General Board Game Concepts

Éric Piette, Matthew Stephenson, Dennis J.N.J. Soemers and Cameron Browne
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, the Netherlands
{eric.piette,matthew.stephenson,dennis.soemers,cameron.browne}@maastrichtuniversity.nl

***Abstract*—Many games often share common ideas or aspects between them, such as their rules, controls, or playing area. However, in the context of General Game Playing (GGP) for board games, this area remains under-explored. We propose to formalise the notion of "game concept", inspired by terms generally used by game players and designers. Through the Ludii General Game System, we describe concepts for several levels of abstraction, such as the game itself, the moves played, or the states reached. This new GGP feature associated with the ludeme representation of games opens many new lines of research. The creation of a hyper-agent selector, the transfer of AI learning between games, or explaining AI techniques using game terms, can all be facilitated by the use of game concepts. Other applications which can benefit from game concepts are also discussed, such as the generation of plausible reconstructed rules for incomplete ancient games, or the implementation of a board game recommender system.**

***Index Terms*—General Game Playing, Concepts, Game Features, Ludii, Transfer learning, Explainable AI, Board games**

I. Introduction

Since the foundation of Artificial Intelligence (AI), games have often been used as testbeds for major advancements [1]. This can be explained by their simplicity and popularity, but also because, as humans, we can perceive the problems (game interfaces) and their solutions (strategies). In the field of game AI, two agent types can be distinguished. Dedicated Game AI agents designed for a single game (or a small set of games) and General Game AI agents built to play any arbitrary game.

Game-specific agents have been studied since the beginning of the AI field [2], [3] and are often used as benchmarks of well-known problems. For example, many single-player games, commonly called puzzles, are interesting illustrations of planning problems (e.g. Rubik's Cube [4] or Sokoban [5]) or Constraint Satisfaction Problems [6]. Similarly, multi-player games are representations of many different AI research areas (Knowledge Representation, Reinforcement Learning, Transfer Learning, etc.) which lead to the development of many game AI agents trying to outperform human players. Nowadays, computer players for classical board games like Chess [7], Go [8], Shogi [9] or Checkers [10], can defeat expert human players. Even for stochastic games like Backgammon [11], games with incomplete information like Poker [12], or video games like Starcraft II [13], computer players have reached an excellent level of play. The success of these dedicated game-playing programs has demonstrated their capacity to outperform humans, marking a milestone in artificial intelligence research [8], [14]. However, they are often highly reliant on game-specific knowledge and expertise, preventing them from playing other games as effectively.

Building general game AI agents that perform well on any board game, rather than being specialised, is the aim of General Game Playing (GGP). These general game AI agents are implemented without knowing the games they will play in advance and consequently, they cannot use specific pre-defined knowledge about them when playing the game. This differs greatly from dedicated game AI agents, as the AIs developed for specific games are often biased by human-developed heuristics. Unfortunately, humans still tend to outperform AI in general board games. This can be explained by the human capacity for understanding common aspects between games and consequently being able to learn new games quicker by detecting common points with previously played games.

This paper introduces board game concepts for GGP through the Ludii general game system. These concepts are expressed in game terms commonly used by game players and designers, making them an interesting mechanism for providing human-understandable explanations of different AI techniques. These concepts can be associated with several levels of abstraction, such as the game itself, a game trial, a game state, or individual moves. This opens many new possibilities for different AI research topics, including agent selection, transfer learning between games, AI explainability, as well as game generation, reconstruction, and recommendation.

The remainder of this paper is organised as follows: Section II describes the field of GGP and the different general game systems proposed so far. Section III describes the Ludii system as well as the ludeme-based language used to describe the games and the reasons behind its use for identifying game concepts. Section IV highlights the notion of game concepts in the context of Ludii, how they are modelled and computed for each game. Section V proposes several GGP research directions which can benefit from our proposed game concepts. Finally, Section VI concludes the paper.

II. Background

A. General Game Playing

In Artificial Intelligence, the General Game Playing (GGP) challenge [15] is to develop computer players that understand the rules of previously unknown games, and learn to play these games well without human intervention in situations where its knowledge of the games rules, objectives and strategies is limited to the game description. Developing these general

978-1-6654-3886-5/21/$31.00 ©2021 IEEE

agents, and the AI techniques behind them, is vitally important in the development of real-world agents which can deal with unpredictable and novel situations. For this reason, General Game Playing is seen as a necessary step on the way to Artificial General Intelligence (AGI) [16].

Historically, the first GGP model is defined in 1968 by Jacques Pitrat [17] to describe two-player games with complete information and rectangular boards. Then, only from the '90s, other work on General Game systems appeared such as SAL [18], Metagamer [19], Hoyle [20], and Morph-II [21]. After the turn of the millennium, more modern General Game systems started appearing such as Multigame [22] in 2001 and Zillions of Games [23] in 2002.

B. General board Game Languages

Many GGP systems describe their games using a standardised game description language. Different game descriptions are written in different formats and with different levels of abstraction, lending themselves to different kinds of knowledge representation, reasoning, and learning approaches (such as for performance reasons).

