To appear in the Proceedings of the Second Game Programming Workshop in Japan, September 1995.

The Integration of Cognitive Knowledge into Perceptual Representations in Computer Go

Jay Burmeister [α β], Janet Wiles [α β], Helen Purchase [α]
[α] Department of Computer Science [β] Department of Psychology
The University of Queensland, QLD 4072, Australia

e-mail: jay@cs.uq.edu.au
HTTP: http://www.psy.uq.edu.au/~jay/research/research_page.html

Abstract

This project targets one facet of Go programming - how to relate simple patterns of stones (e.g., kogeima links) to high level properties (e.g., the outcome of a ladder) for a given context (e.g., the presence or absence of a ladder-breaker stone). The numeric representation generated by a simple influence function provides the "perceptual representation". The conditions under which a ladder is won constitutes "cognitive knowledge". As Cognitive Scientists, our goal is to design algorithms for integrating cognitive knowledge about Go into perceptual representations which may be used by others to build Go programs.

Typical artificial intelligence (*AI*) approaches to Go programming are based on the translation of perceptual information (modelled by pattern recognition processes) to symbolic form (modelled by rule-based systems) for access by symbolic reasoning processes. In this project we explore the converse process - seeking to integrate cognitive knowledge into a core perceptual representation which can then be accessed by symbolic reasoning processes. The underlying theory is an extension of Hofstadter's theory of high-level perception, originally applied to letter-string analogies in the Copycat project (Chalmers, French and Hofstadter, 1990).

We demonstrate the concrete instantiation of our theory in four simulations. The core algorithm we used in the simulations integrated cognitive knowledge by directly modifying the numeric information contained in the perceptual representation of the board. The four simulations each implemented different variations of our core algorithm (i.e., static, additive, multiplicative and context-sensitive modifications of influence values).

Introduction

This project concerns the integration of perceptual and cognitive knowledge. The development of a theory for such an integration first requires an understanding of the representation (i.e., the format of the information) that comprises both perceptual and cognitive knowledge with such representations being necessarily domain specific.

Definitions of Perception and Cognition

Perception can be viewed as a process whereby a system, either human or machine, senses its environment and forms an internal representation appropriate to the system's needs. In symbolic AI terms, perception is the formation of a symbolic representation. Forming a representation would require sensing and deciphering the state of the environment and instantiating system variables, slots or memory with the information obtained. In cognitive science terms, a perceptual process is any process that is involved in forming a "mental model" of a system's environment. Such processes would include processes involved in gathering raw data e.g. visual detectors, or in filtering or encoding the data into a final representation. Similarly, the entire process of creating the representation is often referred to as "the perceptual process".

In both AI and cognitive modelling, massively parallel numeric representations are used e.g. matrices, Fast Fourier Transforms, Gabor filters, and neural nets.

Cognition can be viewed as high-level operations that are performed on a system's internal representation to satisfy its needs or achieve its goals. If considered at all, researchers in AI use the term "cognition" to refer to a system's manipulation of its internal symbolic representation in response to a particular problem presented to it. Cognitive processes may include processes ranging from those involved in implementing search heuristics and control strategies to those involved in machine learning, theorem proving, planning, natural language processing etc. In both AI and cognitive modelling, representations are typically symbolic and use the "traditional symbols" of high level languages such as Lisp.

The Traditional AI Approach to Forming Representations is Problematic

A vast gulf exists between the numeric representations formed by perceptual processes and the symbolic representations used by cognitive processes. The traditional AI view is that there is a one-way flow of information from perceptual processes to cognitive processes. Thus, information is processed in a bottom-up manner and translated into symbolic rules in a high-level language e.g. Lisp. In AI approaches to cognitive modelling, the perceptual process is usually omitted from the model and the existence of a "representational module" which forms the system's initial representation is assumed.

There are three problems with the assumption of a representation module. The first is the dual problem of deciding what information is relevant and how it should be organized. The second is that implicit in the assumption of a representational module is the objectivist view that there is a single correct representation for objects, relations and events. The third is that a representation module does not allow a context-sensitive representation to be formed (see Chalmers et al., 1990, for an in depth treatment of these issues).

