A Cognitive Map for an Artificial Agent

Unmesh Kurup
Rensselaer Polytechnic Institute
Carnegie 013, 110 8th Street
Troy, NY 12180
kurupu@rpi.edu

B. Chandrasekaran
The Ohio State University
591 Dreese, 2015 Neil Ave
Columbus, OH 43210
chandra@cse.ohio-state.edu

Abstract

We show how a general-purpose cognitive architecture augmented with a general diagrammatic component can represent and reason about Large-scale Space. The diagrammatic component allows an agent built in this architecture to represent information both symbolically and diagrammatically as appropriate. Using examples we show (a) how the agent's bimodal representation captures its knowledge about large-scale space as well as how it learns this information while problem solving and (b) the agent's flexibility when it comes to using learned information and incorporating new information in solving problems involving large-scale space.

Introduction

An agent based on a general purpose cognitive architecture has the ability to work on a range of problems. Agents based on task-specific structures are usually restricted to particular task and problem domains. The advantages of task-specific knowledge such as faster solution times can however be realized in the general architecture case through the use of general-purpose learning mechanisms that account for the formation of task-specific knowledge from the general underlying representations. Newell has made a comprehensive case for the development of unified general architectures for solving the problem of general human cognition (Newell 1990). Soar (Laird, Newell et al. 1987) and ACT-R (Anderson and Lebiere 1998) have been two of the architectures that have tried to rise to the challenge of generality. For our purposes it is not their differences that are important but what they share. Their central representational framework is symbolic, or more precisely, predicate-symbolic. In the predicate-symbolic view, the agent's knowledge, goals etc are represented in terms of symbol structures that describe the world of interest in terms of properties of and relations between individuals in the world.

Cognitive models built in Soar and ACT-R have been very effective in showing how a general purpose cognitive architecture can produce task-optimal behavior, and how learning provides efficiencies over time and across tasks. However, work using these architectures has tended to focus more on certain kinds of tasks over others. Among the tasks that have received less attention are those that deal with the representation of and reasoning about large-scale space. As it happens, for any given spatial reasoning task, a model builder can represent the task-specific spatial information symbolically, treating the extraction of such symbolic information to perceptual processes outside the architecture. On the other hand, it has been proposed that there are certain forms of perceptual representation that belong inside the cognitive architecture itself (Chandrasekaran 2002). In this paper, we demonstrate how an agent based on a general cognitive architecture, albeit one with an additional diagrammatic representation, can represent and reason about space. We further show the agent can learn during problem solving and show transfer of learning within and between tasks.

Representing and Reasoning about Large-Scale Space

In 1948, (Tolman 1948) proposed that animals have an internal representation of large-scale space which he called the cognitive map. In 1960, (Lynch 1960) produced his seminal study of the environment in which he identified *landmarks, routes, nodes*, *districts* and *edges* as the features that are important in building a cognitive map. Since then there have been a number of models, both descriptive models without commitment to mechanisms, and computational models that propose mechanisms, which have been proposed to account for various phenomena associated with the representation of space (add refs). A variety of behavioral/psychological studies have also aided the development of these models by providing a set of characteristics or behaviors that a model should posses. Chief among them is the understanding that the cognitive map is less of a map and more of a collage (Tversky 1993). That is, representations of large-scale spaces are not holistic but stored in pieces and that these pieces are brought together as needed during problem solving. Such a representation allows the agent to be flexible with respect to representing inconsistent and incomplete information.

Knowledge of large-scale space can come from multiple sources. The most common, of course, being personal experience of navigation in space. We automatically build representations of our environment as we traverse them. A second, and important, source is maps. Our knowledge of large environments, such as the spatial extent and geographical locations of the fifty states of the USA, originated from our use of maps. Representations, originating from either source, are combined and modified in various ways and for various purposes during problem solving. In this paper, we focus on spatial reasoning tasks that involve an external map rather than the agent moving about in space. This is because biSoar, being built on Soar, is a theory of high-level cognition, and navigating in the world requires perception of the external world (as well as motor systems to act on it), capabilities which biSoar, and cognitive architectures in general, are not intended to capture. However, we believe our proposal can be suitably extended in the case when such abilities become available.