1) GDL: Since 2005, the *Game Description Language* (GDL) [24] and the GGP-Base platform [25] proposed by the Stanford Logic Group[1] has become one of the main standards for academic research in GGP. The purpose of GDL is to provide a generic language for representing any board game [26], including collaborative games and games with simultaneous actions. GDL describes the games in a language inspired by Prolog, using first-order logical clauses. An extension named GDL-II [27] has been developed to handle games with partial observations and stochastic actions, and another extension named GDL-III [28] for epistemic games. This formalism and platform have provided a high-level challenge which has led to important research contributions [29], [30] – especially in *Monte Carlo tree search* (MCTS) enhancements [31], [32], with some original algorithms combining constraint programming, MCTS, and symmetry detection [33]. These techniques have lead to the development of several General Game AI agents that perform well on specific games (e.g. Gamer [34], CadiaPlayer [35], ClunePlayer [36], WoodStock [37], etc.).

Unfortunately, GDL also has some negative points. This includes poor efficiency [38], the fact that each element of the game has to be defined tabula rasa and cannot easily be taken from previously modelled examples, and verbose game descriptions that are not representative of any game-related aspects that human players typically use. The opaque and low-level character of GDL descriptions, without any shared semantics or high-level concepts being explicitly recognisable between games, forms a challenge for tasks such as the creation of mappings between games for transfer learning, or the generation of human-understandable explanations about any aspect of a game.

2) RBG: The Regular Boardgames (RBG) system [39] proposes the idea of encoding piece movement for games using a regular language. This formalism can describe any finite deterministic game with perfect information excluding simultaneous moves. RBG game descriptions are typically much short than their GDL counterparts, making it much easier to model complex board games (e.g. Chess, Go and Arimaa). RBG also runs substantially faster than GDL, averaging over 50 times as many playouts during a recent comparison [38].

3) Ludii: The Ludii system[2], named after its predecessor LUDI [40], is a complete general board game system, that can model games as a tree of ludemes. The decomposition of games into their conceptual units of game-related information *ludemes* [41] results in more intuitive games descriptions compared to prior GGP systems. The number and variety of board games that are implemented within Ludii also go far beyond those presented by most prior general game systems [42]. Ludii is capable of modelling the full range of playable GDL games, as well as stacking games, boardless games, and games with hidden information. This, combined with high efficiency compared to other general systems [38], makes Ludii the ideal system for investigating the new game concepts-based research directions discussed in this paper.

C. Related Work

In the context of game AI research, prior work on feature extraction has been predominantly focused on video games. This includes more general frameworks such as the General Video Game AI (GVGAI) system [43], [44], but also specific video games such as Angry Birds [45] and Starcraft [46]. These papers used their extracted features to predict the performance of different agents on unknown games, creating a portfolio or ensemble agent that combines the strengths of multiple AI techniques.

Despite the demonstrated advantages of this approach, little work has been done on feature extraction for general board games. Banerjee and Stone [47] proposed a reinforcement learning game player interacting with the GGP-base system, which can transfer knowledge learned in one game to expedite learning in other games. This work is based on value-function transfer [48] where general features are extracted from the state space of a previous game and matched with the different state space of a new game. This work showed good performance for low-level features on small games, but only for a limited set of specific types of source-target pairs where appropriate state space mappings can be automatically detected based on GDL game descriptions [49].

Some of the best General Game Playing players based on GDL have shown the importance of heuristics and game features by using them to improve their performance during GGP competitions. ClunePlayer used heuristic evaluation functions to represent simplified games as abstract models, incorporating the most essential aspects of the original game to construct tailored heuristics [36]. CadiaPlayer used template matching

[1]Stanford Logic Group: logic.stanford.edu/

[2]The source code of Ludii is available at github.com/Ludeme/Ludii

to identify simple board game features, such as square tiling or specific piece types [35]. Each of these agents won the International General Game Competition [50], in 2005 for ClunePlayer and 2007, 2008, and 2012 for CadiaPlayer.

III. Ludii

Ludii[3] uses a *class grammar* approach [51] to provide a direct link between the keywords in its game descriptions and the underlying Java code that implements them. The core of Ludii is a ludeme library, consisting of several classes, each implementing a specific ludeme. Ludemes are used to define both the *form* of the game (rules and equipment) and its *function* (legal moves and outcomes for the end state).

Each game description is composed of three different sections which define the relevant information about the players (number, facing direction, etc.), the equipment (board, pieces, etc.), and the rules of the game. The rules of the game can also be decomposed into four specific sub-sections:

- **Meta-rules** applied to each state (e.g. no repetition).
- **Starting rules** defining the initial state.
- **Playing rules** defining the legal moves for a state s.
- **Ending rules** defining the conditions under which the game's outcome is determined.

An important benefit of this representation is that it allows each game description to be expressed in a compact and human-understandable manner by hiding unnecessary implementation details [52]. Figure 1 shows Ludii game descriptions for the board games Amazons[4] and Havannah,[5] alongside the graphical interfaces of playouts run on these games.

