The International General Game Playing Competition

Michael Genesereth, Yngvi Björnsson

■ *Games have played a prominent role as a test bed for advancements in the field of artificial intelligence ever since its foundation over half a century ago, resulting in highly specialized world-class game-playing systems being developed for various games. The establishment of the International General Game Playing Competition in 2005, however, resulted in a renewed interest in more general problem-solving approaches to game playing. In general game playing (GGP) the goal is to create game-playing systems that autonomously learn how to play a wide variety of games skillfully, given only the descriptions of the game rules. In this paper we review the history of the competition, discuss progress made so far, and list outstanding research challenges.*

General game players are systems able to play strategy games based solely on formal game descriptions supplied at run time. (In other words, they don't know the rules until the game starts.) Unlike specialized game players, such as Deep Blue (Campbell, Hoane, and Hsu 2002), general game players cannot rely on algorithms designed in advance for specific games; they must discover such algorithms themselves. General game playing expertise depends on intelligence on the part of the game player rather than intelligence of the programmer of the game player.

General game playing (GGP) is in many ways similar to autonomous planning. Domain-independent problem solving is at the core of both. The description of a game in the game description language (GDL) (Love, Hinrichs, and Genesereth 2006) is similar to that in the languages used by planners (such as PDDL); and the overall goal is the same — to achieve a state with specified properties. One obvious difference is that, in GGP, there are opponents, which complicates the process of determining an ideal course of action. Another difference is that, in GGP, there is an execution environment, making it possible for a game player to interleave planning and execution. Also, in GGP, there are time constraints, which make it essential for players to act even when they are unsure which courses of action are best. In the last two respects GGP is more similar to reactive than classical planning, in that it has to commit to a

single next action based on the current context before having a complete plan available. However, whereas reactive planning has, at least traditionally, concentrated on myopic techniques that allow for an immediate reaction (measured in milliseconds) then GGP allows for an in-depth deliberation on each move (measured in tens of seconds), as for playing games skillfully it is necessary to look many moves ahead.

General game playing is also related to game theory, since both are concerned with games. However, again, there are differences. In game theory, a game corresponds to a game tree, and there is little or no attention to how games are communicated to game players. In general game playing, the problem description is essential; different game descriptions can be written in multiple ways, each lending themselves to a different kind of knowledge representation, reasoning, and learning approaches (such as for performance reasons). So it is not only a question of how to reason but also how to do so efficiently in real time. Also, game theory often makes assumptions about the rationality of the players, whereas, in general game playing, these assumptions are less common — the opponents might not be rational at all or they may have crashed or lost connectivity to the game manager.

General game playing is an interesting application in its own right. It is intellectually engaging and more than a little fun. But it is much more than that. It provides a theoretical framework for modeling discrete dynamic systems and for defining rationality in a way that takes into account problem representation as well as complexities like incompleteness of information and resource bounds. It has potential practical applications in areas where these features are important, for example, in enterprise management and computational law. It is also concerned with applications of AI technology in the real world, such as how to learn from experience and act autonomously in novel environments in real time. More fundamentally, it raises questions about the nature of intelligence and serves as a laboratory in which to evaluate competing approaches to artificial intelligence.

The International General Game Playing Competition

In order to promote progress on GGP, the AI community in 2005 established the International General Game Playing Competition, and it has run annual competitions ever since (Genesereth, Love, and Pell 2005). The competitions are typically associated and colocated with either the AAAI conference or IJCAI each year.

The Computational Logic Group at Stanford University is the main organizer of the International GGP Competition. The number of participants in the competition has been stable at around 10 to 15 entries annually; for example, 11 teams from six different nations participated in the 2012 competition. The competition consists of two phases: a preliminary and a final. The preliminaries are open to everyone. A wide variety of games are played and the top 8 teams advance to the finals. The finals always take place on site (AAAI/IJCAI) and are played using a playoff format with two agents matched against each other. Each playoff match typically consists of three different games, with the winner advancing from the quarterfinal, to the semifinal, and to the final. In the last couple of years a double-elimination playoff format has been used, giving the agents that lose a regular playoff match a second chance to play on in a so-called loser bracket.

The organizers compose and select the games that are played in the competition to highlight the different aspects of GGP. In the preliminaries, single-agent, two-player, and multiplayer games are played; the playoffs, because of their pairing format, are however restricted to two-player games. The games can be turn-based or simultaneous-move, zero-sum or non-zero-sum, and range in complexity from being simple puzzles to challenging chesslike games. The games are often interesting variants of existing board games, for example, checkers played on a cylindrical board, or tic-tac-toe played in parallel on nine different boards.