Generality comes at a price. Currently, biSoar agents cannot outperform more task-specific proposals for representing and reasoning about space. Instead, our focus in this paper is in showing how agents built in the biSoar architecture are flexible and versatile. Using examples we show how information about large-scale space can be represented in a piece-meal fashion in biSoar's underlying bimodal representation. We then show how an agent, during the course of problem solving, learns these pieces of information. We use two sets of examples to show biSoar's flexibility and versatility. In the first one, we show how information learned by an agent in one task (route-finding) can be used to solve problems in a different but similar task (geographic recall). In the second example, we show how the agent can incorporate information from multiple sources during an episode of route-finding.

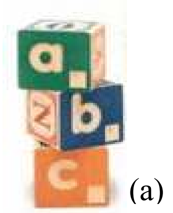
biSoar

To create biSoar (Kurup and Chandrasekaran 2006), a general-purpose cognitive architecture, Soar was augmented with the Diagrammatic Reasoning System (DRS), a domain-independent system for representing diagrams (Chandrasekaran, Kurup et al. 2005). The diagrammatic component of the state, encoded in DRS, is part of the agent's *internal* representation, just as the predicate symbolic component is in an agent's state representation in current theories of cognitive architecture. The content can come from various sources, recall from memory, imagination by composing elements from memory, or from an external image of a diagrammatic representation. DRS of an external diagram is an abstraction of the external diagram: regions in the image intended to be points are abstracted in DRS as points but with the same location of the intended point, regions intended to be curves are abstracted into the intended curves, symbolic annotations abstracted to symbols associated with the DRS element, and so on. The perceptual routines operate on the DRS elements, whether they were generated internally or from external representations. Of course, creating a DRS corresponding to an external diagram requires image processing routines, such as those that do background subtraction or edge detection, to go from an image to a collection of objects with their spatial extents. Such external processing, however, is not part of the theory of the bimodal internal state, nor of the operation of the cognitive architecture. In our examples that involve an agent interacting with an external map, we assume that such image processing routines are available to the agent and focus on how the result of these routines, the diagrammatic representation, is represented and manipulated in spatial reasoning tasks

While a physical diagram (on a screen or on paper) is an image that contains diagrammatic objects, each to be interpreted as a *point*, *curve* or a *region*, the diagram is viewed as a configuration of diagrammatic objects. Note too that while in the physical diagram all the objects are regions, so that they can be perceived, DRS captures the intended diagram. If an object in the physical diagram appears as a circle, in DRS it would be treated as a Euclidean point object with just location to characterize it. DRS is domain-independent – the only objects are points, curves or regions. Interpreting them in domain terms is the job of the user of the representation. The objects in DRS have associated with them information about their spatiality -- locations for point objects, and representations that are functionally equivalent to the sets of points that constitute the objects for curves and regions. Associated with the DRS are a set of perception and diagram construction/modification capabilities; following (Ullman 1984), these are called *routines*. All these routines are visual, but we use the more general term so that it will apply to the more general multi-modal view.

Perception Routines take diagrammatic elements as arguments and return information about specified spatial properties or spatial relations. There are two types of perception routines: the ones in the first type re-perform the figure-ground separation on the image – rather than on the DRS – perceiving emergent objects (e.g., the two sub-regions that emerge when a curve intersects a region.) Routines of the second type return specified spatial properties of objects, e.g., the length of a curve; and evaluate specified spatial relations between objects, e.g., Inside(Region1, Region2). These routines work from descriptions in DRS. DRS thus is an intermediate representation that supports reconstituting the image, a capability needed for emergent object identification, and also the perceptual routines that perceive properties of and relations between objects.

Routines that help in constructing or modifying the diagram are *action routines.* They create diagrammatic objects that satisfy specific perceptual criteria, such as "a curve object that intersects a given region object," and "a point object inside the region object." The sets of routines are open-ended, but routines that are useful across a number of domains are described in (Chandrasekaran, Kurup et al. 2004), which also contain more information on DRS.



Working Memory:
Block (A), Block (B), Block (C), On (A,B), On (B,C)

Selected Operator: None

(b

Figure 1: (a) Blocks World and (b) Soar's representation of the world in (a).