A Less Problematic Approach: The Copycat Model

Research undertaken by members of Hofstadter's research group at Indiana University criticises the traditional one-way flow of information approach to representation formation. In the Copycat project (Chalmers et al., 1990), the domain of analogical mapping between letter-strings was used to show how low-level and high-level perceptual processes can interact to form a context-sensitive representation. The problems solved by Copycat are examples of tasks that require an interaction of perceptual and cognitive processes.

The following example illustrates how perception and cognition interact in Copycat. If the string *ppqrss* was to be analogically mapped to the string *aamnxx*, it would be most natural to view the second string as *aa-mn-xx* and the first string as *pp-qr-ss*. However, if *ppqrss* was to be analogically mapped to *aijklx*, it would then be more likely that the second string would be viewed as *a-ijkl-x* and the first string as *p-pqrs-s*. Thus, the way in which the string *ppqrss* is viewed depends on the content of the string to which it is being analogically mapped.

In the Copycat model, human perception is described as being composed of low-level and high-level perception. Low-level perception filters and forms representations from the large amount of sensory information present in the environment. High-level perception is responsible for extracting meaning from the representation passed to it from low-level perceptual processes and creating various representations based on that information that are then used when performing high-level (cognitive) tasks. We believe that high-level perception as defined by Hofstadter and his colleagues lies at the interface between perception and cognition; it receives bottom-up information from low-level perceptual processes and top-down information from cognitive processes.

To develop a theory of integration of perceptual and cognitive knowledge, the nature and utility of cognitive knowledge needs to be re-examined. The Copycat theory of perception stimulates such a re-examination.

To appear in the Proceedings of the Second Game Programming Workshop in Japan, September 1995.

Perception and Cognition in Go

The only *link* between stones which is recognized in the rules of Go is between horizontally or vertically adjacent stones. In practice however, there are many virtual links between stones which are not horizontally or vertically adjacent which are recognized by experienced Go players. An example of a virtual link is the kogeima link (described below).

The stones involved in the virtual links are considered to be *connected* since an appropriate placement of additional stones would enable the intervening points to be filled. The conditions under which two *strings*[1] can be considered to be connected by a virtual link depend on contextual factors which range from physical to abstract. The physical factors include the placement of friendly and hostile stones both locally and far across the board, and the abstract factors include the opponent's goals and plans etc. Thus, as in the Copycat model, the way in which stones, strings and virtual links are perceived depends on contextual factors which could be determined in a top-down manner.

Influence Functions

To evaluate proposed board positions, programs use heuristics such as estimating the territory controled by each side. *Influence functions* are heuristics which propagate patterns of positive and negative numeric values from each stone to show how much influence they have on surrounding points. Typically, white stones generate negative influence and black stones generate positive influence. The influence values are summed at each point to give an overall influence value which indicates the relative size of black and white influence for each point. Influence functions are typically distanced-based with the influence exerted by a stone decaying as distance increases. There are various ways of propagating influence. The algorithm described by Chen (1989) radiates influence onto unoccupied points whereas others e.g. Zobrist (1969) and Ryder (1971) radiate influence to occupied and unoccupied points of the board alike. *Influence maps* contain the numeric influence values for a given board configuration.

Connectivity

Connectivity is a (computational) measure of how likely it is that two stones can be successfully linked together to form a string. For any two stones of the same colour, if there are no stones of the opposite colour between them, it seems reasonable that they could be connected by the appropriate placement of additional stones. One of the heuristic ways of determining the level of connectivity between two strings is the use of influence values. The higher the influence values between two stones, the more likely it is that they can be linked into a single string whereas the lower the influence values between stones the more likely that they will be cut by the opponent.

A threshold of connectivity can be set by a programmer (depending on implementation specific details) which will serve to classify links between strings as either secure or threatened. If there is a path of unoccupied board points between two stones (called *connection points*) whose influence values do not fall below the threshold of connectivity, the link can be considered to be secure. The converse holds if every potential connection path between two stones has at least one point whose influence value is below the threshold of connectivity.

