

Lane, P. C. R., Cheng, P. C-H. & Gobet, F. (1999). Learning perceptual schemas to avoid the utility problem. *Proceedings of the Nineteenth SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, pp.72-82. Cambridge, UK: Springer.

Learning Perceptual Schemas to Avoid the Utility Problem

Peter C.R. Lane, Peter C-H. Cheng and Fernand Gobet
ESRC Centre for Research in Development, Instruction and Training,
School of Psychology, University of Nottingham,
University Park, NOTTINGHAM NG7 2RD, UK
{pcl,pcc,frg}@psychology.nottingham.ac.uk
www.psychology.nottingham.ac.uk/research/credit

Abstract: This paper describes principles for representing and organising planning knowledge in a machine learning architecture. One of the difficulties with learning about tasks requiring planning is the utility problem: as more knowledge is acquired by the learner, the utilisation of that knowledge takes on a complexity which overwhelms the mechanisms of the original task. This problem does not, however, occur with human learners: on the contrary, it is usually the case that, the more knowledgeable the learner, the greater the efficiency and accuracy in locating a solution. The reason for this lies in the types of knowledge acquired by the human learner and its organisation. We describe the basic representations which underlie the superior abilities of human experts, and describe algorithms for using equivalent representations in a machine learning architecture.

1. Introduction

As computers are applied to increasingly complex tasks, the extent and organisation of knowledge required for success becomes an additional complex task in its own right. For this reason, an important branch in the study of Artificial Intelligence deals with algorithms which learn and organise their own knowledge directly from experience. In this paper, we are interested in those domains where a system must find a solution path between some start and some goal state. The task of “speed-up” learning refers to the acquisition of information which improves the efficiency of finding solution paths. Examples of tasks where such learning is beneficial include: board games, controlling a mobile robot, various scheduling and planning problems, and the construction of proofs of mathematical theorems. Two of the more successful speed-up learning systems are PRODIGY [1] and SOAR [2]. These systems learn rules by generalising an example start state to as wide a class of potential start states as possible; future problems from within this class may have the same solution path applied to them without incurring the cost of its rediscovery. However, as shown by Minton [1], such systems can suffer from the “utility problem”, where system efficiency is degraded by the cost of managing the collection of learnt rules.

Such considerations do not seem to apply to human learners, in which increasing knowledge is typically accompanied by increasing accuracy and efficiency. We therefore consider whether machine learning algorithms can be improved by including what is known about human expert representations. Research on skill acquisition in humans shows that expert representations differ from those of novices in two ways. First, experts have a large number of *perceptual chunks*, which enables them to rapidly identify and categorise a sample problem [3]. Second, they associate these perceptual chunks with possible operator sequences in knowledge structures known as *schemas* [4]. These schemas encode domain-specific knowledge based on the expert's experience, and it is the propagation of knowledge through schemas which distinguishes expert behaviour from that of the novice; the expert can use information perceived in the problem statement to retrieve a suitable schema, which then guides the formation of a solution, whereas the novice often resorts to search strategies. However, earlier work on human expertise does not provide a computational model for learning such schemas directly from experience. In this paper, we address this problem with a set of principles and computational algorithms for learning these representations.

This paper is organised as follows. In Section 2, an example domain requiring the computation of unknown quantities within electric circuits is used to illustrate the way in which humans acquire a skill. Section 3 describes the main findings on expert representations, i.e. perceptual chunks and schemas. Section 4 shows how machine learning algorithms can be designed to learn and work with such representations. Finally, Section 5 returns to the utility problem, and discusses how the model for expert representations can be used to provide efficient indexing into domain-specific knowledge for problem solving.

2. Computing Unknowns in Electric Circuits

In this section we describe a typical task in which speed-up learning may be observed. We discuss two representations for solving problems, the more conventional algebraic method and a diagrammatic representation. The purpose of this section is not to argue for the benefits of the latter, which has been done elsewhere [5,6,7,8,9]; instead this section provides an example of the kind of representation which humans find easiest to master. This permits us to draw some conclusions about the mechanisms underlying the human learning process, which may then be implemented in a machine learning algorithm.

