Let the System Learn a Game: How Can FCA Optimize a Cognitive Memory Structure

William Dyce, Thibaut Marmin, Namrata Patel, Clement Sipieter, Guillaume Tisserant, and Violaine Prince

University Montpellier 2 and LIRMM-CNRS
Montpellier, France
{William.Dyce,Thibaut.Marmin,Namrata.Patel,
Clement.Sipieter}@etud.univ-montp2.com
{Tisserant,Prince}@lirmm.fr

Abstract. The goal of this article is to study the contribution of FCA (Formal Concepts Analysis) to (1) optimize (2) organize (3) discover new concepts or a better operation of the semantic memory of an Artificial Intelligence (AI) system based on a cognitive approach. The system has been applied to game modeling (here the Reversi board game), since games are a very good experimental field for performance evaluation. After describing the COGITO project, which tries to assess the pros and cons of cognitive modeling over pure operational but non explicative paradigms in games modeling, the paper stresses out the benefits of FCA in providing a better abstraction, and a more reliable way to handle conflictual knowledge.

Key words: Artificial Intelligence, Cognitive Modeling, Games, Semantic Memory, Formal Concepts Analysis

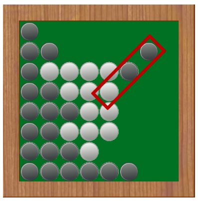
1 Introduction

One of the old dreams of Artificial Intelligence (AI) was to substitute humans with AI systems, in most of the chores involving problem solving. During the past fifty years, two methodological tracks have been extensively explored: Either imitating human behavior, a trend that naturally leads researchers to mimic the human cognitive structure, seen as the outcome of a natural selection [9]; Or definitely assessing that humans and computers are utterly different, and designing algorithms fitted to machines, thus bypassing human skills in problem solving [12]. If the first trend seems nowadays set aside because of its too many failures, this paper attempts to revive some of its claims, by constraining the project to a very simple task. This task, a REVERSI board game [7], has interesting basic properties:

– In a cognitive approach, games require different mechanisms: Capturing an input, trying to map it with the present memory state, learning it if new, and exploiting the integrated shape through reasoning when playing a new game. Thus, the behavior of a cognitive-based system could easily be tracked in its different steps.

– A totally different approach, the Minimax (also called the Von Neuman theorem, [11]), has given very good results. However, the Minimax is a way to win in a zero sum play, not a way to learn or to understand.
– This situation enables the evaluation of the pros and cons of a cognitive approach, versus a pragmatically performant but non explicative method, for a given task (even if more or less biased).

Fig. 1. A noteworthy pattern on a Reversi Board using the *aligned* predicate



This paper describes a part of a more extensive project named COGITO (both a research team, and an implemented software), restricted to the management and operation of the semantic memory , and the mechanisms that acquire (i.e. learn) or exploit (i.e. play) the knowledge required to learn to play a Reversi game.These aspects are implemented and functional. The goal of this article is, after describing the founding assumptions and the selected cognitive model, to study the contribution of FCA (Formal Concept Analysis) to (1) optimize (2) organize (3) discover new concepts or a better operation of the memory.

2 Designing a Reversi Board Game and Player

Figure 1 shows a Reversi board, with its black and white pieces. The aim of the game is to transform the adversary's pieces into one's own by placing the piece in such a way that it blocks the other's expansion. Thus, the play relies on noteworthy patterns that help the player develop winning strategies. There is a very scarce literature in AI applied to Reversi. In fact, a more modern and Japanese version, named Othello, has much more interested researchers. Rosenbloom [8] was the first to implement an Othello program (IAGO). Then, Lee and Mahajan have enhanced the program performances in their BILL program [4]. Another software, Logistello, has been developed by Buro, [1] who has further provided asurvey of Othello evolution [2]. All implementations were based on minimax evaluation functions. Later on, the game has been modeled with neural networks by [3], as a tentative approach to introduce cognitive-based models. Our attempt is the first to step from a performance-based software into a reasoning-based one.

2.1 The Game Requirements and their Modeling

In a computational framework, the play has been modeled with **predicates** that are the founding elements which compose these noteworthy patterns. The retained basic predicates are the following:

1. $isMine(x)$: For the system, its own pieces
2. $isOpp(x)$: The adversary's pieces
3. $isEmpty(x)$: A position on the board which can possibly be occupied by a next move
4. $isEdge(x)$: A noteworthy position on the edge of the board. The piece that occupies it is harder to take.
5. $isCorner(x)$: Also a noteworthy position.
6. $near(x, y)$: Defines neighborhood. Might lead to a capture, if x are not of the same color as y.
7. $aligned(x, y, z)$: Three pieces on a same line, either vertical, horizontal, or even a diagonal. Allows to capture the two other pieces, if z is not of the same color as x and y.

