Strategizing: A meeting of methods

Sujata Ghosh

Indian Statistical Institute, Chennai
E-mail: sujata@isichennai.res.in

This paper discuses a bridging technique of the different perspectives on modeling strategic reasoning, namely, experiments, logics and computational cognitive models. Empirical studies describe human strategic behavior. Logical studies on one hand facilitate the study of properties of such reasoning processes, on the other hand pave the way for implementation through formal languages. Computational cognitive models explore the essence of cognitive functionalities in the realm of strategic reasoning. Bridging these methodologies bring out a fresh perspective in terms of integrating different aspects of strategizing under one common standpoint.

Keywords: Strategic reasoning; Experiments; Logic; Computational cognitive models

1. Introduction

Strategies are everywhere, not just in ‘real’ games like chess and bridge, but also in many social interactions in daily life: How can you create a win-win solution in *negotiations* in contexts as local as the sale of your house, or as global as an international treaty aimed at fighting global warming? How do you decide as a new company entering the market whether to give your articles a high price or a low one, reckoning with the strategy of the existing *competing* company? How do you decide to which place in New York to go and try to meet your friend, when only the meeting time and city have been *coordinated* in advance, and you cannot communicate? Unlike when playing a chess game, we usually do not know explicitly all the relevant details, leading to strategizing under partial information. Choosing between possible meeting points where a friend might show up, pricing of commodities without knowing the rival prices, or divulging fallback positions in negotiations without knowing its effects are natural examples.

1.1. *Modeling strategic reasoning in various fields*

Because strategies play out in so many different areas of life, the study of strategies has become an integral part of many areas of science: game theory itself, which is usually viewed as part of economics; ethics and social philosophy; the study of multi-agent systems in computer science; evolutionary game theory in biology; strategic reasoning in cognitive science; and the study of meaning in linguistics. There are various signs of interdisciplinary cooperation between these fields based on plausible viewpoints on the basic similarities between the perspectives on strategies. Let us first briefly describe some of the different perspectives on strategies.

Game theory: In this area of economics, strategies and their dependence on information form the main topics of study. One of the main focus of study is on strategies bringing about equilibrium play in conflicting as well as cooperating situations. Many such important concepts, such as Nash equilibrium, sub-game perfect equilibrium, sequential equilibrium and extensions and modifications thereof have been developed over the years by the game theorists while modelling various situations as games. In these games the players are considered with varied information content, e.g. perfect, imperfect, incomplete and their strategies are modelled accordingly. Such concepts have been developed by game theorists [1] and adopted by the other areas. More recently, work on psychological game theory [2] has gained momentum which considers the effect of human emotions on their strategies. This line of work often provides justifiable solutions to problems that come up with the orthodox assumptions of rationality on the players as cited by experimental studies on games. [3,4]

Mathematics: In mathematics, and in particular in set theory, game theory has received a warm reception. Set theorists have investigated infinite games, focusing mainly on the question of *determinacy*: does a particular game have a winning strategy for one of the players? This question has important repercussions for descriptive set theory. [5] The *axiom of determinacy* (AD) which states that 'a certain type of two player perfect information infinite games is determined, that is, one of the two players will always have a winning strategy' is inconsistent with the *axiom of choice* (AC), but certain related set of axioms which are actually consistent with AC have led to interesting results in terms of provability inside ZFC (Zermelo-Fraenkel set of axioms with AC) and more recently, in terms of the relationship to large cardinal axioms. [6]

Philosophy: David Hume (1740) was probably the first person to mention the role of *mutual knowledge* in *coordination* in his account of convention in *A treatise of Human Nature*. He argued that without the required amount mutual knowledge, beneficial social conventions would disappear. Subsequently, there have been a few researchers who noted the importance of the role of common knowledge in reasoning about one another in certain situations. However, Morris Friedell [7] and

David Lewis,[8] in the late 1960s, were among the firsts who studied the concept of *coordination* using game-theoretic methods. For this, they formally introduced the concept of *common knowledge*, which came to be profitably used in economics. In the 1980s, Michael Bratman started the philosophical analysis of *intentions, plans, and practical reason*.[9] Flowing out of this line of work, the notions of *action* and *agency* form integral parts of the study on strategies.[10]

Multi-agent systems: In the begin of the 1990s, the field of multi-agent systems, investigating teamwork and other social interactions among software agents, started to flower. Part of the investigations concentrated on agents' planning and intentions, inspired by Bratman's philosophical work. The study of strategic reasoning forms another crucial ingredient of multi-agent systems. Agents, on the basis of some information, reason to devise strategies for ensuring maximal gain. Using the languages of logic and game theory, the models of strategic reasoning in multi-agent systems have led to new insights into the dynamics of observation, updating of knowledge and belief, preference change, and dialogues.[11] Researchers have also developed decidable/tractable formulations of strategies from the viewpoint of both strategy synthesis and strategy verfication.[12]

Logic: Modeling social interaction has brought to the fore various logical systems to model agents' knowledge, beliefs, preferences, goals, intentions, common knowledge and belief.[13] When interactions are modelled as games, reasoning involves analysis of agents' long-term powers for influencing outcomes. While researching intelligent interaction, logicians have been interested in the questions of how an agent selects a particular strategy, what structure strategies may have, and how strategies are related to information. Thus, logicians have devised logical models in which strategies are 'first class citizens', rather than unspecified means to ensure outcomes.[14,15]

Cognitive science: In addition to idealized game-theoretic and logical studies on strategies in interactive systems, there have also been experimental studies on players' strategies and cognitive modelling of their reasoning processes. The classical game-theoretic perspective assumes that people are rational agents, maximizing their own utility by applying strategic reasoning. However, many experiments[3,4,16,17] have shown that people are not completely rational in this sense. Players may be altruists, giving more weight to the opponent's payoff; or they may try to salvage a good cooperative relation in case they meet the other agent again in future. Also, due to cognitive constraints such as working memory capacity,[18,19] people may be unable to perform optimal strategic reasoning, even if in principle they are willing to do so.Various cognitive models have been put forward to model such boundedly rational reasoning capabilities.[18]

Linguistic semantics and pragmatics: The concepts of strategies also play a role in language use and interpretation. For example, pragmatics can be explained in terms of a sender and receiver strategizing to understand and be understood, on the basis of concise and efficient messages.[20] Evolutionary game theory has been used to explain the evolution of language; for example, it has been shown that in signaling games, evolutionarily stable states occur when the sender's strategy is a one-one map from events to signals, and the receiver's strategy is the inverse map.[21]

1.2. *Logical approaches to modeling strategic reasoning*

The questions of how an agent selects a particular strategy, what structure strategies may have, and how strategies are constituted, are of utmost importance in the context of interaction situations, and as we see above, these questions arise in different forms in different subject areas. For a joint perspective on *strategizing* in related areas towards forming more realistic models of social and intelligent interaction, the need of the hour is to widen our scope of understanding to seemingly orthogonal viewpoints on strategizing, in other words, strategic reasoning. As a case in point, we describe below a meeting of game theory and logic for providing models of strategic interaction.

