Beyond Analytical Modeling, Gathering Data to Predict Real Agents' Strategic Interaction

Rustam Tagiew (tagiew@informatik.tu-freiberg.de)

Institute for Computer Science of TU Bergakademie Freiberg, Germany

Abstract. This paper presents research proposals on the interdisciplinary research infrastructure for understanding human reasoning in game-theoretic terms. Strategic reasoning impacts human decision making in social, economical and competitive interactions. The provided introduction summarizes concepts from AI, game theory and psychology. First result is a concept of interdisciplinary game description language as a part of the focused interdisciplinary research infrastructure. The need of this domain-specific language is motivated and is aimed to accelerate the current developments. As second result, the paper provides a summary of ongoing research and its significance.

1 Introduction

Different scientific disciplines predict outcomes of human social, economical and competitive interactions on the granularity level of individual decisions [1, p.4]. Autonomous intelligent systems, which perceive, decide and act in an environment according to their preferences, are called *agents* in AI, MAS and sociology. People and implemented artificial agents are called as *real agents.*

A *rational* agent makes always decisions, whose execution has according to his subjective estimation the most preferred consequences for him [2,3]. Level of intelligence impacts the quality of the subjective estimation. Rationality justifies agents' decisions and predictions of other agents' decisions. In *strategic interactions*, agents are *rational* and apply mutually and even recursively the concept *rationality.* Game theory predicts outcomes of strategic interactions [3]. Further, *game* is a notion [4] for the formal structure of a concrete strategic interaction. Agents involved in a game are called *players.*

A game in *normal form* consists of *strategies* (sets of decisions) and players' preferences over outcomes [3]. *Finite normal form games* contain a finite number of outcomes. Game theory is there to provide *equilibria* id est to solve games. An equilibrium is an irrevocable combination of players' strategies – none of the players can improve his outcome by altering his chosen strategy. If players' preferences are defined by their *payoff functions*, the equilibria of finite normal form games are guaranteed [5]. A payoff or also utility function assigns a numeric value as magnitude of preference to every outcome. Calculation of equilibria is generally NP-hard [6]. *GAMBIT* is a software library for this task [7].

In the case of players, who are intelligent enough to solve their game, an equilibrium should (immediately) occur independently of players' builds or even real identities. This is fundamentally different to the common AI approach (referenced further as *game playing*), where programs compete the last decades in playing chess e.g. still without achieving any irrevocable solution [2, chap.5]. Although the existence of an equilibrium is proven for chess, analytical abilities of none of the present game theorists suffices to show at least, whether the equilibrium implicates the win of the whites. If at least one of the players is not able to solve the game, an equilibrium is not guaranteed to occur.

Although the modern game-theoretic textbooks contain mostly fictive scenarios, the reader is asked to resist the conclusion that real-life cases can not be formalized as games. Contrary to parlor games, a concrete strategic interaction with lacking explicit rules is to be formalized as a game. Manual formalization is not guaranteed to be accurate and mostly results in far too simplified *"toy games"* as criticized in AI literature [8]. Scaling down the space of games can enable automatic formalization [9]. A player faces this problem as well as a strict outsider like a game theorist. He has to answer the question correctly, which interacting agents really exist in the environment and in what kind of game he is involved in. It is called *incomplete information* in game-theoretic terms [3], if this question can not be clarified.

Incomplete information transforms to *imperfect information*, if a probability distribution over all possible formalizations of a game can be provided [10]. Poker e.g. is a game of imperfect information, where every player is supposed to be aware about the probability distribution over the possible hands, but does not know the current hands. It is not obligatory that a game becomes a *common knowledge* (CK) among its players – it is also possible that a player bears his own variant of the game in mind, he misconceives to be involved in. As a reminder, CK is that, what everybody knows and everybody knows that everybody knows it and infinitely prepending "everybody knows that" [3].