Working Memory: Block (A), Block (B), Block (C), On (A,B), On (B,C) Selected Operator: None	A B C

Figure 2: biSoar representation of the world shown in 1(a)

Cognitive State in Soar

Soar's representations are predicate-symbolic. The cognitive state in Soar is represented by the contents of Soar's WM and operator, if any, that has been selected. Fig 1(b) shows Soar's cognitive state representation of the blocks world example in 1(a).

Cognitive State in biSoar

The cognitive state in biSoar is bimodal – it has both symbolic and diagrammatic parts. Fig 2 shows the bimodal representation of the world depicted in Fig 1(a). Working memory in biSoar is represented as a quadruplet, with each Identifier, Attribute, Value triplet augmented with a diagrammatic component in DRS that represents the visualization (metrical aspect) of the triplet. Since not all triplets need to be (or can be) visualized, the diagrammatic components are present only as needed. States represent the current or potential future state of interest in the world and the symbolic and the diagrammatic part may represent related or distinct aspects of the world. However, the diagrammatic representation is "complete" in a way that the symbolic representation is not. For example, from the symbolic representation alone it is not possible to say without further inference whether A is above C. But the same information is available for pick up in the diagram with no extra inference required. This has advantages (for instance in dealing with certain aspects of the Frame Problem) and disadvantages (over-specificity).

Bimodal LTM and Chunking

There are two questions that have to be answered in an implementation of Long Term Memory (LTM) – how are elements put into LTM (i.e., learned) and how are elements retrieved from LTM. In the case of Soar the answers to these two questions are chunking for learning and a matching process that matches the LHS of a LTM rule to WM for retrieval.

Chunking **-** Chunking simply transfers the relevant contents of WM to LTM. In the case of biSoar, chunking transfers to LTM both symbolic and diagrammatic elements present in WM.

Matching - In the case of Soar the retrieval process is straightforward because matching (or even partial matching when variables are present) symbols and symbol structures to each other is an exact process; either they match or they don't. When the cognitive state is bimodal, WM has metrical elements in addition to symbols. Matching metrical elements to each other (say a curve to another curve) is not an exact process since two metrical elements are unlikely to be exactly the same. Matching metrical elements would require a different approach like a non-exact process that can match roughly similar elements in a domain-independent manner (since the matching should be architectural). It may also turn out that only calls to perceptual routines are present in LTM while matching metrical elements is a more low-level cognitive process present only in stimulus-response behavior. For now we take the latter approach where the LHS of biSoar rules contain only symbol structures while the RHS contains calls to the diagram that execute perceptual routines. The results of executing these routines appear as symbol structures in the symbolic side at the end of a decision cycle. We think that this approach can account for many of the diagrammatic learning capabilities that are required in

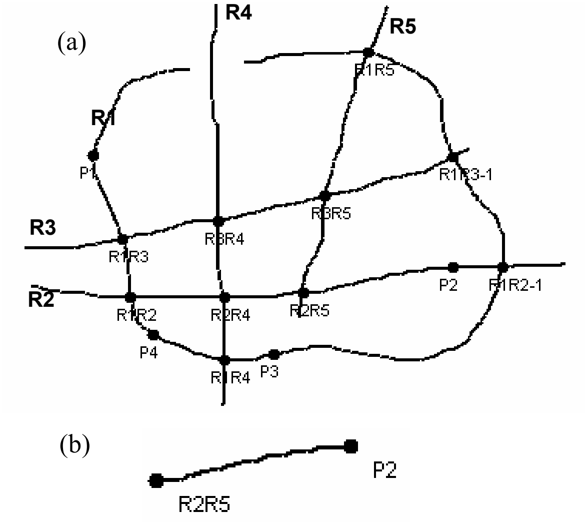


Figure 3: (a) A map of main highways in Columbus, OH showing routes R1...R5 and locations P1...P4. Intersections of routes also form additional locations. (b) The DRS representation of the route from R2R5 to P2

models of cognition except in cases where goal specifications contain irreducible spatial components, such as might be the case in the problem solving of a sculptor.