To implement the game, one needs to:

– Acquire the board 'state', also called the **board configuration**. It requires the coordinates of all pieces, and which predicates each piece instantiates.
– Map the present board to a set of stored noteworthy patterns that express playing strategies.
– Choose the best and thus perform a **move**.
– Learn moves from the other player in order to enhance the system abilities.

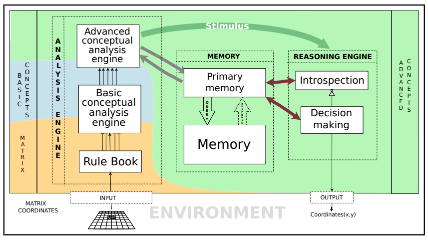


Fig. 2. The General Schema of the Implemented Cognitive Structure

2.2 The Implemented Cognitive Structure: Memory and Reasoning

The theoretical computational model underlying the design of such a requirement set is provided in figure 2. The board is described as a matrix, and is transmitted, from the *environment* to the *Rulebook* module, through an I/O module. The latter determines all the possible moves, and generates the set of all resulting matrices, to be transmitted to the *Basic Conceptual Analyzer*. This package transforms the matrices into a logical set of first order formulas, using the basic predicates defined above. The outcome is a set of facts in a logical format, transmitted to the *Advanced Conceptual Analyzer*. The latter maps the possible patterns of the board with the already stored noteworthy patterns. Then, it launches the *reasoning module* which chooses between the possible moves, according to the set of board configurations and recognized noteworthy patterns. This choice is based on an evaluation associated with the pattern, representing the number of times it has figured in a winning game (in the form of a 'probability of winning' with the appropriate formula). This cognitive structure schema has been largely inspired from the 'artificial consciousness model' in [10]. The latter involves a much larger set of elements and relationships. The COGITO project work has mostly focused on the memory and reasoning parts of the model. As seen here, the primary memory is a temporary buffer that stores the results of perceived inputs (short term memory). The memory module contains two other parts: An *episodic memory*, storing games and moves as they have been played during different sessions, and the *semantic memory*, keystone of this contribution. Both have exactly the same structure, and the episodic memory content is 'appended' to the semantic memory.

3 The Semantic Memory Structure

3.1 Memory as a Graph Structure

Fig. 3. The Semantic Memory Matrix

	RPBS_1	RPBS_2	RPBS_3	RPBS_4	RPBS_5
CBS_1	●	●		●	
CBS_2	●		●	●	●
CBS_3	●	●	●		
CBS_4		●		●	●
CBS_5	●		●		●
CBS_6				●	
CBS_7	●	●	●	●	
CBS_8		●	●	●	

The semantic memory stores:
(1) *Objects*, that represent board configurations met during different games. These objects are implemented as classes named **Complete Board States** or **CBS**.
(2) *Attributes*, for those noteworthy patterns added, all along, by the reasoning module introspective part, and named **Relevant Partial Board States** or **RPBS**.
(3) Relationships between boards and patterns, i.e. between CBS and RPBS.
Figure 3 is a representation of the semantic memory content. As such, this has naturally led us to consider two possible approaches for shaping and formalizing the semantic memory:
(1)A bi-part graph, where objects and attributes are nodes, and their edges standing for their mutual relationships, such as in figure 4. The prefix 'master' seen in this graph allows typing any graph (from the episodic memory), appended to the stored parts of the long term semantic memory (Figure 5 shows how the semantic memory is upgraded with parts coming out of the episodic memory). The root node suggests here an artificial referential element, neutral to the relationship 'edge'.
(2) A conceptual model obtained after applying FCA.
The graph representation has been implemented here with a graph oriented

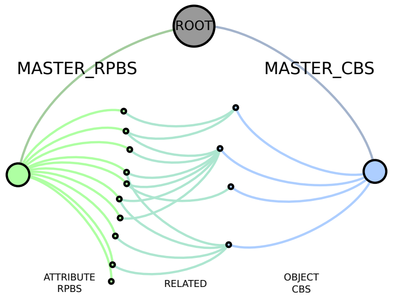


Fig. 4. A Graph Representation of the Semantic Memory

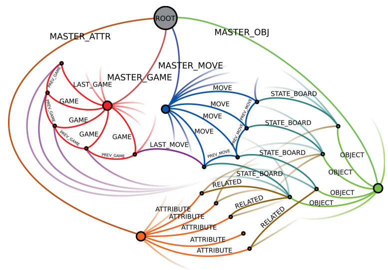


Fig. 5. Appended Graphs of the Episodic Memory, Completing the Semantic Memory Graph

DBMS (an open source system called Neo4j), and exploited. It works quite well in most of the cases.