CK of the game would not exist during a poker round, if Alice hides cards in her sleeve unknown to her opponent Bob. Alice considers that Bob's actions should conform the equilibria calculated in the original game and she calculates the equilibria of a *global game* consisting of the original game and the "cheated" game [11]. Players may have an intractable number of mutual nested believes about the details of their game. For instance, Alice could reckon to some extent that Bob also cheats, Bob could suspect Alice of considering him as a cheater and so on. An intractable number of mutual nested believes results in an intractable size of the global game, where no game-theoretic solution is guaranteed to be calculated or even to exist in the case of infinity. Therefore, cases are concentrated on, where a game of imperfect information is its players' CK.

This paper concentrates on the relevance of strategic reasoning in real agents' decision making, which causes an (immediately) occurring equilibrium, if the unbounded intelligence is assumed. In the cases of absent strategic reasoning, game theory can still predict the direction of convergence id est towards an equilibrium. This is proposed by Price [12] for prediction of stochastic processes, more precisely populations in biological terms, and is called *evolutionary game theory*

[13]. A convergence is also observed in the case of using *reinforcement learning* instead of strategic reasoning [14, e.g.]. The assumption of this paper is that the usage of the concept strategic reasoning can be extended by newer methods. This assumption does not negate the existence of reinforcement learning e.g. in real agents' strategic interactions and is discussed next in the case of people.

Although the real human preferences are a subject of philosophical discussions [15], the application of strategic reasoning assumes that they can be captured in concrete interactions as required for modeling rationality. The consideration of people as rational agents is disputed at least in psychology [16, pp.527–530], where even a scientifically accessible argumentation exposes the existence of stable and consistent human preferences as a myth [17]. Since the last six decades nevertheless, the common scientific standards for *econometric* experiments are that subjects' preferences over outcomes can be insured by paying differing amounts of money [18]. However, insuring preferences by money is criticized by the term *homo economicus* as well.

The ability of identifying other agents and of modeling their reasoning corresponds with the psychological term *ToM* (Theory of Mind) [19], which lacks almost only in the cases of autism. For application of strategic reasoning, subjects as well as researchers, who both are supposed to be non-autistic people, may be then able of modeling of others' strategic reasoning too. In Wason task at least, subjects' reasoning does not match the researchers' one though [20]. People may use no logic at all [21], but also mistake seriously in the calculus of probabilities [22].

The data of econometric experiments does not match the equilibria of games according to which they are conducted [23,24]. That means that the strategic reasoning according to the global (researchers' point of view) game does not arise among the subjects due to a set of reasons, which should be clarified. There is a need for more than only the pure analytical game theory, because even people familiar with game theory are observed to deviate from equilibria in multiple cases [24]. It is gathering and analyzing of data from experiments and field studies. This paper steps further – it proves the potential of making the interdisciplinary research on real agents' strategic interactions more efficient. Like in bio-informatics [25], it is supposed to be done by an *interdisciplinary research infrastructure* – domain specific languages and common tools.

The paper is organized as follow. Next section summarizes the main concepts for the interdisciplinary research infrastructure. Then, the section 3 presents detailed the hitherto research. At the end, the results are concluded in order to figure out the remaining construction sites.

2 Conceptualizing Interdisciplinary Research Infrastructure

A conceptualization of the already partially existing (interdisciplinary) research infrastructure for real agents' strategic interactions follows. It aims to provide an elaborate overview and an exhaustive motivation. Artificial agents are also included into consideration, although this paper concentrates mostly on people. People can be replaced by artificial agents in order to simulate or to intervene human strategic interactions. Whether simulation or intervention – in both cases, artificial agents can cut costs, allow a direct control of their builds, can be numerously deployed and are almost unlimited in period of use.

In order to avoid incomplete information, which can make the application of strategic reasoning intractable, an explicit formalization is to be used. Because a formalization of a concrete strategic interaction can be inaccurate, it is reasonable to execute it inversely. A concrete strategic interaction has to be created out of an already existing game. This is *game realization* and a software-based game realization is a *game implementation* [1, p.108]. Game realization is the almost always unmentioned action after *mechanism design* [2, p.632]. Mechanism design is inverse to game solving – adjusting of a game to already predetermined desired equilibria. If game realization is impossible, mechanism design is futile.