The study of strategic reasoning forms a crucial ingredient of the research area of intelligent interactive systems. Agents devise their strategies on how to interact so as to ensure maximal gain in the interaction process modeled as games. Their strategic reasoning is influenced by their knowledge, beliefs and intentions as well. Various logical studies of games and strategic reasoning have led to new insights into the dynamics of information,[22,23] updating of knowledge and belief,[24–28] preference change,[29–31] and processes of strategic interaction.[15,32–38] Studying *structures* in strategies, namely union, intersection, sequential composition and *response strategies* has led to describing top-down strategizing in large games played by perfectly-informed players with full as well as limited resources.[15,34,38] All these have now merged into broader studies of formal models of society, where computer science meets decision theory, game theory, and social choice theory.

1.3. *Cognitive science approaches to modeling strategic reasoning*

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).[39] Nowadays, cognitive scientists construct fine-grained theories about human reasoning strategies,[40,41] 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.[42] Cognitive models developed within this framework aim to explain certain aspects of cognition by assuming only general cognitive principles. Cognitive models of simple games exist in which it is important to know the opponent's behavior,[43–45] however, they do not take complex strategic reasoning into account.

The usual game-theoretic perspective assumes that people are rational agents, optimizing their gain by applying strategic reasoning. However, many experiments have shown that people are not completely rational in this sense. For example, McKelvey and Palfrey[4] have shown that in a traditional centipede game (cf. Figure 1) participants do not behave according to the Nash equilibrium reached by backward induction. In this version of the game, the payoffs are distributed in such a way that the optimal strategy is to always end the game at the first move. However, in McKelvey and Palfrey's experiment, participants stayed in the game for some rounds before ending the game: in fact, only 37 out of 662 games ended with the backward induction solution. One interpretation of this result is that the game-theoretic perspective fails to take into account the reasoning abilities of the participants. That is, perhaps, due to cognitive constraints such as working memory capacity, participants are unable to perform optimal strategic reasoning, even if in principle they are willing to do so. Thus, building up computational cognitive models for strategic reasoning, as suggested in our work,[46] provides a way to incorporate people's beliefs and constraints within the scope of their reasoning processes and also enhances our understanding of choices of strategies found in the empirical studies done by other researchers[4] as well as ourselves.[16,17]

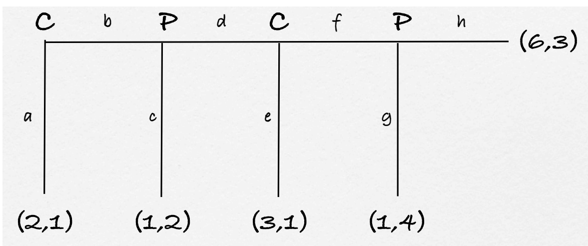

Figure 1. An example of a centipede game. For a detailed description, see 47.

In 48, a comparative study of various cognitive architectures has been provided. The quality of a cognitive model depends on its simplicity, its fit to the experimental data and predictions. These models sometimes have numerical parameters which can be tweaked, giving different outcomes. One can also specify the initial knowledge and actions of the model in a varied manner, resulting in different ways of modeling a single task. An advantage of having cognitive models, besides having statistical models, is that cognitive models can be broken down into mechanisms. Another advantage of a cognitive model is that one can compare the model's output with human data, and acquire a better understanding of individual differences. Strategic reasoning in complex interactive situations consists of multiple serial and concurrent cognitive functions, and thus it may be prone to great individual differences. Such differences become explicit in the modeling because of the reduction mechanism of the cognitive architecture of ACT-R.[42] It provides an excellent architecture to model learning and the application of cognitive skills. The theory has been validated by behavioral and neuro-scientific research.[49,50] From this computational cognitive modelling perspective, decision strategies like backward induction for turn-taking games have been used to capture second-order social reasoning.[45,51] For modelling strategic reasoning in such games, both declarative memory (to retrieve successive steps) and working memory (to store temporarily) have been used.

1.4. *Building bridges between empirical, logical and cognitive modeling*

In recent years, many researchers have questioned the idealization that a formal model undergoes while representing social reasoning methods (e.g. see [52,53]). Do formal methods represent human reasoning satisfactorily or should we concentrate on empirical studies and models based on those empirical data? A tension exists between the normative aspect of logic and the descriptive aspect of cognitive science.[54] A methodology for resolving this tension has been provided in our work in [46,55] which deals with human strategic reasoning in games of perfect information. Rather than thinking about formal and cognitive modeling as separate, one can consider them to be complementary and investigate how they can aid each other to bring about a more meaningful model of real-life scenarios. Game experiments will lead us to the behavioral strategies of humans having varying amounts of information. Such strategies will be modeled as logical formulas constituting a descriptive logic of strategies which will help in the construction of cognitive models. The cognitive models will predict human strategies which can be tested against the human data available from the experiments. The following is a schematic diagram of the process.

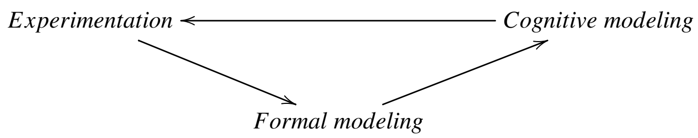

Figure 2. A schematic diagram of the bridging technique between different methodologies.

In [55], we provided an attempt to bridge the gap between experimental studies, cognitive modeling, and logical studies of strategic reasoning. In particular, a first study of a cognitive model of strategic reasoning that is constructed with the aid of a formal framework is discussed there. In [46], we extended the language that we introduced in [55] to represent strategies by a new belief component, so that we can describe reasoning about the opponent at a more fine-grained level. A new architecture PRIMs,[56] based on ACT-R, was used as the basis of the computational cognitive models. Actual implementations were made with respect to some strategy formulas and predictions were made based on the simulations about the data of the experiment reported in [16], closing the circle depicted in Figure 2.

