Decision-making and strategic thinking through analogies

Alexandre Linhares[1, 2]

1. Center for Research on Concepts and Cognition, Indiana University, Bloomington

2. EBAPE / Getulio Vargas Foundation, linhares@clubofrome.org.br
Praia de Botafogo 190 office 509, Rio de Janeiro 22250-900, Brazil

Abstract. *When faced with a complex scenario, how does understanding arise in one's mind? How does one integrate disparate cues into a global, meaningful whole? Consider the chess game: how do humans avoid the combinatorial explosion? How are abstract ideas represented? The purpose of this paper is to propose a new computational model of human chess intuition and intelligence. We suggest that analogies and abstract roles are crucial to solving these landmark problems. We present a proof-of-concept model, in the form of a computational architecture, which may be able to account for many crucial aspects of human intuition, such as (i) concentration of attention to relevant aspects, (ii) how humans may avoid the combinatorial explosion, (iii) perception of similarity at a strategic level, and (iv) a state of meaningful anticipation over how a global scenario may evolve.*

Keywords: *Representation; perception; computational model; analogy; reasoning; chess*

I. Introduction

When faced with a complex scenario, how does understanding arise in one's mind? How does one integrate disparate cues into a global, meaningful, whole? Consider the chess game: how do humans avoid the combinatorial explosion? How can strategically similar situations be perceived as such? What about the problem of intuition? What type of information-processing is involved behind our subcognitive, intuitive sense of a position? Could a computer ever acquire this genuine, intuitive, understanding of a chessboard? Could it have a meaningful state of intuitive anticipation over how a particular scenario may develop? This last question has perhaps been best posed by Atkinson (1993):

> ‘‘The master's superior play is due to ‘sense of position’, an intuitive form of knowledge gained from experiencing a great variety of chess situations. Intuitive knowledge is perhaps the most important component of expertise in any human discipline, yet intuition has been one of our least understood phenomena. Intuition is an automatic, unconscious part of every act of perception. It is often associated with emotion: an intuitive realization can ‘feel right’ and evoke a sense of satisfaction or, equally, can elicit that sinking feeling of impending frustration. Intuition is nonverbal, yet trainable. Intuition is not part of present-day machine chess’’.

This meaningful ‘sense of a position’ comes from *abstract analogical perception*: Consider the chess positions shown in Figure 1. In the eyes of a chess expert, positions 6 and 10 are similar (Linhares and Brum 2007). Yet, to "the eyes" of a computer program, these positions are remarkably different. These positions display (i) different numbers of pieces, (ii) different types of pieces; (iii) different tree depth; (iv) different tree breadth, (v) different locations of pieces, and (v) different distances between pieces. Position 10 is solvable within a brief tree expansion, while position 6, which demands a computation of 20+ plies of the rapidly expanding game tree before any improvement in the evaluation function can be achieved, demands substantial tree expansion. However, when we *understand* these positions and see the checkmate, a remarkable similarity at a strategic-vision level emerges (Linhares and Brum, 2007). *How can strategic similarity be perceived? Where is the similarity to be found?*

Similarity at an abstract, strategic, level, is perceived not only in "calm" positions, but also in positions in which exchanges take place (8 and 20), and even in checkmates (A and B). This is the first issue we consider: to intuitively understand a position, a computational model needs to be able to account for our ability to perceive abstract, strategic, similarity.

A second issue is: How can combinatorial explosions be avoided? Position 6 demands over 20 plies before any increase in the evaluation function, so how could it be processed in few clock cycles?

Finally, there is the issue of how to measure scientific progress in developing cognitive models of chess. Is move quality still the standard to look forward to, when we do not want to construct a world champion, but rather to understand and model human intuition? In the appendix, we propose a number of constraints to illuminate this issue.

This paper proposes a new framework with which we may approach these landmark problems. A new cognitive theory of decision-making in chess, in the form of an evolving computer architecture, is proposed—which perhaps may lead to innovative research avenues in problems of understanding and semantics beyond the domain of chess.

In the next section, we will propose, following Hofstadter and FARG (1995), a new computational model for chess that is centered in analogy-making.

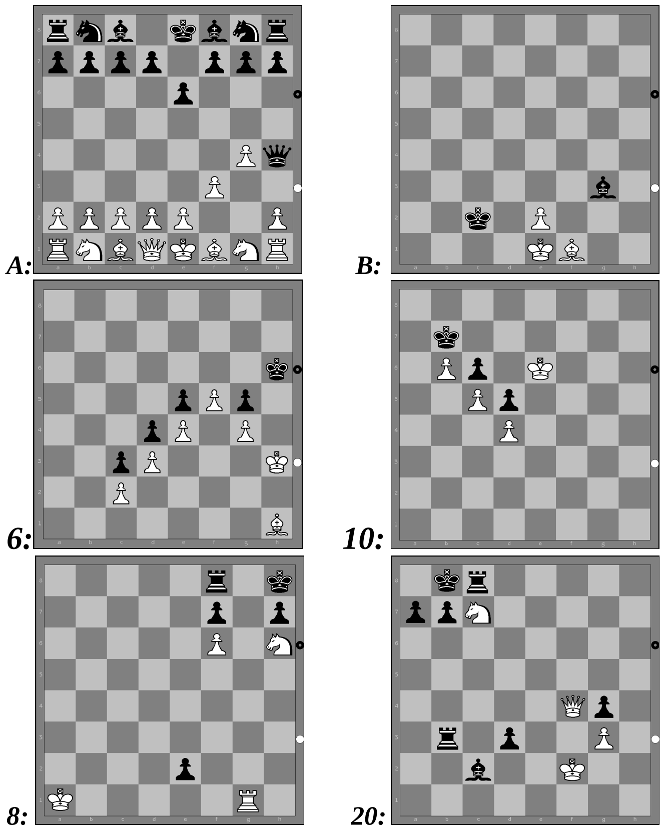


Figure 1. *Analogy enables the application of familiar strategies to new, unforeseen, situations. To demonstrate this, Linhares and Brum (2007) conducted an experiment to study "strategically similar" pairs of positions: In this display, pairs (A,B), (6,10), and (8,20) are shown. Here we can observe that analogy plays a significant role in chess cognition—experts report that these pairs are 'very similar', despite the large number of superficial differences. The positions in each pair differ in piece sets, number of pieces, position of pieces, and the underlying search tree (from the point of view of the previous ply), so how can they be similar to each other? How can a computational model perceive such abstract strategic similarity? For readers less familiar with chess, the solutions to these positions follow. Positions A and B present the fastest possible mate and its "most economical" representation (Ban, 1997, p.12). White to move and win in the remaining positions: In positions 6 and 10, black cannot defend from the simultaneous attack of white's passed pawn and king. Note that in position 6 the white king will move across the board, to eventually threaten the E5 pawn. These movements expand the game tree, involving over 20 plies. Yet, the strategic similarity to the situation in position 10 is remarkable, even with such large differences in search tree size. Positions 8 and 20 involve exchanges. In position 8—a variant of a position used in (Charness et al., 2001)—, white moves rook to g8 check, black rook captures white rook, white knight captures black pawn at f7 checkmate. In position 20, white moves knight to a6 check (by knight and by queen), black king escapes to a8, white moves queen to b8 check, black rook captures white queen, white knight returns to c7 checkmate. These variations of "Philidor's mate" display high strategic similarity and no similarity at a surface appearance level. Analogical mechanisms enable humans to perceive such strategic similarities.*

II. Capyblanca: an example architecture

What are the elements of an architecture that could display understanding of a chess position? How should it process information? What does the process look like? The following sections present our proposals.

2.1 Architectural elements

The architecture consists of the following elements:

Pressing urges and codelets – The computational processes constructing the representations on short-term memory are subcognitive processes named *codelets*. The system perceives a great number of subtle pressures that immediately invoke subcognitive urges to handle them. These urges will eventually be executed. Some of these processes may look for particular objects, some may look for particular relations between objects and create bonds between them, some may group objects into chunks, or associate descriptions to objects, etc. The collective computation of these impulsive processes, at any given time, stands for the working memory of the model. These processes are involuntary, as there is no conscious decision required for their triggering. (As George Lakoff puts it, if one asks you "not to think of an elephant", it is too late, you already have done so, in an involuntary way.) They are also automatic, as they know how to do their job without asking for help. They are fast, with only a few operations carried out. They accomplish direct connections between their micro-perceptions and micro-actions. Processing is also granular and fragmented – as opposed to a linearly structured sequence of operations that cannot be interrupted (Hofstadter and FARG 1995; Linhares 2005). Finally, they are functional, associated with a subpattern, and operate on a subsymbolic level (but not restricted to the manipulation of internal numerical parameters as opposed to most connectionist systems).

Coderack: a list of parallel priorities—Each codelet executes a local, incremental change to the emerging representation, but the philosophy of the system is that all pressing urges are perceived simultaneously, in parallel. So there is, at any given point in time, a list of subcognitive urges ready to execute, fighting for the attention of the system and waiting probabilistically to fire as a codelet. This list of parallel priorities is called the coderack in Hofstadter and FARG (1995). The name is due to an analogy with a coatrack in which any coat may be taken out of at any point in time.

A workspace that interacts with external memory—This is the working short-term memory of the model. The workspace is where the representations are constructed, with pressing urges waiting for attention and their corresponding processes swarming over the representation, independently perceiving and creating many types of subpatterns. Common examples of such subpatterns are relations between objects, awareness of abstract roles played by objects, and so on.