In real-life cases, games can be realized by physical conditions, by non-participating agents, by participating agents themselves or by a subset of the 3 previous instances [26]. If participating agents are responsible for a (partial) game realization, they should prefer the compliance with rules over the advantages from "cheating". A non-participating agent responsible for game realization can be modeled as rational too. For instance, the attorney in *prisoner's dilemma* keeps his word in order to not risk his reputation, where prisoner's dilemma is supposed to be familiar to the reader.

In the case of game implementation, the software can be divided into fractions: *game management*, game-solving algorithms, game-playing algorithms and auxiliary algorithms. Game management is the part of software, which executes the rules and calculates outcomes [27]. It may also record in order to gather data [28]. Human-computer interfaces aka *proxy agents* [29] are examples of auxiliary algorithms. Additionally, game implementation minus game-solving/playing algorithms is called *game infrastructure* [1, p.53].

In the case of explicit rules, the question about the form of games arises. The normal form is one of the two most general forms to express games in *non-co-operative game theory*. In *co-operative game theory* or also *coalitional game theory* [30], players may covenant and group into *coalitions*. The process of negotiating and the way of ensuring the agreements themselves are not issues of co-operative game theory. From non-co-operative game-theoretic point of view, disadvantages should arise for the players, who break the agreements. From both points of view, a player makes a rational decision, whether it is his own behavior or an agreement about a co-operative behavior. Both points of view are considered to be equal [3]. Due to the fact that the pure analytical game theory does not suffice, the interest is focused a more detailed form – the *extensive form* – the second most general to express games in non-co-operative game theory. In contrast to others, the extensive form captures separately the actions' sequences and their alternating subsequences in a representation known as *game tree* in AI. Therefore, the actions' sequences can be called as *root paths*. The normal form and the coalitional formalization are skipped as argued because of their higher abstraction. Also skipped is the consideration of continuous games [31], where equilibra can be acquired by solving of differential equations and no general form exit.

The theoretical appropriateness of the extensive form does not result a computational one – the problem is the inappropriate size of a games represented in extensive form. For instance, one can consider that the rules of chess are sent as a game tree via network. It is rather an *Interdisciplinary Game Description Language* (IGDL) having expressive power of the extensive form, which is needed for 4 rough categories of considered instances within and beyond game implementations. These categories are **A**) game-solving algorithms, **B**) game-playing algorithms, **C**) non-solving/playing algorithms (game management, mechanism design, auxiliaries etc.) and **D**) scientific human users (sociologists e.g.).

Less important issues are skipped out of this consideration for IGDL – issues like adjustments for *evolutionary mechanism design* [32] and for non-scientific human users. Reducing the size compared with an equivalent representation in extensive form is called further *compactness* [1, p.65]. Compactness is not the only criterion for IGDL. For the categories A–D, one can summarize the partial criteria as following:

1. **Computational speed-up.** Regularities like symmetries can be used in order to reduce the computation time of equilibria [33]. This feature is captured by formalisms called *succinct games* and should be also provided by IGDL. Reduction of computation time by using compact representation applies also to game-playing algorithms [34, e.g.] and can be considered in the case of game management.
2. **Re-usability & comparability.** A language for games forces the re-usability and also comparability of game-playing algorithms as suggested by Pell [35]. This applies also for game-solving algorithms [7] and for non-solving/playing algorithms algorithms [36].
3. **Interdisciplinary human usability.** IGDL should prevent the scientific manual game-theoretic formalization from resulting in “toy games” as criticized in AI literature [8]. A graphical representation of a game may improve the usability even more [37].
4. **Decidability.** This feature should be provided for IGDL in order to ensure that the calculation of outcomes definitely terminates [38]. This also important for game-solving/playing algorithms to be able to calculate the consequences of actions.
5. **General compact interchange format.** For interfacing instances of the categories A–D, IGDL should satisfy the need of a compact interchange format. At same time, IGDL should be as general as possible, where the facility to express n-person games of imperfect information is most general. Finally, instances of all the mentioned categories should be at least theoretically IGDL-compatible in order to skip reformatting, id est to facilitate their efficient mutual integration.
6. **Time.** Time remains the issue disregarded by the extensive form. As one can conceive by comparing game playing in fast chess and in normal chess,