1. locate the starting & destination locations in the map
2. make the starting location the current location
3. Find the routes on which the current location lies
4. For each route, find the directions of travel
5. for each route and direction of travel, find the next location
6. calculate the Euclidean distance between these new locations and the destinations
7. pick the location that is closest to the destination and make that the current point
8. repeat 3-8 until destination is reached

Figure 4: The wayfinding strategy used by the biSoar agent

Representing Large-Scale Space in BiSoar

Soar's LTM is in the form of rules. Each rule can be thought of as an if-then statement where the condition (if) part of the rule matches against the existing conditions in WM and the action (then) part of the rule describes the changes to be made to WM. Thus, Soar's LTM is arranged to respond to situations (goals) that arise as part of problem solving. This makes sense because, ideally, a majority of an agent's rules are learned as part of problem solving and hence in response to a particular situation. If the situation (or a similar one) arises in the future, Soar can now use this learned rule in response. For the route-finding scenario, the agent has knowledge about various locations on the map and about routes between these locations, presumably as a result of learning from previous problem solving episodes. The agent's knowledge of the area consists of bimodal rules in the following form:

If goal is find_destination and at location A and traveling in direction Dx on route Rx, then destination is location B, diagram is DRSx

Where DRSx represents the spatial extent of the section of the route that runs from Lx to Ly along route Rx and is represented using DRS. So, for example, for the map in Fig 3(a), the spatial relationship between locations R2R5 and P2 would be expressed as

If goal is find_destination and at R2R5 and traveling Right on Route R2, then destination is P2, diagram is DRS1

Where DRS1 is shown in Fig 3(b). The directions available are Right, Left, Up and Down though this choice of directions is arbitrary. For convenience, we represent the above information in a more concise form as follows:

Gx,Lx,Dx,Rx → Ly,DRSx

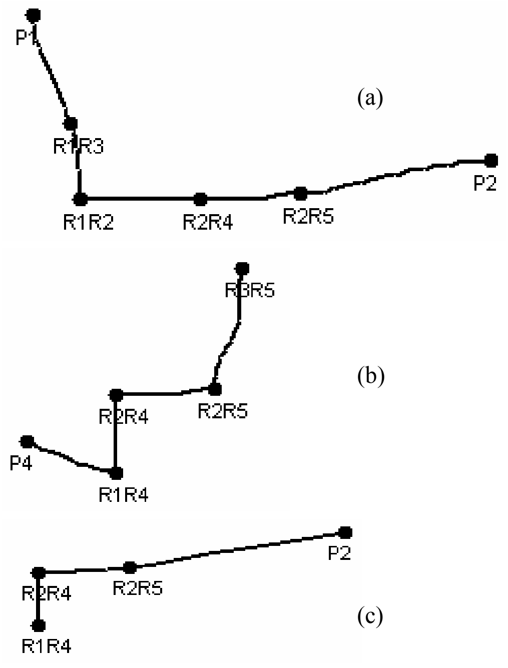


Figure 5: Routes found by the agent from (a) P1 to P2 b) P4 to R3R5 and (c) R1R4 to P2

Route-finding & Learning Using BiSoar

We created an agent to perform route-finding tasks given a map using the simple strategy shown in Fig 4. The agent finds a route by locating the starting and destination points and finds a path by moving to the next point on the route that is the closest to the destination using a simple Euclidean distance measure. Some of the steps of the algorithm require information from the diagrammatic component. This information is of two types - in certain cases it is symbolic, such as the answer to the query "On what route(s) is the point located?" while in other cases, the information is diagrammatic, like for the question "What's the route between the point P1 and R1R3?"

Each step of the algorithm is implemented such that it becomes a sub-goal for the biSoar agent. As a result, when the agent solves a sub-goal, the architecture automatically learns a chunk (rule) that captures the solution to the task. For example, corresponding to step 5 of the algorithm, the agent finds that if you move down from P1 along route R1, you reach R1R3. The next time the agent is faced with this sub-goal, the chunk that was learned is used to answer it. Fig 5(a) shows the path found by the agent from P1 to P2. Table 1 shows the information learned by the agent as a result of that task. Fig 5(b) similarly shows the result of route-finding by the agent between the locations P4 and R3R5 and Table 2, the information learned by the agent

Within-task Transfer – To show within-task transfer of learning, we ran the agent on route-finding tasks with an external map. As a result of learning, the agent acquired a number of chunks. The external map was then removed and the agent was giving a new route-finding task, one in which the starting and destination points were present in the previous tasks but never as a pair between which a route had to be found. For example, with the external map available, the agent was asked to find routes from P1 to P2 and from P4 to R3R5. The external map was then removed and the agent was asked to find a route from R1R4 to P2. The agent using information learned during the previous two tasks (Tables 1 & 2), found the route in Fig 5(c).