Table 1 shows the winners of the competition over the years, mostly different players in different years with the notable exception of CadiaPlayer, which has won three times.

In recent years, the competition has included a man-machine demonstration match pitting the competition winner against a human player. While the human player won the first of these demonstrations, the computer has won all of the matches since. In 2012, CadiaPlayer, in addition to defeating the other competitors, also defeated the human race (represented by Chris Welty) in the postcompetition Carbon versus Silicon matchup. (As a consolation prize, the human was awarded two bottles of Scotch, in part to ease his disappointment at letting down the human race.)

A related development is the availability of a massive open online course (mooc) of general game playing, aimed at exposing the field to tens of thousands of students and preparing them to participate in the annual competition. The first of these moocs is scheduled to run on the Coursera platform in the spring of 2013.[1]

Brief Overview of General Game Playing

General game playing is concerned with finite, synchronous games. These games take place in an

environment with finitely many states, with one distinguished initial state and one or more terminal states. In addition, each game has a fixed, finite number of players; each player has finitely many possible actions in any game state, and each state has an associated goal value for each player. The dynamic model for general games is synchronous update: all players move on all steps (although some moves could be no-ops), and the environment updates only in response to the moves taken by the players.

Because all games in GGP are finite, it is possible, in principle, to describe such games in the form of lists of states and actions and tables or graphs to express legality, goals, termination, and update. Unfortunately, such explicit representations are not practical in most cases. Even though the numbers of states and actions are finite, they can be extremely large; and the tables relating them can be larger still. For example, in chess, there are thousands of possible moves and more than 10^{40} states (Shannon 1950).

In the vast majority of games, states and actions have composite structure that allows us to define a large number of states and actions in terms of a smaller number of more fundamental entities. In chess, for example, states are not monolithic; they can be conceptualized in terms of pieces, squares, rows and columns and diagonals, and so forth. By exploiting this structure, it is possible to encode games in a form that is more compact than direct representation. The game description language supports this by relying on a conceptualization of game states as databases and by relying on logic to define the notions of legality, reward, termination, and so forth. For a reference, simple games like tic-tac-toe can be coded in GDL in less than 50 lines of code, whereas more complicated games, like chess or checkers, may require several hundreds lines of code.

The process of running a game goes as follows. Upon receiving a request to run a match, a program called a *game manager* first sends a start message to each player to initiate the match. The start message lists the name of the match, the role the player is to assume (for example, white or black in chess), a formal description of the associated game (in GDL), and the start clock and play clock associated with the match. The start clock determines how much time remains before play begins. The play clock determines how much time each player has to make each move once play begins. Once game play begins, the game manager sends play messages to each player to get their plays, and it then simulates the results. This part of the process repeats until the game is over. The manager then sends a stop message to each player.

Having a formal description of a game is one thing; being able to use that description to play the game effectively is something else. Since game descriptions are written in logic, game players obviously require some degree of automated reasoning. The good news is that there are powerful reasoners for GDL. The bad news is that such reasoners do not, in and of themselves, solve the real problems of general game playing, which are the same whatever representation for the game rules is used, namely, dealing with indeterminacy and size and multigame commonalities.

Year	Game Player	Developer(s)
2005	Cluneplayer	Jim Clune
2006	Fluxplayer	Stephan Schiffel, Michael Thielscher
2007	CadiaPlayer	Yngvi Björnsson, Hilmar Finnsson
2008	CadiaPlayer	Yngvi Björnsson, Hilmar Finnsson
2009	Ary	Jean Mehat
2010	Ary	Jean Mehat
2011	TurboTurtle	Sam Schreiber
2012	CadiaPlayer	Yngvi Björnsson, Hilmar Finnsson

Table 1. Winners of the International General Game Playing Competition

Progress in the Field

Over the years of the competition, general game players have become more sophisticated and significantly more powerful. There is no question that today's players can easily beat players developed early on. Partly, this has been due to tuning and tweaking, but there have also been significant innovations that have dramatically improved performance. The following are notable in this regard.

Game-Independent Heuristics

The first GGP programs introduced game-independent heuristics to deal with limited search (Kuhlmann and Stone 2006; Clune 2007; Schiffel and Thielscher 2007). These included things like mobility (the number of legal moves), inverse mobility (limiting the opponents' freedom, and goal proximity (similarity of intermediate states to goal states). While such heuristics are generally better than random play, they do not perform well in all games. Learning of game-independent heuristics is still an important research area in GGP (Kirci, Sturtevant, and Schaeffer 2011).