time given for making decisions impacts them. Therefore, time is needed to be included in IGDL in order to ascertain the time dependent details by the explicit rules. Otherwise, the durations of actions' sequences e.g. may depend on the current game implementation and not be given explicitly in advance conjoined with the game.

Some examples for usage of IGDL can be provided. As 1st example, IGDL-compatible chess playing algorithms can be incorporated into system, which compete in playing other chess-like games described in IGDL. As 2nd example, a IGDL-based game editor can be used to allow non-computer scientists to set-up their own experiments. As 3rd example, data gathered in experiments conducted according to a game described in IGDL can be compared with the equilibria calculated by IGDL-based game-solving algorithms for the same game. As 4th example, game described in IGDL can be easily forwarded to the IGDL-based game-playing algorithms for an approximative solution, if IGDL-based game-solving algorithms fail to output timely. As 5th example (proposed by [39]), the data of the experiments conducted based on IGDL can be better compared or even stored in a central web database like the state of art in bio-informatics. Games described in a special language with and without conjoined time dependent details are called further *game descriptions*.

3 Summarizing Ongoing Research

IGDL is the desired domain-specific language, which has already some precursors and these precursors are summarized in this section. It is also possible that the ongoing research will bear different concurrent versions of IGDL. In order to assess the hitherto approaches better, a rough categorization of the used formal means is provided. Such categories are functions, graphs, logic, Petri-nets and so on. Tab.1 contains the regarded approaches and their rough categories. The ability to describe *simultaneous moves* (id est actions) is subset to more general imperfect information, because other players' actions can be unobserved only during a simultaneous execution. In *deterministic* games, it is impossible to describe a probability distribution over possible subsequences of actions.

In discrete non-co-operative game theory, there are different approaches for compact game forms, whose aim is computational speed-up [33]. The most important of them chronologically ordered are *congestion games* [40], *sequential form* [8], *graph games* [41], *local effect games* [42] and *action graph games* [43]. The software *GAMUT* can generate random games of these and other kinds, where the extensive form is included [52]. Computational speed-up of sequential form in solving 2-person-games of imperfect information is used in GAMBIT [7].

GAme LAnguage (Gala) [44] is developed in order to provide an interface to the game-solving algorithms. Factually, Gala affords Prolog-based game descriptions, where a game of extensive form or of normal form can be generated. The generated game-theoretic representation can be forwarded to GAMBIT or other game-solving algorithms. The main improvement of Gala compared to these representations is the game descriptions' compactness as perceived at least at the

Table 1. Overview of the precursors for IGDL [1, p.69]. Numbers in the **Crit.**-subrow are the satisfied criteria from the section 2, The **Used**-subrow shows the categories of used software. The **Means**-row includes the rough categories of the means used for describing games. “perfect” concerning information is to be inferred in case of missing “imperfect” and “simultaneous moves”.

Approach **Citation**	**Crit.** **Used**	**Means**	**Class of games**
congestion games [40]	1,4 A,C	functions	subset of n-person games, simultaneous moves
sequential form [8]	1,4 A,C	matrices	2-person games of imperfect information
graph games [41]	1,4 A,C	graphs, functions	n-person games, simultaneous moves
local effect games [42]	1,4 A,C	functions	subset of n-person games, simultaneous moves
action graph games [43]	1,4 A,C	graphs, functions	n-person games, simultaneous moves
Gala [44]	2 A	logic	n-person games of imperfect information
MAID [45]	1,4 A,B	Bayes-nets	n-person games of imperfect information
continuous games [46]	6 A	functions	subset of 2-person games of imperfect information
timed games [47,48]	6 B	functions	2-person games
GDL [38]	1,2,4 B,C	logic	deterministic n-person games, simultaneous moves
GDL-II [49]	1,2,4,5 B,C	logic	n-person games of imperfect information
game Petri-nets [50]	1 A	Petri-nets	deterministic n-person games, simultaneous moves
PNSI [51]	2,4–6 A–C	Petri-nets	n-person games of imperfect information
SIDL2.0 [1, p.98]	2,5,6 C	logic	n-person games of imperfect information
z-Tree-language [36]	2,3,5,6 C,D	imperative language	n-person games of imperfect information