A semantic associative network undergoing constant flux—The long-term memory of the system is embedded into a network of nodes representing concepts with links between nodes associating related concepts. This network is a crucial part for the formation of a chain reaction of conceptual activation: any specific concept, when activated, propagates activation to its related concepts, which will in turn launch top-down expectation-driven urges to look for additional information relevant to each active concept (Linhares 2005). For example, if a bishop is found at a certain position, this perception will trigger associated concepts (and codelets) to look at the diagonals, and to construct a chess relationship structure if a piece is found. This mode of computation not only enforces a context-sensitive search but also is the basis of a chain reaction of activation spreading – hence the term 'active symbols' (Hofstadter 1979, 1985). This network is called the slipnet in Copycat. One of the most original features of the slipnet is the ability to "slip one concept into another", in which distances between concepts change through time (for details see Hofstadter 1995, Mitchell 1993).

All of these architectural elements operate on many different levels of organization, some more ‘concrete’, tied to the specifics of a position (i.e., *percepts*), and others more ‘vague’, abstractions of the content of a position (i.e., *concepts*). This is our next subject.

2.2 Multiple-leveled analysis

The architecture employs many different levels of analysis, notably:

- *Squares*—this is the most basic information processed by the system. Capyblanca is able to ‘saccade eyes’ to empty squares, to recognize that pieces may be present in its’ “peripheral vision”, and to register that a particular piece has been found at a particular square;

- *Pieces*—After a piece is found, codelets may trigger associated codelets to search for the possible trajectories that the piece may go to on the next moves; other codelets will attempt to gather an estimate of the initial “spheres of influence” of the piece, leading to another level of mobilities;

- *Piece mobilities*—This level of information processing is mainly concerned with each piece’s possible influence across the board. An initial map of movement is developed from the starting piece square. This map may be modified in due course by other codelets gathering additional information;

- *Trajectories*—If a piece A is found (randomly at first) to have a potential trajectory towards a piece B, then a codelet may create a data structure to encompass the trajectory between both pieces. Trajectories eventually created will lead to the creation of chess relations;

- *Chess relations*—If attack or defense relations are found, the system registers them in a proper structure, beyond the mere trajectory involved (note that multiple-steps attacks and defenses are registered, just as immediate, single step, relations). These relations will be measured by an *intensity function,* which takes the following form:

  $$I = 10^{-n} v^{-1}(p_1) v(p_2) d^{-1}(p_1, p_2),$$

  where p_1 represents the origin piece, p_2 the destination piece of a chess relation, $v(p_i)$ is the current system value of piece *i*, and v^{-1} its inverse function, d^{-1} is the inverse function of the minimum distance, in number of squares, that p_1 must travel to reach p_2. Intensity might be lowered by a codelet after this initial estimate, by increasing *n* (or the converse may be true, given the proper circumstances). Intensities are established by (i) the value of the pieces involved in a relationship; (ii) the distance between pieces; and (iii) the mobility (or lack thereof) a piece. For example, if a low-value pawn attacks a high-value queen in 1 move, this role of attacker receives a greater priority over the relation of a high-value queen attacking over 3 moves a low value pawn. If an attack can be intercepted, the relation also loses intensity (by other codelets manipulating *n*—note that any of the functions can be changed by codelets at any time).

- *Abstract roles*—Chess relations with the highest intensities will form a set of *abstract roles*: the current interpretation of the system about what is important in a chess position, and which pieces are involved. An example of information processing at the ‘abstract role’ level might be: if there is a double check, the only possible response is to escape with the king. This inference might thus trigger codelets, at the piece mobility level, to check whether the king has an escape route. Reasoning at the abstract role level, we propose, is what enables chess players to intuitively understand a position and to perceive abstract strategic similarity to positions that are different in all superficial aspects (Linhares and Brum 2007).

2.3 A summary of the Code

In this subsection we will provide more details about the code and the underlying architecture of the project.

The code for the current version of Capyblanca is open-sourced under a GPL license. It is hosted at capyblanca.googlecode.com. Capyblanca has been developed using object-oriented pascal and compiles with the (freely downloadable) Turbo Delphi 2005. With this initiative, we hope to contribute to the scientific community by letting those interested not only replicate the results in detail, but also improve the architecture and explore different design decisions which we have not been able to. (A major task, discussed in the conclusion, concerns the integration of learning algorithms.)

A major point concerning FARG architectures is that programs do not rely on procedure calling; they do, instead, launch tasks which are to be handled asynchronously (Hofstadter and FARG, 1995). As mentioned, the task scheduler is known as the coderack, for processes may be triggered from any part of the task queue. It is impossible to detail the thousands of lines of the whole code here, but by focusing on some of the units involved, we may have a better grasp of how the system is developed, how it works, and how one might be able to further develop it and extend the architecture.

Pascal code is divided over units. The major units involved in the current implementation of Capyblanca are detailed below.

- ***MainUnit:*** Creates the graphical user interface; loads particular chess positions into memory; initializes working memory, external memory, and the slipnet; lets the user test the system; includes some basic tests.

- ***Slipnet Unit:*** Creates basic semantic nodes and their associations (for example, a piece can be a guardian or an attacker--each of these roles has a corresponding node)

- ***ExternalMemoryUnit:*** Creates a chess position; includes basic code for piece movement, potential trajectories that each piece might access, and the piece's "spheres of influence". Includes code that will attach abstract roles to pieces (or detach them, eventually--see fig. 8)

- ***WorkingMemoryUnit:*** Creates the representation for the "ImaginedBoard", that is, the board in the system's "mind's eye". Notice that this is not equivalent to external memory. For example, external memory starts with a complete position with all pieces, while working memory is gradually filled with bottom-up, data-driven information glanced from external memory, or "imagined", top-down, expectation-driven trajectories and roles.

- ***SpecialFunctions Unit:*** GUI-related; displays only parts of graphics associated with the current representation of trajectories.

- ***BasicImpulse Unit:*** Implementation of an abstract class "TImpulse", which has the basic data structures and associated methods for use on more specific subclasses.

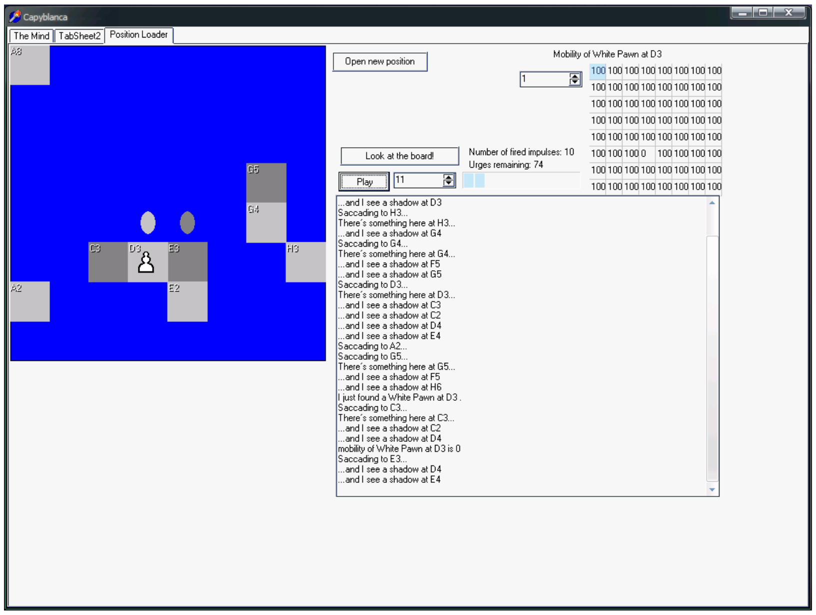
- ***ImpulseSquareAndPiece Unit:*** The impulses implemented here work at a low level, "looking" at squares in the board, creating structures in STM if a piece is found, and finding trajectories and relations between a piece and either a square or a piece.

- ***AbstractImpulses Unit:*** Handles the processing of abstract roles, such as the role of attacker, or guardian, or, in the case of a pawn, of a potential promotion, etc. Creates the corresponding roles for further processing.

- ***ImpulseConsiderations Unit:*** This is the most abstract level of processing in Capyblanca. This unit attempts to model "abstract thoughts", i.e., considerations which are NOT tied to an specific piece or square, and are in their most general form. One example is: "the only solution to a double check is to escape with the king". In this example, nothing is said about the types of pieces attacking the king, their positions, the color that is under check, etc. Another example, that of a piece that finds itself having to juggle between two different, incompatible roles, are presented in the sample run detailed below (in the case of the black king of position 6).

2.4 An example execution

In this section we describe an actual execution of Capyblanca in position 6, a variation of Wilkins' (1980) position. This is a much harder position, in fact, than that used by Wilkins, with increased search depth and breadth. How could a computer understand the position and escape from searching the tree in order to find white's advantage? Figures 2—11 provide a careful look into the intertwined perception and interpretation processes carried out by the model.



