

MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
ARTIFICIAL INTELLIGENCE LABORATORY

AI Working Paper 286

April, 1986

## Cognitive Cliches

David Chapman

### Abstract

This paper is an exploration of a wide class of mental structures called *cognitive cliches* that support *intermediate methods* that are moderately general purpose, in that a few of them will probably be applicable to any given task; efficient; but not individually particularly powerful. These structures are useful in representation, learning, and reasoning of various sorts. Together they form a general theory of special cases.

A cognitive cliché is a pattern that is commonly found in representations and, when recognized, can be exploited by applying the intermediate methods attached to it. The flavor of the idea is perhaps best conveyed by some examples: TRANSITIVITY, CROSS PRODUCTS, SUCCESSIVE APPROXIMATION, CONTAINMENT, ENABLEMENT, PATHS, RESOURCES, and PROPAGATION are all cognitive cliches.

A.I. Laboratory Working Papers are produced for internal circulation, and may contain information that is, for example, too preliminary or too detailed for formal publication. It is not intended that they should be considered papers to which references can be made in the literature.

# 1 Introduction

We can evaluate an AI technique or system according to three criteria. First, generality: what percentage of interesting problems can be attacked using this method? Then, power: if the method is applied, will it solve the problem? Finally, efficiency: when implemented, does the technique run in finite time on problems of realistic size?

Existing AI techniques can be divided into two rough categories, general and special purpose. Then best general purpose systems pass the generality and power criteria. Consider theorem proving: almost any problem can be represented as a theorem to be proved, and if a complete theorem prover is run to completion it will always give the right answer. Another example is search: most AI problems can be reformulated as a search problem, and breadth-first search will always find a solution. Unfortunately, no one has been able to write a theorem prover or general search program that runs in realistic amounts of time on realistic problems; in fact it is impossible to write a theorem prover that is always correct and always halts. Theorem proving and search are extreme cases on all three counts, but general purpose AI techniques have never been used in systems that solve real world problems, because they do not pass the efficiency criterion.

Some special purpose methods have great power (can solve all or most problems in their domain of applicability) and may be efficient enough for real-world use, but these have always depended heavily on details of their domain: they do not generalize. While special purpose AI methods may be extremely useful, they are ultimately unsatisfying for two reasons: an engineering reason, that the cost of building each system based on special purpose methods is very high, so that a more general system would have greater potential; and a scientific reason, that people seem to satisfy all three criteria.

I believe that the tradeoff between these three criteria is inevitable; that there are no "big hammers," or general powerful efficient procedures, in the mind. I do not believe that anyone can ever write an efficient general theorem prover or search heuristic.

There are two ways out of this conundrum, both of which need to be explored. First, it may be that an apparently general purpose effective system in fact consists of a lot of special purpose systems that have been constructed by a learning mechanism. Thus, the system gives the appearance of being general purpose by incrementally extending itself to solve new problems. Second, it may be that one or more of the criteria can be relaxed without drastically affecting the usefulness of the resulting system. I propose to weaken somewhat the requirements of power and generality while maintaining efficiency and usefulness.

This paper is an exploration of a wide class of mental structures called *cognitive cliches* that support *intermediate methods* that are moderately general purpose, in that a few of them will probably be applicable to any given task; efficient; but not individually particularly powerful. These structures are useful in representation, learning, and reasoning of various sorts. Together they form a general theory of special cases.

A cognitive cliché is a pattern that is commonly found in representations and, when recognized, can be exploited by applying the intermediate methods attached to it. The

flavor of the idea is perhaps best conveyed by some examples: TRANSITIVITY, CROSS PRODUCTS, SUCCESSIVE APPROXIMATION, CONTAINMENT, ENABLEMENT, PATHS, RESOURCES, and PROPAGATION are all cognitive cliches.

Problem solving often proceeds by recognizing the form of the problem, and knowing what solution type will work for that problem type. Cliches are partial characterizations of problems which suggest parts of solutions. If many cliches can be recognized in a problem, many intermediate methods can cooperate in solving it. The overall paradigm for use of a cognitive cliché is that it is first recognized in a representation that instantiates it, and then competence attached to the cliché is applied to the representation. Attached competence might aid in planning, analogy, or prediction, for example.