examples from Gala's software package. Due to full-scale Prolog, Gala does not provide decidability. Therefore, the generation of extensive form games from Gala game descriptions must not terminate.

There is a subset of game-playing algorithms, which is not only aimed to play games, but also to simulate human (strategic) reasoning in them. Simulating human reasoning falls in the subject of *cognitive science*. There are currently two different approaches, where the human strategic reasoning has to be expressed in a general language in order to facilitate an efficient comparability of the models. The first is based on *cognitive architectures* [53][54], which are languages for models of general human reasoning. The second is based on *Multi-Agent Influence Diagrams* (MAID) [45]. The second factually conforms the game-theoretic point of view on strategic interactions and provides the alternative language MAID for describing games. MAID are shown to be expressive enough to represent n-person games of imperfect information in the algorithms for game playing. MAID can be also transformed to extensive form in order to solve them [55].

The previously discussed literature does not mention the inclusion of time dependent details in game descriptions, what some theoretical approaches from game theory, *concurrency theory* [56] and *control engineering* [48] aim. A current work [46] in game theory extends extensive form games of perfect information and continuous time [57] to such of imperfect information – *continuous games*. In [57][46], only a subset of such games is regarded – the set of player's actions is always the same. Generally, a point of time is assigned to every action and time grows strictly over a sequence of actions.

Game Description Language (GDL) succeeded in sparking an international programing competition on general game playing [38]. A concrete game description in GDL is sent to an artificial game player and never to its human programmer – the programmer knows only the structure of GDL. This satisfies the criterion 2. For the criterion 5, generality of IGDL's precursors is a trend – either it will be possible to describe n-person games of imperfect information in IGDL or IGDL should be extended to facilitate that. This trend caused the development of GDL-II [49], which is an extension of GDL for n-person games of imperfect information. Like GDL, GDL-II is based on Datalog, which is a version of Prolog [58]. Datalog guarantees decidability by banning functions, limiting variables' ranges and restricting recursion. Due to the decidability, the existing game management algorithm based on GDL-II is guaranteed to terminate. However, the ban of functions worsen compactness. For instance, if arithmetic addition is needed to describe actions' consequences in a game, the result for every required summands' combination must be separately defined in the game description.

There are no time dependent details included explicitly in GDL-II game descriptions. GDL and GDL-II describe games in a way *STRIPS-like* [59] languages do. In languages for planning tasks, STRIPS-like descriptions can be replaced by descriptions based on Petri nets [60]. *Petri Nets for Strategic Interaction* (PNSI) is a game description language, which is proposed chronologically between GDL and GDL-II [37]. PNSI uses basic Petri nets instead of logic. Petri nets are also known being used in game theory to describe subclasses of games [50]. PNSI provides decidability [1, p.89]. The advantages of PNSI compared with GDL-II are the graphical representation of Petri nets and the ability to describe games of *equidistant time*. Equidistant time means that the game management algorithm for PNSI pauses exactly for one *chronon* between two *game states* [51], where a game state is also a node of its game tree. In this context, a chronon is a constant period of time, which is explicitly known to players. During a chronon, players' actions can be submitted. The game management algorithm for GDL-II is not of equidistant time, because the next state is calculated exactly after the submission of the last action, if it is inside the allowed time period. The time point of the last submission may vary depending on players. Of cause, GDL-II is supposed be also able to describe games of equidistant time, if its game management algorithm is modified as proposed for PNSI.