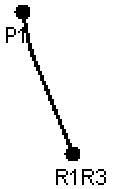
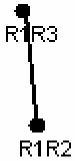
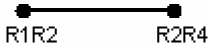
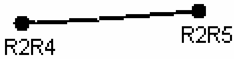
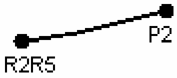
Gon,P1 → R1	
Gon,R1R3 → R1,R3	
Gon,R1R2 → R1,R2	
Gon,R2R4 → R2,R4	
Gon,R2R5 → R2,R5	
Gdir,P1,R1 → up,down	
Gdir,R1R3,R1 → up,down	
Gdir,R1R2,R2 → right,left	
Gdir,R2R4,R2 → right, left	
Gdir,R2R5,R2 → right,left	
Gdest,P1,R1,down → R1R3	
Gdest,R1R3,R1,down → R1R2	
Gdest,R1R2,R2,right → R2R4	
Gdest,R2R4,R2,right → R2R5	
Gdest,R2R5,R2,right → P2	

Table 1: rules learned by the agent in finding the route from P1 to P2

.

Between-task Transfer – To show between-task transfer, we had an agent trained on one type of task (route-finding) perform a different spatial task (geographic recall). As an example, the agent in the within-task transfer example, with the knowledge in tables 1 and 2, was asked to find the geographic relationship between two points that it had encountered during the tasks. The agent's strategy was to recreate (not just recall) the map using the learned information, and extract the relationship between the locations from the re-created map. Fig 6 shows the map created by the agent.

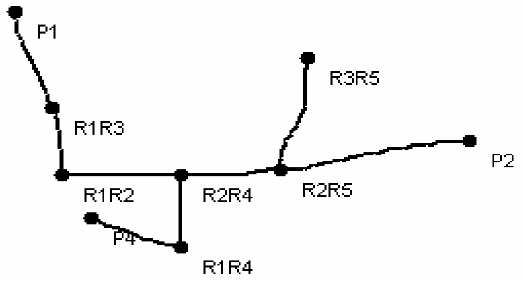


Figure 6: Map created by the agent for geographic recall

Predicting Paths – To show how the agent can incorporate information from multiple sources, we gave the biSoar agent an incomplete version of the earlier map (as shown in Fig 7) with route R5 missing. The agent was also given the symbolic information that there is a path from R2R5 in the up direction, but because the diagram was incomplete it is not clear what the destination of traveling along that route is. The goal of the agent was to find a route from P2 to P3 taking into account possible paths. Fig 8 shows the route found by the agent. The strategy shown in Fig 4 is slightly modified so that the agent makes predictions about destinations. In step 5 of the strategy, when the agent checks the external representation to find the destination from R2R5 on R5 in the "up" direction, it finds that the information is not available in the map. In response, the agent creates a straight line path from R2R5 in that direction and sees that it intersects R3. It names this point INT1 and proposes a route that goes through INT1. Comparing the map in Fig 3(a) and the route found in Fig

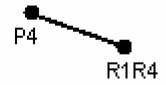
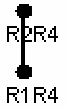
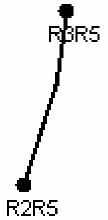
Gon,P4 → R1	
Gon,R1R4 → R1,R4	
Gdir,P4,R1 → left,right	
Gdir,R1R4,R4 → up	
Gdir,R2R5,R5 → up	
Gdest,P4,R1,right → R1R4	
Gdest,R1R4,up → R2R4	
Gdest,R2R5,up → R3R5	

Table 2: rules learned by the agent in finding the route from P4 to R3R5

7(b), we can see that the straight line assumption by the agent results in a slightly different route R5 than what would have been found using the complete map.