Section 2 of this paper explicitly defines the notion of a cognitive cliché and gives some examples. Section 4 talks about building mental models from cliches, and section 3 explains how to use such a model.

This working paper is deficient in many respects. It is a very shallow overview of a lot of semi-digested ideas. I have nothing concrete to back up the claims I make. Perhaps more important, many of the more interesting ideas, particularly a lot of work on describing specific cliches and their properties, are not presented at all, and will have to wait for later papers.

This paper is a follow-on to [2], which describes a partial implementation of some of the ideas herein.

This paper has no serious related work section. The problem is that much of what I have read over the past six years has contributed to the theory, and even if I could remember all the influences, they could not be summarized in a paper of this length. I have been particularly influenced by the Programmer's Apprentice theory of programming cliches (from which I derive the word) [13]. Minsky and Papert describe "concept germs" which seem similar to cognitive cliches, but are only sketchily defined [11]. diSessa's theory of "phenomenological primitives" is also similar to that of cognitive cliches [5]. I have been much influenced by work on naive physics, particularly [8].

Another deficiency of this paper is that it does not discuss what I see as the outstanding problems with the theory. Some of these are the lack of connection to a theory of perception; the lack of an explanation of how cliches relate to domain-specific knowledge; and the lack of a theory of how cliches themselves arise developmentally or are acquired.

## 2 Cognitive Cliches

### 2.1 What is a cliché?

A cognitive cliché is a class of structures

- which are components of mental models
- which occur in many unrelated domains

- which can be recognized in several kinds of input data, and
- for which several sorts of reasoning can be performed efficiently.

For methodological reasons, a cliché should also be

- rigorously definable.

The cliché theory has both empirical and methodological content. The empirical claims are that

- there actually are many structures that fit the definition of cliché given above; and
- these structures with their associated intermediate competence are sufficient to explain much common-sense reasoning.

The methodological content is a *format constraint* in the sense of [1]: a requirement that theories of mind be of a certain sort. The format constraint is that

- theories of reasoning should consist of efficient techniques for computing properties of instances of specific cognitive clichés.

The power of the theory is that the *same* set of structures appears to be useful in many forms of reasoning in many domains. By the term “cognitive” clichés, I mean to emphasize their domain-independence. There are also clichés applicable only to specific domains, commonly occurring patterns that can be used to reason about programs or electronics or mechanics or whatever. Much of what is said in this paper is true of domain-specific clichés as well as general cognitive ones; I have chosen to concentrate on the latter because they offer a solution to the intermediate competence problem.

Clichés are tinker-toy pieces that can be assembled to form “runnable” mental models which perform reasoning tasks. In fact people typically use multiple, overlapping, incomplete models, in contrast to current AI technology, which tries to give programs unitary, modular, and complete theories.

Cognitive clichés are not intended as a theory of representation. They are patterns that can be found in any representation scheme: logic, frames, semantic networks, or whatever. Unconstrained representations in any language are very hard to compute with. If, for instance, all the terms in a semantic net or set of logical axioms are replaced with gensyms, it becomes very difficult for humans to understand or use the result. If such an unlabeled representation can be computed with, it is usually only by recognizing the intended meanings of the symbols, and then computing with those. Cognitive clichés are common patterns in representation that can be recognized and then used in reasoning. They can be thought of as a set of well-understood *models* for representations which have useful precomputed properties. Thus, computation can be done in effect semantically in the model, rather than syntactically in the theory.

## 2.2 Links between cliches

Cliches are connected by a variety of types of links into a complex web. These links are important in a variety of types of reasoning, and some their uses will be described in section 3. The link types are *specialization*, *extension*, *abstraction*, *isoonticity*, and *approximation*. Specialization and extension are defined conventionally in a standard logical framework. Abstraction and isoonticity are taken from McAllester's symmetric set theory [12]. Approximation is a concept new to this paper.

One cliche *specializes* another if every instance of the first is an instance of the second. For instance, TOTAL ORDER specializes PARTIAL ORDER.

One cliche *extends* another if each instance of the first would be an instance of the second, except that it has extra parts. (This can be defined rigorously in model theory by considering the signatures of first order structures. The intuition is clear.) Typically, extension is combined with specialization, because there is some constraint between the new and the old parts. For example, POOL extends and specializes RESOURCE: a pool is like a resource, except that it also involves a set (the *items*), the order relation of the resource is constrained to be the cardinality of the items, the producer to be a SET-INSERTION on the items, the consumer to be a SET-REMOVAL on the items, and the threshold to be that the items be NON-EMPTY.