Moves in Ludii are represented using an atomic model. At each state s, the ludemes describing the playing rules are evaluated and return a list of k legal moves $\mathcal{M}$: $\langle m_1, \ldots, m_i, \ldots, m_k \rangle$. In Ludii, the transition between two successive states s_i and s_{i+1} is possible thanks to a sequence of atomic actions A_i applied on s_i. Such a sequence is modelled as a move m: $\langle a_1, \ldots, a_i, \ldots, a_n \rangle$ with n the number of actions in A_i. Thanks to this representation, it is possible to associate with each move m the concepts triggered by the list of actions defining m. Similarly, concepts can be associated with any state s, by computing the concepts involved at s.

IV. Game Concepts

A concept is an abstract representation that can be shared between different objects or ideas highlighting common points. Previous work (see Section II-C)) commonly uses the term "game feature" to designate a high-level aspect of games. However, in a board game context but also for the more general purpose of this paper, we are going to use the term "game concept" due to the existing use of the term "feature" by previous work on state-action features for board games [53].

Every game concept in Ludii is defined by its name, its category, its data type (numerical or binary) and its computation type. Each concept's name is selected to be as close as possible to terms used by human players or game designers. Version 1.2.0 of Ludii, used in this paper, currently implements 428 distinct game concepts. To organise these concepts, we propose a taxonomy[6] inspired by the Core Subject Taxonomy for Mathematical Sciences Education outlined by the Mathematical Association of America [54].

A. Categories

The seven main concept categories are:

- **Properties**: Concepts related to the format of the game (time model, information type, symmetries, players, ...).
- **Equipment**: Concepts related to the board (shape, tiling, graph, ...) and pieces (tile, dice, large piece, ...).
- **Rules**: Concepts related to each rule type (meta, start, play, end).
- **Math**: Concepts relating to fields of Mathematics (Arithmetic, Comparison, Logic, Algorithmic, ...).
- **Metrics**: Concepts describing well-know game metrics (game length, branching factor, ...).
- **Visual**: Concepts describing a game's graphical style.
- **Implementation**: Concepts describing game implementation details.

Thanks to this taxonomy, it is possible to focus only on a specific subset of categories depending on the specific application or research field. For example, many game-playing applications would have no use for the visual category, while explainable AI research will likely disregard the implementation category.

B. Data Type

A game concept can be numerical or binary. Binary concepts are used to show the existence of concepts in games, while numerical concepts are used to quantify them.

A binary game concept is activated if a specific ludeme, or a combination of ludemes, is used in the game's description. As an example, the concept Stochastic describes games involving chance elements. It can be activated by multiple ludemes in isolation, such as rolling dice (roll) or the use of any random value (value Random 1 5). Contrary to this, the concept Hop Capture will be triggered if the game involves a (move Hop ...) ludeme and a capturing effect within it, such as the ludeme (remove ...).

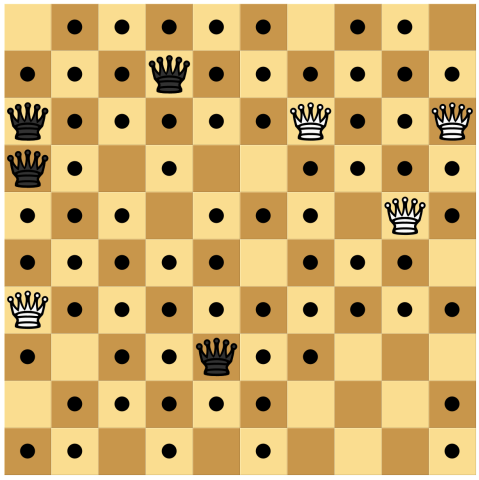
Rather than being activated or not, numerical game concepts are instead associated with a value. This can be an integer, such as the number of players (Num Players) or the number of playable sites (Num Playable Sites), or a float, such as the average number of possible directions per site (Num Directions). Some numerical concepts also correspond to the frequency of binary concepts, such as the average number of times a specific terminal state is reached, e.g., in Havannah, the frequency of winning with a loop (Loop End) compare to connected regions (Connection End), or the average number of times a specific move type was made, e.g. In Amazons, the frequencies of the concepts Slide or Shoot.

[3]Ludii is available at ludii.games/download.php

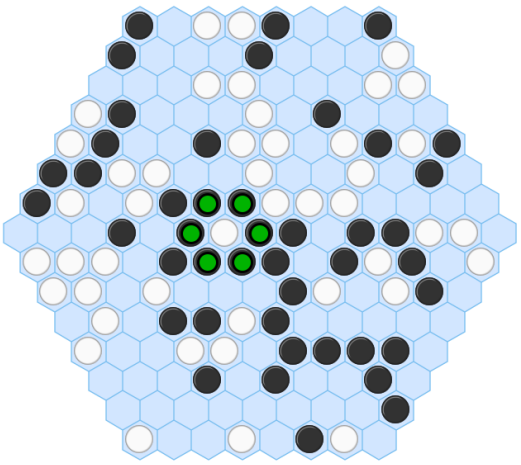
[4]Game of the Amazons: en.wikipedia.org/wiki/Game_of_the_Amazons