For PNSI, there exist an algorithm that can generate games of extensive form from game descriptions as in the case of Gala [51]. Therefore, PNSI provides an interface to game solving algorithms. A game of extensive form generated based on a PNSI game description is a slightly modified *state transition system* of the underlying basic Petri net. In this context, a state transition system is an oriented graph consisting of game states, where every edge represents a progression in time. There is still no algorithm for GDL-II to generate games of extensive form. A state transition system can be also generated for GDL-II game descriptions [49]. Therefore, generation of extensive form games is supposed to be also possible for GDL-II game description.

PNSI suffers of insufficient compactness as well as GDL-II but in a different way. The arithmetic addition and subtraction are banned in Datalog, but inherently supported by basic Petri nets. On the other hand, basic Petri nets require every state to be decoded as a vector of natural numbers and do not have other operations than the addition and the subtraction. The game description of the parlor game *Nim* needs in PNSI much less space than in GDL-II [1]. The opposite for chess, a GDL-II chess description needs less.

Dropping the criterion of decidability may dramatically improve compactness. *Strategic Interaction Definition Language* (SIDL) is based on ISO-Prolog, does not provide decidability and attains for example games a higher compactness [1, p.98]. Of cause, the widely used game description in an imperative language can be also mentioned. However, the game management part of software aka *game server* is then required to send its own code in order to provide explicit rules [1, p.54].

For scientific human users beyond computer science, there is an ongoing development of user-friendly software for experiments [61]. *RatImage* [62] and *TEEC* [61] are examples of the first generation of such software. They are libraries, which facilitate programming. The second generation provides already domain-specific languages for the game management and the layout of human-computer interfaces. These are *ComLabGames* [63] and *z-Tree* [36]. z-Tree is the most used [39]. There is a z-Tree-language, which is actually an imperative language, in which the game and also the human-computer interfaces can be described. This language does not provide decidability. It has no relations to game solving or playing algorithms.

4 Conclusion

A large-scale view on the problem of understanding human strategic reasoning is presented. The elaborated solution is the development of the interdisciplinary research infrastructure. This research infrastructure is proposed to make the interdisciplinary research more efficient, as it is already observed in similar interdisciplinary problems. As an underline of the large-scale view, there are matters chained from different sources, which have been never cited together before. For instance, GDL-II and z-Tree are such matters.

The elaborated concept is the domain-specific language IGDL. A summary of its precursors shows that none of these is developed enough to satisfy IGDL's full outline. There is still no language, which incorporates the graphical representation like PNSI, compactness improvements like GDL-II and a proven human usability like z-Tree-language.

References

1. R. Tagiew, Strategische Interaktion realer Agenten: Ganzheitliche Konzeptualisierung und Softwarekomponenten einer interdisziplinären Forschungsinfrastruktur, Ph.D. thesis, TU Bergakademie Freiberg (2011).
2. S. Russel, P. Norvig, Artificial Intelligence, Pearson Education, 2003.
3. M. J. Osborne, A. Rubinstein, A course in game theory, MIT Press, 1994.
4. O. Morgenstern, J. von Neumann, Theory of Games and Economic Behavior, Princeton University Press, 1944.
5. J. Nash, Non-cooperative games, Annals of Mathematics (54) (1951) 286 – 295.
6. I. Gilboa, E. Zemel, Nash and correlated equilibria, Games and Economic Behavior 1 (1989) 80–93.
7. R. D. McKelvey, A. M. McLennan, T. L. Turocy, Gambit: Software tools for game theory, version 0.2007.01.30, gambit-project.org (2007).
8. D. Koller, N. Megiddo, B. von Stengel, Fast algorithms for finding randomized strategies in game trees, in: STOC, 1994, pp. 750–759.
9. A. Barbu, S. Narayanaswamy, J. M. Siskind, Learning physically-instantiated game play through visual observation, in: ICRA, IEEE, 2010, pp. 1879–1886.
10. J. C. Harsanyi, Games with incomplete information played by bayesian players, Management Science 14 (1967) 59–182, 320–334, 486–502.
11. S. Morris, H. S. Shin, Global games: theory and applications, in: M. Dewatripont, L. Hansen, S. J. Turnovsky (Eds.), Advances in Economics and Econometrics, Cambridge University Press, 2003, Ch. 3.
12. J. Maynard Smith, G. R. Price, The logic of animal conflict, Nature 246 (1973) 15–18.
13. J. Hofbauer, K. Sigmund, The Theory of Evolution and Dynamical Systems, Cambridge University Press, 1988.
14. J. B. Pollack, A. D. Blair, Co-evolution in the successful learning of backgammon strategy, Machine Learning 32 (3) (1998) 225–240.