2.1 Algebraic and Diagrammatic Circuit Representations

We consider a problem solving domain in which subjects are given circuit diagrams consisting of a battery and a number of resistors. Various numerical quantities will be supplied, such as the voltage of the battery and the resistance of some of the resistors. Then some unknown quantity will be asked for, such as the total current drawn from the battery.

Consider the three resistor circuit in Figure 1(a). The battery provides a voltage of 12V, and the resistors have the indicated resistances. In order to compute the total

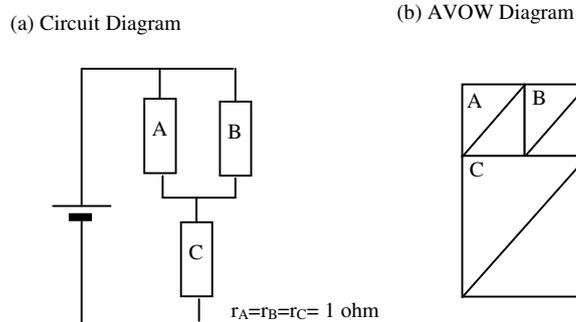


Figure 1 : Equivalent circuit and AVOW diagrams.

current drawn from the battery, one of several approaches may be taken: the individual currents drawn by the different resistors may be calculated, and the total worked out, or the total resistance of the three resistors may be computed, exploiting the property that complex networks may be replaced with equivalent sub-networks. In the given case, it is easiest to compute the total resistance of the circuit. This requires the following recognition: two of the resistors form a parallel network, which is in series with the third resistor. The resistance of the parallel network can be computed as 0.5 ohm from the rule $1/r_p = 1/r_A + 1/r_B$. Then this can be combined with the resistance of the third resistor using the rule that, for resistors in series, $r_T = r_p + r_C$; thus the total circuit resistance is 1.5 ohm and the total current drawn will be $I_T = V_T/r_T = 8A$.

This is an example of the standard algebraic approach for solving such problems. Note that the solver must remember rules for resistors in parallel and series, as well as recognise circuit decompositions which enable these rules to be applied in sequence. Also, the entire procedure would be different if numerical quantities had been supplied for different parameters. In contrast, we describe a diagrammatic representation for electric circuits, known as AVOW diagrams [6].

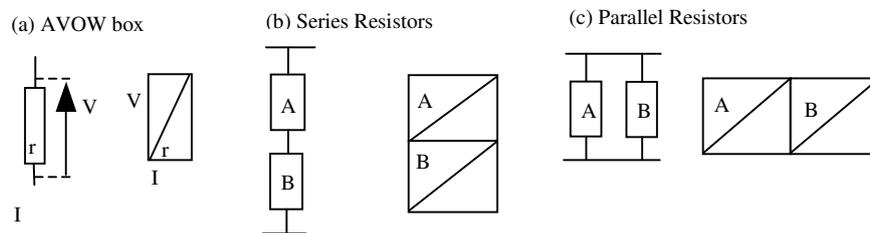


Figure 2 : An AVOW box (a), and rules for their composition: (b) series and (c) parallel resistors.

An AVOW diagram is composed of AVOW boxes, each AVOW box being a

diagrammatic representation of a resistor (or load) within an electric circuit, as shown in Figure 2(a). A resistor has the properties of voltage (V), current (I) and resistance (r). These properties are represented in the AVOW box by scaling the indicated dimension, voltage being the height, current the width, and resistance the gradient of the box's diagonal. It can be seen that the relation of the gradient to the box's height and width encapsulates Ohm's law ($r=V/I$), and also that the area of the box represents the power expended in the resistor ($P=I*V$).