[5]Havannah: en.wikipedia.org/wiki/Havannah

[6]The proposed taxonomy can be found at ludii.games/searchConcepts.php



```
(game "Amazons"
    (players 2)
    (equipment {
        (board (square 10))
        (piece "Queen" Each (move Slide (then (moveAgain))))
        (piece "Dot" Neutral)
    })
    (rules
        (start {
            (place "Queen1" {"A4" "D1" "G1" "J4"})
            (place "Queen2" {"A7" "D10" "G10" "J7"})
        })
        (play
            (if (is Even (count Moves))
                (forEach Piece)
                (move Shoot (piece "Dot0"))
            )
        )
        (end (if (no Moves Next) (result Mover Win)))
    )
)
```



```
(game "Havannah"
    (players 2)
    (equipment {
        (board (hex 8))
        (piece "Marker" Each)
    })
    (rules
        (play (move Add (to (sites Empty))))
        (end
            (if
                (or {
                    (is Loop)
                    (is Connected 3 SidesNoCorners)
                    (is Connected 2 Corners)
                })
                (result Mover Win)
            )
        )
    )
)
```

Fig. 1: Completed games of Amazons and Havannah, on the Ludii system, along with their ludeme-based game descriptions.

C. Computation

Within the Ludii software, game concepts can be obtained in two different ways. One is done near-instantaneously during game compilation (compilation concepts), the other requires running playouts (playout concepts). This distinction is important depending on the intended application. For example, tasks that have to be responsive cannot use the concepts that require playouts, as these take significantly longer to compute.

Compilation concepts are based only on the ludemes used to describe static properties of the game, such as the dimensions of the game board. During the compilation process of Ludii games [51], the existence of specific ludemes or combinations of ludemes trigger each of these concepts. Binary concepts are simply activated by the existence of ludemes while numerical concepts instead have their values set, such as the number of component types (NUM COMPONENT TYPES).

Each atomic action in Ludii can return the concepts triggered by their application to a game state, consequently, game concepts can be associated with each move played instantaneously, due to the atomic-action representation of a move. However, at a higher level such as the game, playout concepts require more time to be computed. All concepts relating to the frequency of specific binary concepts are based on playouts. Each playout corresponds to a sequence of moves called a trial τ: $\langle m_1, \ldots, m_i, \ldots, m_{ter} \rangle$ with m_{ter} the last move played reaching the terminal state s_{ter}. Thanks to the association of each move to a set of binary concepts, each trial τ can compute the frequency of the concepts involved. By running many playouts and consequently obtaining many trials for a single game, the values of the frequency game concepts can be computed by averaging the frequency of the concepts in each trial. Similarly, all the concepts belonging to the Metrics category, such as GAME LENGTH or BRANCHING FACTOR, are also computed using these playouts. For all playout concepts, it is important to note that their values are dependent on the playout type, which can be obtained randomly or with different game-playing agents.

Table I shows the values for a subset of concepts on

TABLE I: Selection of game concepts detected on some popular games.

Game	Players	PlayableSites	Checkmate	SquareTiling	HexTiling	AddMove	Capture	ConnectionEnd	Stochastic
Amazons	2	64	×	✓	×	✓	×	×	×
Backgammon	2	28	×	×	×	×	✓	×	✓
Chess	2	64	✓	✓	×	×	✓	×	×
Chinese Checkers	6	121	×	×	✓	×	×	×	×
Go (19x19)	2	361	×	✓	×	✓	✓	×	×
Havannah	2	169	×	×	✓	✓	×	✓	×
Hex (11x11)	2	121	×	×	✓	✓	×	✓	×
Oware	2	14	×	×	×	×	✓	×	×
Shogi	2	95	✓	✓	×	×	✓	×	×
Snakes and Ladders	4	100	×	✓	×	×	×	×	✓
Tic-Tac-Toe	2	9	×	✓	×	✓	×	×	×
Xiangqi	2	90	✓	✓	×	×	✓	×	×

several popular games. Due to the large number of possible game concepts, we refer readers to the webpage of each available Ludii game at ludii.games/library.php. Each game entry lists all the compilation concepts that were detected, while playout concepts were computed by running 10,000 random playouts for each game. For example, for Amazons and Havannah, the games shown in Figure 1, the concepts are shown at ludii.games/concepts.php?gameId=52 and ludii.games/concepts.php?gameId=372, respectively. We can observe that for Amazons the frequencies of the concepts SLIDE and SHOOT are both 50%, while for Havannah the frequencies of the concepts CONNECTIONEND and LOOPEND are 27% and 73%, respectively.

V. GGP RESEARCH DIRECTIONS

Having now described the different game concept categories and data types, as well as how they are computed, we turn to the different possible applications and research directions that can utilise them.

A. Agent Selection

Hyper-heuristic approaches aim to select the best agent, heuristic or strategy for a given task from a pre-defined set of choices [55]. Such approaches, especially for game AI research, typically involve identifying features between games that give an indication of how these different choices will perform [56], [57]. Our proposed game concepts may provide such features, if a correlation between the values for certain concepts and the performance of different agents can be identified.

Although not yet proven, it is highly likely that the existence of certain concepts or concept values would affect the abilities of different game-playing agents. A larger board and more pieces would typically indicate a greater branching factor, which affects the performance of certain AI techniques more than others. Likewise, agents that require full playouts to compare different moves (e.g. Monte Carlo tree search) may perform worse on games that take longer to complete, compared to agents that use defined state evaluation heuristics (e.g. Minimax).