15. L. Stevenson, D. L. Haberman, Ten Theories of Human Nature, OUP USA, 2004.
16. M. W. Eysenck, M. T. Keane, Cognitive Psychology: A Student's Handbook, Psychology Press, 2005.
17. M. H. Bazerman, D. Malhotra, Economics wins, psychology loses, and society pays, in: D. De Cremer, M. Zeelenberg, J. K. Murnighan (Eds.), Social Psychology and Economics, Lawrence Erlbaum Associates, 2006, Ch. 14, pp. 263–280.
18. E. H. Chamberlin, An experimental imperfect market, Journal of Political Economy 56 (1948) 95–108.
19. R. Verbrugge, L. Mol, Learning to apply theory of mind, Journal of Logic, Language and Information 17 (2008) 489–511.
20. P. C. Wason, Reasoning, in: B. M. Foss (Ed.), New horizons in psychology, Penguin Books, 1966, pp. 135–151.
21. M. Oaksford, N. Chater, The probabilistic approach to human reasoning, Trends in Cognitive Sciences 5 (2001) 349–357.
22. D. Kahneman, P. Slovic, A. Tversky, Judgment Under Uncertainty: Heuristics and Biases, Cambridge University Press, 1982.
23. R. Pool, Putting game theory to the test, Science 267 (1995) 1591–1593.
24. C. F. Camerer, Behavioral Game Theory, Princeton University Press, 2003.
25. H. Nakamura, S. Date, H. Matsuda, S. Shimojo, A challenge towards next-generation research infrastructure for advanced life science, New Generation Computing 22 (2004) 157–166.
26. R. Tagiew, General game management agent, CoRR abs/0903.0353.
27. M. R. Genesereth, N. Love, B. Pell, General game playing: Overview of the aaai competition, AI Magazine 26 (2) (2005) 62–72.
28. R. Tagiew, Towards a framework for management of strategic interaction, INSTICC Press, 2009, pp. 587–590.
29. Y. Gal, A. Pfeffer, Modeling reciprocal behavior in human bilateral negotiation, in: AAAI, AAAI Press, 2007, pp. 815–820.
30. D. Ray, A Game-Theoretic Perspective on Coalition Formation, Oxford University Press, 2007.
31. R. Isaacs, Differential Games, John Wiley and Sons, 1965.
32. S. G. Phelps, Evolutionary mechanism design, Ph.D. thesis, University of Liverpool (2007).
33. C. H. Papadimitriou, The complexity of finding nash equilibria, Ch. 2, pp. 29–50.
34. S. Schiffel, M. Thielscher, Fluxplayer: A successful general game player, in: AAAI, AAAI Press, 2007, pp. 1191–1196.
35. B. Pell, Metagame: a new challenge for games and learning, in: H. van den Herik, L. Allis (Eds.), Heuristic programming in artificial intelligence 3–the third computer olympiad, Ellis-Horwood, 1992.
36. U. Fischbacher, z-Tree: Zurich toolbox for ready-made economic experiments, Experimental Economics 10 (2007) 171–178.
37. R. Tagiew, Multi-agent petri-games, in: CIMCA/IAWTIC/ISE, IEEE, 2008, pp. 130–135.
38. N. Love, T. Hinrichs, D. Haley, E. Schkufza, M. R. Genesereth, General game playing: Overview of the aaai competition, Tech. rep., Stanford University (2008).
39. S. Gächter, Improvements and future challenges for the research infrastructure in the field 'experimental economics', SSRN eLibrary Working Paper No.56.
40. R. Rosenthal, A class of games possessing pure-strategy nash equilibria, International Journal of Game Theory 2 (1973) 65–67.
41. M. Kearns, M. Littman, S. Singh, Graphical models for game theory, in: Proceedings of UAI, 2001.