Mobility of White Pawn at D3

100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

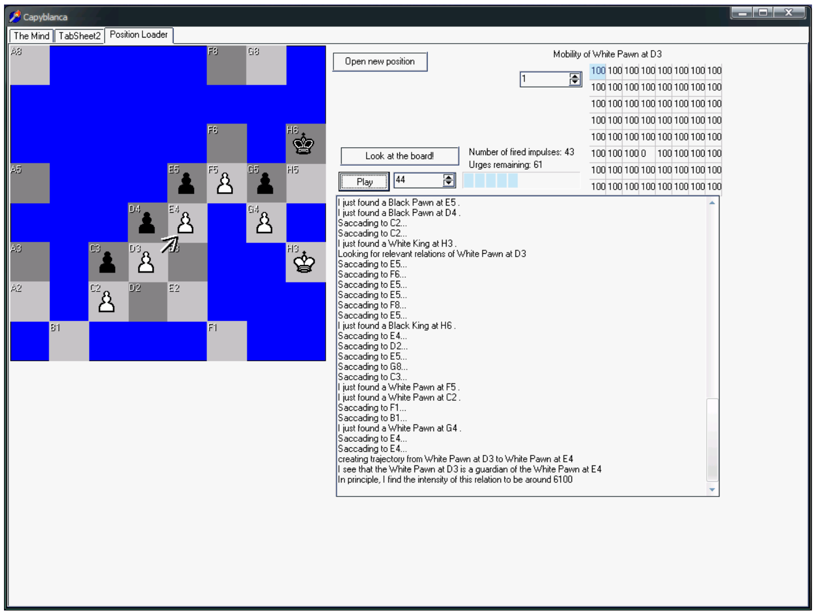
1

11

Number of fired impulses: 10
Urges remaining: 74

```
...and I see a shadow at D3
Saccading to H3...
There´s something here at H3...
...and I see a shadow at G4
Saccading to G4...
There´s something here at G4...
...and I see a shadow at F5
...and I see a shadow at G5
Saccading to D3...
There´s something here at D3...
...and I see a shadow at C3
...and I see a shadow at C2
...and I see a shadow at D4
...and I see a shadow at E4
Saccading to A2...
Saccading to G5...
There´s something here at G5...
...and I see a shadow at F5
...and I see a shadow at H6
I just found a White Pawn at D3 .
Saccading to C3...
There´s something here at C3...
...and I see a shadow at C2
...and I see a shadow at D4
mobility of White Pawn at D3 is 0
Saccading to E3...
...and I see a shadow at D4
...and I see a shadow at E4
```

Figure 2. *State 11 of an execution of Capyblanca. How do experts rapidly perceive what is relevant in a position? How can they immediately zoom in the important areas of the board? In Capyblanca, the mechanisms that enable such behavior are not explicitly pre-programmed, but emerge from the interactions of small, micro-perceptions of codelets. At start, the idea is to model the process akin to the first seconds of perceiving a position: the system has only 64 codelets, one for each board square, to eventually 'saccade eyes' (i.e., invest computational effort, or attention) to it. Since codelets are drawn at random from the coderack, one might expect that the system would randomly scan the board. That is not the case: the board is usually scanned in a way that concentrates priority to the "most important" areas. If a piece is found by a codelet, that codelet will increase the priority given to codelets looking at the squares that the piece can move to in subsequent moments of the game. To model an ability of peripheral vision, if there exists a piece in one of the (at most 8) squares surrounding a saccaded square, then a 'shadow' is perceived (i.e., a representation that something is in there, but the system is unable at that point to state exactly what). In figure 4, for example, after codelet 11, we can see some random squares which have been saccaded to, and also, a first piece is found, which increases the urgencies to look at that piece's potential trajectories. Two shadows are also found, and now the urgency to 'look at those squares' is increased. As with humans, unexpected perceptions in peripheral vision bring an urge to focus on them, and this is reflected in the system by an increased urgency to look at that square.*



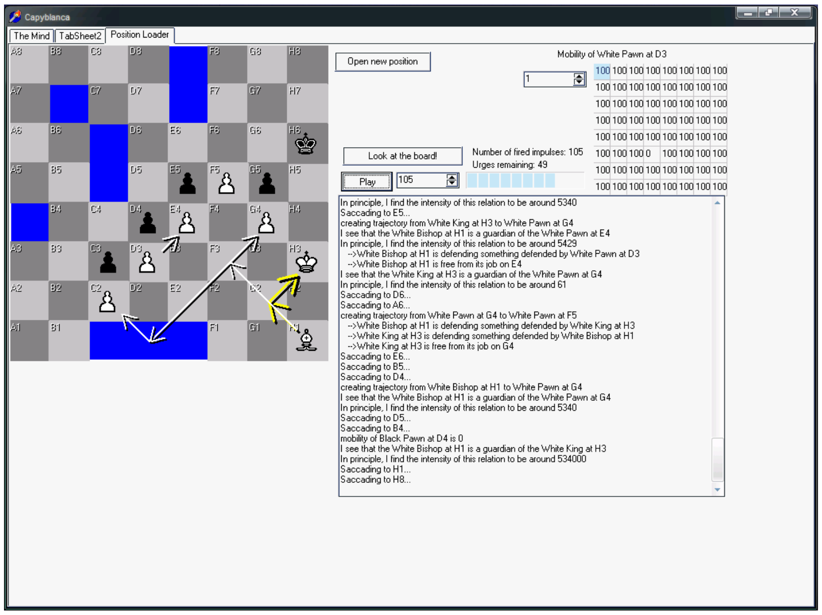
Mobility of White Pawn at D3

100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

```
1
44
Number of fired impulses: 43
Urges remaining: 61
```

```
I just found a Black Pawn at E5 .
I just found a Black Pawn at D4 .
Saccading to C2...
Saccading to C2...
I just found a White King at H3 .
Looking for relevant relations of White Pawn at D3
Saccading to E5...
Saccading to F6...
Saccading to E5...
Saccading to E5...
Saccading to F8...
Saccading to E5...
I just found a Black King at H6 .
Saccading to E4...
Saccading to D2...
Saccading to E5...
Saccading to G8...
Saccading to C3...
I just found a White Pawn at F5 .
I just found a White Pawn at C2 .
Saccading to F1...
Saccading to B1...
I just found a White Pawn at G4 .
Saccading to E4...
Saccading to E4...
creating trajectory from White Pawn at D3 to White Pawn at E4
I see that the White Pawn at D3 is a guardian of the White Pawn at E4
In principle, I find the intensity of this relation to be around 6100
```

Figure 3. *Some codelets afterwards, at state 55, all pieces but except bishop have been registered. The system has also perceived a first chess relation, even if a large empty region of the board—given by the odd-colored squares—remains still unattended. At this stage, the system is processing information at different levels, notably five: squares, shadows, pieces, piece mobilities, and immediate chess relations. Each one of those types of information may trigger subsequent codelets and significantly affect the upcoming processing. This shows how attention is quickly thrown into the 'relevant' areas of the board. Herein lies an interesting first point concerning the system: a key mechanism in low-level perception (pieces on squares) starts from a random order of 'eye saccading', to non-random attention shifts towards the most important areas of the board (as has been documented by Charness et al, 2001). It is an emergent effect, a consequence of subtle shifts in urgency given the system's acquisition of information. It is not pre-programmed a priori and at each run the process will be different—there is no efficient algorithm or data structure to rapidly register the pieces. It is actually possible, in principle, that the architecture might find the pieces only after saccading to all remaining empty squares. This theoretical possibility, however, does not arise in simulations, and the large-scale, statistically emergent, effect of attention placing is robust. In this run, merely 24 out of the 64 squares have been 'saccaded to', but these tend to be the most important ones.*



Mobility of White Pawn at D3

100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

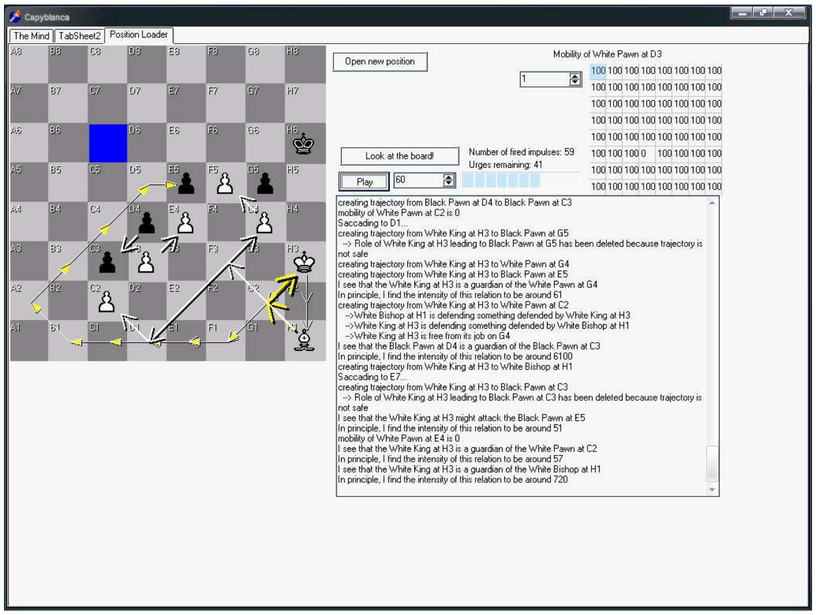
1

105

Number of fired impulses: 105
Urges remaining: 49

```
In principle, I find the intensity of this relation to be around 5340
Saccading to E5...
creating trajectory from White King at H3 to White Pawn at G4
I see that the White Bishop at H1 is a guardian of the White Pawn at E4
In principle, I find the intensity of this relation to be around 5429
   -->White Bishop at H1 is defending something defended by White Pawn at D3
   -->White Bishop at H1 is free from its job on E4
I see that the White King at H3 is a guardian of the White Pawn at G4
In principle, I find the intensity of this relation to be around 61
Saccading to D6...
Saccading to A6...
creating trajectory from White Pawn at G4 to White Pawn at F5
   -->White Bishop at H1 is defending something defended by White King at H3
   -->White King at H3 is defending something defended by White Bishop at H1
   -->White King at H3 is free from its job on G4
Saccading to E6...
Saccading to B5...
Saccading to D4...
creating trajectory from White Bishop at H1 to White Pawn at G4
I see that the White Bishop at H1 is a guardian of the White Pawn at G4
In principle, I find the intensity of this relation to be around 5340
Saccading to D5...
Saccading to B4...
mobility of Black Pawn at D4 is 0
I see that the White Bishop at H1 is a guardian of the White King at H3
In principle, I find the intensity of this relation to be around 534000
Saccading to H1...
Saccading to H8...
```