After determining the performance of a set of agents for a given set of games, we can attempt to train models to predict the best agent for a game based on its concepts. If successful, this can provide a portfolio agent which can predict the best agent on a new game using its concept values. A similar approach has been previously presented in [58], which uses the ludemes within the games – rather than their concepts – to predict the performance of different general-game-playing heuristics. It is likely that the game concepts provide additional information which can help give more accurate predictions, thus leading to better portfolio agent performance.

B. Transfer Learning

Transfer learning research in games, and more generally reinforcement learning [59]–[61], focuses on transferring learned objects such as heuristics, value functions, policies, etc. from one or more source domains, to one or more target domains. This may involve transferring trained weights and/or representations – which may be explicit features [53], or learned representations such as those in hidden layers of deep neural networks.

Transfer of heuristics or value functions, which are functions of game states, between any pair of games requires that those games have identical state spaces, or compatible state representations such that we can create mappings between their state spaces. Similarly, transfer of policies, which are functions of game states and actions, also requires identical action spaces or compatible action representations that enable mappings between action spaces. There has been some research towards transferring value functions in GGP based on general features of the shape of a lookahead search tree [47], as well as value function transfer in a limited set of specific types of source-target pairs where appropriate state space mappings can be automatically detected based on GDL game descriptions [49].

Ludii's object-oriented, game-independent state and action representations [52] has already been demonstrated to facilitate straightforward mappings between state and action spaces of different games, allowing for effective transfer of deep neural networks with both policy and value heads between many different pairs of games [62]. However, even when such mappings are possible, transfer learning is sometimes still ineffective (or even detrimental, a phenomenon known as negative transfer [63]) due to significant differences in

the goals or optimal strategies between games. For example, the state and action representations of *Hex* and *Misère Hex* – which is *Hex* with an inverted win condition – are very similar, but neither policy nor value functions work well when transferred directly (although learned features, or hidden representations, can still be useful).

Game concepts may be viewed as "summarising" a problem or task description for a game, and hence also used to analyse in which aspects any pair of games is similar or different. This can be used to predict whether or not transfer may be successful, or more specifically to predict which aspects – only policies, or only value functions, or only features/learned representations, etc. – may successfully transfer [64], [65].

C. Explainability

Explainable AI (XAI) is an emerging field in Artificial Intelligence that has become increasingly important in recent years [66]. XAI attempts to bring transparency to how AI agents perform, by providing human-understandable explanations for the decisions they make. For many AI agents, the strategies played are generally obscure and difficult to decipher. For example, Deep Reinforcement Learning based agents such as AlphaGo [67] outperforming the best Go players, are playing strategies that are still analysed by Go experts [68] years after its victory on Lee Sedol [69]. It can be similarly difficult to understand the strategies followed by agents that do not explicitly use understandable domain knowledge, such as general game agents or puzzle solvers.

Correlations between concepts associated with played moves, and concepts associated with the states in which moves are played, may provide human-understandable insights – expressed in "game terms" – of agents' strategies. For example, identifying specific concepts triggered by moves played only from states always corresponding to the same set of concepts can explain the strategy followed by an AI agent.

For General Game agents, the good or bad performance of AI techniques/agents on games could be explained by the concepts shared between games. This explanation could be provided at a game agent level but also for the different parameters or guiding search techniques to some subsets of games but not others. For example, agents based on MCTS techniques or an Alpha-Beta search without any pre-defined knowledge, are generally achieving different performances based on the rules of the games requiring more or less time to be computing at each state (e.g. Chess-like games are generally in favour of Alpha-Beta agents because of the checkmate rule).

D. Move Evaluation

Concepts associated with moves can be used as a means of classifying them based on high-level categories, such as distinguishing capture moves from moves that do not capture. This may provide basic evaluations of the usefulness of different moves, and hence used to guide search algorithms. For example, such concepts could be used for move ordering in minimax-based agents, or to bias move selection in playouts using techniques such as FAST [32]. Such guidance will typically be fairly basic due to the high-level nature of concepts, but also highly efficient because the concepts can be extracted directly from moves generated by Ludii. This process does not require any additional computation, as would be required for features of a successor state.

E. Game Generation

Board game generation through the use of an evolutionary algorithm has been shown to provide interesting new games [40], [70]. One improvement to this work might be to generate games that result in some particular combination of game concepts that a specific player might find appealing. Game concepts could also be an important aspect of fitness functions for evaluating generated games, particularly for search-based approaches in the same vein as previous work on puzzle generation [71]. Concepts can also help to direct the selection of specific ludemes within a generated games description, by favouring ludemes that are associated with desirable concepts.

The use of game concepts to generate games can also provide new challenges. For example, the generation of games that use some novel combination of concepts that have not been previously created, perhaps leading to new and exciting gameplay properties. Similarly, the simplification of games by removing a particular concept from its rules could provide games that help teach specific rules to beginners. It has been shown in prior work [72] that agents learn faster when independently trained on separate simpler versions of a large game. This is useful in cases where training time is limited, for example in AI competitions [42], [50].

F. Game Reconstruction

As described by the Digital Ludeme Project (DLP) [73], there are a large number of historical games for which the rules are unknown or incomplete. A process similar to that of game generation could be used to reconstruct the rules of an incomplete game. By looking at the concepts used in known nearby games, we can use these to predict which concepts our incomplete game might have. In essence, for each game concept, we can calculate a probability that it was included in an incomplete game, based on the concepts present in other similar games.