The AVOW boxes are combined into an AVOW diagram for an entire circuit using simple rules of composition. In order to represent two series resistors, two AVOW boxes are aligned vertically, as shown in Figure 2(b). Similarly, two parallel resistors are represented by aligning the boxes horizontally, as shown in Figure 2(c). The alignment rules encapsulate Kirchhoff's Laws which govern the flow of current and distribution of potential differences in electric circuits. For the completed AVOW diagram to be a well-formed representation of the circuit, it must be a rectangle completely filled with AVOW boxes with no overlap or gaps. This requirement captures an important abstraction used in circuit analysis: a collection of resistors in a circuit can be regarded as equivalent to a single resistor. In the same way, the composite AVOW diagram is also an AVOW box, containing all the information for this equivalent single resistor. Just as with the single AVOW box, the resistance of the total AVOW diagram can be found by measuring the gradient of the total rectangle's diagonal. The geometrical nature of this constraint on the final AVOW diagram and the rules for composing separate AVOW boxes mean that it is very natural for humans to work within this representation, a fact which has important pedagogical implications.

As an example of their use, we return to the example problem in Figure 1(a). In order to construct an equivalent AVOW diagram for this circuit, three AVOW boxes must be drawn, each requiring its dimensions (height, width, gradient) and position in the diagram to be determined: the final diagram is shown in Figure 1(b). Human subjects typically proceed as follows. Beginning with resistor A, the only known quantity is its resistance, which determines the gradient of the diagonal of its AVOW box. In this instance the resistance is 1, so a square box is drawn. Similarly, for resistor B a further square AVOW box is required, and it is drawn alongside that of resistor A with the same height because the two resistors are in parallel. Resistor C is also represented by a square AVOW box, but this time, because it is in series with resistors A and B, it is drawn below these two boxes but with their combined width. Finally, the *scale* of the diagram is found by recognising that the height of the final diagram represents the voltage across the entire circuit, i.e. the 12V supplied by the battery. Hence, by measuring the width of the combined box and scaling, the current drawn from the battery is determined.

2.2 Comparison

It is evident that the steps followed by a solver using the diagrammatic representation are qualitatively different from those followed by the solver using algebra. The major difference being the immediate goal: with the AVOW diagrams, the goal is to correctly draw an equivalent representation of the circuit. That this is

sufficient for then providing a solution to the given problem is a property of the AVOW diagram representation, specifically, its encoding of domain laws in the geometry of the representation. Cheng [5,6] discusses this topic in more depth.

From studies on human subjects [7], it is clear that learners using AVOW diagrams generalise the perceptual information in the diagrams more than the numerical information. For instance, circuits with the same structure are considered similar, and not circuits requiring the same sequences of algebraic manipulations, i.e. learners rely on the perceptual properties of the diagrams. It is only whilst in the process of producing a scaled diagram that the learner will incorporate the numerical quantities in the given circuit. The studies show that the improvement in ability of subjects is based, to a large extent, on the acquisition and retrieval of perceptual information relating to circuit topologies and AVOW diagrams. Similar studies with subjects using algebraic solution techniques have not shown such a strong improvement. Hence the claim that perceptual processes are an important source of the power in speed-up learning observed in humans.

3. Expert Representations: Chunks and Schemas

Research on skill acquisition in humans has shown that experts have a large number of *perceptual chunks*, which enables them to rapidly identify and categorise a sample problem [3]. Second, they associate these perceptual chunks with possible operator sequences in knowledge structures known as *schemas* [4]. In this section we briefly describe the acquisition of perceptual chunks and their association with information in schemas.

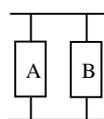
3.1 Perceptual Chunking Theory

The use by experts of perceptual chunks to discriminate between examples from their domain of expertise was first described in the seminal work of de Groot [3]. He found that one of the most effective tests (beyond playing ability) which distinguished expert from novice chess players was the recall of chess positions presented for short periods of time; the expert could recall typical game positions almost exactly after only a few seconds presentation, whereas the novice could recall only a few pieces. This ability of the expert was not due simply to a better recall of visual phenomena, as experts were only marginally better than novices with board positions with random placements of pieces [10]. These results have been replicated and extended upon in work by Gobet and Simon [11], work which also provides a computational model of this ability. The model, CHREST, learns a discrimination network of chunks relating to typical configurations of pieces. These configurations may be rapidly identified in novel board positions, and enable the whole board to be retained as a relatively small number of chunks, each chunk containing several pieces. Tests show that CHREST conforms very closely to the performance of human subjects, with different abilities of chess player corresponding to differing numbers of acquired perceptual chunks.