Figure 4. *State 160: note that the Bishop's "moves" are not moves in the tree-searching sense (with plies followed by the opponent). Instead, they are intended to model subcognitive pressures: the first fleeting seconds one perceives this chessboard and the Bishop's constrained area. After duly recognizing the pieces, but before fully scanning the board, Capyblanca has found that the bishop, the most powerful piece in the position, is "trying to get out", but that its mobility is restricted. One information that is displayed on the figure are the trajectories leading to the king and to the unprotected pawns. But there are two additional types of representational structure active at this point: mobility and roles. After a piece has been recognized and (some of) its potential trajectories discovered, a 'mobility matrix' is constructed to account for that piece's 'spheres of influence' across the board. This information will enable many upcoming events, and, more importantly, will restrain the possibilities for consideration (not only for that piece, but also, globally across the board). The other type of representation structure active at this point is that of an 'abstract role'. For example, the bishop is seen to play the role of 'guardian' of the unprotected pawns and of the king's square. The current intensity of these roles is displayed by the thickness of the arrows.*



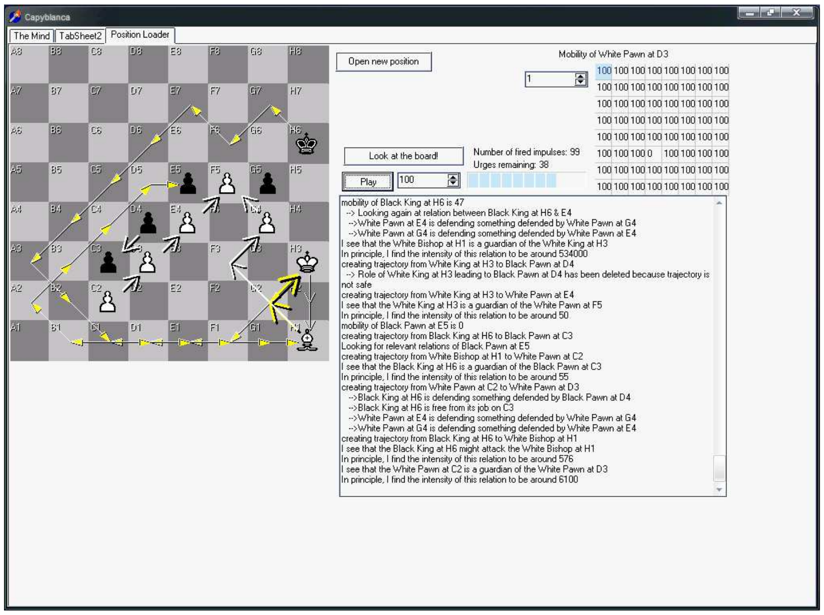
Mobility of White Pawn at D3

100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

Number of fired impulses: 59
Urges remaining: 41

```
creating trajectory from Black Pawn at D4 to Black Pawn at C3
mobility of White Pawn at C2 is 0
Saccading to D1...
creating trajectory from White King at H3 to Black Pawn at G5
  --> Role of White King at H3 leading to Black Pawn at G5 has been deleted because trajectory is
not safe
creating trajectory from White King at H3 to White Pawn at G4
creating trajectory from White King at H3 to Black Pawn at E5
I see that the White King at H3 is a guardian of the White Pawn at G4
In principle, I find the intensity of this relation to be around 61
creating trajectory from White King at H3 to White Pawn at C2
   -->White Bishop at H1 is defending something defended by White King at H3
   -->White King at H3 is defending something defended by White Bishop at H1
   -->White King at H3 is free from its job on G4
I see that the Black Pawn at D4 is a guardian of the Black Pawn at C3
In principle, I find the intensity of this relation to be around 6100
creating trajectory from White King at H3 to White Bishop at H1
Saccading to E7...
creating trajectory from White King at H3 to Black Pawn at C3
  --> Role of White King at H3 leading to Black Pawn at C3 has been deleted because trajectory is
not safe
I see that the White King at H3 might attack the Black Pawn at E5
In principle, I find the intensity of this relation to be around 51
mobility of White Pawn at E4 is 0
I see that the White King at H3 is a guardian of the White Pawn at C2
In principle, I find the intensity of this relation to be around 57
I see that the White King at H3 is a guardian of the White Bishop at H1
In principle, I find the intensity of this relation to be around 720
```

Figure 5. *After codelet 220, the white king is found, through random chance, to be able to attack the pawn at E5, and proper trajectories and roles are created. Note that the pawn at d4 might also have had been a possible candidate, for, at that stage, Capyblanca had yet to find that square d4 was protected by black's pieces. The system is yet to find that this new trajectory is not sustainable. In this point, all different hierarchical levels are simultaneously being processed by the architecture. There are codelets still 'waiting' to saccade to square c6, codelets 'waiting' to look for a piece's potential trajectories, codelets 'waiting' to create the 'spheres of influence' matrix of a certain piece, codelets 'waiting' to create a trajectory for a certain chess relation, codelets waiting to create a role for a certain type of chess relation, among others.*



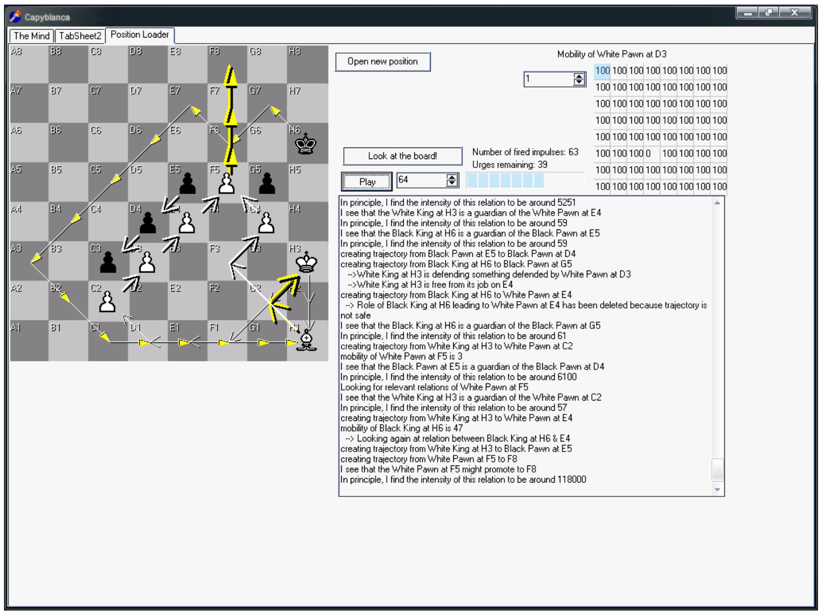
Mobility of White Pawn at D3

100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

Number of fired impulses: 99
Urges remaining: 38

```
mobility of Black King at H6 is 47
  --> Looking again at relation between Black King at H6 & E4
   -->White Pawn at E4 is defending something defended by White Pawn at G4
   -->White Pawn at G4 is defending something defended by White Pawn at E4
I see that the White Bishop at H1 is a guardian of the White King at H3
In principle, I find the intensity of this relation to be around 534000
creating trajectory from White King at H3 to Black Pawn at D4
  --> Role of White King at H3 leading to Black Pawn at D4 has been deleted because trajectory is
not safe
creating trajectory from White King at H3 to White Pawn at E4
I see that the White King at H3 is a guardian of the White Pawn at F5
In principle, I find the intensity of this relation to be around 50
mobility of Black Pawn at E5 is 0
creating trajectory from Black King at H6 to Black Pawn at C3
Looking for relevant relations of Black Pawn at E5
creating trajectory from White Bishop at H1 to White Pawn at C2
I see that the Black King at H6 is a guardian of the Black Pawn at C3
In principle, I find the intensity of this relation to be around 55
creating trajectory from White Pawn at C2 to White Pawn at D3
   -->Black King at H6 is defending something defended by Black Pawn at D4
   -->Black King at H6 is free from its job on C3
   -->White Pawn at E4 is defending something defended by White Pawn at G4
   -->White Pawn at G4 is defending something defended by White Pawn at E4
creating trajectory from Black King at H6 to White Bishop at H1
I see that the Black King at H6 might attack the White Bishop at H1
In principle, I find the intensity of this relation to be around 576
I see that the White Pawn at C2 is a guardian of the White Pawn at D3
In principle, I find the intensity of this relation to be around 6100
```

Figure 6. *After codelet 320, the black king, again through random chance, is found to be able to attack the white bishop. Trajectories and roles are created. Note the different boldness in the arrows, indicating what Capyblanca perceives to be intensity of the threats and defenses. This configuration remains relatively stable (under slight changes) until codelet 384, when a furiously fast restructuring of the perception is triggered by a "local event".*



