CHREST Manual
(version 4.0.0)

Peter Lane

August 7, 2012

Abstract

CHREST is a symbolic cognitive architecture explaining how experience affects our ability to remember, categorise and think about the world. This document contains a technical description of CHREST, including details of its main processes and links to how these are implemented. A separate user guide is available, which is a gentler introduction to using the CHREST software.

Contents

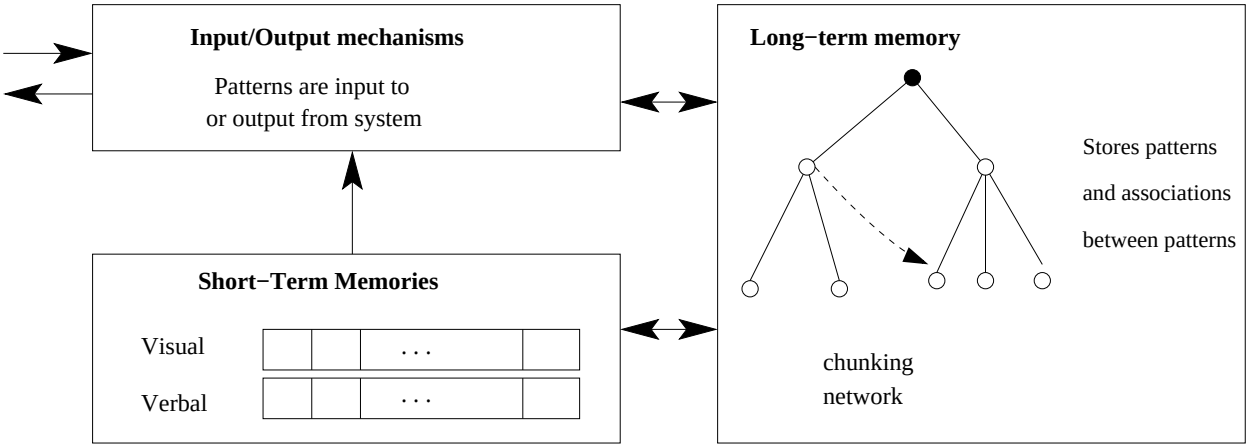

Figure 1: An overview of CHREST

1 Overview of CHREST Architecture

The CHREST architecture may be broadly divided into four components. Interactions with the world pass through an input/output component, which is responsible for interpreting inputs as patterns and for converting action patterns into operations on the world. Patterns are passed through the discrimination network, which acts as an index into long-term memory. Chunks retrieved from long-term memory are placed into the short-term (or working) memory, which is a limited capacity store. Figure 1 provides an overview of these components and their connections.

The distinctive aspect and role of CHREST is in accessing appropriate long-term memory structures given a perceived stimulus, such as an image or spoken sentence. The architecture explains how memories are accessed given some input, and how certain kinds of associations can be constructed across elements of declarative or procedural memory when primed by perception. The *discrimination network* provides the index into long-term memory, and is constructed in close association with the perceptual mechanisms, which control eye movements or how verbal information is stored for rehearsal.

CHREST thus has links to the following kinds of memories, but does not provide a fully-worked out theory of them:

- Procedural memory: unconscious memory of skills
- Declarative memory: conscious memory of facts
  - Episodic memory: facts about one's own life
  - Semantic memory: facts about the world and their connections

With only a small emphasis on procedural and declarative memory, CHREST has a different focus than architectures such as Soar [18, 19] or ACT-R [1]. However, with a greater emphasis on learning and the link between learning and perception, CHREST provides an explanation of how experience affects our ability to remember, categorise and think about the world around us.

This document describes in more detail each component of CHREST. We begin with the long-term memory structures: the discrimination network, the semantic memory and templates. Then we cover the perception module, based on Perceiver, and the short-term or working memory, followed by explaining the mechanisms for including emotional responses in learning and retrieval. Problem-solving is introduced through a look at the CHUMP mechanisms and the SEARCH simulation. We bring these components together by explaining how to control and work with a CHREST-based agent. Finally, we introduce some example domains and models.

2 Discrimination Network

3 Semantic Memory and Templates

3.1 Semantic Links

In EPAM, Simon's model of Memory from which CHREST is inspired, nodes are connected by tests used in the discrimination process. Although nodes are also supposed to be connected by semantic links, a common idea in cognitive psychology, this is not implemented in EPAM. CHREST 2 introduced the idea of semantic links, to make the model a more plausible theory of human memory.