To have a sustainable bridging technique between these methodologies, it is not enough to construct computational cognitive models corresponding to certain strategy formulas, but to come up with a translation system which, starting from a strategy represented in formal logic, automatically generates a computational model in the PRIMs cognitive architecture, which can then be run to generate decisions made in certain games, e.g. perfect informations games in our case. Such a translation system has been developed in [57,58] based on centipede-like games, a particular kind of perfect informations games. Since we wanted to make predictions on experiments like those reported in [16,17] which are based on centipede-like games, we concentrated on these games. These games are like centipede games (cf. Figure 1) in almost all respects, the only difference being that unlike centipede games, the sum of the points of players may not increase in the subsequent moves.[47]

In the remainder of the paper, based on the work reported in [46,55,57,58], we provide a systematic study of the endeavors underlying each of these *schematic arrows* depicted in Figure 2 providing pointers towards the complementary contributions of these methods in modeling human strategic behavior. To build up our study, we start with describing a particular kind of game experiment,[16,17] a strategy logic[46] and a computational cognitive model.[56]

2. Experiment, logic and computational cognitive model

In this section we briefly describe an experiment done in two phases, a language for describing strategic reasoning and a cognitive architecture, which are all essential ingredients for our study on the meeting of different methodologies involved in modeling strategic behaviour of the humans.

2.1. *Experiments*

The experiments that we describe here are experiments on perfect information games reported in [16,17] to analyze human strategic behavior. Experimental studies in behavioral economics have shown that the backward induction outcome is often not reached in large centipede games (cf. Figure 1). Instead of immediately taking the 'down' option, people often show partial cooperation, moving right for several moves before eventually choosing 'down'.[4,59,60] Accordingly, in the experiments reported in [16,17], our main interest was to examine participants' behavior in centipede-like games following a deviation from backward induction behavior by their opponent right at the beginning of the game. We primarily asked the following questions:

(1) Are people inclined to use forward induction when they play such games?
(2) If not, what are they actually doing? What roles are played by risk attitudes and cooperativeness versus competitiveness?
(3) Do people take the perspective of their opponents and make use of theory of mind?
(4) Can they be reasonably divided into types of players?

The experiments were conducted at the Institute of Artificial Intelligence (ALICE) at the University of Groningen, The Netherlands. A group of 50 Bachelor's and Master's students from different disciplines at the university took part in each phase of the experiment. The participants had little or no knowledge of game theory, so as to ensure that neither backward induction nor forward induction reasoning was already known to them. The participants played the finite perfect-information games in a graphical interface on the computer screen (cf. Figure 3). In each case, the opponent was the computer, which had been programmed to play according to plans that were best responses to some plan of the participant. The participants were instructed accordingly. In each game, a marble was about to drop, and both the participant and the computer determined its path by controlling the orange and the blue trapdoors: The participant controlled the orange trapdoors, and the computer controlled the blue trapdoors. 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.

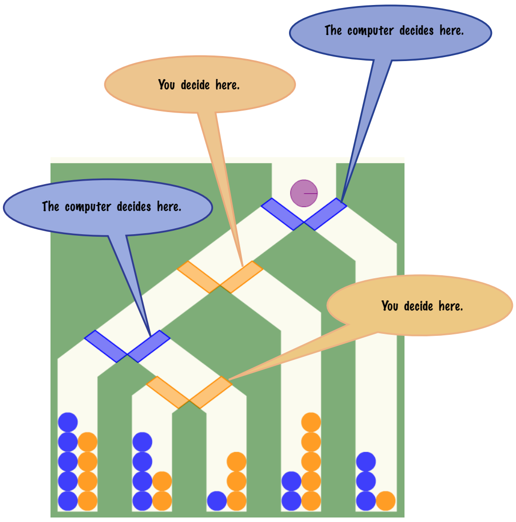

Figure 3. Graphical interface for the participants. The computer controls the blue trapdoors and acquires pay-offs in the form of blue marbles (represented as dark grey in a black and white print), while the participant controls the orange trapdoors and acquires pay-offs in the form of orange marbles (light grey in a black and white print).

In the experiment reported in [16], we investigated whether people are inclined to use forward induction in centipede-like games, rather than backward induction. We found that in the aggregate it appeared that participants showed forward induction behavior in response to deviation from backward induction behavior by their opponent, the computer, right at the beginning of the game. However, there exist alternative explanations for the choices of most participants; for example, choices could have been based on the extent of risk aversion that participants attributed to the computer in the remainder of the game, rather than to the sunk outside option that the computer has already foregone at the beginning of the game. Cardinal effects seemed to play a role as well: a number of participants might have been trying to maximize expected utility. For these reasons, the results of the experiment did not provide conclusive evidence for forward induction reasoning on the part of the participants.

So, we followed up with a similar experiment, reported in [17], where we designed centipede-like games with new payoff structures in order to make such cardinal effects less likely. We asked a number of questions to gauge the participants' reasoning about their own and the opponent's strategy at all decision nodes of a sample game. Even though in the aggregate, participants in the new experiment still tend to slightly favor the forward induction choice at their first decision node, their verbalized strategies most often depend on their own attitudes towards risk and those they assign to the computer opponent, sometimes in addition to considerations about cooperativeness and competitiveness.

The tasks that the participants had to perform in these experiments are mentioned in the table below in the order given there.

Step 1	Introduction and instructions.
Step 2	Practice Phase: 14 games.
Step 3	- Experimental Phase: 48 game items, divided into 8 rounds of 6 different games each, in terms of isomorphism class of pay-off structures. - Each of the 6 games occurs once in each round; in [16], these games occur in the same order in each round; in [17], these games occur in a random order in each round. - Question on computer's behavior in several rounds: For [16] Group A in rounds 3, 4, 7, 8; Group B in rounds 7, 8. For [17] Group A in the middle of the game at certain rounds; Group B at the end of the game in certain rounds.
Step 4	Final Question(s): In [16], the question was about possible future moves of the computer; in [17], the questions were regarding decisions at all nodes of a sample game.

2.2. *Logic*

With regard to the logical framework, we describe the one proposed in [46]. We restrict to the strategy specification language discussed there and present the same as that is the main ingredient of the formal framework involved in our bridge-building method. We start with describing extensive form games.

2.2.1. *Extensive form games*

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

Let N denote the set of players; we use i to range over this set. For the time being, we restrict our attention to two player games, 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.

2.2.2. *Game trees*

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

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

2.2.3. *Strategies*

A *strategy* for player i is a function μ^i which specifies a move at every game position of the player, i.e. $\mu^i : S^i \to \Sigma$. A strategy μ^i can also be viewed as a subtree of T where for each node belonging to player i, there is a unique outgoing edge and for nodes belonging to player $\bar{\imath}$, every enabled move is included.

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. $\sigma^i : S^i \rightharpoonup \Sigma$. As above, a partial strategy σ^i can also be viewed as a subtree of T where for some nodes belonging to player i, there is a unique outgoing edge and for other nodes belonging to player i as well as nodes belonging to player $\bar{\imath}$, every enabled move is included.

2.2.4. *Syntax for extensive form game trees*

We now build a syntax for game trees. 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 *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(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 4 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 $j : 1 \leq j \leq k$, $T_j = (S_j, \Rightarrow_j, \widehat{\lambda}_j, s_{j,0})$.