*Isoonticity* is a relation definable (to my knowledge) only in McAllester's symmetric set theory. (Each *o* is pronounced separately; *iso*, "same" + *ontic*, being.) In McAllester's framework, two objects are isoontic if they are both "views of the same thing;" or if they "carry the same information." For details, see [12]. I use the word somewhat more broadly: two cliches are isoontic if for each instance of the one, there is an instance of the other to which it is isoontic in McAllester's sense. The intuition is perhaps best given by some examples. EQUIVALENCE RELATIONS and PARTITIONS are isoontic: for each equivalence relation, there is a partition that "carries the same information" and from which the equivalence relation can be recovered. For another example, TUPLES can be defined in (at least) two equivalent ways: as FUNCTIONS from INITIAL SEGMENTS OF THE INTEGERS, or as nested PAIR structures. These two views of tuples are isoontic. A CHOICE FUNCTION has as its domain a family of sets; it maps each set to a member of that set (the one "chosen"). The set of choice functions on a family is isoontic to the cross product of the family.

*Abstraction*, as I use the term, is also defined using symmetric set theory. An object abstracts another if it contains strictly less information about the same things. Again, I say that a cliche abstracts another if every instance of the first abstracts some instance of the second. For example, SET abstracts SEQUENCE. Every sequence has an underlying set; if you know that set, you know something but not everything about the sequence. If you know the sequence, you know nothing more if you are handed the set.

The notion of *approximation* can not be precisely defined; it is a purely heuristic notion. One cliche approximates another if every instance of the first can usefully be reformulated (viewed) as an instance of the second, but this reformulation is *not* correctness preserving. In general this relation is not symmetric; its converse is *com-*

*plication*. Large DISCRETE ORDERS and CONTINUA symmetrically approximate each other. Adding a DIMENSION to a space is often a way to complicate it. ORDERS often are reformulated as CYCLES. A PROPAGATION complicates an INCREMENT. A problem involving a CONTAINER can often be reformulated by approximating the container as a MUTABLE SET. And so on.

### 2.3 Examples of cognitive cliches

I estimate that there are more than two hundred and fewer than a thousand cognitive cliches; I've identified about a hundred. There are perhaps between three and three hundred things you know about each cliche. This section presents a small sampler of cliches to give the flavor of the full set. For some of them I give some interesting property they have.

I have given each cliche a name, spelled in small capitals: RESOURCE. Whenever possible the English meaning coincides with the rigorous one, but the names are mnemonic only. Since this is not analytic philosophy, I don't care what a resource "really" is. In cases where no common-sense word was available, I have often used a technical word: CROSS PRODUCT, ENERGY WELL. Some names I've had to make up: CATOOLYST.

The most basic cliches together form *naive mathematics*. Naive mathematics is an *irrational reconstruction* of parts of mathematics. It is that of which mathematics is the formal sublimation. Mathematical cognitive cliches are the central, core ones from which many of the others are built. Naive mathematics concentrates on finite set theory, combinatorics, and algebra. People, like computers, are good at manipulating finite, discrete structures.

Naive mathematical cliches are all quite familiar: PARTIAL and TOTAL ORDERS, EQUIVALENCE RELATIONS, MULTISSETS, SEQUENCES, ONE-TO-ONE and MONOTONIC FUNCTIONS. COMPOSITION, ASSOCIATIVITY, HOMOMORPHISM, WEIGHTED AVERAGE and INTERVALS are naive mathematical cliches. Most AI programs in fact incorporate some of these cliches in their operations; some even represent the cliches explicitly. But no one has systematically built a toolkit of these simple mathematical ideas that can be assembled into runnable mental models.

Some very abstract cliches are not familiar as part of mathematics, often because they deal with change. For example, REVERSIBILITY of actions is a very useful property to notice. Some combinations of cliches are themselves cliches, because they support intermediate methods not obviously derivable from the parts. For instance, we can precisely define a RESOURCE as consisting of a state variable whose value must be a member of a TOTAL ORDER, a *consumer* action, which DECREASES the variable relative to the order, and a *dependent* action, which is ENABLED only when the value of the variable is greater than some THRESHOLD. Reasoning about resources is extremely important in planning, prediction, and diagnosis. A POOL is a resource in which the order is the CARDINALITY of some SET, and the consumer REMOVES elements from the set.