Computational Problems With Using Influence to Determine Connectivity in Go

Although determining the connectivity of between strings is a relatively easy perceptual task for a moderately experienced Go player, algorithms used in Go programs are not, on the whole, very successful in determining connectivity between strings. Influence functions are necessarily local and do not take into account global factors which affect connectivity. Influence functions, while suitable for short range interactions, do not adequately capture the effect

1. We use the term "string" to denote one or more stones that are horizontally or vertically adjacent.

of factors beyond the local region, nor are they sensitive to the high level (or symbolic) knowledge possessed by even a beginner human player.

The problem of connectivity is an example of the interface between perceptual and cognitive issues in Go.

Ladders

One particular sub-task of Go that requires the integration of perceptual and cognitive knowledge is the readout of *ladders*. Influence functions, although useful for determining local connectivity, can not alone be used to evaluate the outcome of a ladder. The importance of stones outside the local context is difficult to evaluate other than at a global level.

The points on which the black and white stones involved in a ladder are played are called the *ladder-points*. If a player already has a stone on any of the ladder points before trying to trying to cut an opponent's kogeima link, the kogeima link would be successfully cut. Such a stone is called a *ladder-breaker.* Thus, the context of the ladder points needs to be taken into consideration in evaluating the strength and security of a kogeima link.

The winner of a ladder can be determined a priori by competent human players and therefore the losing player will not initiate it. Other than when played by novices, ladders do not appear in normal play which makes them difficult to learn experientially.

Since a ladder-breaker stone can be remote from the initial ladder site, it is difficult for simple distance-based heuristics to recognise them as having an effect. Thus, knowledge derived from a local influence function would be misleading since it would not contain context-sensitive knowledge pertaining to ladder-breakers which fell outside its scope.

Computational Solutions to Ladders

Typically in Go programs, ladders are solved by tactical means whereby a special algorithm is used to determine the outcome of a ladder once it has been recognized (Fotland, 1993). Due to the structure of ladders and the nature of the moves involved (forcing moves) it is relatively easy for a program to trace out the path of a ladder (which may be up to 80 moves).

However, the success of programs at incorporating tactical knowledge about the results of a ladder into strategic considerations is not so good. A human player having recognised that two stones that were kogeima linked are potentially cut by a ladder-breaker, will "see" that friendly influence at that point is somewhat lower than it first appears. Programs however, will need explicit intervention to "see" the consequences of the ladder-breaker as affecting the influence around the kogeima linked stones. The consequences of not "seeing" the effect that the ladder-breaker has on the local influence is that when a program makes decisions based on the influence values, it is not taking into consideration the effect of the ladder-breaker. A human player in the same situation would easily be able to incorporate the effect of the ladder-breaker and would consequently make sounder tactical and strategic decisions.

Simulations: Ladder-breakers and Influence Functions

The issues raised above highlight the use of a specific domain as a test bed for generating and testing ideas for cognitive theories. In summary, the issues concern the integration of cognitive knowledge into a perceptual representation, and the domain of study is connectivity problems in the game of Go. The specific task chosen is to determine the connectivity of two stones that are kogeima-linked given the presence or absence of ladder-breakers. Our goal is to demonstrate through simulation a method for instantiating the theory in algorithm design[1].

Perceptual and Cognitive Information

The first step in modelling the task is to develop an understanding of the information available at each stage in solving the task, and the form of such information (i.e., the represen-

1. In order to focus attention on the method of integrating knowledge, we have concentrated on designing very simple algorithms and have bypassed the multitude of issues required for a complete Go program.

tation). Since we are interested in the integration of cognitive knowledge into a perceptual representation, we start by specifying just what information comprises cognitive knowledge in the domain and what format the perceptual representation will have.

The perceptual representation in our simulations corresponds to the influence map, which is a numeric representation of the proximity of black and white stones from any position on the board. Influence maps are a heuristic device that, failing any further knowledge, can form the basis of connectivity decisions. For any two stones of the same colour, if there are no stones of the opposite colour between them, it seems reasonable that they could be connected by appropriate placement of additional stones. The influence heuristic, while adequate for short range
interactions, does not adequately capture the effect of factors beyond the local region, nor is it sensitive to the high level (or symbolic) knowledge of even a beginner human player.

The cognitive knowledge in our simulations corresponds to knowledge about the kogeima link, and possible factors beyond the local region of the "linked" stones. More specifically, the content of the cognitive knowledge is the existence of ladder-breaker stones and their effect on the connectivity of the link (i.e., any stone placed in the path of a ladder which changes the likelihood of the link being made or actually leads to it being broken).