  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 $s \in S_{T_j}$, $\widehat{\lambda}_h(s) = \widehat{\lambda}_j(s)$;
  - $\Rightarrow_h = \bigcup_{j:1 \leq j \leq k}(\{(s_x, a_j, s_{j,0})\} \cup \Rightarrow_j)$.

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

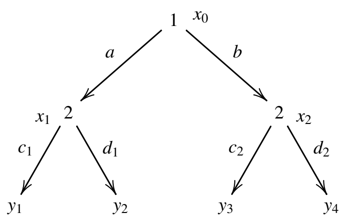

Figure 4. Extensive form game tree. The nodes are labelled with turns of players and the edges with the actions. The syntactic representation of this tree can be given by:
$h = ((1, x_0), a, t_1) + ((1, x_0), b, t_2)$, where
$t_1 = ((2, x_1), c_1, (2, y_1)) + ((2, x_1), d_1, (2, y_2))$;
$t_2 = ((2, x_2), c_2, (2, y_3)) + ((2, x_2), d_2, (2, y_4))$.

2.2.5. *Strategy specifications*

We are now ready to describe the strategy specification language. The main case specifies, for a player, which conditions she tests before making a move. In what follows, 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 T_h, player i *believes* at node x that ψ holds'. We reiterate our disclaimer from [46]. 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.

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 $(r \leq q)$ to denote that 'the rational number r is less than or equal to the rational number q'. The syntax of strategy specifications is given by:

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

where $\psi \in BPF_b(P^i)$. The basic idea is to use the above constructs to specify properties of strategies as well as to combine them to describe a play of the game. For instance, the interpretation of a player i's specification $[p \mapsto a]^i$ where $p \in P^i$, is to choose move a at every game position belonging to player i where p holds.

At positions where p does not hold, the strategy is allowed to choose any enabled move. The strategy specification $\eta_1 + \eta_2$ says that the strategy of player i conforms to the specification η_1 or η_2. The construct $\eta_1 \cdot \eta_2$ says that the strategy conforms to specifications η_1 and η_2.

2.3. *Computational cognitive model*

We now provide a brief description of the cognitive architectures at the basis of our computational cognitive models. We first provide a description of ACT-R based on which the architecture of PRIMs has been developed, followed by a description of PRIMs, especially pointing out the differences from ACT-R. The description of ACT-R is based on what we provided in [55], and that of PRIMs in [46].

2.3.1. *ACT-R*

ACT-R, *Adaptive Control of Thought - Rational*, is an integrated theory of cognition as well as a cognitive architecture that many cognitive scientists use.[42] It consists of modules that link with cognitive functions, for example, vision, motor processing, and declarative processing. Each module maps onto a specific brain region. Furthermore, 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. Consequently, if a model has to keep track of more than one piece of information, it has to move the pieces of information back and forth between two important modules: *declarative memory* and the *problem state*. Moving information back and forth comes with a time cost, in some cases causing a cognitive bottleneck.[18]

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.[61] Anderson[42] provided a formalization of the mechanism that produces the relationship between the probability and speed of retrieval. If the activation value drops below a certain minimal value (the retrieval threshold), the related information is no longer accessible. In that case, the system will report a retrieval failure after a constant time factor. If the activation value is above the retrieval threshold, the information is accessible. Then, the higher the activation value, the faster the retrieval will be. As soon as a chunk is retrieved from declarative memory, it is placed into the declarative module's buffer. As mentioned earlier, each ACT-R module has a buffer that may contain one chunk at a time. On a functional level of description, the chunks that are stored in the various buffers are the knowledge structures of which the cognitive architecture is aware.

The *problem state* module also contains a buffer that can hold one chunk in which information can be temporarily stored. 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*.[42] If two or more production rules match a particular game state, the production rule with the highest utility is selected.

2.3.2. *PRIMs*

PRIM, the *primitive elements theory*, is a recent cognitive theory developed by Taatgen, who implemented it in the computational cognitive architecture PRIMs.[56] 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 rather than ACT-R as cognitive architecture in [46] to model more sophisticated reasoning strategies.

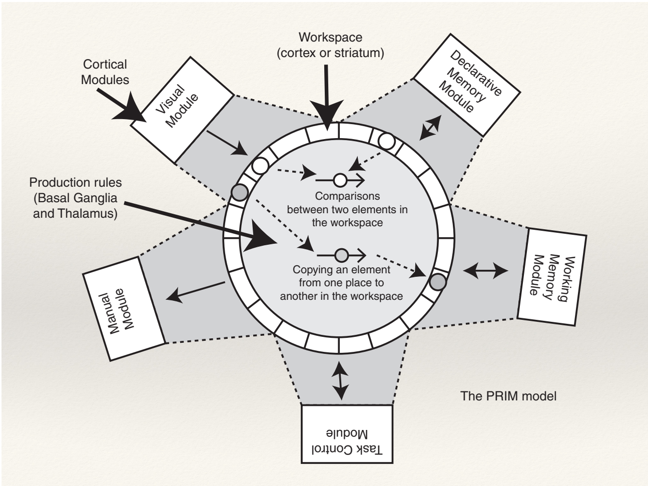

Figure 5. The PRIMs cognitive architecture, from [56]. Reprinted with permission from the author. For this work, PRIMs models of strategic reasoning are used as 'virtual participants' that play centipede-like games.

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 5). 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[56,62] showed the predictive power of PRIMs by modeling a variety of transfer experiments such as text editing, arithmetic, and cognitive control. The architecture has been used to model children's development of theory of mind,[63] transfer between the 'take the best' heuristic and the balance beam task,[64] and children's mistakes in arithmetic.[65] PRIMs models can also be run to predict the estimated time to complete certain tasks. Like ACT-R, PRIMs models cognitive resources as being bounded.

3. Details of the bridging techniques

To describe the essentials of the bridging between experiments, logic and computational cognitive models we will mainly focus on the centipede-like games presented in Figure 6, which are the Games 1, 4 and 1′ used in the experiment reported in [17].