TRADE-OFFS may give rise during planning to a CONTINUUM of possible solutions

to a problem. In a trade-off there are two quantities which you can control, but not independently; they are constrained to be related by a MONOTONIC DECREASING FUNCTION.

Many cognitive cliches are spatial. Minsky, in unpublished work, has proposed a cliche theory of shape. There are a few dozen shape cliches like BLOBS, ELBOWS, T-JUNCTIONS, and WEDGES. Other shapes are represented by combining and modifying cliched shapes. There's a lot you know about each shape cliche: for example, two L-SHAPES are best fit together by turning one upside-down and making the two elbows FACE each other CATTY-CORNERS. Knowledge like this might be used in a VLSI layout system. In robot navigation, to pass an OBSTACLE, you must go UNDER, OVER, AROUND, or THROUGH it; there are no other alternatives.

The spatial cliches IN and BETWEEN approximate each other. If the objects which something is between can be seen as comprising a single object, it can be considered to be in that object. Conversely, if something is in something else, it can be thought of as between the sides or walls of it. Between is both a mathematical concept ( $y$  is between  $x$  and  $z$  if  $x < y < z$ ) and a spatial concept (for which the definition is less clear). This provides a portal between the mathematical and the spatial domains.

Consider a space  $S$  with a function  $f$  on it; let  $v$  be in the range of  $f$ . A PROPAGATION is a process which changes  $f$  to be  $v$  on some elements of  $S$  adjacent to those that already have an  $f$ -value of  $v$ . Propagation suggests a limit analysis in which all CONNECTED COMPONENTS of the space that have any  $v$ -valued points will be entirely  $v$ -valued. There are planning techniques for controlling propagation: consider a microbiologist who must infect a culture selectively with one micro-organism and not others. He must assume that any object that comes into contact with another that may be infected, will be infected. Such techniques are also needed in diagnosis, in which causes must be localized.

The cliche CONTAINER is important in many common-sense domains; and is often used non-physically. One thing you know about containers is that you can't displace the contained arbitrarily without removing it from the container. A container, if displaced, may displace the contents with it. This suggests a plan: to move something, put it in a container, then move the container.

The idea of a fluid FLOW is used analogically in an enormous number of domains in which there is no literal liquid. ATTACHMENT, CREATING, ENERGY WELLS, OSCILLATORS, REGENERATION, OPPOSITION, and RESISTANCE are other physics-like cliches. Each of these can be precisely defined and used abstractly in non-physical domains.

I'll give now a single incomplete example of a mental model assembled out of cognitive cliches. A zipper is a SEQUENCE of PAIRS of teeth. There is a ONE-TO-ONE-CORRESPONDENCE between teeth on one side and teeth on the other. Each tooth is BETWEEN two others. Teeth from each side ALTERNATE with the other. Each tooth is ADJACENT to two and OPPOSITE one other. Zipping is a MONOTONIC PROPAGATION along the TOTAL ORDER that is the isoontic view of the sequence. The WAVEFRONT of the propagation is at the handle. At the end of the zipper there may be a tab that

acts as an OBSTACLE to propagation. The zipper ATTACHES the two sides. If you approximate the DISCRETE zipper order by a CONTINUUM, you would get a ziplock bag. The zipper typically serves to OPEN and CLOSE a CONTAINER, and so acts as a sort of GATE. And so on.

### 3 Exploiting Cliches

In this section I sketch techniques for using a mental model built out of cliché instances. Presumably such a model is derived from cliché analysis (see section 4). There are very many ways of using clichés; in every type of reasoning I've looked at, at least some clichés seem to have attached reasoning methods.