Integration of Cognitive Knowledge into a Perceptual Representation

In the simulations, the integration of cognitive knowledge into the perceptual representation is realised by the modification of the numeric information contained in the perceptual representation. Thus the numeric values returned by the influence function (the perceptual representation) are modified around the kogeima link to reflect the effect of the ladder-breaker stones at various distances away from the kogeima link (cognitive knowledge).

METHODS

An initial simulation was carried out in which the influence values for a board configuration containing the beginning of a ladder was calculated to provide a baseline for comparison. The influence values for additional board positions containing ladder-breaker stones at increasing distances from the stones in the initial configuration were also computed. Four simulations replicating the above procedure were then carried out. In each simulation, selected influence values were modified using different methods.

Influence Function

An influence function based on the one described in (Chen 1989) was implemented in AKCL Lisp. The influence exerted from stone *st* to point *pt* was:

$$influence(st, pt) = m * f^{(d(st, pt) - 1)}$$

where:

m, the maximum value = 64,

f, the decay factor = 0.5,

d is the distance between *st* and *pt* in unoccupied points, and

influence < 1 was ignored (i.e. d(*st*, *pt*) > 7 was ignored).

The influence propagated from each white and black stone was summed at each point to give the final influence value for every unoccupied point on the board. Black stones propagated positive influence whilst white stones propagated negative influence.

Initial Position

A 13 x 13 board position is shown in Figure 1. A kogeima link exists between the black stones at (G,3) and (H, 5). The Black player must play two stones on the connection points between (G, 3) and (H, 5) to physically link the two stones that are kogeima linked, and has three options to achieve this: (H, 3) and (H, 4); (G, 4) and (H, 4); (G, 4) and (G, 5). The influence map for the initial position is superimposed on the stones in Figure 1 with the four connection points shown within a box and the ladder-points shown between two dashed lines.

The Ladder-breaker Stones

Ladder-breaker stones were added to the initial board position in one of three positions - *close* (F,6), *medium* (D, 8) and *far* (B,10).

	1	2	3	4	5	6	7	8	9	0	1	2	3	
A	**0**	**0**	**3**	**3**	**2**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	A
B	**0**	**0**	**7**	**6**	**6**	**4**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	B
C	**0**	**1**	**15**	**13**	**11**	**8**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	C
D	**1**	**3**	**31**	**25**	**21**	**15**	**5**	**2**	**0**	**0**	**0**	**0**	**0**	D
E	**3**	**7**	**61**	**49**	**41**	**30**	**9**	**5**	**2**	**0**	**0**	**0**	**0**	E
F	**6**	**3**		**110**	**88**	**62**	**20**	**9**	**5**	**2**	**0**	**0**	**0**	F
G	**-37**			**124**	**128**	**100**	**26**	**12**	**6**	**3**	**1**	**0**	**0**	G
H	**-1**	**-14**	**67**	**86**			**12**	**3**	**1**	**0**	**0**	**0**	**0**	H
I	**-4**	**-14**	**20**	**16**	**2**			**-53**	**-27**	**-14**	**-7**	**-3**	**-2**	I
J	**-3**	**-9**	**7**	**2**	**-11**	**-76**	**-83**	**-36**	**-18**	**-9**	**-5**	**-3**	**-1**	J
K	**-1**	**-5**	**3**	**1**	**-6**	**-38**	**-42**	**-18**	**-9**	**-5**	**-3**	**-1**	**0**	K
L	**-1**	**-2**	**1**	**0**	**-3**	**-20**	**-21**	**-9**	**-5**	**-3**	**-1**	**0**	**0**	L
M	**-1**	**-1**	**1**	**0**	**-2**	**-10**	**-11**	**-5**	**-3**	**-1**	**0**	**0**	**0**	M
	1	2	3	4	5	6	7	8	9	0	1	2	3	

Figure 1 Influence map for the initial board position. A kogeima link exists between the black stones at (G,3) and (H, 4). The ladder-points are contained between the two dashed lines. Connection points are shown enclosed i.e. (H,3), (H,4), (G,4) and (G,5).