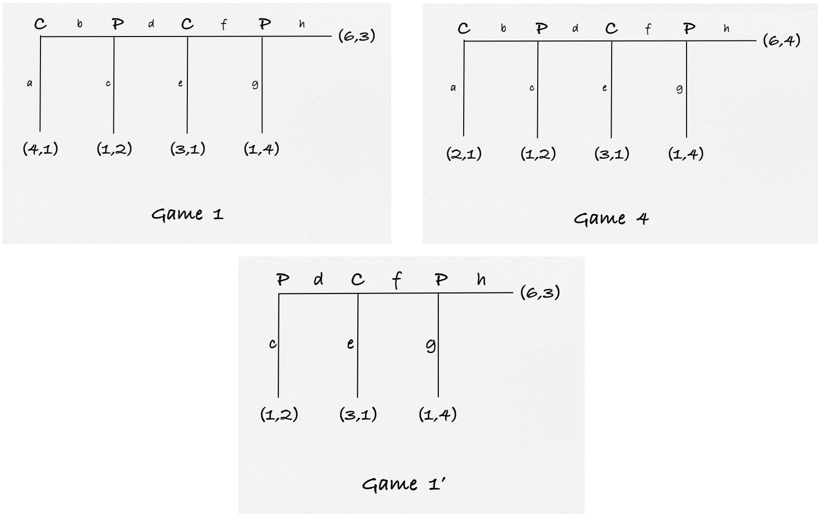

Figure 6. Games 1, 4 and 1′ from [17]

3.1. *Experimentation ⟶ Formal modeling*

Empirical studies on games can describe human strategic behavior. Logical studies on one hand facilitate the study of properties of such reasoning processes in games, on the other hand pave the way for implementation through formal languages. It is this second feature that we utililize for our bridging methodology. We will use a suitably defined formal language to express human strategic behavior as found in the empirical studies. As a case in point, we consider some of the strategies used in playing games 1, 1′ and 4 as reported in [17], and represent them in the language described in Section 2.2.

We start with fixing the preliminary 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 games 1 and 4 of Figure 6, with C playing at n_1 and n_3, and P playing at the remaining two nodes n_2 and n_4. Since game 1′ is a subgame of game 1, the three decision nodes are denoted by n_2, n_3 and n_4. We have four belief operators for the games 1 and 4, namely two per player. We abbreviate some formulas that describe the payoff structure of the games:

$\alpha := \langle d\rangle\langle f\rangle\langle h\rangle((u_C = p_C) \wedge (u_P = p_P))$
(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 descriptions to describe the payoff structures of the games in Figure 6:

$$\varphi_1 := \alpha_1 \wedge \beta_1 \wedge \gamma_1 \wedge \delta_1 \wedge \chi_1 \qquad \varphi_4 := \alpha_4 \wedge \beta_4 \wedge \gamma_4 \wedge \delta_4 \wedge \chi_4$$

$$\varphi_{1'} := \alpha_{1'} \wedge \beta_{1'} \wedge \gamma_{1'} \wedge \delta_{1'}$$

Let ψ_i^j denote the conjunction of all the order relations of the rational payoffs for player i ($\in \{P, C\}$) given in Game j ($\in \{1, 4, 1'\}$) of Figure 6.

Strategy specifications describing forward induction (extensive form rationalizable (EFR))[66] reasoning of player P at the node n_2 in games 1 and 4 are as follows:

$$\eta_P^1 : [(\varphi_1 \wedge \psi_P^1 \wedge \psi_C^1 \wedge \langle b^-\rangle\mathbf{root} \wedge \mathbb{B}_{g1}^{n_2,P}\langle d\rangle\neg e \wedge \mathbb{B}_{g1}^{n_2,P}\langle d\rangle\langle f\rangle g) \mapsto d]^P$$

$$\eta_P^4 : [(\varphi_4 \wedge \psi_P^4 \wedge \psi_C^4 \wedge \langle b^-\rangle\mathbf{root} \wedge \mathbb{B}_{g4}^{n_2,P}\langle d\rangle e \wedge \mathbb{B}_{g4}^{n_2,P}\langle d\rangle\langle f\rangle g) \mapsto c]^P$$

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

$$\zeta_P^1 : [(\varphi_1 \wedge \psi_P^1 \wedge \psi_C^1 \wedge \langle b^-\rangle\mathbf{root} \wedge \mathbb{B}_{g1}^{n_2,P}\langle d\rangle e \wedge \mathbb{B}_{g1}^{n_2,P}\langle d\rangle\langle f\rangle g) \mapsto c]^P$$

$$\zeta_P^4 : [(\varphi_4 \wedge \psi_P^4 \wedge \psi_C^4 \wedge \langle b^-\rangle\mathbf{root} \wedge \mathbb{B}_{g4}^{n_2,P}\langle d\rangle f \wedge \mathbb{B}_{g4}^{n_2,P}\langle d\rangle\langle f\rangle h) \mapsto d]^P$$

We note here that in game 1, P has a unique extensive form rationalizable strategy and a unique backward induction strategy and they are not identical, whereas in game 4 all strategies can be considered as both backward induction and extensive form rationalizable strategies. Thus the formulas for game 4 provide some example cases.

We now describe some other simple strategies of players with different kinds of restrained reasoning capabilities. Note that such limited reasoning is ubiquitous in our daily life (see e.g. [45]). 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. [67]). Such a player-strategy can be described for game 1′ as follows:

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

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. [68]). Their strategy in game 1′ can be described as follows:

$$\chi_P^{1'} : [(\alpha_{1'} \wedge \beta_{1'} \wedge \delta_{1'} \wedge \gamma_{1'} \wedge (1 \leq 2) \wedge (2 \leq 4) \wedge (2 \leq 3) \wedge \mathbf{root}) \mapsto d]^P$$

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 formula, and α, β, δ, and γ for the χ formula. In fact, one could ignore the payoffs for C for the χ formula. We will come back to these strategies in Section 3.3 when we validate the model predictions with the experimental results. The experimental findings [17] showed that c was played about 35% in game 1, about 28% in game 4, and about 41% in game 1′ at the first decision node for the participant P.

3.2. *Formal modeling ⟶ Cognitive modeling*

Computational cognitive models provide ways to explore the essence of cognitive functionalities in the realm of strategic reasoning. The current method of constructing computational cognitive models is basically ad hoc - created by hand. For example, decision strategies like backward induction for turn-taking games have been used to capture second-order social reasoning.[45,51] Using the language described in Section 2.2, Top[57] has devised a method that automatically generates cognitive models from strategy formulas, without human intervention. In this section, we describe the main points of this approach - for details, see [58]. The models are constructed based on the PRIMs architecture.

In what follows, we provide for each component in the logical language the corresponding behaviour of a PRIMs model generated using that component.