The most basic form of reasoning I call "calculation." Calculation is the "running" of a mental model. Different clichés structures have different things they can calculate. You can apply a function to an argument; construct the partition that is the isoontic view of a specific equivalence relation; enumerate the elements of a set; count the cardinality of a multiset; remove an element from a pool; find the reverse of an action or the optimal value of a trade-off; optimally pack a set of clichéd shape instances; or find the amount of a flow against a known resistance. Each of these things is very simple, and computationally inexpensive as compared with general inference (theorem proving). Every AI system does some of these things; a comprehensive package of routines to perform such calculations efficiently and uniformly would be very useful. In fact, calculation is used in all the other types of cliché-based reasoning I've looked at. Simulation and prediction, for example, are just calculation turned sideways in time.

Analogy is taken to be a fundamental mental operation, and considerable research in AI and cognitive science has been devoted to it. Cognitive clichés suggest some ways of approaching analogy that are either new or have received little previous attention in AI.

There are clichéd analogies. This is the point of [9], and the same point is made by [11]. Approximation links are standard reformulations (analogies) between cognitive clichés, and can be extended to reformulations of whole mental models. [10] uses techniques like this in the automatic algorithm design.

Cliches can help decrease the computational complexity of analogy matching. Naively, an analogy system might match every representation against every other representation; but this is generally too expensive. An alternative is to index representations by their most important features and only match representations that are similar. [7] suggests higher-order relations are the important features to match; clichés, as particularly interesting higher order relations, might be the best choice. The specialization and extension lattice of clichés can be extended to a lattice of mental models built from them. Then, to find good analogies for a representation, you can climb up the lattice a ways, and then drop down again to find something similar. (This idea is not unique to me; it occurred to several people in our lab independently some years ago.)

Cliches can be used in planning in a number of ways. In [2] I describe an implemented



system in which “canned” plans are attached to cliches. In [3] I discuss an approach to solving the frame problem using cognitive cliches. A planner has to maintain a model of the world and its changes over time; such a model can be built out of cliches, and cliche computation can be used to discover things in the model.

Taxonomy is the problem of finding useful PARTITIONS. Useful partitions often have structure to them; that is, the EQUIVALENCE CLASSES themselves participate in cognitive cliches. Let us *raise* any representation by replacing elements with equivalence classes. Then HIERARCHICAL CLASSIFICATIONS are the raises of TREES; MULTI-FEATURE CLASSIFICATIONS, the raises of CROSSPRODUCTS; dichotomies the raises of PAIRS.

## 4 Cliche Analysis

One of the two fundamental components of the cliche paradigm is cliche analysis, the subject of this section. Cliche analysis is the process of building a mental model by examining a representation, recognizing instance of cliches, and making connections between them. Cliche analysis can be viewed as a type of learning: when you have built a mental model, you know something new that will often be useful in achieving your goals.

Cliche analysis can be thought of as proceeding in two phases, though in fact they must be intermixed. The first phase is recognition, which is the process of examining the input representation for the patterns which allow one to instantiate cliches that index (model) the representation. An example is checking all presented pairs of a dyadic relation to see if it is symmetric. The second phase is cliche elaboration, which extends a mental model by adding cliche instances which model the input simply by virtue of other cliche instances modeling the input. For example, if it is known that a relation is a finite total order, one can elaborate the model by adding a finite sequence that isoontically models the same relation. In fact, some patterns of cliche connections may only suggest, rather than imply, elaborations: in this case, elaboration will call recognition as a subroutine to verify that the suggested cliche will model the input.

Recognition has generally been thought of as a matter of perception, and in fact some forms of cliche recognition do have a perceptual taste to them. Whether it is better to think of recognition as learning or perception seems to depend on the input to the recognizer: if it is quantified, abstract knowledge structures (like a text-book), recognition is like learning; if the input is sense-data or the output of other perceptual systems, recognition is like perception. I'll discuss both forms of recognition in this section.

Cliche elaboration is a function from models to models, and so does not depend on the input representation formalism, while recognition typically depends heavily on the representation. It is useful to distinguish the two for this reason: only one elaborator need be written, while a new recognition module might be needed for every representation the mind is expected to deal with.

The system I reported on in [2] has one cliché recognition module and no elaboration. The one form of input it can recognize is presented tuples of relations. This is the form of input that most classical AI learning systems have been given. My system was able to learn to stack blocks by experimentally taking actions in a blocks world, and being told the effects in the form of tuples of the *on* relation. Each tuple consists of a relation symbol and some arguments. Using simple induction coupled with directed experimentation, the system was able to recognize such clichés as equivalence and order relations, increment functions, compositions, stacks and queues, and inverse functions.