Two Methods for Selecting Modification Points

Two methods for selecting points for modification were used. The *simple* method was to select the four connection points for modification. The *extended* method was to select the four modification points and the inner two lines of ladder points up to the ladder-breaker stone for modification.

Four Methods for Modifying Influence Values

In the *Static* method the selected influence values were changed to a static value below the threshold of connectivity. The threshold of connectivity was arbitrarily set to +30 for black connectivity.

In the A*dditive* method the selected influence values had a fixed value added to them. The additive value used was -60.

In the *Multiplicative* method the selected influence values were multiplied by a constant factor and rounded up. The multiplier used was 0.2.

In the *Context-sensitive* method the selected influence values by a factor depending on whether they were positive or negative.The formula used was:

if *inf* * *colour* > 0 then

inf' = 1.2 * *inf*

else

inf' = 0.2 * *inf*

where:

inf is the initial influence value of a particular point,

inf' is the modified influence value of a particular point (rounded up), and *colour* is +1 if ladder-breaker is black and -1 if white.

The Four Simulation Studies

Four simulations were conducted in which each simulation had both a selection and modification method associated with it. The *Static* simulation used simple selection and static modification. The *Additive* simulation used extended selection and additive modification. The *Multiplicative* simulation used extended selection and multiplicative modification. The *Context-sensitive* simulation used extended selection and context-sensitive modification.

RESULTS

Influence maps were generated for each ladder-breaker position in each simulation (resulting in 12 maps of which only a sample are included here). From the maps, it was observed that as the ladder-breaker stone was moved further away from the kogeima link, it had a decreasing effect on the influence values at the four connection points. In the furthest position, the ladder-breaker stone had no effect at all on the influence values of the connection points (see Figure 2). Thus, it was observed that local influence functions do not capture the known effects of distal stones.

	1	2	3	4	5	6	7	8	9	0	1	2	3	
A	**0**	**0**	**3**	**2**	**0**	**-3**	**-8**	**-16**	**-32**	**-64**	**-32**	**-16**	**-8**	A
B	**0**	**0**	**6**	**4**	**2**	**-4**	**-16**	**-32**	**-77**	○	**-64**	**-32**	**-16**	B
C	**0**	**1**	**15**	**12**	**9**	**4**	**-6**	**-19**	**-38**	**-64**	**-32**	**-16**	**-8**	C
D	**1**	**3**	**31**	**25**	**20**	**13**	**0**	**-7**	**-16**	**-32**	**-16**	**-8**	**-4**	D
E	**3**	**7**	**61**	**49**	**41**	**6**	**1**	**1**	**-6**	**-16**	**-8**	**-4**	**-2**	E
F	**6**	**3**	●	**110**	**18**	**12**	**19**	**7**	**1**	**-6**	**-4**	**-2**	**-1**	F
G	**-37**	○	●	**25**	**26**	**100**	**26**	**11**	**4**	**-1**	**-1**	**-1**	**0**	G
H	**-1**	**-14**	**13**	**17**	●	●	**12**	**3**	**0**	**-2**	**-1**	**0**	**0**	H
I	**-4**	**-14**	**20**	**16**	**2**	○	○	**-53**	**-27**	**-15**	**-7**	**-3**	**-2**	I
J	**-3**	**-9**	**7**	**2**	**-11**	**-76**	**-83**	**-36**	**-18**	**-9**	**-5**	**-3**	**-1**	J
K	**-1**	**-5**	**3**	**1**	**-6**	**-38**	**-42**	**-18**	**-9**	**-5**	**-3**	**-1**	**0**	K
L	**-1**	**-2**	**1**	**0**	**-3**	**-20**	**-21**	**-9**	**-5**	**-3**	**-1**	**0**	**0**	L
M	**-1**	**-1**	**1**	**0**	**-2**	**-10**	**-11**	**-5**	**-3**	**-1**	**0**	**0**	**0**	M
	1	2	3	4	5	6	7	8	9	0	1	2	3	

Figure 2 Influence map for the Context-sensitive simulation with the ladder-breaker stone in the far position (B,10). Modified connection and ladder points are shown enclosed in a solid line. The area that is affected by the influence of the ladder-breaker stone is to the right and above the dashed line. Notice that the four connection points are outside this area i.e., their influence values are unaffected by the influence propagated by the ladder-breaker.