Learning Weights on Game Playing Heuristics

To deal with the deficiencies of game-independent heuristics, some early players utilized the start

clock period to play games and assign weights to different general heuristics, and these weights were then used during the play clock period to differentiate moves. This helped quite a bit and led Cluneplayer to victory in the first competition (Clune 2007). Unfortunately, the method is error prone. In the final game of the second competition, Cluneplayer heavily weighted inverse mobility of its opponent. Sadly, it was a variant or checkers with forced moves, and the best way Cluneplayer could find to limit its opponent's moves was to sacrifice pieces (in most cases without the opportunity to recapture).

Monte Carlo Tree Search

The most significant improvement in GGP came from the introduction of Monte Carlo tree search (MCTS) methods (Finnsson and Björnsson 2008). Rather than using general heuristics, MCTS uses run-time statistics to estimate the quality of a state, dropping a number of random depth charges to the bottom of the game tree and averaging the results. However, instead of selecting moves uniformly at random, the smartness of MCTS comes from using more informed stochastic selection strategies both in the game tree and in the rollouts. The effect was dramatic. Suddenly, automated general game players began to perform at a high level. Using this technique CadiaPlayer won the competition three times. Almost every general game playing program today uses some version of MCTS. Important research directions in MCTS in GGP include automated learning of simulation search control (Finnsson and Björnsson 2010) and effective parallelization algorithms (Méhat and Cazenave 2011).

Structural Analysis

In many games it is possible to discern structure that can be used to decrease the combinatorics of the game. Consider, for example, the game of hodgepodge, which is a combination of traditional games. If the player does not recognize that it is made up of independent subgames, it is going to search a space in which the branching factor is the product of the branching factors of the individual games. If it is able to factor the game description, it can solve the subgames independently and dramatically decrease search cost. In many cases, it is possible to find such factors in time proportional to the size of the game description rather than the size of the game graph. So there is substantial economy to be gained in doing such analysis.

The competition has just begun emphasizing games with structure of this sort. Algorithms for finding such structure have been published in the literature (Cox et al. 2009; Günther, Schiffel, and Thielscher 2009; Schiffel 2011) but so far have not been used effectively in competition.

Compilation

Game descriptions are written in logic and automated reasoning techniques can be used in generating game trees. However, GDL descriptions are very simple (essentially pure Prolog) and can be compiled into more efficient programs. Compilation does not change the asymptotic behavior of the players, but it can improve performance by orders of magnitude. Moreover, since games are finite and completely described, game descriptions are equivalent to Boolean circuits. The upshot is that they could, in principle, be compiled into hardware using field programmable gate array for even more performance improvement. While this has not yet been tried in competition, it remains a powerful idea.

Conclusion

Unfortunately, while some interesting technology has emerged from work on GGP, it has not yet found widespread application outside of GGP. It is still early in the field, and this is likely to change as games are chosen that more closely resemble real-world problems.

Perhaps the biggest problem with GGP at the moment is the name. It suggests that the task is frivolous, when in point of fact many real-world problems can be cast as games. The organizers have repeatedly toyed with the idea of renaming the competition General Problem Solving, but it seems that the name GGP is too entrenched to allow this.

Meanwhile, work goes on. There is already a well-established research community working on GGP, resulting in numerous publications including several doctoral theses (Clune 2008; Schiffel 2011, Finnsson 2012). Also, in addition to the International competition, several other GGP events are regularly hosted, including various national GGP competitions and the biennial GIGA workshop. Even with the current formulation of the field there is still room for progress. Once this subsides, there are several variations waiting in the wings.

General Game Playing with Incomplete Knowledge

In current GGP, players do not know the moves of their opponents (in advance), but they know the full details of the game world. In GGP with incomplete knowledge they do not even have complete information about the game world. For example, they may not know the initial state (as in Battleship). Or there may be probabilistic elements, as in card games. Already two languages have been developed for such games, IGDL and GDL-II (Thielscher 2011), and there are some rudimentary players capable of playing games described in these languages.

Inductive General Game Playing

The main innovation in inductive general game playing (IGGP) is that the players are not provided with rules but only instances of games and they are left to induce the rules for themselves. Early research work in this direction is already underway (Björnsson 2012; Kaiser 2012)

Really General Game Playing

Really general game playing (RGGP) takes this progression one step farther. In RGGP, the players are given a characterization of sensors and effectors and a utility meter. Their goal is to function in the world in such a way as to maximize their utility, knowing nothing else about the world. This is not likely to be worked on soon, though some students have experimented with various approaches that could be applied.