Although most of the detailed modelling work has been confined to work on chess, the basic hypothesis that experts possess a large number of perceptual chunks has

been validated in many domains, e.g. ice hockey, football and electric circuits. Estimates for the number of chunks acquired by an expert are typically of the order of 50,000 chunks [12]. The large number of chunks enable the expert to do two things. First, the expert can use the chunks as an efficient indexing mechanism into knowledge relevant for problem solving. Second, the chunks help the expert to ‘zero-in’ on unique features of the current problem. For example, in the chess domain, a piece in an unusual position will not have an associated chunk, priming the expert to consider the consequences of his position compared with the more familiar situation.

Template for parallel resistors



Possible rules and actions:

If r_A and r_B known, then $1/r_{Total} = 1/r_A + 1/r_B$

If I_A and I_B known, then $I_{Total} = I_A + I_B$

If V_{Total} known, then $V_A = V_B = V_{Total}$

Figure 3 : An example of a schema.

3.2 Knowledge for Problem Solving

According to Sweller [13], a *schema* is simply a “structure which allows problem solvers to recognize a problem state as belonging to a particular category of problem states that normally require particular moves”. This implies that “certain problem states can be grouped, at least in part, by their similarity and the similarity of the moves that can be made from those states.” Analysis of the performance of subjects and their protocols [6,4] has provided considerable evidence of the kinds of information contained in schemas; it is the nature of this information and its propagation which largely distinguishes expert behaviour from the novice. An example of a schema is given in Figure 3. The template provides the perceptual pre-conditions for determining whether this schema applies in the current situation. Various slots determine different behaviours depending on different constraints found in the problem. Such schemas have been shown to support flexible problem solving strategies and explain how experts can employ a direct forward-inferencing strategy when solving relatively simple problems [4]. The collection together of domain knowledge in schemas is similar to that achieved by frames [14]. The difference here is the emphasis on perceptual knowledge for forming the template.

However, in spite of such work on the content of schemas, and even their use in intelligent tutoring systems [15], no substantive computational model has been proposed for learning such representations. We address this question in the next section.

4. Learning Perceptual Schemas: CHREST+

The previous two sections have outlined the principles which underlie an efficient representation for supporting skill acquisition, and also the mechanisms thought to

explain the superior performance of human experts. There are two central underlying mechanisms: the acquisition of perceptual chunks and the acquisition of knowledge for problem solving. This section considers these in turn, illustrating how they may be implemented and combined in a machine learning architecture.

4.1 Learning Perceptual Chunks

The heart of the expert's representation is an efficient indexing mechanism to a large number of perceptual chunks. Here we describe an effective algorithm for learning and indexing this knowledge. The algorithm, CHREST [16,11], builds up a discrimination network from the presented examples. It functions in a similar manner to EPAM [17], but has some important differences. The discrimination network is learnt as follows.

The network consists of a collection of chunks, with each chunk representing a meaningful group of basic elements. For example, in the circuit domain, the basic elements are the resistors and connections; the chunks are the collections of resistors which comprise circuits and sub-circuits. The chunks are developed as the discrimination network grows through the processes of discrimination and familiarisation. Essentially, each node of the network holds a chunk of information about an object in the world. The nodes are interconnected by links into a network, with each link representing the result of applying a test to the object. When trying to recognise an object, the tests are applied beginning from the root node, and the links are followed until no further test can be applied. The information held in the stored chunk is then compared with that in the current object: the object matches the stored chunk if the chunk is a more general description than that of the object. At the node reached during sorting, if the stored chunk matches that of the object then familiarisation occurs, in which the chunk is specialised by adding more details of the features in that object. If the current object and the chunk at the node reached differ in some feature, then discrimination occurs, which adds a new node and a new link based on the mis-matched feature. Therefore, with discrimination, new nodes are added to the discrimination network; with familiarisation, the resolution of chunks at those nodes is increased.