The values of connection points (as shown in the enclosed region in Figure 1) in each of the simulations for the close, medium and far positions have been summarized in Table 1. The influence map for Context-sensitive simulation with the ladder-breaker stone in the far position is shown in Figure 6 with the modified points being enclosed in single lines and the area affected by the influence of the ladder-breaker stone above and to the right of the dashed line

DISCUSSION OF THE SIMULATIONS

The extent of the affect of influence from a ladder-breaker in the far position is shown by the dashed line in Figure 2. The influence values of the connection points in the initial position (Figure 1) are not affected by the influence of a ladder-breaker in the far position and thus, any decision regarding whether or not to try to connect the kogeima linked stones based on the corresponding influence map is likely to be misguided.

	Close (F,6)			**Medium (D,8)**			**Far (F,10)**		
Initial Position		**108**	**96**		**123**	**126**		**124**	**128**
	63	**78**		**67**	**86**		**67**	**86**	
Static		**29**	**29**		**29**	**29**		**29**	**29**
	29	**29**		**29**	**29**		**29**	**29**	
Additive		**48**	**36**		**63**	**66**		**64**	**68**
	3	**18**		**7**	**26**		**7**	**26**	
Multiplicative		**22**	**19**		**25**	**25**		**25**	**26**
	13	**16**		**13**	**17**		**13**	**17**	
Context-sensitive		**22**	**19**		**25**	**25**		**25**	**26**
	13	**16**		**13**	**17**		**13**	**17**	

Table 1 Influence values for the connection points for the initial position and the four simulations for the close, medium and far positions. (See Figure 1 to locate the connection points close (F,6), medium (D,8) and far (B,10) on the board).

In the Static simulation, only the connection points were statically modified to below the threshold of connectivity. This method of modification directly reflects the knowledge that the ladder-breaker can cut the connection. However, this method arbitrarily removes the information necessary to make discrimination between the relative influence values of the connection points.

In the Additive simulation, the addition of a fixed value of 60 only resulted in reducing the influence values of the connection points below the threshold of connectivity when the ladder-breaker was in the close position. To achieve the same result in the medium and far positions it would have been necessary to use a fixed value of around 90. However, using an additive value of 90 would have resulted in the two furthermost connection points being reduced to negative influence i.e. they would be considered to be under White influence rather than Black influence. In the close position, using of 90 would have resulted in all four connection points being modified to White influence values.

In the Multiplicative simulation, the problem associated with a simple additive modification were solved; all connection points in all positions were reduced to below the threshold of connectivity. However, the influence values of the ladder points close to the ladder-breaker were also reduced. Such a reduction of influence values close to the ladder-breaker stone would indicate that white influence was diminished close to the ladder-breaker when in fact it should remain the same or possibly increase.

In the Context-sensitive simulation, the selected influence values were modified in a context-sensitive manner. When the influence value was that of the ladder-breaker stone, the influence value was increased by 20%. When the influence value was that of the opponent, the influence value was reduced to 20% of its original value. This context-sensitive modification resulted in the connection points being reduced below the threshold of connectivity whilst raising the white influence of points close to the white ladder-breaker stone.

The values for the connection points for the Multiplicative and Context-sensitive simulations shown in Table 1 are identical. The reason for the similarity is that close to the kogeima link, the values were modified in the same manner i.e. multiplied by 0.2. Where the two simulations differ is in the influence values close to the ladder-breaker stone e.g., in the Multiplicative simulation, (B,9) was -13 and (C,9) was -6 (compare to Figure 2).

The results derived from Context-sensitive simulation are, in our opinion, the closest of those from the four simulations to the intuitions of human Go players. The influence around the kogeima link is affected far more in relative terms than the influence close to the ladder-breaker stone, however, the influence values close to the ladder-breaker stone are also somewhat affected.