For More Information

For more details on general game playing and the International GGP Competition, visit the competition website (games.stanford.edu). Other valuable GGP resources include www.general-game-playing.de and www.ggp.org.

Notes

1. See www.coursera.org/course/ggp.

References

Björnsson, Y. 2012. Learning Rules of Simplified Boardgames by Observing. In *Proceedings of the 20th European Conference on Artificial Intelligence* (ECAI-12), 175–180. Amsterdam, The Netherlands: IOS Press.

Campbell, M.; Hoane A. J.; and Hsu, F.-H. 2002. Deep Blue. *Artificial Intelligence* 134(1–2): 57–83.

Clune, J. 2007. Heuristic Evaluation Functions for General Game Playing. In P*roceedings of the 22nd AAAI Conference on Artificial Intelligence* (AAAI-07), 1134–1139. Menlo Park, CA: AAAI Press.

Clune, J. E. 2008. Heuristic Evaluation Functions for General Game Playing. Ph.D. dissertation, Department of Computer Science, University of California Los Angeles.

Cox, E.; Schkufza, E.; Madsen, R.; and Genesereth, M. 2009. Factoring General Games Using Propositional Automata. Paper presented at the IJCAI-09 Workshop on General Game Playing (GIGA-09), July 11, Pasadena, CA.

Finnsson, H. 2012. Simulation-Based General Game Playing. Ph.D. dissertation, School of Computer Science, Reykjavik University.

Finnsson, H., and Björnsson, Y. 2008. Simulation-Based Approach to General Game Playing. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence* (AAAI-08), 259–264. Menlo Park, CA: AAAI Press.

Finnsson, H.. and Björnsson, Y. 2010. Learning Simulation Control in General Game-Playing Agents. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence* (AAAI-10), 954–959. Menlo Park, CA: AAAI Press.

Genesereth, M. R.; Love, N.; and Pell, B. 2005. General Game Playing: Overview of the AAAI Competition. *AI Magazine* 26(2): 62–72.

Günther, M.; Schiffel, S.; and Thielscher, M. 2009. Factoring General Games. Paper presented at the IJCAI-09 Workshop on General Game Playing (GIGA-09), July 11, Pasadena, CA.

Kaiser, L. 2012. Learning Games from Videos Guided by Descriptive Complexity. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence* (AAAI-12), 963–970. Menlo Park, CA: AAAI Press.

Kirci, M.; Sturtevant, N.R; and Schaeffer, J. 2011. A GGP Feature Learning Algorithm, *Künstliche Intelligenz* 25(1): 35–41.

Kuhlmann, G., and Stone, P. 2006. Automatic Heuristic Construction in a Complete General Game Player. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence* (AAAI-06), 1457–1462. Menlo Park, CA: AAAI Press.

Love, N.; Hinrichs, T.; and Genesereth, M. 2006. General Game Playing: Game Description Language Specification. Technical Report, April 4, 2006, Computer Science Department, Stanford University.

Méhat, J., and Cazenave, T. 2011. A Parallel General Game Player. *Künstliche Intelligenz* 25(1): 43–47.

Schiffel, S. 2011. Knowledge-Based General Game Playing. Ph.D. dissertation, Department of Computer Science, Technische Universität Dresden.

Schiffel, S., and Thielscher, M. 2007. Fluxplayer: A Successful General Game Player. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence* (AAAI-07), 1191–1196. Menlo Park, CA: AAAI Press.

Shannon, C. E. 1950. Programming a Computer for Playing Chess. *Philosophical Magazine* 41(314): 256–275.

Thielscher, M. 2011. GDL-II. *Künstliche Intelligenz* 25(1): 63–66.

Michael Genesereth is an associate professor in the Computer Science Department at Stanford University. He received his Sc.B. in physics from MIT and his Ph.D. in applied mathematics from Harvard University. Professor Genesereth is most known for his work on computational logic and applications of that work in enterprise computing, computational law, and general game playing. He is the current director of the Logic Group at Stanford and founder and research director of CodeX (the Stanford Center for Legal Informatics). He initiated the International General Game Playing Competition in 2005.

Yngvi Björnsson is an associate professor in the School of Computer Science at Reykjavik University. He received his Ph.D. in computer science from University of Alberta, Canada. He has been active in the computer games research community for many years and is the coauthor of the CadiaPlayer GGP agent. He is a cofounder and the current director of the CADIA research lab at Reykjavik University.