The basic idea is that nodes that have a similar internal representation (in the case of CHREST, patterns of chess pieces) are connected together by links.

3.1.1 How are semantic links created?

Chunks stored in STM are compared pairwise, and if their images (internal representation) overlap with at least the minimum of elements set by the `similarity threshold` parameter, a link is created.

Example:

Assume that pointers to Node-1 and Node-2 are stored in STM. The images of the nodes are as follows:

```
node-1-image:  ("Pc4" "Pd5" "Pe4" "Pf3" "Nc3" "Be3")
node-2-image:  ("Pd5" "Pe4" "Nc3" "Be3")
```

assume also that the `similarity threshold` is set to 4.

In this case, 4 pieces-on-squares are members of both images, so the criterion is met, and a semantic link is created. A pointer to node-1 is added to the list of semantic links for node-2, and a pointer to node-2 is added to the list of semantic links for node-1.

3.1.2 How are semantic links used during pattern recognition?

When recognising a pattern, the program uses the discrimination net as before. The difference is that when a leaf node is reached, this node is not returned immediately, but the program searches through the nodes connected by semantic links to see if some node has richer information, as calculated by the size of the image and the number of slots in the template (see below). The depth of search is controlled by the parameter `maximum semantic distance`. This parameter is set to 1 currently: only nodes one semantic link away from the node reached by discrimination can be selected. (This parameter is not accessible in the GUI.)

The effect of the semantic links is to make all search paths to one of the nodes in the semantic network always reach the most informative semantically-equivalent node. As these connections and relations are computed based on the current state of the network, subsequent learning in other parts of the network may alter the node retrieved for any given input pattern.

3.2 Templates

The idea of templates is discussed in detail in [15]. The basic idea is that some chunks, which occur often when playing or studying games, evolve into more complex data structures, which possess slots supporting the storage of variable information. Templates are also supposed to point to other types of information: moves, plans, evaluation of the position described in the template, and so on, but this has not been implemented yet.

3.2.1 How templates are created

Slots are created for types of information that recur often. There are two sources for the information: nodes reached via test links, and nodes reached via semantic links. For the nodes located at a minimum-level

A B

Table 1: Heuristics

in the network, CHREST looks for if some information (type of piece, square or chunk) recurs at least a minimum-number-of-occurrences. If this is the case, a slot is created for this type of piece, square or chunk.[1]

Note that in the present version, templates are created periodically during the learning phase: the whole net is scanned and templates are created when possible. This is done mainly for efficiency and simplicity reasons. Later versions will probably have nodes checked for templates when they are recognized during the learning phase.

3.2.2 How templates are used during a memory task

During the presentation of a position, the program recognizes patterns of pieces on squares, and outputs a node. If the node is a template, the program tries to fill in slots of this template with pieces on squares in the pattern that do not belong to the image of the node. In addition, the program tries to fill in slots of the hypothesis with each pattern newly perceived. During the reconstruction phase, information in slots is used as any other information stored in STM.

In the current version, slots may be filled in with any type of information that shares the piece, square or chunk labeling the slot. A possibly more plausible mechanism is that slots may be rapidly instantiated only with information already present in the net. With method (a) of construction, for example, slots could be filled in only with values that appear in at least one of the links coming from the template-node.

4 Perceiver

The Perceiver is responsible for gathering information from a visual stimulus. Input from the simulated eye is in the form of *patterns*, containing a sequence of *item-on-square*.

Attention is controlled using a set of heuristics. In line with studies on expertise, the heuristics vary slightly between novice and experts. An arbitrary dividing line is placed between novice and experts at 2000 nodes in visual LTM.

The heuristics are as follows:

LTM

Neighbouring piece The heuristics are applied, as shown in Table 1.

5 Working Memory

6 Emotions

7 Problem solving

CHUMP [12] and SEARCH [10] provide two complementary aspects of problem-solving behaviour. Currently, CHUMP is implemented within CHREST, and SEARCH is currently being integrated. A process-based model of SEARCH is available, simulating its behaviour.

Problem-solving behaviour in CHREST follows the proposal of Chase and Simon [4] that "skill in playing chess depends on:

[1] Note: slots for chunks are not currently implemented.

1. recognising familiar chunks in chess positions when playing games, and
2. exploring possible moves and evaluating their consequences.