General Discussion and Conclusions

In this paper we have provided an existence proof that it is possible to integrate cognitive knowledge into perceptual representations, and demonstrated how it could be achieved. We can now reflect on the wider consequences of the representations chosen, and the lessons learned for a general theory of the integration of perceptual and cognitive knowledge

The key questions for theoretical development concern the representations of perceptual and cognitive knowledge. The perceptual representation is the influence map, which is described as “numeric” as it consists of a matrix of numeric values which correspond to the influence of stones. The influence map is one way of viewing the board (a simple function of the raw positions of the stones). All the questions that might be asked of the system concerning connectivity could be answered from the influence map representation alone (albeit incorrectly in some circumstances). Knowledge of those circumstances in which the influence function will produce a misleading representation constitute higher-level knowledge, which we are calling “cognitive knowledge”. Hence, the cognitive knowledge that provides deeper insights into connectivity is additional to the basic perceptual knowledge. The cognitive knowledge necessary to understand the connectivity situations is used to create processes that operate over the perceptual representation.

It is an important point to consider how different the approach of integrating cognitive knowledge into a perceptual representation outlined above is from the traditional one of translating perceptual information into traditional symbolic rules, which can be easily encoded in a high-level language such as Lisp. The integration approach would facilitate encoding of the cognitive knowledge in a form easily understood by the designer (and the cognitive theorist) and would defer to a later stage of processing the consequences of the cognitive information. At first glance, this process may seem attractive -- it certainly facilitates the encoding of an expert’s cognitive knowledge (once it has been extracted from the expert). The problem of translating the implications of the presence of the ladder-breaker stones into modification of the influence map would disappear, replaced by the converse one of translating the information contained in the influence map into Lisp. From an encoding perspective, the integrations (perceptual-to-cognitive or cognitive-to-perceptual) seem symmetric (from the history of AI, at the very least we know that both are hard!).

From our studies, and those of Hofstadter and his colleagues, we know there is a critical difference. Although determining the semantics of cognitive knowledge for integration into perceptual representation is hard, it does seem tractable (at least in the domain of connectivity problems in Go). By contrast there is a major (and we believe insuperable) problem in determining the semantics of the perceptual information at an early stage, which we can explain as follows: whenever information is translated from one encoding to another (perceptual to cognitive, or cognitive to perceptual), the translation in and of itself makes some information easier to access, and some harder, or possibly inaccessible. Such loss is inevitable as all brain processes to some degree can be viewed as extracting salient features from their input data and making them available for further processing or action. The critical insight for this discussion is that the information that should be extracted can not be adequately determined prior to its actual use. This insight is the key motivator for the Copycat model and is well illustrated in Copycat's letter-string puzzles (Chalmers et al., 1990). That is, attempting to translate perceptual information into a context-independent cognitive form is putting the cart before the horse.

The simulations have served as an existence proof that cognitive knowledge can be integrated into a perceptual representation, and have also served to focus our attention on aspects of the tasks that might otherwise have passed unnoticed. Cognitive knowledge pertaining to ladder-breakers and their implications cannot be encoded in a bottom-up perceptual process due to the inherent combinatorial explosion that would result for large regions. Nor can such cognitive knowledge be efficiently learned by trial and error processes, as the situation concerning ladders is integral to even a beginner's assessment of the board, yet cannot be learnt by observation.

Acknowledgements

We thank the Department of Computer Science at the University of Queensland for the facilities to carry out this project and for providing a Summer Research Associate position to the first author and the University of Queensland for subsequently providing a University of Queensland Postgraduate Research Scholarship to the first author.

References

Chalmers, D. J., French, R. M., and Hofstadter, D. R. High-level perception, representation, and analogy: A critique of artificial intelligence methodology. Technical Report 49, Center for Research on Concepts and Cognition, Indiana University, Bloomington, Indiana, March 1990.

Chen, K. Group identification in computer Go. In D. N. L. Levy and D. Beal, (Eds.), *Heuristic Programming in Artificial Intelligence: the First Computer Olympiad*, 195 - 210. Ellis Horwood, Chichester, 1989.

Fotland, D. Knowledge representation in the many faces of Go. Anonymous ftp from bsdserver.ucsf.edu - Go/comp/article. Z, 1993.

Ryder, J. L. *Heuristic Analysis of Large Trees as generated in the Game of Go*. PhD thesis, Department of Computer Science, Standford University, August 1971.

Zobrist, A. L. A model of visual organisation for the game Go. In *Proceedings of the Spring Joint Computer Conference*, **34**, 103 - 112, 1969.