- $\langle a^+\rangle$ and $\langle a^-\rangle$: A model translated from a formula containing these operators uses focus actions[63] to move its visual attention to the specified location. Focus actions take time to complete similar to human gazing, causing these operators to increase the model's reaction time.
- **root**: When a strategy formula contains the proposition **root**, the PRIMs model will visually inspect the specified node to determine whether it is the root of the tree.
- $\mathbf{turn}_i$: When a strategy formula contains a proposition $\mathbf{turn}_i$, where i is C or P, the PRIMs model will read the player name from the specified node in the game tree, and compare it to i.
- $(u_i = q_i)$: The proposition $(u_i = q_i)$ states that player i's payoff is equal to q_i at a certain location. The PRIMs model will compare q_i to a value in its visual input. Because this value may be required for future comparisons, it is also stored in an empty slot of working memory.
- $(r \leqslant q)$: A PRIMs model cannot instantly access each value in a visual display: it has to remember them by placing them in working or declarative memory before it can compare them. A proposition $(u_i = q_i)$ causes such a value to be stored in working memory. A proposition $(r \leqslant q)$ then sends two of these values from working memory to declarative memory, to try and remember which one is bigger. When a model is created, its declarative memory is filled with facts about single-digit comparisons, such as $(0 \leqslant 3)$ and $(2 \leqslant 2)$.
- $\mathbb{B}^{(i,x)}_{\mathbb{h}}$ and a: To describe the PRIMs model behavior, let us consider an example of a belief formula:

$$\mathbb{B}^{(C,n1)}_{g1}\langle b^+\rangle c$$

This formula can be read as 'In Game 1, at node 1, player C believes that after playing b, c will be played'. To verify such a belief, a model employs a plan of action similar to the ones used by models in [69]. When a model is created, it contains several strategies in its declarative memory. When a model verifies a belief, it sends a *partial* sequence of actions to declarative memory, corresponding to the assumptions of the belief, in an attempt to retrieve a *full* sequence of actions, which conforms to a strategy. Using the formula above as an example, the assumptions of the belief are that b is played. Therefore the model sends the sequence b to declarative memory. All sequences b-c, b-d-e, b-d-f-g and b-d-f-h could be retrieved, depending on the strategies present in declarative memory. However, only *b-c* verifies $B_{g1}^{(C,n1)}\langle b^{+}\rangle c$. All the other sequences falsify it.

As we have mentioned in Section 3.1, strategies such as BI and EFR can have multiple solutions in one game, when there are payoff ties (cf. game 4 in Figure 6). In this case one has to exhaustively list all solutions for the specified strategy. Because of this, the translation system as reported in [58] allows for a strategy to consist of a list of multiple strategy formulas. The PRIMs model generated from this list tries to verify each formula in it, using the behaviour described above, until it finds one it can verify, and play the action prescribed by the formula it verified. There is no need to specify what the model has to do when it cannot verify any of the formulas in the list - the list is exhaustive, and at least one of the formulas holds.

A pertinent question here could be as follows: How to make this exhaustive list of strategy formulas? Doing it with hand is just moving the burden from the ad hoc construction of computational cognitive models to the ad hoc listing of strategy formulas. This could be taken care of quite easily by implementing a software for generating this exhaustive list of formulas, given the alphabet of the language, formation rules and the finite game(s) under consideration.

While describing the bridging methodologies, (i) experiment to logic, and (ii) logic to cognitive modeling, we have only used the syntax of the logic as presented in Section 2.2. We neither needed the semantics of the logic, nor any properties within the purview of logic. So we could have easily restricted to some suitable formal language doing away with the logic aspect altogether. The reason that we do not do so, and logic is an integral part of this bridge-building technique is that, presenting the whole logical framework, that is, the language plus semantics provides a sanity check on, for example, the expressivity of the language. Moreover, without the semantics it would not have been possible to attach the respective roles of the different language components in the PRIMs models. Because of space restrictions we did not present the semantics of the language described in Section 2.2, it is available in [46].

3.3. *Cognitive modeling ⟶ Experimentation*

Towards completing the full circle in our schematic diagram of Figure 2, based on the work done in [58], we now provide a comparison of how the computational cognitive models built by hand as well as by the translation system described in Section 3.2 performed with respect to the human participants in the experiment reported in [46].

We consider four models: a handmade myopic and own-payoff model, and an automatically generated myopic and own-payoff model (cf. Section 3.1). These automatically generated myopic and own-payoff models are generated from the myopic and own-payoff strategy formulas for game 1′. These models play Game 1′ (see Figure 6) only. They play against computer opponents who play pre-specified moves. The models play as player P, whereas the computer opponent plays as player C. Each model was run 50 times, where it plays 50 games, to simulate 50 virtual participants who play 50 games each. Reaction times and decisions were recorded.

The reaction times for the four models, as well as the human participants in [46], can be found in Figure 7. It indicates that the myopic models are faster than the own-payoff models, and the generated models are faster than the handmade models. Human players tend to be faster than these models, but they may not use the myopic and own-payoff strategies, which is why there is no point in making direct comparisons for now.

Figure 7. Reaction times for handmade and automatically generated myopic and own-payoff models, as well as the human participants in [46], when making their first decision in Game 1′.

The abilities of the translation system were also investigated by looking at novel automatically generated models playing games 1 and 4 (see Figure 6). For both games, two models were generated: one that uses backward induction (BI), and one that uses extensive-form rationalizability (EFR). These strategy formulas not only contain payoffs and comparisons, like the myopic and own-payoff strategy formulas, but also contain beliefs. Because both BI and EFR strategies have multiple solutions in game 4, exhaustive strategy formulas are required to describe these strategies.

The reaction times for the BI and EFR models can be found in Figure 8. These exhaustive models are a lot slower than the myopic and own-payoff models. Furthermore, reaction times in Game 1 are faster than reaction times in Game 4. It seems that reaction times are a function of the number of formulas required to create the exhaustive strategy formula: for both BI and EFR, in Game 1, only one formula is needed. In Game 4, BI requires two formulas, and EFR requires four formulas. To test this, a simple linear regression using number of formulas was performed to predict reaction times. A significant regression equation is found ($F(1, 189) = 432.6$, $p < 2.2 \cdot 10^{-16}$), with an R^2 of 0.696. Predicted reaction time in milliseconds is equal to 10401 + 50453·(number of formulas).

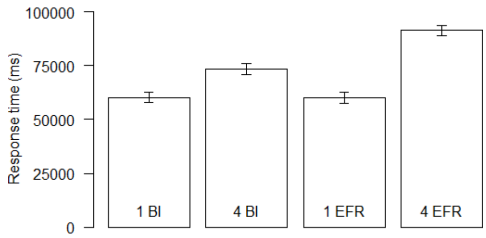

Figure 8. Reaction times for automatically generated BI and EFR models, when making their first decision. Here, '1 BI' denote the reaction time for the BI model in Game 1.

4. Conclusion and future work

In this paper we have presented a broad overview of a bridging methodology between the so-called orthogonal approaches towards modeling human reasoning - descriptive reasoning methods vis-à-vis normative or idealized reasoning methods. In [55], these ideas were presented for the first time, and we followed it up with certain extensions in [46], providing the first ever description of how the cycle depicted in Figure 2 can be completed in terms of connecting the different methods. Based on subsequent work [17,57,58] we provide here more detailed explorations of the underlying ideas for each of the arrows in Figure 2. They provide further proofs of concept of the methodology under consideration.