There are two distinguishing features of CHREST which are important to its efficiency. First, information is not confined to the leaf nodes of the discrimination network (which is similar to the organisation in UNIMEM [18]). Hence, each node will contain information describing one perceptual chunk, and its descendant nodes will contain information about closely related chunks. The second feature is that CHREST uses perceptual chunks as tests. Thus, when discrimination occurs, instead of simply taking a single feature from the current object for use as a test, CHREST will search the network for a node which represents the mismatched sub-chunk; the information at that node can then be used as a test on the new link.

These two features have an important consequence for the use of CHREST to rapidly discriminate perceptual information: the node in the discrimination network relating to the currently perceived chunk will suggest, via its links, where the next piece of perceptual information should be looked for. Also, because the tests are themselves acquired perceptual chunks, the information looked for will not be at

the lowest level of resolution, but instead reflect the amount of knowledge of the system. It is usual for experts to require far fewer pieces of perceptual information to acquire a far greater amount of conceptual information, and this behaviour is captured quite effectively in CHREST [11].

4.2 Learning Knowledge for Problem Solving

The previous section described an algorithm which learns perceptual chunks. As suggested by previous researchers, e.g. [3,17,4], we assume that these chunks are the right information to be problem templates in a schema, as discussed in Section 3.2. What we require is a learning mechanism to associate with each template information relevant for solving the problems which it matches. As an example, we

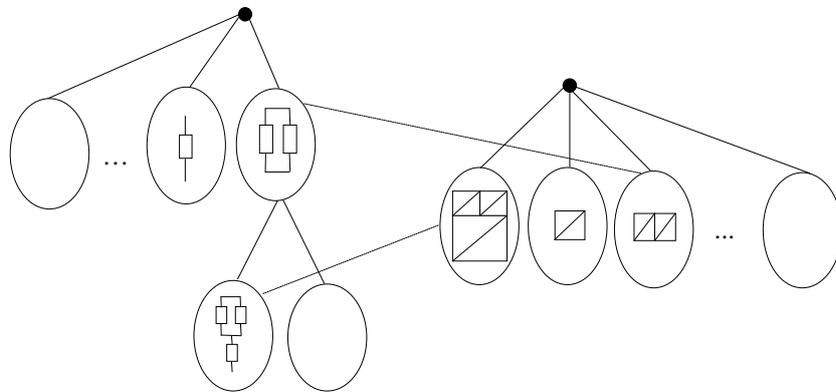


Figure 4 : Multiple discrimination networks, showing two equivalence links and the inheritance structure. Chunks are illustrated at some of the nodes.

consider the learner attempting to acquire schemas for the circuit domain from Section 2, and we will assume that the learner is working with AVOW diagrams.

We consider a learner in a supervised situation provided with a sample circuit and its equivalent AVOW diagram. The discrimination process for perceptually distinguishing instances from each of these representations will tend to build up separate networks, one for the circuits and one for the AVOW diagrams. This process can be performed by CHREST. We now use the methods for combining multiple networks described in [19] to provide CHREST with the ability to associate the node for the circuit diagram with the node for the AVOW diagram. This link between two equivalent representations is known as an *equivalence link*, and we call the extended model CHREST+. The result of this process is illustrated in Figure 4, where some equivalent nodes in discrimination networks for the two representations are shown.

If the learner sees the same circuit in a later setting, either on its own or as part of a larger circuit, then that circuit's chunk will be retrieved from the discrimination network. The equivalence link enables the learner to retrieve a representation for the final form of the AVOW diagram. This mechanism means the learner can decompose complex circuits into sub-circuits which have been encountered

previously. As was shown in Section 2.1, the visual pattern of an AVOW diagram works as a template, as it applies to a variety of possible quantities in the circuit. This means that each schema generalises to a wide range of possible problems, thus supporting rapid speed-up learning. However, other representations for solution paths may be used. For instance, if the learner is using algebra, the sequence of algebraic manipulations may be encoded as an ordered list and, through variabilisation, the particular equations may be generalised to a wider class of circuits. This ordered list may be discriminated by CHREST in the same way as the AVOW diagrams, and equivalence links added between the circuit and algebraic nodes. However, because the algebra does not generalise across instances as well as the AVOW diagrams, the complete system will not learn as rapidly, just as is the case with human subjects.