G. Game Recommender Systems

Concepts can also be used to recommend new games to human players based on their preferences. If a player favours games with certain concepts, such as asymmetric games with hidden information, but dislike others, such as games with capture moves, then we can find and suggest other games in Ludii that also contain these desired combinations. Our concepts could also be combined with other board game recommendation systems, such as those developed for the online board game geek database [74]–[77], to create an even more extensive system.

Game concepts could also be used as searching criteria to find appropriate benchmark games for AI agents. Many AI

techniques are not suitable or applicable to certain games types, such as algorithms dedicated to deterministic games versus agents dedicated to stochastic games with hidden information. This applicability problem can be more specific, mainly when the techniques are based on some specific heuristics or functions such as knowledge designed for some specific game goals. The search for good benchmarks can be time-consuming and would be much easier with the use of game concepts showing the properties of games in a recommender system. Identifying smaller subsets of games that give near equal information about an agent compared to the full set has been previously investigated for the GVGAI framework [78]. By comparing the concepts present within the Ludii game corpus, we can achieve a similar means of creating a smaller game subset that still provides maximal concept coverage.

VI. Conclusion

This paper refines the notion of game concepts to General Board Game Playing. Such concepts are defined using common terms which are understood by game players and designers. These concepts can be identified automatically for games in Ludii thanks to the ludeme representation, and are associated with any level of game abstraction. Game concepts provide several interesting research possibilities, including the creation of a portfolio agent via agent or heuristic selection, transfer learning between games, explainability of AI techniques and strategies, Game Generation, Game Reconstruction and Game Recommendation.

Future work over the next few years will primarily involve identifying more game concepts and implementing their detection within the Ludii system, as well as investigating the research directions proposed in this paper. In the long term, the ludeme philosophy used to represent games could be extended to other domains, such as protein folding, physics simulations or chemical reactions, by adapting the notion of ludemes to conceptual units of each of these topics. Consequently, concepts specific to these fields could be identified and detected by similar techniques, providing a human understandable explanation for any technique applied in these fields.

Acknowledgment

This research is part of the European Research Council-funded Digital Ludeme Project (ERC Consolidator Grant #771292) run by Cameron Browne at Maastricht University's Department of Data Science and Knowledge Engineering.

References

[1] G. N. Yannakakis and J. Togelius, *Artificial Intelligence and Games*. Springer, 2018, http://gameaibook.org.

[2] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. Dev.*, vol. 3, no. 3, pp. 210–229, 1959.

[3] ——, "Programming computers to play games," *Adv. Comput.*, vol. 1, pp. 165–192, 1960.

[4] R. E. Korf, "Finding optimal solutions to rubik's cube using pattern databases," 1997, pp. 700–705.

[5] A. Junghanns and J. Schaeffer, "Sokoban: Enhancing general single-agent search methods using domain knowledge," *Artificial Intelligence*, vol. 129, p. 2001, 2001.

[6] C. Piette, É. Piette, M. Stephenson, D. J. N. J. Soemers, and C. Browne, "Ludii and XCSP: playing and solving logic puzzles," in *IEEE Conference on Games*. IEEE, 2019, pp. 630–633.

[7] T. Romstad, M. Costalba, J. Kiiski, and G. Linscott, "Stockfish: Strong open source chess engine," https://stockfishchess.org/, 2008.

[8] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[9] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[10] J. P. Patist and M. Wiering, "Learning to play draughts using temporal difference learning with neural networks and databases," in *Proc. 13th Belgian-Dutch Conf. Mach. Learn.*, 2004, pp. 87–94.

[11] G. Tesauro, "Temporal difference learning and td-gammon," *Commun. ACM*, vol. 38, no. 3, pp. 58–68, 1995.

[12] "Computer poker: A review," *Artif. Intell.*, vol. 175, no. 5, pp. 958–987, 2011.

[13] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, Ç. Gülçehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. P. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, "Grandmaster level in starcraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[14] M. Campbell, A. Hoane, and F. hsiung Hsu, "Deep blue," *Artificial Intelligence*, vol. 134, no. 1, pp. 57–83, 2002.

[15] M. Genesereth and M. Thielscher, *General Game Playing*. Morgan & Claypool, 2014.

[16] R. Canaan, C. Salge, J. Togelius, and A. Nealen, "Leveling the playing field: Fairness in ai versus human game benchmarks," in *Proceedings of the 14th International Conference on the Foundations of Digital Games*, ser. FDG '19. New York, NY, USA: Association for Computing Machinery, 2019.

[17] J. Pitrat, "Realization of a general game-playing program," in *IFIP Congress (2)*, 1968, pp. 1570–1574.

[18] M. Gherrity, "A game-learning machine," Ph.D. dissertation, Univerity of California at San Diego, 1993.

[19] B. Pell, "A strategic metagame player for general chess-like games," *Computational Intelligence*, vol. 12, 1996.

[20] S. L. Epstein, "Identifying the right reasons : Learning to filter decision makers," in *AAAI'94*. AAAI Press, 1994, pp. 68–71.