However, observation has shown the following liabilities, that were transformed into requirements for an FCA modeling:

- The abstraction level is quite low, and still too close to the operational requirements of the game.
- When reasoning on patterns as pure attributes, any composition of patterns inherits the valuation of its members. For instance, if two 'winning' RPBS, when associated, could generate a conflict, the present approach would not detect it.
- It is possible that the information appended from the episodic memory and some already existing parts of the semantic memory, turn out to be redundant. The present model does not prevent such a situation, neither does it cure it.

3.2 The Contribution of FCA to the Semantic Memory Organization

FCA (Formal Concepts Analysis) [6] helps organizing and structuring information presented as a collection of objects and their properties. Figure 3 shows that the semantic memory content is a very good candidate for such a design. Thus, it has been performed on the CBS/RPBS matrices presented in Figure 3, using Concept Explorer (Conexp, [13]) an open source concept lattice builder. A recent work on a neighboring application, related to semantic neural decoding [5] has encouraged such an attempt. Human cognitive structures are neural, and an imitative model, such as our Cognitive Semantic Memory, would probably benefit from the same achievements. The discussed improvements are those described in the following subsections.

Reducing Redundancy, Optimizing Decision Making, Evaluating Patterns Quality and Discovering New Patterns In Figure 6, three concepts introduce more than five noteworthy patterns (RPBS). This helps to reduce redundancy, as mentioned previously, by merging these patterns into one, without any loss of information on the whole set of acquired data. Let us note that FCA concepts are sets of CBS sharing the same properties. This means that games could be put into categories, and a classification of winning games and their hierarchy, can be extracted from this work (see next subsection).

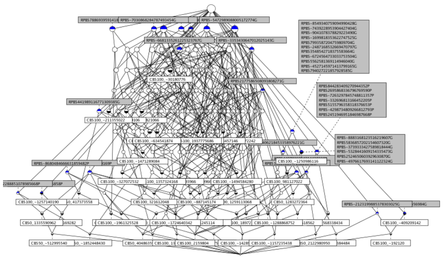


Fig. 6. An overview of the Concepts Lattice: Boards are Objects, and Patterns, Attributes

Also, a hierarchy among the RPBS seems to appear. This might lead to a more intelligent search on the matching RPBS (in the reasoning module of Figure 2) by reducing the number of homomorphisms applied to each board. Thus, the decision making might take less time, without reducing the number of RPBS in the search space.

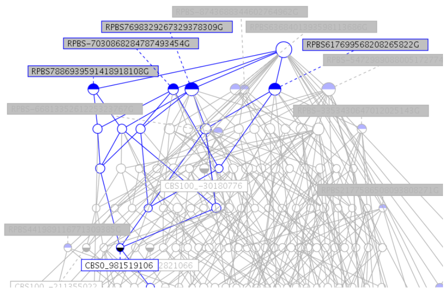


Fig. 7. noteworthy Patterns emerging as Common Parts of Concepts, i.e. Game Boards

The RPBS introduced in the parent concepts of a common concept (introducing at least one CBS) have potentially a common subpart, that could be extracted as a new (and efficient) RPBS (see Figure 7).
Also, RPBS have been weighted, in the COGITO system, according to their presence frequency in winning, respectively losing games. We have tried to build lattices based on this feature. Figure 8 shows the organization of the winning games. Such a piece of information is crucial since it might help modifying the RPBS weights.

Discovering New Rules and Strategies Moreover, and unexpectedly, Conexp has helped us find rules featured as follows, that might be caused by inclusions of RPBS:

$$r < n > RPBS_{id}(w) => < m > RPBS_{id1}, RPBS_{id2}, ...RPBS_{idk} \quad (1)$$

where:

1. r is the rank of the rule. The better the rank, the more reliable the rule.
2. $< n >$ represents the number of times the RPBS identifier ($RPBS_{id}$) in the hypothesis appears in a CBS.
3. m is the number of times the conclusion is found.
4. w represents the confidence associated to the rule. if w is equal to 100, it means that each time the $RPBS_{id}$ is found, there is a 100 percent chance that it is followed by the $RPBS_{id1}, RPBS_{id2}, ...RPBS_{idk}$ of the conclusion. w is the representation of n/m. The following extract shows a few among those that have been found by the system.