<i>Given quantities</i>	<i>Solution path for the unknown V_A</i>
$I_A r_A$	Compute $V_A = I_A r_A$
$I_B r_B$	Compute $V_B = I_B r_B$, then infer $V_A = V_B$
$I_B I_C r_A$	Infer $I_A = I_C - I_B$, then compute $V_A = I_A r_A$
$V_{TOT} V_C$	Infer $V_A = V_{TOT} - V_C$
$V_{TOT} I_C r_C$	Compute $V_C = I_C r_C$, then infer $V_A = V_{TOT} - V_C$

Table 1: Some possible problems from the circuit in Figure 1(a).

5. Avoiding the Utility Problem?

This paper is concerned with the organisation of knowledge for efficient speed-up learning. Earlier work has shown that systems such as PRODIGY [1] and SOAR [2] suffer from the utility problem, in which system efficiency is degraded by the cost of managing the collection of learnt rules. As Minton [1] explains: “To be useful, the cumulative benefits of applying the knowledge must outweigh the cumulative costs of testing whether the knowledge is applicable.” In Section 3 we argued that experts use chunks of perceptual information to index domain-specific information stored within schemas. In this section we discuss the efficiency of retrieving the relevant schema based on perceptual information.

As an example, let us consider the circuit in Figure 1(a). Restricting consideration to voltage (V), current (I) and resistance (r), this circuit contains at least 12 quantities: those for the individual resistors and values for the whole circuit. Any one of these may be used as the target unknown for a given problem. For example, assuming the unknown is the voltage across resistor A, V_A , Table 1 lists some possible problems and their solution path. One such table would be required for each of the other 11 quantities.

Imagine now a learning system confronted with a sequence of circuit problems and solution paths such as in the table. The question is how to store this information for later recall. Note that each solution path must be accompanied by its preconditions: in the second example in the table, the resistors A and B must be in parallel. As can

be seen, the amount of information which even a simple circuit structure may generate is large. The utility problem now arises because of the cost of locating, within all this information, the piece which relates to a novel situation. Such a system is described by Kieras [20], where explanations for solution paths are used to build up knowledge such as that in Table 1. Kieras notes that “in terms of CPU time, the pattern matching required to instantiate [these rules] can overwhelm the savings from the faster processing of explanations”. In other words, the utility problem arises. The assumption in many psychological models using production-rules is that all the productions are matched in parallel, but this condition is not met by designers of practical systems which usually possess a single or restricted number of processors.

In contrast, Section 3 has argued that experts do not directly store information relating to solution paths. Instead, they use easily indexed chunks of perceptual information as a means of referencing information about the kind of problems which arise from similar circuits. Thus, a schema for the circuit in Figure 1(a) will contain a perceptual chunk for the circuit as a whole, and then pointers to the possible solution paths contained in Table 1. The expert therefore does not immediately attempt to recognise a possible solution path out of all those theoretically possible, but instead locates a similar looking circuit: a task of much reduced complexity.

It should also be noted that this method facilitates reuse of previously learnt information. For instance, in recognising the example circuit, the expert, in passing, will identify the top two resistors as a parallel circuit. Any schema for that parallel circuit will thus also be indexed. This means that the information in the second line of Table 1 may be inferred from the parallel circuit schema and not require relearning for more complex examples.

There is still the question of matching the possible solution paths in the schema to the specific circuit quantities; for large circuits the number of possible paths can become large. It is at this point that the advantage of a diagrammatic representation may be seen. The AVOW diagrams discussed in Section 2.1 provide a unified representation for all the possible solution paths given in Table 1: independent of the target unknown and the given quantities, the problem solver must draw a correctly scaled AVOW diagram for the circuit. In every case, a similar series of operations must be carried out, and obtaining the quantity of the unknown is reduced to a simple measurement of the relevant dimension. This unification of solution paths into a diagram means that separate algebraic and inference techniques need not be learnt for every possible solution path. It also means that sequences of operators are automatically fitted to the values provided in the problem. This enables the solver to skip unnecessary steps [4,15].