42. K. Leyton-Brown, M. Tennenholtz, Local-effect games, in: IJCAI, Morgan Kaufmann, 2003.
43. N. Bhat, K. Leyton-Brown, Computing nash equilibria of action-graph games, in: Proceedings of UAI, 2004.
44. D. Koller, A. Pfeffer, Representations and solutions for game-theoretic problems, Artificial Intelligence 94 (1997) 167–215.
45. D. Koller, B. Milch, Multi-agent influence diagrams for representing and solving games, in: IJCAI, Morgan Kaufmann, 2001, pp. 1027–1034.
46. Y. Sannikov, Games with imperfectly observable actions in continuous time, Econometrica 75 (2007) 1285–1329.
47. R. Alur, D. L. Dill, A theory of timed automata, Theoretical Computer Science 126 (1994) 183–235.
48. C. J. Tomlin, J. Lygeros, S. S. Shankar, A game theoretic approach to controller design for hybrid systems, Proceedings of the IEEE 88 (2000) 949–970.
49. M. Thielscher, A general game description language for incomplete information games, in: AAAI, AAAI Press, 2010, pp. 994–999.
50. J. Clempner, Modeling shortest path games with petri nets, Internaltional Journal of Applied Mathematics and Computer Science 16 (2006) 387–397.
51. R. Tagiew, On multi-agent petri net models for computing extensive finite games, in: N. T. Nguen, R. Katarzyniak, A. Janiak (Eds.), ICCCI (SCI Volume), Vol. 244, Springer, 2009, pp. 243–254.
52. E. Nudelman, J. Wortman, Y. Shoham, K. Leyton-Brown, Run the gamut: A comprehensive approach to evaluating game-theoretic algorithms, in: AAMAS, IEEE, 2004, pp. 880–887.
53. F. E. Ritter, D. P. Wallach, Models of two-person games in ACT-R and Soar, in: ECCM, Nottingham University Press, 1998, pp. 202–203.
54. R. L. West, C. Lebiere, D. J. Bothell, Cognition and Multi-Agent Interaction, Cambridge University Press, 2006, Ch. 5.
55. D. Koller, B. Milch, Multi-agent influence diagrams for representing and solving games, Games and Economic Behavior 45 (2003) 181–221.
56. L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, M. Stoelinga, The element of surprise in timed games, in: CONCUR, Springer, 2003, pp. 144–158.
57. L. K. Simon, M. B. Stinchcombe, Extensive form games in continuous time: Pure strategies, Econometrica 57 (1989) 1171–1214.
58. E. Dantsin, T. Eiter, G. Gottlob, A. Voronkov, Complexity and expressive power of logic programming., in: Conference on Computational Complexity, IEEE, 1997, pp. 82–101.
59. R. Fikes, N. Nilsson, Strips: a new approach to the application of theorem proving to problem solving, Artificial Intelligence 2 (1971) 189–208.
60. D. Zhang, Planning with petri nets, in: Robotics and Automation, IEEE, 1991, pp. 769–775.
61. S. Geisler, E. Ponick, Teec: Ein java toolkit für ökonomische experimente, in: M. H. Breitner, B. Bruns, F. Lehner (Eds.), Neue Trends im E-Learning, Physica-Verlag HD, 2007, pp. 93–106.
62. K. Abbink, A. Sadrieh, Ratimage - research assistance toolbox for computer-aided human behavior experiments, Discussion Paper Serie B 325, University of Bonn, Germany (1995).
63. P. Kese, R. A. Miller, V. Prasnikar, D. Zupanic, Comlabgames, comlabgames.com (2004).