It is evident that an in-depth study of these connections between different methods also provide novel insight into the individual methods. The experimental findings on strategic reasoning of humans suggest for incorporating newer operators in the logical languages to deal with such kind of reasoning. For example, risk-taking and risk-averseness form a major part of the considerations of the participants in the experiments reported in [16,17]. This suggests that a graded belief operator might model the participants' behavior in a better way in comparison to the belief operators that we use in the current work. From the computational cognitive modeling perspective, it is useful to note that the performance of automated models is better than the corresponding handmade models (cf. Figure 7). This could speed up research to a great extent in this area. Also, it is found that the reaction times of the models are too slow compared to human reaction times (cf. Figure 7). This suggests the need of further development of the focus actions in the PRIMs model.[58] Finally, from the experimentation viewpoint, given the exhaustive list of strategy formulas and the corresponding cognitive models, it is useful to have a comparative study of these strategies empirically and then to have a comparison with the predictions of the cognitive models.

Marr[70] 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.

According to Isaac et al.,[71] 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[72] 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[73] 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). Through our bridging methodology we show that this study based on logic, experiment and computational cognitive model can play a fruitful role at all these levels and at the interfaces between them.

A natural continuation for this line of work is to model partially-informed agents' strategies which provide interesting challenges - incorporating information structures and memory restrictions leading to evolution of strategies in formal frameworks, testing and introducing such information states in experimental subjects, and modelling such states as cognitive modules. As before, one should start with developing logical frameworks, possibly using automata-theoretic techniques.

Then one can produce computational cognitive models based on these formal frameworks, and finally validate those models based on game experiments that have been performed earlier to investigate human strategic reasoning under partial information. The focus would be to overcome interdisciplinary modelling challenges and construct computationally useful and efficient models of human society.

Bibliography

1. M. Osborne and A. Rubinstein, *A Course in Game Theory* (MIT Press, Cambridge, MA, 1994).
2. G. Attanasi and R. Nagel, A survey of psychological games: Theoretical findings and experimental evidence, in *Games, Rationality and Behaviour: Essays on Behavioural Game Theory and Ex- periments*, eds. A. Innocenti and P. Sbriglia (Palgrave McMillan, 2008) pp. 204–232.
3. C. Bicchieri, Common knowledge and backward induction: A solution to the paradox, in *TARK '88: Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning About Knowledge*, ed. M. Vardi (Morgan Kaufmann Publishers Inc., 1988).
4. R. McKelvey and T. Palfrey, An experimental study of the centipede game, *Econometrica* **60(4)**, 803 (1992).
5. A. Kanamori, *The Higher Infinite*, second edn. (Springer, Hiedelberg, 2008).
6. P. Koellner and W. H. Woodin, Large cardinals from determinacy, in *Handbook of Set Theory*, eds. M. Foreman and A. Kanamori (Springer-Verlag, 2010) pp. 2–202.
7. M. Friedell, On the structure of shared awareness, *Behavioral Science* **14**, 28 (1969).
8. D. Lewis, *Convention: A Philosophical Study* (Harvard University Press, 1969).
9. M. Bratman, *Intention, Plans, and Practical Reason* (CSLI Publications, 1999).
10. J. F. Horty and N. Belnap, The deliberative stit: A study of action, omission, ability and obligation, *Journal of Philosophical Logic* **24**, 583 (1995).
11. M. Wooldridge, *An Introduction to Multi-agent Systems*, second edn. (John Wiley & Sons, 2009).
12. J. Pilecki, M. Bednarczyk and W. Jamroga, Synthesis and verification of uniform strategies for multi-agent systems, in *Computational Logic in Multi-Agent Systems. CLIMA 2014 Proceedings*, ed. N. B. et. al., LNCS, Vol. 8624 (Springer, 2014).
13. J. van Benthem, *Logic in Games* (MIT Press, 2016).
14. K. Chatterjee, T. Henzinger and N. Piterman, Strategy logic, in *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR 07)*, eds. L. Caires and V. Vasconcelos, LNCS, Vol. 4703 (Springer Verlag, Berlin, 2007).
15. R. Ramanujam and S. Simon, A logical structure for strategies, in *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, eds. G. Bonanno, W. van der Hoek and M. Wooldridge, Texts in Logic and Games, Vol. 3 (Amsterdam University Press, Amsterdam, 2008) pp. 183–208.
16. S. Ghosh, A. Heifetz and R. Verbrugge, Do players reason by forward induction in dynamic perfect information games?, in *Proceedings of the Fifteenth Conference on Theoretical Aspects of Rationality and Knowledge, TARK*, ed. R. Ramanujam, EPTCS, Vol. 2152015.

17. S. Ghosh, A. Heifetz, R. Verbrugge and H. de Weerd, What drives people's choices in turn-taking games, if not game-theoretic rationality?, in *Proceedings of the Sixteenth Conference on Theoretical Aspects of Rationality and Knowledge, TARK*, ed. J. Lang, EPTCS, Vol. 2512017.
18. J. Borst, N. Taatgen and H. van Rijn, The problem state: A cognitive bottleneck in multitasking, *Journal of Experimental Psychology: Learning, Memory, & Cognition* **36**, 363 (2010).
19. 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*, 2014.
20. A. Benz, G. Jäger and R. van Rooij (eds.), *Game Theory for Pragmatics* (Palgrave Macmillan, London, 2006).
21. G. Jäger, Applications of game theory to linguistics, *Language and Linguistics Compass* **2**, 406 (2008).
22. J. Gerbrandy, Bisimulation on planet Kripke, PhD thesis, University of Amsterdam1999.
23. J. van Benthem, J. van Eijck and B. Kooi, Logics of communication and change, *Information and Computation* **204**, 1620 (2006).
24. J. van Benthem, Games in dynamic-epistemic logic, *Bulletin of Economic Research* **53**, 219 (2001).
25. G. Bonanno, Belief revision in a temporal framework, in *New Perspectives on Games and Interaction*, eds. K. Apt and R. van Rooij, Text in Logic and Games, Vol. 4 (Amsterdam University Press, Amsterdam, 2009).
26. J. van Benthem, Dynamic logic for belief revision, *Journal of Applied Non-Classical Logic* **17**, 129 (2007).
27. T. Ågotnes and N. Alechina, The dynamics of syntactic knowledge, *Journal of Logic and Computation* **17**, 83 (2007).
28. W. Jamroga and W. van der Hoek, Agents that know how to play, *Fundamenta Informaticae* **63**, 185 (2004).
29. F. Liu, Changing for the better: Preference dynamics and agent diversity, PhD thesis, University of Amsterdam2008.
30. O. Roy, Thinking before acting, PhD thesis, University of Amsterdam2008.
31. P. Girard, Modal logic for belief and preference change, PhD thesis, Department of Philosophy (Stanford University), (Stanford, CA, USA, 2008).
32. R. Parikh, The logic of games and its applications, *Annals of Discrete Mathematics* **24**, 111 (1985).
33. W. van der Hoek, A. Lomuscio and M. Wooldridge, On the complexity of practical ATL model checking, in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 06)*, eds. P. Stone and G. Weiss (ACM Inc., New York, 2006).
34. S. Ghosh, Strategies made explicit in dynamic game logic, in *Proceedings of the Workshop on Logic and Intelligent Interaction, ESSLLI*, eds. J. van Benthem and E. Pacuit2008.
35. J. van Benthem, In praise of strategies, in *Games, Actions and Social Software*, eds. J. van Eijck and R. VerbruggeTexts in Logic and Games, FoLLI subseries of LNCS (Springer Verlag, Berlin, 2012) pp. 96–116.