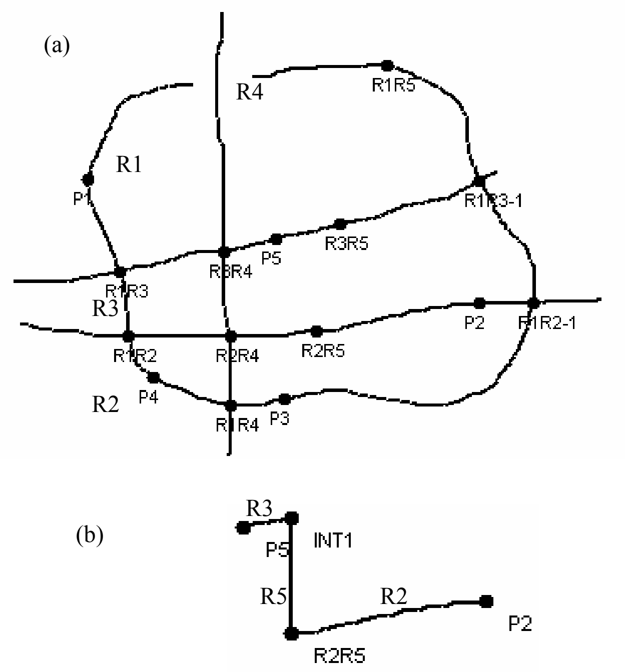


Figure 7: (a) Map without route R5 for the predicting paths task. (b) Route found between P2 and P5

Conclusion

Representations of large-scale space in general purpose architectures are usually limited to topological graph-like representations due to constraints imposed by the underlying predicate-symbolic representation. Reasoning, most commonly route-finding, then proceeds via the application of a graph-traversal algorithm. In our bimodal architecture, biSoar, large-scale space is represented using both symbolic and diagrammatic representations. This bimodal representation provides a richer representational format that can be used to solve a wider range of spatial reasoning tasks. At the same time, the diagrammatic representations are not specific to large-scale space but part of a more general approach to understanding cognition as multi-modal. In the case of large-scale space reasoning, such an approach has the following benefits. First, it captures not only the topological but also the metrical aspects of space. Depending on the task, either or both of the representations can be used to solve the problem. Second, an agent in this architecture can learn both symbolic and diagrammatic elements via chunking. This information can then be used to solve similar and related tasks. Third, the agent's representation of the space is not holistic in nature. Instead it is spread over a number of rules and smaller diagrammatic pieces. This allows the agent to function under the presence of inconsistencies as well as include information from multiple sources during problem solving. The lack of a consistent single representation also makes it easier for the agent since it does not have to maintain consistency as new information comes in. In these respects, the agent's representation is similar to human spatial representation. Lastly, the presence of a metrical representation allows the agent to reason about space in a way that topological representations cannot, namely, in predicting destinations of paths or finding shortcuts or novel paths.

Acknowledgements

Advanced Decision Architectures Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0009.

References

Anderson, J. R., and Lebiere, C. 1998. *The Atomic Components of Thought*, Lawrence Erlbaum Associates.

Chandrasekaran, B. 2002. Multimodal Representations as Basis for Cognitive Architecture: Making Perception More Central to Intelligent Behavior. *Intelligent Information Processing*, Kluwer Academic Publishers.

Chandrasekaran, B.; Kurup, U.; Banerjee, B. 2005. A Diagrammatic Reasoning Architecture: Design, Implementation and Experiments. AAAI Spring Symposium *Reasoning with Mental and External Diagrams: Computational Modeling and Spatial Assistance*.

Chandrasekaran, B.; Kurup, U.; Banerjee B.; Josephson, J.; Winkler, Robert. 2004. An Architecture for Problem Solving with Diagrams. *Diagrammatic Representation and Inference conference*, Springer-Verlag.

Kurup, U., and Chandrasekaran, B. 2006. Multi-modal Cognitive Architectures: A Partial Solution to the Frame Problem. *28th Annual Conference of the Cognitive Science Society*, Vancouver.

Laird, J. E.; Newell, A.; Rosenbloom, P. 1987. Soar: an architecture for general intelligence. *Artificial Intelligence* 33(1): 1-64.

Lynch, K. 1960. *The Image of the City*. Cambridge, MIT Press.

Newell, A. 1990. *Unified theories of cognition*. Harvard University Press.

Tolman, E. C. 1948. *Cognitive Maps in Rats and Man*. Psychological Review 55: 189-208.

Tversky, B. 1993. Cognitive maps, cognitive collages, and spatial mental model. *Spatial information theory: Theoretical basis for GIS*. U. Frank and I. Campari. Heidelberg-Berlin, Springer-Verlag: 14-24.

Ullman, S. 1984. *Visual Routines*. Cognition 18: 97-159.