Mobility of White Pawn at D3

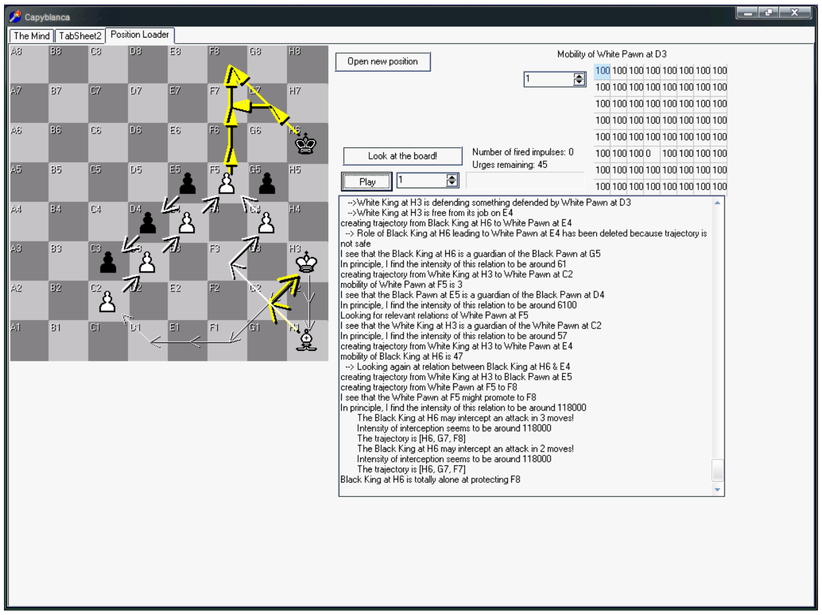
1

100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

Number of fired impulses: 63
Urges remaining: 39

```
In principle, I find the intensity of this relation to be around 5251
I see that the White King at H3 is a guardian of the White Pawn at E4
In principle, I find the intensity of this relation to be around 59
I see that the Black King at H6 is a guardian of the Black Pawn at E5
In principle, I find the intensity of this relation to be around 59
creating trajectory from Black Pawn at E5 to Black Pawn at D4
creating trajectory from Black King at H6 to Black Pawn at G5
  -->White King at H3 is defending something defended by White Pawn at D3
  -->White King at H3 is free from its job on E4
creating trajectory from Black King at H6 to White Pawn at E4
  --> Role of Black King at H6 leading to White Pawn at E4 has been deleted because trajectory is
not safe
I see that the Black King at H6 is a guardian of the Black Pawn at G5
In principle, I find the intensity of this relation to be around 61
creating trajectory from White King at H3 to White Pawn at C2
mobility of White Pawn at F5 is 3
I see that the Black Pawn at E5 is a guardian of the Black Pawn at D4
In principle, I find the intensity of this relation to be around 6100
Looking for relevant relations of White Pawn at F5
I see that the White King at H3 is a guardian of the White Pawn at C2
In principle, I find the intensity of this relation to be around 57
creating trajectory from White King at H3 to White Pawn at E4
mobility of Black King at H6 is 47
  --> Looking again at relation between Black King at H6 & E4
creating trajectory from White King at H3 to Black Pawn at E5
creating trajectory from White Pawn at F5 to F8
I see that the White Pawn at F5 might promote to F8
In principle, I find the intensity of this relation to be around 118000
```

Figure 7. *After codelet 384, the white king's attack on e5 is found to be easily blocked by the black king, thus the system has destroyed that role (and associated trajectory). In the next codelet, though, the system perceives the potential passed pawn promotion, a "local event" which triggers a fast, global, chain reaction of repercussions across the entire board. These repercussions follow very rapidly, so we must "slow down" this narrative to account for rapid events in between the execution of a mere 5 (or less) codelets in the following figures.*



Mobility of White Pawn at D3

1

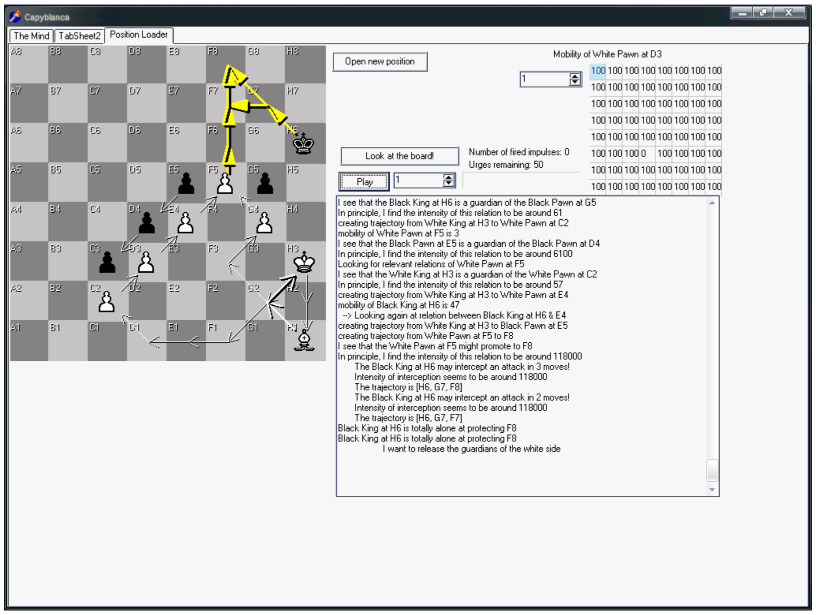
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

Number of fired impulses: 0
Urges remaining: 45

```
  -->White King at H3 is defending something defended by White Pawn at D3
  -->White King at H3 is free from its job on E4
creating trajectory from Black King at H6 to White Pawn at E4
  --> Role of Black King at H6 leading to White Pawn at E4 has been deleted because trajectory is
not safe
I see that the Black King at H6 is a guardian of the Black Pawn at G5
In principle, I find the intensity of this relation to be around 61
creating trajectory from White King at H3 to White Pawn at C2
mobility of White Pawn at F5 is 3
I see that the Black Pawn at E5 is a guardian of the Black Pawn at D4
In principle, I find the intensity of this relation to be around 6100
Looking for relevant relations of White Pawn at F5
I see that the White King at H3 is a guardian of the White Pawn at C2
In principle, I find the intensity of this relation to be around 57
creating trajectory from White King at H3 to White Pawn at E4
mobility of Black King at H6 is 47
  --> Looking again at relation between Black King at H6 & E4
creating trajectory from White King at H3 to Black Pawn at E5
creating trajectory from White Pawn at F5 to F8
I see that the White Pawn at F5 might promote to F8
In principle, I find the intensity of this relation to be around 118000
      The Black King at H6 may intercept an attack in 3 moves!
      Intensity of interception seems to be around 118000
      The trajectory is [H6, G7, F8]
      The Black King at H6 may intercept an attack in 2 moves!
      Intensity of interception seems to be around 118000
      The trajectory is [H6, G7, F7]
Black King at H6 is totally alone at protecting F8
```

Figure 8. *After codelet 388, the discovery of the major threat (in terms of intensity) emanating from the passed pawn triggers a search for black's potential interceptions to the promotion. Those triggered codelets have high urgency and rapidly discover that the black king is able to intercept the trajectory and defend from the promotion. Again, trajectories and roles are created. But, in this case, the processes also perceive that the black king is the only piece able to restrain the pawn from promoting, and hence restrain the black king's mobility from squares in which it would be unable to respond to that threat: the black*

king's spheres of influence are reduced, therefore, to a subset of the squares in the rectangle defined by [c5, h8].



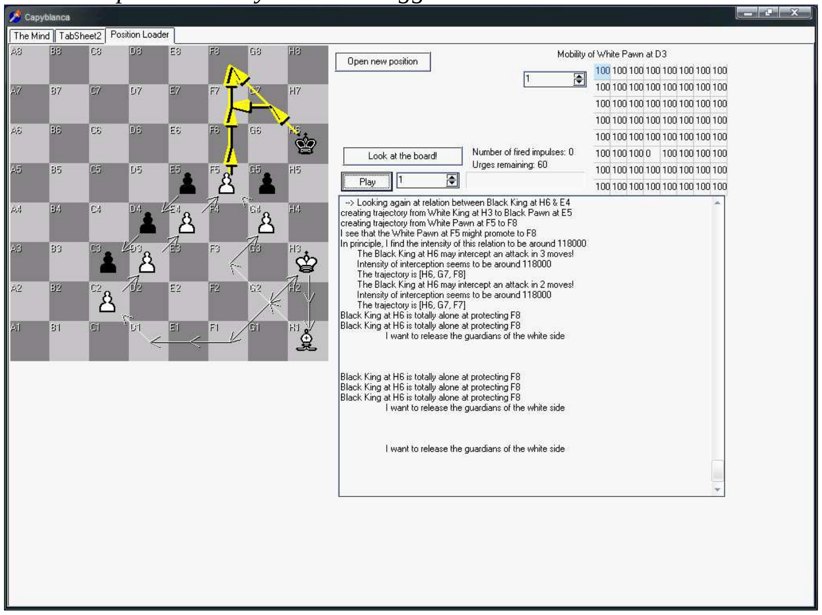
Mobility of White Pawn at D3

100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

Number of fired impulses: 0
Urges remaining: 50

```
I see that the Black King at H6 is a guardian of the Black Pawn at G5
In principle, I find the intensity of this relation to be around 61
creating trajectory from White King at H3 to White Pawn at C2
mobility of White Pawn at F5 is 3
I see that the Black Pawn at E5 is a guardian of the Black Pawn at D4
In principle, I find the intensity of this relation to be around 6100
Looking for relevant relations of White Pawn at F5
I see that the White King at H3 is a guardian of the White Pawn at C2
In principle, I find the intensity of this relation to be around 57
creating trajectory from White King at H3 to White Pawn at E4
mobility of Black King at H6 is 47
  --> Looking again at relation between Black King at H6 & E4
creating trajectory from White King at H3 to Black Pawn at E5
creating trajectory from White Pawn at F5 to F8
I see that the White Pawn at F5 might promote to F8
In principle, I find the intensity of this relation to be around 118000
      The Black King at H6 may intercept an attack in 3 moves!
      Intensity of interception seems to be around 118000
      The trajectory is [H6, G7, F8]
      The Black King at H6 may intercept an attack in 2 moves!
      Intensity of interception seems to be around 118000
      The trajectory is [H6, G7, F7]
Black King at H6 is totally alone at protecting F8
Black King at H6 is totally alone at protecting F8
            I want to release the guardians of the white side
```

Figure 9. *Since the black king, and by consequence the black side, can no longer attack the white side's unprotected pawn, two states later, after codelet 390, the roles of 'guardians' of the white pieces start to lose intensity: the black king, now restrained, cannot "go there" anymore, and neither can any other black piece. In the absence of need, the 'guardian' roles in the white side lose intensity. This triggers a search for other relevant roles those pieces might have. Note that this inference concerning intensity of the guardian relations is achieved by information processing at the abstract role level (the black king is no longer a threat). The reader may notice that processing at these later stages has been triggered mostly by 'expectation-driven,' top-down, codelets: to find new roles for the white side's pieces, for instance, codelets at the trajectory level and piece mobility level are triggered.*