36. D. Walther, W. van der Hoek and M. Wooldridge, Alternating-time temporal logic with explicit strategies, in *Proceedings of the XIth Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, ed. D. Samet (ACM Inc., New York, 2007).
37. S. Ghosh, R. Ramanujam and S. Simon, Playing extensive form games in parallel, in *Proceedings of the 11th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA XI)*, eds. J. Dix and others, LNCS, Vol. 6245 (Springer, Berlin, 2010).
38. R. Ramanujam and S. Simon, Dynamic logic on games with structured strategies, in *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR-08)*, eds. G. Brewka and J. Lang (AAAI Press, Menlo Park, CA, 2008).
39. G. Polya, *How to Solve It. A New Aspect of Mathematical Method* (Princeton University Press, Princeton, N.J., 1945).
40. M. Lovett, A strategy-based interpretation of Stroop, *Cognitive Science* **29**, 493 (2005).
41. I. Juvina and N. Taatgen, Modeling control strategies in the n-back task, in *Proceedings of the 8th International Conference on Cognitive Modeling*, eds. R. Lewis, T. Polk and J. Laird (Psychology Press, 2007).
42. J. Anderson, *How Can the Human Mind Occur in the Physical Universe?* (Oxford University Press, New York (NY), 2007).
43. C. Lebiere, D. Wallach and R. West, A memory-based account of the prisoner's dilemma and other 2x2 games, in *Proceedings of Third International Conference on Cognitive Modeling*, eds. N. Taatgen and J. Aasman (Universal Press, 2000).
44. R. West, C. Lebiere and D. Bothell, Cognitive architectures, game playing, and human evolution, in *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*, ed. R. Sun (Cambridge University Press, New York (NY), 2006) pp. 103–123.
45. 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**, 455 (2014).
46. S. Ghosh and R. Verbrugge, Studying strategies and types of players: Experiments, logics and cognitive models, *Synthese* , 1 (2017).
47. R. Rosenthal, Games of perfect information, predatory pricing and the chain-store paradox, *Journal of Economic theory* **25**, 92 (1981).
48. N. Taatgen and J. Anderson, The past, present, and future of cognitive architectures, *Topics in Cognitive Science* **2**, 693 (2010).
49. J. Anderson, D. Bothell, M. Byrne, S. Douglass, C. Lebiere and Y. Qin, An integrated theory of the mind, *Psychological Review* **111 (4)**, 1036 (2004).
50. J. Anderson and C. Lebiere, The Newell test for a theory of cognition, *Behavioral and Brain Science* **26**, 1 (2003).
51. L. van Maanen and R. Verbrugge, A computational model of second-order social reasoning, in *Proceedings of the 10th International Conference on Cognitive Modeling*, eds. D. Salvucci and G. Gunzelmann2010.
52. B. Edmonds, Social intelligence and multi-agent systems, in *Invited talk, The Multi-Agent Logics, Languages, and Organisations Federated Workshops*, 2010.
53. R. Verbrugge, Logic and social cognition: The facts matter, and so do computational

models, *Journal of Philosophical Logic* **38**, 649 (2009).
54. J. van Benthem, Logic and reasoning: Do the facts matter?, *Studia Logica* **88**, 67 (2008), Special issue on logic and the new psychologism, edited by H.Leitgeb.
55. S. Ghosh, B. Meijering and R. Verbrugge, Strategic reasoning: Building cognitive models from logical formulas, *Journal of Logic, Language and Information* **23**, 1 (2014).
56. N. A. Taatgen, The nature and transfer of cognitive skills, *Psychological Review* **120**, 439 (2013).
57. J. Top, Bridging the gap between logic and cognition: A translation method for centipede games, Master's thesis, University of Groningen (2017).
58. J. Top, R. Verbrugge and S. Ghosh, An automated method for building cognitive models for turn-based games from a strategy logic, *Games* **9** (2018), 44.
59. R. Nagel and F. F. Tang, Experimental results on the centipede game in normal form: An investigation on learning, *Journal of Mathematical Psychology* **42**, 356 (1998).
60. C. Camerer, *Behavioral Game Theory* (Princeton University Press, 2003).
61. J. R. Anderson and L. J. Schooler, Reflections of the environment in memory, *Psychological Science* **2**, 396 (1991).
62. N. A. Taatgen, Between architecture and model: Strategies for cognitive control, *Biologically Inspired Cognitive Architectures* **8**, 132 (2014).
63. 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*, 2015.
64. L. Gittelson and N. Taatgen, Transferring primitive elements of skill within and between tasks, in *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*, (Cognitive Science Society, 2014).
65. T. Buwalda, J. Borst, H. v. d. Maas and N. Taatgen, Explaining mistakes in single digit multiplication: A cognitive model, in *Proceedings of the 14th International Conference on Cognitive Modeling*, 2016.
66. D. Pearce, Rationalizable strategic behaviour and the problem of perfection, *Econometrica* **52**, 1029 (1984).
67. T. Hedden and J. Zhang, What do you think I think you think? Strategic reasoning in matrix games, *Cognition* **85**, 1 (2002).
68. M. Raijmakers, D. Mandell, S. van Es and M. Counihan, Children's strategy use when playing strategic games, *Synthese* **191**, 355 (2014).
69. C. Stevens, J. Daamen, E. Gaudrain, T. Renkema, J. Top, F. Cnossen and N. Taatgen, Using cognitive agents to train negotiation skills, *Frontiers in Psychology* **9** (2018).
70. D. Marr, *Vision* (Freeman and Company, New York, 1982).
71. A. Isaac, J. Szymanik and R. Verbrugge, Logic and complexity in cognitive science, in *Johan van Benthem on Logic and Information Dynamics*, (Springer, 2014) pp. 787–824.
72. G. Baggio, M. v. Lambalgen and P. Hagoort, Logic as Marr's computational level: Four case studies, *Topics in Cognitive Science* **7**, 287 (2015).
73. R. Cooper and D. Peebles, Beyond single-level accounts: The role of cognitive architectures in cognitive scientific explanation, *Topics in Cognitive Science* **7**, 243 (2015).