[21] R. L. Ucsc-Crl and R. Levinson, "Morph ii: A universal agent: Progress report and proposal," Tech. Rep., 1994.

[22] J. Romein, "Multigame - an environment for distributed game- tree search," Ph.D. dissertation, Vrije Universiteit Amsterdam, 2001.

[23] J. Mallet and M. Lefter, "Zillons of games: Unlimited board games & puzzles." hhttps://www.zillions-of-games.com, 1998.

[24] M. R. Genesereth, N. Love, and B. Pell, "General game playing: Overview of the AAAI competition," *AI Magazine*, vol. 26, no. 2, pp. 62–72, 2005.

[25] S. Schreiber, "Games-base repository," http://games.ggp.org/base/, 2016.

[26] M. Thielscher, "The general game playing description language is universal," in *Proceedings of the Twenty-second International Joint Conference on Artificial Intelligence, IJCAI-11*, 2011, pp. 1107–1112.

[27] S. Schiffel and M. Thielscher, "Representing and reasoning about the rules of general games with imperfect information," *Journal of Artificial Intelligence Research*, vol. 49, pp. 171–206, 2014.

[28] M. Thielscher, "GDL-III: A description language for epistemic general game playing," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1276–1282.

[29] M. Świechowski, H. Park, J. Mańdziuk, and K.-J. Kim, "Recent advances in general game playing," *The Scientific World Journal*, 2015.

[30] Y. Björnsson and S. Schiffel, "General game playing," in *Handbook of Digital Games and Entertainment Technologies*. Singapore: Springer Singapore, 2016, pp. 1–23.

[31] H. Finnsson and Y. Björnsson, "Simulation-based approach to general game playing," in *The Twenty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 2008, pp. 259–264.

[32] ——, "Learning simulation control in general game-playing agents," in *The Twenty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 2010, pp. 954–959.

[33] F. Koriche, S. Lagrue, É. Piette, and S. Tabary, "Constraint-based symmetry detection in general game playing," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 280–287.

[34] S. Edelkamp and P. Kissmann, "Gamer, a general game playing agent," *KI - Künstliche Intelligenz*, vol. 25, no. 1, pp. 49–52, 2011.

[35] H. Finnsson and Y. Björnsson, "Cadiaplayer: Search-control techniques," *Künstliche Intell.*, vol. 25, no. 1, pp. 9–16, 2011.

[36] J. E. Clune, "Heuristic evaluation functions for general game playing," *KI - Künstliche Intelligenz*, vol. 25, no. 1, pp. 73–74, 2011.

[37] F. Koriche, S. Lagrue, E. Piette, and S. Tabary, "Woodstock : un programme-joueur générique," *Rev. d'Intelligence Artif.*, vol. 31, pp. 281–310, 2017.

[38] É. Piette, D. J. N. J. Soemers, M. Stephenson, C. F. Sironi, M. H. M. Winands, and C. Browne, "Ludii – the ludemic general game system," in *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*, ser. Frontiers in Artificial Intelligence and Applications, G. D. Giacomo, A. Catala, B. Dilkina, M. Milano, S. Barro, A. Bugarín, and J. Lang, Eds., vol. 325. IOS Press, 2020, pp. 411–418.

[39] J. Kowalski, M. Maksymilian, J. Sutowicz, and M. Szykuła, "Regular boardgames," in *The Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 2019.

[40] C. B. Browne, "Automatic generation and evaluation of recombination games," Ph.D. dissertation, Queensland University of Technology, 2009.

[41] D. Parlett, "What's a ludeme?" in *Game Puzzle Design*, vol. vol. 2, 2016, pp. 83–86.

[42] M. Stephenson, É. Piette, D. J. N. J. Soemers, and C. Browne, "Ludii as a competition platform," in *CoG*. IEEE, 2019, pp. 634–641.

[43] D. Perez-Liebana, S. Samothrakis, J. Togelius, S. M. Lucas, and T. Schaul, "General video game ai: Competition, challenges, and opportunities," in *Proc. 30th AAAI Conf. Artif. Intell.*, ser. AAAI'16, 2016, pp. 4335–4337.

[44] P. Bontrager, A. Khalifa, A. Mendes, and J. Togelius, "Matching games and algorithms for general video game playing," in *AIIDE*, 2016.

[45] M. Stephenson and J. Renz, "Creating a hyper-agent for solving angry birds levels," in *AIIDE*. AAAI Press, 2017, pp. 234–240.

[46] D. Churchill and M. Buro, "Portfolio greedy search and simulation for large-scale combat in starcraft," in *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, 2013, pp. 217–224.

[47] B. Banerjee and P. Stone, "General game learning using knowledge transfer," in *The 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 672–677.

[48] B. Banerjee, G. Kuhlmann, and P. Stone, "Value function transfer for general game playing," in *ICML workshop on Structural Knowledge Transfer for Machine Learning*, 2006.

[49] G. Kuhlmann and P. Stone, "Graph-based domain mapping for transfer learning in general games," in *Machine Learning: ECML 2007*, ser. Lecture Notes in Computer Science (LNCS), J. Kok, J. Koronacki, R. Mantaras, S. Matwin, D. Mladenič, and A. Skowron, Eds., vol. 4071. Springer, Berlin, Heidelberg, 2007, pp. 188–200.