Hence, expertise depends *both* on the availability in memory of information about a large number of frequently recurring patterns of pieces, and upon the availability of strategies for highly selective search in the move tree." [10]

8 Controlling a Model

9 Examples

The following examples demonstrate the use of CHREST in three domains taken from published work. More examples are provided in the 'examples' folder of the distributed software.

9.1 Chess recall

Based on book [5] and article [16].

9.2 Language learning

Based on EPAM-VOC [17].

9.3 Iowa gambling task

Based on Marvin's project [14].

10 Recommended reading

The following publications are recommended for understanding the current status of CHREST and related models.

- A general description of chunking mechanisms in human learning.[13]

10.1 Historical background

The following publications may be of interest for understanding the development of CHREST and its precursors.

10.1.1 CHREST and related models

Version 1 of CHREST was released in 1993 and published as:

- F. Gobet (1993). Les memoires d'un joueur d'échecs. [9]
- F. Gobet (1993). A computer model of chess memory. [8]

A variant of the first version of CHREST was described in chapter 8 of [5]. The main differences in comparison with the first version were that the internal representation does not play any more the role of a retrieval structure, and that the eye movements were simulated much more closely than in the earlier version.

Version 2 of CHREST was released in 1995 and included the ideas of semantic links between nodes and templates. Important publications for version 2 include:

- F. Gobet and H.A. Simon (1996), Templates in chess memory: A mechanism for recalling several boards. [15]
- F. Gobet, Memory for the meaningless: How chunks help. [11]

CHUMP is a version of CHREST which learns to select moves by pure pattern recognition (no search):

- F. Gobet & P. Jansen (1994). Towards a chess program based on a model of human memory. [12]

Version 3 of CHREST was released in 2009, and used in modelling some further results in chess expertise, particularly short-term memory effects.

- R. Ll. Smith, F. Gobet and P.C.R. Lane (2007), An investigation into the effect of ageing on expert memory with CHREST. [26]
- R. Ll. Smith, P.C.R. Lane and F. Gobet (2008), Modelling the relationship between visual short-term memory capacity and recall ability. [28]
- R. Ll. Smith, F. Gobet and P.C.R. Lane (2009), Checking chess checks with chunks: A model of simple check detection. [27]

Version 4 of CHREST is due for release in late 2012, and is a complete reimplementation of CHREST from Lisp into Java. Version 4 includes the components of earlier versions of CHREST, such as the discrimination network, semantic links, templates, perceiver and working memory, additionally supports CHUMP [3, 12], and includes the emotion-handling mechanisms, as developed by Marvin Schiller [14].

10.1.2 Precursors of CHREST

CHREST's discrimination network is at heart similar (but not identical) to that proposed by the EPAM theory.

- E.A. Feigenbaum & H.A. Simon (1962). A theory of the serial position effect. [6]
- E.A. Feigenbaum, & H.A. Simon (1984). EPAM-like models of recognition and learning. [7]
- H.B. Richman & H.A. Simon (1989). Context effects in letter perception: Comparison of two theories. [20]
- H.B. Richman & H.A. Simon (1994). EPAM simulations of short-term memory. [22]
- H.B. Richman, H.A. Simon & F. Gobet (1991, July). Applying retrieval structures to chess memory. [23]
- H.B. Richman, J. Staszewski & H.A. Simon (1995). Simulation of expert memory with EPAM IV. [21]

EPAM-based models have been used in chess:

- G.W. Baylor & H.A. Simon (1966). A chess mating combinations program. [2]
- H.A. Simon & M. Barenfeld (1969). Information processing analysis of perceptual processes in problem solving.[25]
- H.A. Simon & K.J. Gilmartin (1973). A simulation of memory for chess positions. [24]

References

[1] J. R. Anderson and C. Lebière, editors. *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum, 1998.

[2] G.W. Baylor and H.A. Simon. A chess mating combinations program. In *Proceedings of the 1966 Spring Joint Computer Conference*, pages 431–447, 1966.

[3] T. Bossomaier, J. Traish, F. Gobet, and P. C. R. Lane. Neuro-cognitive model of move location in the game of Go. In *Proceedings of the 2012 International Joint Conference on Neural Networks*, 2012.

[4] W. G. Chase and H. A. Simon. Perception in chess. *Cognitive Psychology*, 4:55–81, 1973.