Mobility of White Pawn at D3

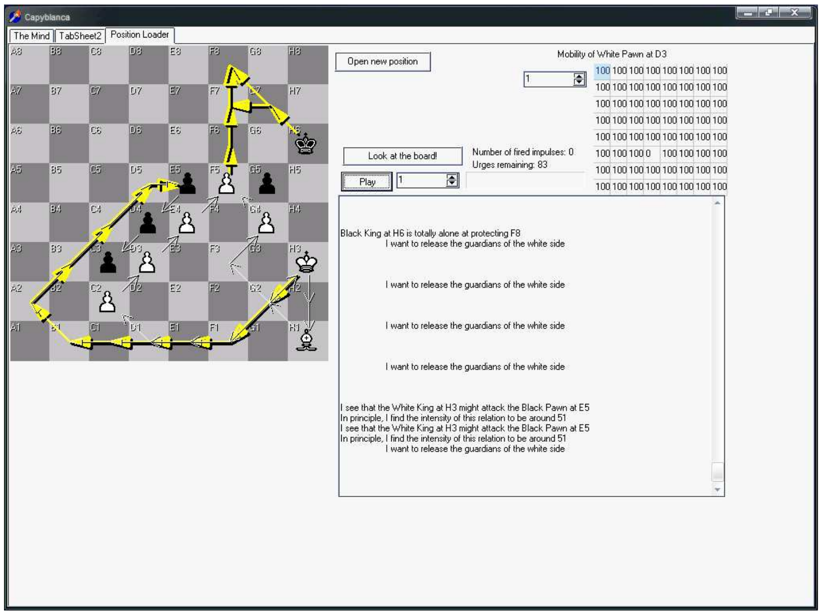
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

Number of fired impulses: 0
Urges remaining: 60

```
  --> Looking again at relation between Black King at H6 & E4
creating trajectory from White King at H3 to Black Pawn at E5
creating trajectory from White Pawn at F5 to F8
I see that the White Pawn at F5 might promote to F8
In principle, I find the intensity of this relation to be around 118000
      The Black King at H6 may intercept an attack in 3 moves!
      Intensity of interception seems to be around 118000
      The trajectory is [H6, G7, F8]
      The Black King at H6 may intercept an attack in 2 moves!
      Intensity of interception seems to be around 118000
      The trajectory is [H6, G7, F7]
Black King at H6 is totally alone at protecting F8
Black King at H6 is totally alone at protecting F8
            I want to release the guardians of the white side



Black King at H6 is totally alone at protecting F8
Black King at H6 is totally alone at protecting F8
Black King at H6 is totally alone at protecting F8
            I want to release the guardians of the white side



            I want to release the guardians of the white side
```

Figure 10. *After codelet 396, all white pieces' 'guardian' roles have very low intensity. For example, intensity of white bishop's defense of the g4 pawn is less than 1/10000 of the intensity of the potential pawn promotion—the reader should note that an arrow's thickness display is affected by orders of magnitude.*

Hence, the bishop is seen at this point as a piece of low consequence, with its constrained mobility and nothing to attack in its 'spheres of influence'.



Mobility of White Pawn at D3

100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	0	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

Number of fired impulses: 0
Urges remaining: 83

```
Black King at H6 is totally alone at protecting F8
            I want to release the guardians of the white side

            I want to release the guardians of the white side

            I want to release the guardians of the white side

            I want to release the guardians of the white side

I see that the White King at H3 might attack the Black Pawn at E5
In principle, I find the intensity of this relation to be around 51
I see that the White King at H3 might attack the Black Pawn at E5
In principle, I find the intensity of this relation to be around 51
            I want to release the guardians of the white side
```

Figure 11. *Nine codelets afterwards, at state 405, the white king—finally free from having to guard from any impending attacker—rediscovers the potential attack towards the D5 pawn. This time, however, there is knowledge of the black king's attachment to its role of guardian from the promotion, and the piece cannot assume a second simultaneous role of preventing the white king's attack. It is at this point that the architecture registers the fact that the joint attack by the white king and the passed pawn cannot be defended single-handedly by the black king. Intensity of the white king's attack soars to a previously unforeseen level. Hence white's advantage becomes clear, and a strategy to secure that advantage has emerged. The system has achieved, in a mere 405 codelets, an understanding of the global scenario. (The execution continues for 300 codelets with this configuration firmly stable—the system stops after some time with no improvement.) During the process, some minor local perceptions have triggered global repercussions which rapidly rearrange the whole view.*

Significant conclusions may be drawn here:

(i) The architecture is rapidly drawn into the 'relevant' parts of the situation, and attention to irrelevant parts is scarce;
(ii) The emergent process negotiates between top-down (expectation-driven) and bottom-up (data-driven) processes, and does not fall prey to a combinatorial explosion;
(iii) Strategically similar chess positions can be explained *from the point of view of sets of abstract roles*.
(iv) For deep understanding of a scenario, the abstraction is *the most efficient level of description*.

The following section elaborates on some of these points' implications.

III. Discussion

This paper has presented the Capyblanca project, a chess model based on the idea that analogy lies at the core of human cognition (Hofstadter 2001). Capyblanca is a psychologically plausible, multi-leveled, (simulated) parallel computational architecture. Starting from isolated pieces found in random squares, it gradually converges towards a context-sensitive interpretation of the global scenario. The architecture displays (i) a high degree of 'entropy', continually constructing and destroying structures; (ii) a high degree of parallelism, with both bottom-up and top-down processes running concurrently; (iii) concurrent processing distributed over multiple levels (pieces perceived, distance relations, chess relations, abstract roles, etc); (iv) and "vagueness" brought by continuous degrees of intensity of abstract roles. It gradually builds an abstract representation, based on the roles perceived, which gather knowledge about what may happen in the position. We have presented a detailed run of the system in position 6, which is combinatorially demanding. This initial project is a step towards a long-term understanding of four landmark problems:

(i) How do chess players concentrate attention to relevant aspects of a situation?

We propose that players are primed to find relevant information very rapidly, as they perceive the board. Capyblanca models this by placing higher urgencies to (i) processes triggered from new bottom-up information, and (ii) top-down processes with high computed intensity. The architecture is rapidly drawn into the 'relevant' parts of the situation, and attention to irrelevant parts is minimum. There is, however, a constant exploration of alternative "counterfactual" pathways. We believe this wasted effort is psychologically plausible, as a part of the priming process itself.

(ii) How do humans avoid the combinatorial explosion in chess?

Capyblanca does not compute the chess tree. Instead, the system gradually builds an interpretation of a position, based on various types of information (pieces, trajectories, mobilities, roles, etc.). It attempts to synthesize information into abstract form: "The black king is not able to defend simultaneously from the passed pawn and the upcoming white king threat". This type of information encoded as "roles" is generalizable, by changing the pieces involved, neither does it concern the accidental trajectories involved. The system, therefore, attempts to find the essence of the strategic situation—exactly what enables humans to play without facing the combinatorial explosion. Instead of a tree-search process, in Capyblanca, elicited cues lead to proper responses, by activating concepts and top-down expectations likely to be relevant to the situation at hand.

(iii) How do humans perceive similarity at a strategic level?

Strategically similar chess positions may be explained by looking at the configurations *from the point of view of sets of abstract roles*. For deep understanding, that abstraction is *the most efficient level of description*. While a skeptical reader might consider that the processing of both *trajectories* and of *roles* may be redundantly inefficient, that is really not the case. It is crucial to process *both* trajectories and roles. The distinction is an important one. Because a particular trajectory is bound to a set of squares, it is not an adequate structure to either perceive the situation in abstract terms or to generalize to other superficially different but strategically similar scenarios. Trajectories lead to chess relations, but are different from them. Chess relations are independent of locations and of particular squares of a trajectory. And chess relations are subtly different from abstract roles: In Capyblanca's evolving interpretation of a position, roles emerge gradually from the set of chess relations, by the process of continuous re-evaluation of the intensity of each relation.

This is a distinction, not an argument. By structuring information in terms of abstract roles it is possible to obtain *abstract representational invariance*. Consider, for example, the notion that, "under a double check, the only response is to escape with the king". This notion is generalizable to an immense set of situations. It would be unthinkable—or at least computationally intractable—to try to describe it in terms of particular pieces, particular squares, and particular trajectories. The most efficient level of description is the abstraction "piece A has a role of attacker towards piece X", "piece B also has a role of attacker towards piece X", and "X is a king". It is irrelevant for understanding, and inefficient for processing, what type of piece A and B are, what their mobilities are like, or even which squares they occupy. Therefore, it is imperative, for an architecture with the ambition to model human understanding of a chess position, to use different representational structures to account for the trajectories involved in chess relations, and separate structures for the abstract roles that are found to be relevant in a particular position. Deep understanding arises by recognizing a familiar, previously seen, configuration of abstract roles. For understanding a scenario, the most efficient level of description is given by a combination of abstract relations representing the most important roles objects play. By reasoning at this level, the system is able to perceive analogies between dramatically different scenarios (Linhares 2008). This enables the eventual emergence of a global, meaningful, understanding of the constraints and key issues underlying the strategic situation one faces—and leads us to our final claim concerning the architecture.

(iv) Finally, how does the system acquire a state of meaningful anticipation on how the global scenario may evolve?

Capyblanca acquires, during a run, a large number of constraints concerning the situation at hand. Some of these constraints concern, for example, mobility of a piece. Mobility of any given piece is not static to the current chess position. It dynamically changes, due to new information. For example, the black king is constrained to a small area of the board after the system perceives that it is the only piece with the role of intercepting the passed pawn. This triggers new information on the other side of the board. Since the black side lost its only piece with the abstract role of a potential threat, all of white's pieces that are seen as guardians (from such potential attack) lose these roles. This eventually leads to a new role for the white king: to move across the board and threaten black's pawn. With this multiple-leveled processing, the system is able to grasp the affordances that each piece, or a combination of pieces, may have. It enables the system to anticipate moves that may follow.