[50] M. R. Genesereth and Y. Björnsson, "The international general game playing competition," *AI Magazine*, vol. 34, pp. 107–111, 2013.

[51] C. B. Browne, "A class grammar for general games," in *Advances in Computer Games*, ser. LNCS, vol. 10068, Leiden, 2016, pp. 167–182.

[52] É. Piette, C. Browne, and D. J. N. J. Soemers, "Ludii game logic guide," *CoRR*, vol. abs/2101.02120, 2021.

[53] C. Browne, D. J. N. J. Soemers, and É. Piette, "Strategic features for general games," in *KEG@AAAI*, ser. CEUR Workshop Proceedings, vol. 2313. CEUR-WS.org, 2019, pp. 70–75.

[54] M. A. of America, *Core Subject Taxonomy for Mathematical Sciences Education*, 2004. [Online]. Available: https://www.maa.org/press/periodicals/loci/joma/subject-taxonomy

[55] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: a survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.

[56] A. Mendes, J. Togelius, and A. Nealen, "Hyper-heuristic general video game playing," *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 94–101, 2016.

[57] J. Li and G. Kendall, "A hyperheuristic methodology to generate adaptive strategies for games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 1, pp. 1–10, 2017.

[58] M. Stephenson, D. J. N. J. Soemers, E. Piette, and C. Browne, "General game heuristic prediction based on ludeme descriptions," *CoRR*, vol. abs/2105.12846, 2021.

[59] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," in *Journal of Machine Learning Research*, S. Mahadevan, Ed., vol. 10, no. 56, 2009, pp. 1633–1685.

[60] A. Lazaric, "Transfer in reinforcement learning: a framework and a survey," in *Reinforcement Learning*, ser. Adaptation, Learning, and Optimization, M. Wiering and M. van Otterlo, Eds., vol. 12. Springer, Berlin, Heidelberg, 2012, pp. 143–173.

[61] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *CoRR*, vol. abs/2009.07888, 2020. [Online]. Available: https://arxiv.org/abs/2009.07888

[62] D. J. N. J. Soemers, V. Mella, É. Piette, M. Stephenson, C. Browne, and O. Teytaud", "Transfer of fully convolutional policy-value networks between games and game variants," *CoRR*, vol. abs/2102.12375, 2021.

[63] W. Zhang, L. Deng, L. Zhang, and D. Wu, "Overcoming negative transfer: A survey," *CoRR*, vol. abs/2009.00909, 2020. [Online]. Available: https://arxiv.org/abs/2009.00909

[64] H. Bou Ammar, "Automated transfer in reinforcement learning," Ph.D. dissertation, Maastricht University, Maastricht, the Netherlands, 2013.

[65] H. Bou Ammar, E. Eaton, M. E. Taylor, D. C. Mocanu, K. Driessens, G. Weiss, and K. Tuyls, "An automated measure of mdp similarity for transfer in reinforcement learning," in *Proceedings of the Interactive Systems Workshop at the American Association of Artificial Intelligence (AAAI)*, 2014, pp. 31–37.

[66] A. Das and P. Rad, "Opportunities and challenges in explainable artificial intelligence (XAI): A survey," *CoRR*, vol. abs/2006.11371, 2020. [Online]. Available: https://arxiv.org/abs/2006.11371

[67] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016.

[68] Y. Zhou, *Rethinking Opening Strategy: AlphaGo's Impact on Pro Play*. CreateSpace Independent, 2018.

[69] C. Metz, "In two moves, alphago and lee sedol redefined the future," https://www.wired.com/2016/03/two-moves-alphago-lee-sedol-redefined-future/, 2016.

[70] C. B. Browne, *Evolutionary Game Design*. Springer, 2011.

[71] B. De Kegel and M. Haahr, "Procedural puzzle generation: A survey," *IEEE Transactions on Games*, vol. 12, no. 1, pp. 21–40, 2020.

[72] G. J. Kuhlmann, "Automated domain analysis and transfer learning in general game playing," Ph.D. dissertation, University of Texas at Austin, 2010. [Online]. Available: http://repositories.lib.utexas.edu/handle/2152/ETD-UT-2010-08-1975

[73] C. Browne, "Modern techniques for ancient games," in *IEEE Conf. Comput. Intell. Games*, 2018, pp. 490–497.

[74] J. Kim, J. Wi, S. Jang, and Y. Kim, "Sequential recommendations on board-game platforms," *Symmetry*, vol. 12, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2073-8994/12/2/210

[75] M. Ion, D. Sacharidis, and H. Werthner, "Designing a recommender system for board games," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, ser. SAC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1465–1467.

[76] G. Cheuque, J. Guzmán, and D. Parra, "Recommender systems for online video game platforms: The case of steam," in *Companion Proceedings of The 2019 World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 763–771.

[77] J. Zalewski, M. Ganzha, and M. Paprzycki, "Recommender system for board games," in *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*, 2019, pp. 249–254.

[78] M. Stephenson, D. Anderson, A. Khalifa, J. Levine, J. Renz, J. Togelius, and C. Salge, "A continuous information gain measure to find the most discriminatory problems for ai benchmarking," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 92–99.