Further, the separation of the solution template from the sequence of solution steps means the system may readily incorporate information about the problem from different sources. Therefore CHREST+ may be used as an additional mechanism to established problem solving techniques; the proposed solution template may enter a further module which considers its sequence of actions based on the template, other sources of planning information and formal look-ahead.

6. Conclusion

The primary conclusion from this paper is that the avoidance by human experts of the utility problem may be explained through efficient computational mechanisms for learning perceptual schemas. These mechanisms can be incorporated into standard machine learning systems, and also provide guidelines for knowledge representation to maximise the effects of learning.

However, this paper does not claim that the use of expert representations will entirely solve the problems of acquiring and using large volumes of information. This paper is instead intended as a contribution towards designing computers which emulate the ability of humans to learn effective and efficient rules for problem solving. As such, it has proposed a method for using expert representations in learning knowledge for problem solving, and also described principles and computational justifications underlying representations which support efficient problem solving.

References

- [1] Minton, S. (1990). Quantitative results concerning the utility of explanation-based learning. *Artificial Intelligence*, 42, 363-391.
- [2] Laird, J. E., Rosenbloom, P. S., and Newell, A., (1986). Chunking in SOAR: The anatomy of a general learning mechanism. *Machine Learning*, 1, 11-46.
- [3] de Groot, A. N. (1946). *Het denken van den schaker* [Thought and choice in chess]. Amsterdam: North-Holland.
- [4] Koedinger, K. R., & Anderson, J. R. (1990). Abstract planning and perceptual chunks: Elements of expertise in geometry. *Cognitive Science*, 14, 511-550.
- [5] Cheng, P. C-H. (1996). Scientific discovery with law-encoding diagrams. *Creativity Research Journal*, 9, 145-162.
- [6] Cheng, P. C-H. (1998). A framework for scientific reasoning with law encoding diagrams: Analysing protocols to assess its utility. In M. A. Gernsbacher & S. J. Derry (Eds.) *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 232-235). Mahwah, NJ: Erlbaum.
- [7] Cheng, P. C-H. (submitted). Electrifying representations for learning: An evaluation of AVOW diagrams for electricity.
- [8] Larkin, J. H. & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11, 65-99.
- [9] Tabachneck-Schijf, H. J. M., Leonardo, A. M., & Simon, H. A. (1997). CaMeRa: A computational model of multiple representations. *Cognitive Science*, 21, 305-350.
- [10] Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, 4, 55-81.
- [11] Gobet, F. & Simon, H. A. (in press). Five seconds or sixty? Presentation time in expert memory. *Cognitive Science*.
- [12] Simon, H. A. & Gilmarin, K. J. (1973). A simulation of memory for chess positions. *Cognitive Psychology*, 5, 29-46.
- [13] Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12, 257-285.
- [14] Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), *The psychology of computer vision*. McGraw-Hill, New York.
- [15] Koedinger, K. R., & Anderson, J. R. (1993). Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as Cognitive Tools*, Lawrence Erlbaum, New Jersey.

- [16] Gobet, F. (1998). Memory for the meaningless: How chunks help. In M. A. Gernsbacher & S. J. Derry (Eds.) *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 398-403). Mahwah, NJ: Erlbaum.
- [17] Feigenbaum, E. A., & Simon, H. A. (1984). EPAM-like models of recognition and learning. *Cognitive Science*, 8, 305-336.
- [18] Lebowitz, M. (1987). Experiments with incremental concept formation: UNIMEM. *Machine Learning*, 2, 103-138.
- [19] Gobet, F. (1996). Discrimination nets, production systems and semantic networks: Elements of a unified framework. *Proceedings of the Second International Conference of the Learning Sciences* (pp. 398-403). Evanston, III: Northwestern University.
- [20] Kieras, D. A. (1993). Learning schemas from explanations in practical electronics. In S. Chipman & A. L. Meyrowitz (Eds.) *Foundations of Knowledge Acquisition: Cognitive Models of Complex Learning*, Kluwer Academic Publishers.