[5] A. D. de Groot and F. Gobet. *Perception and Memory in Chess: Heuristics of the Professional Eye*. Assen: Van Gorcum, 1996.

[6] E. A. Feigenbaum and H. A. Simon. A theory of the serial-position effect. *British Journal of Psychology*, 53:307–320, 1962.

[7] E. A. Feigenbaum and H. A. Simon. EPAM-like models of recognition and learning. *Cognitive Science*, 8:305–336, 1984.

[8] F. Gobet. A computer model of chess memory. In *Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society*, pages 463–468. Hillsdale, NJ: Lawrence Erlbaum, 1993.

[9] F. Gobet. *Les memoires d'un joueur d'échecs*. Fribourg (Switzerland): Editions Universitaires, 1993.

[10] F. Gobet. A pattern-recognition theory of search in expert problem solving. *Thinking and Reasoning*, 3:291–313, 1997.

[11] F. Gobet. Memory for the meaningless: How chunks help. In *Proceedings of the Twentieth Annual Meeting of the Cognitive Science Society*, pages 398–403. Mahwah, NJ: Lawrence Erlbaum, 1998.

[12] F. Gobet and P. Jansen. Towards a chess program based on a model of human memory. In H. J. van den Herik, I. S. Herschberg, and J. W. Uiterwijk, editors, *Advances in Computer Chess 7*, pages 35–60. Maastricht: University of Limbourg Press, 1994.

[13] F. Gobet, P. C. R. Lane, S. J. Croker, P. C-H. Cheng, G. Jones, I. Oliver, and J. M. Pine. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5:236–243, 2001.

[14] F. Gobet and M. Schiller. A manifesto for cognitive models of problem gambling. In B. Kokinov, A. Karmiloff-Smith, and N. J. Nersessian, editors, *European Perspectives on Cognitive Sciences – Proceedings of the European Conference on Cognitive Science*. New Bulgarian University Press, Sofia, 2011.

[15] F. Gobet and H. A. Simon. Templates in chess memory: A mechanism for recalling several boards. *Cognitive Psychology*, 31:1–40, 1996.

[16] F. Gobet and H. A. Simon. Five seconds or sixty? Presentation time in expert memory. *Cognitive Science*, 24:651–82, 2000.

[17] G. A. Jones, F. Gobet, and J. M. Pine. Computer simulations of developmental change: The contributions of working memory capacity and long-term knowledge. *Cognitive Science*, 32:1148–1176, 2008.

[18] J. E. Laird. *The Soar Cognitive Architecture*. MIT Press, 2012.

[19] A. Newell. *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press, 1990.

[20] H. B. Richman and H. A. Simon. Context effects in letter perception: Comparison of two theories. *Psychological Review*, 3:417–432, 1989.

[21] H. B. Richman, J. Staszewski, and H. A. Simon. Simulation of expert memory using EPAM IV. *Psychological Review*, 102:305–330, 1995.

[22] H.B. Richman and H.A. Simon. Epam simulations of short-term memory. Complex Information Paper 521. Carnegie Mellon University, Pittsburgh, PA 15213., 1994.

[23] H.B. Richman, H.A. Simon, and F. Gobet. Applying retrieval structures to chess memory. Paper presented at the International Conference on Memory, Lancaster, 1991.

[24] H. A. Simon and K. J. Gilmartin. A simulation of memory for chess positions. *Cognitive Psychology*, 5:29–46, 1973.

[25] H.A. Simon and M. Barenfeld. Information processing analysis of perceptual processes in problem solving. *Psychological Review*, 76:473–483, 1969.

[26] R. Ll. Smith, F. Gobet, and P. C. R. Lane. An investigation into the effect of ageing on expert memory with CHREST. In *Proceedings of The Seventh UK Workshop on Computational Intelligence*. London, UK, 2007.

[27] R. Ll. Smith, F. Gobet, and P.C.R. Lane. Checking chess checks with chunks: A model of simple check detection. In A. Howes, D. Peebles, and R. Cooper, editors, *Proceedings of the Ninth International Conference on Cognitive Modelling*, 2009.

[28] R. Ll. Smith, P.C.R. Lane, and F. Gobet. Modelling the relationship between visual short-term memory capacity and recall ability. In *Proceedings of the Second UKSim European Symposium on Computer Modelling and Simulation*, pages 99–104. IEEE Computer Society, 2008.