3.1 Generality of the model, and how to measure psychological plausibility.

If Capyblanca were an *ad-hoc* system tied exclusively to the chess domain, it would be hardly interesting as a cognitive model. However, it should be noted that Capyblanca is one project out of a family of cognitive architectures (Mitchell 1993, Foundalis 2006, Hofstadter and FARG 1995, McGraw 1995, Marshall 1999, Rehling 2001, French 1995). Perhaps these architectures may enable us to propose a definition of "understanding", and, as such, refrain from using quotes with the term. An analogy may be of use here. Consider, for example, a light cone in physics: it defines a clear boundary, as there can be no causation between an observer and events outside the light cone. We may define understanding as *relevance within an information-processing cone*: given a representational state at any point in time, the relevance of the set of possible future states of a cognitive system define the understanding, the level of meaningful anticipation, of the system (this notion is close to the memory-prediction framework sketched by Hawkins, 2005). Understanding is not a property of a particular state, of a particular representation. It is a property of a *process*. In Capyblanca, at any point in time, there are pathways the system can follow and pathways the system cannot follow; the more relevant the states in its trajectory cone, the better understanding of the scenario at hand.

In the appendix, we propose that the criteria for measuring advances in cognitive chess models should be *process-oriented*, as opposed to solely *output-oriented*. Instead of measuring scientific progress by studying moves proposed and quality of play, we may want to study the particular processing pathways the architectures go through, and compare those to a human's. Careful manipulations enable us to keep the models constantly close to the goal of psychological plausibility. Processing pathways change rapidly under such manipulations, and those changes, if also reflected on human players, can teach us a lot about the nature of our own cognition. We thus propose to take 'output' from its prominent current standing, in favor of information-processing trajectories. The information-processing pathways, the *trajectories*, are the major results.

In this family of architectures, there is a high degree of parallelism, some of which is bottom-up and data-driven. Registration of bottom-up data leads to expectations and hypothesis (i.e., a bishop primes a player to hypothesize about the contents of the diagonals). Hence, there is a number of simultaneous top-down, hypothesis-driven, processes attempting to gather more meaningful information concerning the scenario at hand, and to constrain the plausible courses of action. A result of this is that the representation is emergent, and cannot be traced to any single, or even to a small set of, individual processes. Another consequence is that of improved robustness of the system: if, at any point, a process has not gathered a meaningful piece of information, which should radically alter the hypotheses and expectations (and therefore all subsequent processing), the architecture may still be able to suddenly obtain such piece of information and trigger the necessary chain reaction. Erroneous, irrelevant "counterfactual" pathways are always being explored by some codelets, but generally self-correct rapidly. Consider, as an example, the white king’s attack prior to the architecture’s perception that no black piece could intercept it (Fig. 5).

3.2 Current limitations of the architecture

As of this writing, the system is, like the original Copycat project, unable to learn (Linhares, 2005). Its objective is solely to model human perception and intuitive understanding of a chess scenario “in an individual agent at a particular time” (Chalmers et al, 1992). If one were to ask how the system models chess expertise, the answer is: it does not.

Currently, top-down expectations need to be explicitly programmed. The system is unable to acquire new strategies; therefore it cannot play a full game of chess. A full rewrite of the system is underway, which should in theory enable Capyblanca to acquire, during a game, new types of roles, new sets of roles in combinations, and use them in upcoming games. However, the mechanisms for such an architecture have numerous design decisions, and are daunting research problems in their own right.

Yet, despite its current limitations, it would be premature to dismiss the current model as not worthy of scientific interest. José Raoul Capablanca reportedly learned to play chess by watching his father play. He won his very first game, against his father; and went on to become world champion. While we do not intend to have a world class chess engine, we want to better understand human cognition using chess as a domain. Four years in development, the Capyblanca project may be, we hope, a starting point in fully addressing this goal. The general theoretical principles given here should be able to account for expertise; as the higher the skill, the more relevant the states in the trajectory cone, and the greater understanding exhibited by the system. Capyblanca accounts for more aspects of human thought in chess than any other model we are aware of, for we propose that, after the opening of a game, analogy lies at the core of chess thought.

Appendix. Why a cognitive chess model should not be evaluated by the quality of proposed moves, but by the psychological plausibility of the process.

If our objective were to create a world chess champion computer player—as it has been for 50 years—, how could we evaluate progress towards that goal? How could we know whether a new proposal was a scientific advance? The historical answer has been, obviously, to look at the results obtained, in terms of quality of play. How well does the new evaluation function play in comparison to the old one? How much performance can be gained by a parallel implementation? How much can be gained by specific hardware, by a new pruning strategy, and so on? By looking at the results obtained, it has been possible to compare alternative approaches. Coupled with Moore's (1965) law, the comparisons between different outputs stemming from different pruning techniques, evaluation functions, parallelization strategies, and custom-designed hardware led, after decades of effort, to Deep Blue's stunning success. Most research has been, up to now, "results-oriented".

That, in itself, is not problematic—what is problematic is the idea that output can tell the whole story. Some methods for analyzing the outputs of cognitive models have been proposed by Hofstadter and FARG (1995, p. 363—364). Here we adapt those for the case of chess-playing programs, and propose some new ones. Consider, for example, comparing a set of strategies obtained by the program in a particular position with those obtained by humans of comparable ELO rating:

- **(i)** The strategies that the program comes up with (especially those with high frequency) should almost always seem plausible to people of a specific level of play;
- **(ii)** The program should occasionally come up with insightful or creative strategies;
- **(iii)** If a given strategy seems obvious to most people, then it should be obvious to the program (and crop up with high frequency);
- **(iv)** If a given strategy seems far-fetched to most people, then in the model it should crop up with low frequency;
- **(v)** If a given strategy seems elegant but subtle to most people, then in the model it should crop up with low frequency;
- **(vi)** If a strategy is considered ugly by most people, then its quality should be rated low by the program;
- **(vii)** If a strategy is considered elegant by most people, then its quality should be rated high by the program;

What if, instead of striving to devise a world champion, what we strive for is a computer that "thinks" like a human player, that "understands" a position without computing its search tree? In this case, what type of data should we look for as a guide to serious scientific progress? Output comparison remains important, but it may not tell us much, since that output, in the form of a move (or a longer-term strategy), could have been obtained by an information-processing method different from that of a human. Deep Blue could pass tests (i)—(vii); but there are tests of psychological plausibility that it could not pass. How can we evaluate psychological plausibility? If a computational architecture is aimed at genuine intelligence, how can we evaluate it as such?

An answer is to study *the process*. We can focus on the process through which the program acquires and transforms information. We should not only look at the final output, but at the steps through which the system passes to obtain that output. The process, or *the trajectory of information-processing*, should be the key data to be compared to human beings[1]. Instead of being *results-oriented*, or *output-oriented*, we may be *process-oriented*, focusing on the *information-processing trajectories*.

[1] Note that processing time is not an issue here. Perhaps a realistic computational model of the human mind may be much faster than a human, perhaps much slower. Since processing times have been, and will probably continue to be, increasing exponentially, it is almost a tautological statement that, one day, humans and machines have had, have, or will have, comparable speeds when undergoing the same process. In the proposed architecture neither speed nor quality of play should be an issue. The real issue should be: does the system realistically model how humans think, given a specific level of play?

	Architecture's trajectory	Architecture's trajectory
Human trajectory	**Positive/positive** **(viii)** The pathways taken to a given strategy should seem plausible from a human point of view; **(ix)** The spectrum of strategies that people come up with should in large part be accessible to the program (in the sense that there should be some theoretical pathway that could lead to it). **(x)** As a situation is gradually tweaked in a series of stages that, to people, clearly strengthen the pressures for certain strategies and weaken those for others, the program's frequencies and quality judgments should reflect this tendency;	**Negative/positive** • in positions that humans usually blunder, the reasons for such errors could be studied, and the program's trajectories could be evaluated: do the program's trajectories also misjudge some subtle pressure? Do they also ignore important information? Was the process (or could it be) led astray for similar reasons than those under human processing?
Human trajectory	**Positive/negative** **(xi)** When various key features of the architecture are disabled (or "lesioned") by taking them one by one out of the program, the behavior should degrade in ways that reflect the theoretical role that each feature was intended to play; • A certain trajectory, which the computer could not, in principle, have processed, is said to have occurred to chess players. Human players should then evaluate whether or not that seemed an idea which they themselves had (or could ever have) subscribed to, when processing the position;	**Negative/negative** • the architecture should not spend effort considering moves of "completely irrelevant" pieces; • the architecture should be able to go through the same information-processing trajectory, and "ignore" a "completely irrelevant" piece, if it is deleted from a particular position;

There are four possible cases for analyzing the information-processing trajectories: We can either take a "positive sample" (i.e., data) from either a human or a program; or a "negative sample" (i.e., a control): manipulations from humans or programs. This leads to four cases:

Positive/positive criteria are those cases in which chess players study the trajectory of information-processing of the architecture and mention that the process seems plausible and sensible. The same sequence of perceptions and concept activations (see below) should lead to a process in which pressures are registered, then vaguely and gradually sorted out. Which pressures are crucial? Which are irrelevant? Which pressures stand in between high and low importance? Do humans agree with the computers' assessment of those pressures? Do these judgments seem psychologically plausible?

We know, for example, that humans do not search the game tree, and "think ahead" rather sporadically (de Groot 1965, Gobet 1998, Gobet et al 2001, Klein 1999). Strong evidence points to the fact that high-level chess playing comes from a deep, meaningful, perception of a position (Chase and Simon 1973, Gobet 1998, Linhares 2005, Linhares and Brum 2007). This process of understanding of a scenario is extremely fast and robust, to the point of making deliberate "thinking ahead" processes unnecessary: Consider, for example, the case of Ron "Suki" King, a world checkers champion, which played in 1998 a simultaneous game against 385 opponents. He beat them all (Myers, 2002). Suppose he had an average of 2 seconds for each position. A mere two seconds; barely enough time to look at the board, have an initial idea, then make a movement. That would give each of his opponents 12 minutes and 30 seconds to consider the reply. In Chess, José Raoul Capablanca[2] once said: "I see only one move. The best one". A computer architecture should model the intuitive perception process to account for these facts.