This means that beyond the relationships between concepts (CBS) and attributes (RPBS), FCA helps discovering possible dependancies amongst attributes themselves, leading to a re-design of the noteworthy pattern notion.
Samples of Derived Rules

```
1 < 7 >    RPBS2451946951846987668P [100 \%]
   ==> < 7 > RPBS-4298734809266812793P   RPBS5155796358318376653P
    RPBS-3326968131664452205P  RPBS-7263297845748811357P
      RPBS2695868336796769590P  RPBS844283409270944352P;
 [..]
58 < 20 > RPBS6368401393598113686G  [95 \%]=>
 < 19 > RPBS6176995682082658 22G;
[..]
```

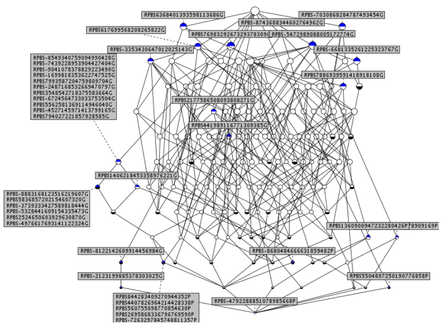


Fig. 8. A Lattice of the Winning Games and the Patterns they used

4 Conclusion

The experiment has shown that FCA can provide very interesting modifications to the initial memory structure. For the moment this step has not been automated, because we wanted to evaluate FCA added value: indeed, it has improved the quality and reliability of the acquired knowledge. However, FCA must not be run on a learning system, since it would bias the learning step (when the system acquires new RPBS and CBS from games played against a human player). The general idea is to re-design the memory with FCA, and this time, automatically, but only after a reasonable number of games where the memory has more or less acquired elements, and to rerun FCA only until another quite large number of games have been played. During the game step, it would be valuable to store the ongoing CBS in a concept. Concepts can be 'weighted' with values expressing their reliability. Thus the ongoing CBS can inherit its parent concept present weight and, benefiting from the lattice structure, drastically reduce the number of possible RPBS (in the reasoning module). Also, it would be interesting to constantly check stabilization in learning, in order to build a sort of a final lattice, which will represent a stable and 'mature' state of the semantic memory. We also anticipate that the final number of concepts will also stabilize. A future experiment will be performed to determine the final lattice size.

References

1. Buro, Michael. "The Othello Match of the Year: Takeshi Murakami vs. Logistello", ICCA Journal, vol 20, 3, P.189-193 (1997).
2. Buro, Michael. The Evolution of Strong Othello Programs", in: Entertainment Computing - Technology and Applications, R. Nakatsu and J. Hoshino (ed.), Kluwer, p. 81-88 (2003).
3. Chong, S.Y., Tan M.K, White, J.D. "Observing the evolution of neural networks learning to play the game of Othello", IEEE Transactions on Evolutionary Computation, Vol 9, 3, p.240-251(2005)
4. Lee,K; Mahajan, S. "The development of a world-class Othello program". Artificial Intelligence, vol. 43, p. 21-36.(1990).
5. Endres, Dominik M.; Foldiak, Peter; Priss, Uta. "An Application of Formal Concept Analysis to Semantic Neural Decoding", Annals of Mathematics and Artificial Intelligence, Vol, 57, 3, Springer-Verlag, p. 233-248 (2010).
6. Ganter, Bernhard; Stumme, Gerd; Wille, Rudolf, eds, "Formal Concept Analysis: Foundations and Applications", Lecture Notes in Artificial Intelligence, no. 3626, Springer-Verlag. (2005)
7. A description of the Reversi Game. http: //en.wikipedia.org/wiki/Reversi.
8. Rosenbloom, P. "A world-championship level Othello program". Artificial Intelligence, vol. 19, p.279-320. (1982).
9. Sweller, John. "Instructional Design Consequences of an Analogy between Evolution by Natural Selection and Human Cognitive Architecture". Instructional Science, 32, 1, 9-31.Springer Netherlands (2004)
10. Tisserant, Guillaume; Maurin, Guillaume; Ndongo, Wand ; Villemot, Anthony. "Rapport sur une conscience artificielle". LIRMM-CNRS research Report.(2010)
11. von Neumann, John. Zur Theorie der Gesellschaftspiele , Mathematische Annalen, vol. 100, p. 295-320(1928).
12. Warwick, Kevin. "March of the machines: the breakthrough in artificial intelligence" Univ. of Illinois. (2004)
13. Yevtushenko Serhiy A. "System of data analysis "Concept Explorer"." (In Russian),Proceedings of the 7th national conference on Artificial Intelligence KII-2000, p. 127-134, Russia, (2000).