Human strategic reasoning in dynamic games: Experiments, logics and cognitive models. In Wiebe van der Hoek, Wesley H. Holliday, and Wen fan Wang, editors, *LORI 2015*, pages 116 –128, 2015.

22. S. Ghosh, A. Heifetz, and R. Verbrugge. Do players reason by forward induction in dynamic perfect information games? In R. Ramanujam, editor, *Proceedings of the 15th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 121–130, 2015.

23. S. Ghosh, B. Meijering, and R. Verbrugge. Strategic reasoning: Building cognitive models from logical formulas. *Journal of Logic, Language and Information*, 23(1):1–29, 2014.

24. S. Ghosh and R. Ramanujam. Strategies in games: A logic-automata study. In N. Bezanishvili and V. Goranko, editors, *Lectures on Logic and Computation - ESSLLI 2010, ESSLLI 2011, Sel. Lecture Notes*, pages 110–159. Springer, 2012.

25. L. Gittelson and N.A. Taatgen. Transferring primitive elements of skill within and between tasks. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*, pages 529–534. Cognitive Science Society, 2014.

26. L. A. Goodman. The analysis of systems of qualitative variables when some of the variables are unobservable. Part IA: modified latent structure approach. *American Journal of Sociology*, pages 1179–1259, 1974.

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

28. T. Halder, K. Sharma, S. Ghosh, and R. Verbrugge. How do adults reason about their opponent? Typologies of players in turn-taking games. In *Proceedings of the 37th Annual Conference of the Cognitive Science Society*, pages 854–859, 2015.

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

30. A. Heifetz and A. Perea. On the outcome equivalence of backward induction and extensive form rationalizability. *International Journal of Game Theory*, 2014. forthcoming.

31. A.M.C. Isaac, J. Szymanik, and R. Verbrugge. Logic and complexity in cognitive science. In *Johan van Benthem on Logic and Information Dynamics*, pages 787–824. Springer, 2014.

32. I. Juvina and N. A. Taatgen. Modeling control strategies in the N-back task. In *Proceedings of the 8th International Conference on Cognitive Modeling*, pages 73–78, New York, NY, 2007. Psychology Press.

33. T. Kawagoe and H. Takizawa. Level-k analysis of experimental centipede games. *Journal Of Economic Behavior and Organization*, 82(2):548–566, 2012.

34. M. van Lambalgen and M. Counihan. Formal models for real people. *Journal of Logic, Language and Information*, 17:385–389, 2008. Special issue on formal models for real people, edited by M. Counihan.

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

36. L. van Maanen and R. Verbrugge. A computational model of second-order social reasoning. In *Proceedings of the 10th International Conference on Cognitive Modeling*, pages 259–264, 2010.

37. J. N. Marewski and K. Mehlhorn. Using the ACT-R architecture to specify 39 quantitative process models of decision making. *Judgment and Decision Making*, 6(6):439, 2011.

38. D. Marr. *Vision*. Freeman and Company, New York, 1982.

39. R. D. McKelvey and T. R. Palfrey. An experimental study of the centipede game. *Econometrica: Journal of the Econometric Society*, pages 803–836, 1992.

40. B. Meijering, N. A . Taatgen, H. van Rijn, and R. Verbrugge. Modeling inference of mental states: As simple as possible, as complex as necessary. *Interaction Studies*, 15(3):455–477, 2014.

41. 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 Conference of the Cognitive Science Society*, pages 1423–1428, 2010.

42. B. Meijering, H. van Rijn, N. A. Taatgen, and R. Verbrugge. I do know what you think I think: Second-order theory of mind in strategic games is not that difficult. In *Proc. 33rd Annual Conf. Cognitive Science Society*, pages 2486–2491, 2011.

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

44. R. Nagel and F. F. Tang. Experimental results on the centipede game in normal form: An investigation on learning. *Journal of Mathematical Psychology*, 42(2):356–384, 1998.

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

46. M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

47. E. Pacuit. Dynamic models of rational deliberation in games. In Benthem et al. [11], pages 3–33.

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

49. A. Perea. *Epistemic Game Theory: Reasoning and Choice*. CUP, 2012.

50. A. Perea. Finite reasoning procedures for dynamic games. In Benthem et al. [11], pages 63–92.

51. G. Polya. *How to Solve It. A New Aspect of Mathematical Method*. Princeton University Press, Princeton, NJ, 1945.

52. M. Raijmakers, D.J. Mandell, S.E. van Es, and M. Counihan. Children's strategy use when playing strategic games. *Synthese*, 191(3):355–370, 2014.

53. R. Ramanujam. Logical player types for a theory of play. In A. Baltag and S. Smets, editors, *Johan van Benthem on Logic and Information Dynamics*, pages 509–528. Springer, 2014.

54. R. Ramanujam and S. Simon. A logical structure for strategies. In *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, volume 3 of *Texts in Logic and Games*, pages 183–208. Amsterdam University Press, 2008.

55. R. Rosenthal. Games of perfect information, predatory pricing, and the chain store. *Journal of Economic Theory*, 25(1):92–100, 1981.

56. A. Rubinstein. Response time and decision making: An experimental study. *Judgment and Decision Making*, 8(5):540–551, 2013.

57. A. Rubinstein. A typology of players: Between instinctive and contemplative. *Quarterly Journal of Economics*, 2016, to appear.

58. N. A. Taatgen. The nature and transfer of cognitive skills. *Psychological Review*, 120(3):439–471, 2013.

59. N. A. Taatgen. Between architecture and model: Strategies for cognitive control. *Biologically Inspired Cognitive Architectures*, 8:132 – 139, 2014.

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

61. R.M. Willems. Re-appreciating the why of cognition: 35 years after Marr and Poggio. *Frontiers in Psychology*, 2:244, 2011.

Appendix A: Instruction sheet

The following instruction sheet was given to each participant of the experiment presented in Section 2 and was explained to them by the experimenter.

- In this task, you will be playing two-player games. The computer is the other player.
- In each game, a marble is about to drop, and both you and the computer determine its path by controlling the orange and the blue trapdoors.
- You control the orange trapdoors, and the computer controls the blue trapdoors.
- Your goal is that the marble drops into the bin with as many orange marbles as possible. The computer's goal is that the marble drops into the bin with as many blue marbles as possible.
- Click on the left trapdoor if you want the marble to go left, and on the right trapdoor if you want the marble to go right.
- How does the computer reason in each particular game?

    The computer thinks that you already have a plan for that game, and it plays the best response to the plan it thinks that you have for that game.

    However, the computer does not learn from previous games and does not take into account your choices during the previous games.
- The first 14 games are practice games. At the end of each practice game, you will see how many marbles you gained in that game, and also the total number of marbles you have gained so far.
- The practice games are followed by 48 experiment games. At the beginning of the experiment games, the total number of marbles won will be set at 0 again. At the end of each experiment game, you will see how many marbles you gained in that game, and also the total number of marbles you have gained so far.
- You will be able to start each game by clicking on the "START GAME" button, and move to the next game by clicking on the "NEXT" button.
- At some points during the experiment phase, you will be asked a few questions regarding what guided your choices.
- There will be a break of 5 minutes once you finish 24 of the 48 experiment games.
- The money you will earn is between 10 and 15 euros and depends on how many marbles you have won during the experiment phase. You will get 10 euros for participation, and each marble you win will add 4 cents to your amount. The final amount will be given to you rounded off to the nearest 5 cents mark.