In the positive/negative test, a manipulation of *the computer process* is made. A particular trajectory, which the computer could not, in principle, have processed, is said to have occurred to chess players. For example, in position 6, players could be told that the program moved the bishop to protect the g4 pawn—a movement which the program, after perceiving the uselessness of the bishop, should not only disregard, but never really entertain in the first place (see section 3.3 below). Chess players should then evaluate whether or not that seemed an idea which they themselves had subscribed to, or could ever, when processing the position. Moreover, as proposed by Hofstadter and FARG (1995) a computational architecture may be 'lesioned' in order to probe whether its parts indeed are responsible for some particular behavior.

In the negative/positive test a manipulation of *the human process* is made. How can that be done? First, by selecting positions in which humans usually fall short of responding adequately. Suppose, for example, that, for a particular level of play, a large number of human players 'fall' for a certain blunder in a certain situation. Let us consider a specific example in Figure A1—which may or may not be representative of a large number of expert players, yet, is an understandable mistake (for humans).

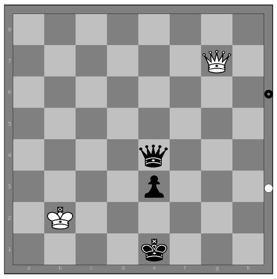


Figure A1. *Batuyev versus Simagin, in Riga, 1954. Simagin, a player of extraordinary skill, playing black, committed a blunder which no tree-searching machine ever could. With material advantage of a passed pawn, it seemed that Simagin could handle a winning endgame. However, in rush to promote, he played pawn to e2. Since his material advantage consisted on the passed pawn, one can only resonate with Simagin's attempt to promote. This led to a mate in three: white queen to g1 check, king escapes to d2, white queen to c1 check, king escapes to d3, queen to c3 checkmate.*

Vladimir Simagin fell for the blunder (as do many other students of the game) because his trajectory of information processing is, at one point or other, misguided by some erroneous distraction or judgment—in this particular case, an overconfident rush towards promotion. Note what we mean by information-processing trajectories: could a tree-searching computer ever fall for this blunder? Since this is a shallow mate in three, there is no computer program of the current tree-search paradigm that would ever fall for such mistake; a small tree search leads to perfect play. In the negative/positive quadrant, the reasons for specific human blunders could be studied, and the program's information-processing trajectories could be evaluated taking this data into account: do the computers' trajectories also misjudge some subtle pressures? Do they also ignore important information? Was the process (or could it be) led astray for reasons similar to those underlying human information-processing (Silva and Linhares 2007)? Only by carefully comparing human information processing trajectories with those trajectories obtained by computers will we be able to evaluate significant advances towards genuine intelligence (Linhares 2000).

Finally, there are negative/negative cases. This is, roughly, the situation that we have at this historical point of research, as computers exhibit trajectories highly different from those of people. Here we can start by manipulating people's processes. Suppose, for example, that chess players are asked which pieces are irrelevant in a given position, and they agree that some pieces are completely irrelevant; and that those

[2] For this and for other remarks, we have named our architecture as Capyblanca, a blend of Capablanca and the Copycat Project (Hofstadter and FARG 1995, Mitchell and Hofstadter, 1990, Mitchell 1993), from which it borrows key ideas.

pieces are never considered in their thought processes. How can we test that? This task is easy: If they mention that a piece is "completely irrelevant", we can take the piece out of the position, and ask them how they would deal with this "new" position—does anything change? For example, suppose that a program from the traditional tree-searching paradigm actually spends computational effort considering moves of those "completely irrelevant" pieces—that would be a blatant failure of this test. To pass the negative/negative type of test, a computational architecture should not (i) spend effort considering moves of "completely irrelevant" pieces, and (ii) should respond in the same way, with the same information-processing trajectory, if a "completely irrelevant" piece is deleted from a position.

To summarize our methodological standpoint, if we want to understand genuine intelligence and human understanding, instead of being solely *output-oriented*, we should be *process-oriented*. Table A1 summarizes these arguments for studying the information-processing trajectories. The bulleted points are original arguments developed here, and points numbered (viii)—(xi) are adapted from Hofstadter and FARG (1995).

Acknowledgement

The author thanks Doug Hofstadter, Harry Foundalis, Helga Keller, Eric Nichols, Hamid Ebkia, José Ricardo de Almeida Torreão, Horacio Hideki Yanasse, Abhijit Mahabal, Francisco Lara-Dammer, James Moor, John Rehling, Daniel Dennett, John Anderson, Neil Charness and Fernand Gobet. Financial support has been provided by the PROPESQUISA program of the Getulio Vargas Foundation.

V. References

Atkinson, G., (1993). *Chess and machine intuition*. Norwood, NJ: Ablex Publishing.
Ban, J. (1997) *The tactics of endgames.* Dover.
Chalmers, D.J., French, R.M., Hofstadter, D.R. (1992) High-level perception, representation, and analogy: a critique of artificial intelligence methodology. *Journal of Experimental and Theoretical Artificial Intelligence* 4 (3), 185-211.
Chase, W.G., and Simon, H.A., (1973). 'Perception in Chess', *Cognitive Psychology* 4, 55-81.
Charness N, Reingold, EM, Pomplun M, Stampe DM, (2001). 'The perceptual aspect of skilled performance in chess: evidence from eye movements.' Memory & Cognition, 29, 1146-1152.
Foundalis, H. (2006) PHAEACO: a cognitive architecture inspired by Bongard's problems. Ph.D. thesis, Indiana University, Bloomington.
French, R.M., (1995) *The Subtlety of Sameness*, MIT Press.
Gobet, F., (1998). Expert memory: a comparison of four theories, *Cognition*, 66, 115-152.
Gobet, F., Lane, P.C.R., Croker, S., Cheng, P.C-H., Jones, G., Oliver, I., Pine, J.M., (2001). 'Chunking mechanisms in human learning'. *Trends in Cognitive Sciences*, 5, 236-243.
de Groot, A.D., (1965). *Thought and choice in chess*. New York: Mouton.
Hawkins, J. (2005) *On Intelligence*, Holt Paperbacks.
Hofstadter, D.R., (1979). *Gödel, Escher, Bach: an Eternal Golden Braid*. New York: Basic Books.
Hofstadter, D.R., (1985). *Metamagical themas*. New York: Basic Books.
Hofstadter, D.R., and FARG, (1995). *Fluid Concepts and Creative Analogies: computer models of the fundamental mechanisms of thought*, New York: Basic Books.
Hofstadter, D.R., (2001) Epilogue: Analogy as the core of cognition. In: *The analogical mind: Perspectives from cognitive science*, ed. D. Gentner, K. J. Holyoak & B. N. Kokinov, pp. 499–538. MIT Press.
Klein, G., (1999) *Sources of Power: how people make decisions*, Cambridge, MA: MIT Press.
Linhares, A. (2000) A glimpse at the metaphysics of Bongard problems, *Artificial Intelligence* 121, 251-270.
Linhares, A. (2005) An active symbols theory of chess intuition, *Minds and Machines*, 15, 131-181.
Linhares, A. (2008) Dynamic sets of potentially interchangeable connotations: A theory of mental objects. Accepted for publication, *Behavioral and Brain Sciences*.

Linhares, A., and P. Brum (2007) Understanding our understanding of strategic scenarios: what role do chunks play? *Cognitive Science* 31 (6) 989—1007.

Marshall, J. (1999) *Metacat: a self-watching cognitive architecture for analogy-making and high level perception*. PhD thesis, Indiana University, Bloomington.

McGraw, G., (1995) *Letter Spirit (part one): emergent high-level perception of letters using fluid concepts*. PhD Thesis, Indiana University, Bloomington.

McGregor, S.J., & Howes, A. (2002) The Role of attack and defense semantics in skilled player's memory for chess positions. *Memory and Cognition*, 30, 707-717.

Mitchell, M., (1993) *Analogy-Making as Perception*, Cambridge: MIT Press.

Mitchell, M., and Hofstadter, D.R., (1990) 'The emergence of understanding in a computer model of concepts and analogy-making'. *Physica D*, vol. 42, 322-334.

Moore, G. (1965) Cramming more components onto integrated circuits. *Electronics* 38(8), April 19. Available online at ftp://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf

Myers, D.G. (2002) *Intuition: its powers and perils*. Yale University Press, New Haven.

Rehling, J.A. (2001). *Letter spirit (part two): modeling creativity in a visual domain*. PhD Thesis, Indiana University, Bloomington.

Silva, J., and Linhares, A., (2007) Cognitive reflection: the 'premature temperature convergence' hypothesis, in *Proceedings of the Twenty-Ninth Conference of the Cognitive Science Society*, Nashville, USA.

Wilkins, D., (1980). 'Using patterns and plans in chess', *Artificial Intelligence*, 14, 165-203.

OUTLINE OF CHANGES IN RESPONSE TO THE REFEREES OF THE “SPECIAL ISSUE ON ANALOGIES” OF **COGNITIVE SYSTEMS RESEARCH**

A previous version of this paper has been submitted to the “special issue on analogies” of the Cognitive System Research Journal. Though one referee recommended publication without revision, unfortunately, the previous piece did not provide enough details concerning the architecture’s implementation, which were deemed essential in order to reproduce the results presented here. Hence, by freely distributing the source code of the project, we expect researchers to be able to not only reproduce these results, but also to attempt design decisions which have not been explored. The code is available under a GPL license, which means that improvements and forks must be publicly available. We hope these steps, alongside the inclusion of subsection 2.3, will satisfy the referee that declined the manuscript for publication.