One of the key techniques used was *temporal abstraction*, which allows one to view an action (function from states of the world to states of the world) as a mathematical function (from objects to objects). Temporal abstraction depends on the *functionality* of operators: that is, applied to the same inputs, they will always have the same effect. In the real world, this is not always true. There are three sources of afunctionality for actions: dependency of effect on state variables (as well as arguments); noise; and changes in the world due to processes other than the system itself. My pilot system was unable to deal with any of these effects; a more powerful recognizer ought to.

I have done a number of experiments that suggest that cliché recognition is both possible for and useful in coping with dependence on hidden state. Dietterich [4] has presented an elegant technique for understanding hidden state variables; his theory has at one point a kludge, that it requires preconceptions of what “types” of actions can exist. These types are exactly cognitive clichés. Another approach might be to use standard finite state machine induction techniques interleaved with recognizing patterns in the state transitions of the induced machines. This could perhaps overcome the efficiency problem due to the NP-hardness of the induction task.

A more interesting form of input is quantified abstract propositional knowledge structure, such as semantic networks or logical formulae. These might be the output of a language module given text or spoken input; or the output of cliché recognition from data; or the output of other learning mechanisms, statistical induction for example. Such knowledge is amorphous and ought to be structured, or formalized, or made explicit. This can be done by fitting clichés to the knowledge. The techniques for recognizing clichés in such knowledge are quite different from those for recognizing clichés in data. No induction is involved; parsing or matching is the image. I don’t have any good ideas about how to do this, but it will probably be important.

Dyer’s BORIS system [6] builds mental models from components much like cognitive clichés, starting with English text input. His approach is to attach to English words the possible structures that they could represent, and to use later constraints to disambiguate in cases of multiple possibilities. I think that a similar approach will work for clichés in general, attaching clichés not to words but to specific predicates. Thus cliché recognition can start where most natural language systems leave off, with a semantic network.

I have done cliché recognition and elaboration by hand starting from high-school level accounts of genetics. Beginning with a text-book, I translated the account into a

quantified semantic network formalism, and then derived the cliché analysis of the text: the output of a recognizer. Comparison of these two representations suggest that the approach of the last paragraph is plausible, and that there is sufficient constraint that consistency checking can disambiguate alternative analyses.

Quantified knowledge can come from other sources than language. For example, many statistical learning algorithms produce such knowledge as output. The difficulty with such algorithms is that it is often very hard to know what to do with the mass of assertions they produce. If clichés can be recognized in them, though, all the cliché-associated competence will be applicable.

## Acknowledgments

Portions of this research have been supported by Thinking Machines, the SRI AI Center, Schlumberger Palo Alto Research, and the MIT Artificial Intelligence Lab. I'd like to thank Danny Hillis, Mike Georgeff, Pat Hayes, and Chuck Rich for making my work possible.

As this paper represents a core-dump of much of my thought over the last six years, innumerable people have contributed ideas in conversation, net mail, and comments on paper drafts. Phil Agre, Margaret Fleck, Chuck Rich, and Dan Weld read the most recent draft and have very usefully applied continued pressure to finish the damn thing.

## References

- [1] Agre, Philip E., "The Methodological Armamentarium of Faculty Psychology." Unpublished manuscript, 1984.
- [2] Chapman, David, "Naive Problem Solving and Naive Mathematics." MIT AI Working Paper 249, June, 1983.
- [3] Chapman, David, *Planning for Conjunctive Goals*. MIT AI TR 802, forthcoming.
- [4] Dietterich, Thomas G., "Learning About Systems That Contain State Variables." *AAAI-84*, pp. 96-100.
- [5] diSessa, Andrea A., "Intuition as Knowledge." Unpublished manuscript, July 5, 1983.
- [6] Dyer, Michael G., *In-Depth Understanding*. MIT Press, Cambridge Mass.: 1983.
- [7] Gentner, Dedre, "Structure-Mapping: A Theoretical Framework for analogy." Undated unpublished manuscript.
- [8] Hayes, Patrick J., *The Second Naive Physics Manifesto*. University of Rochester Cognitive Science Technical Report 10, October, 1983.
- [9] Lakoff, George, and Mark Johnson, *Metaphors We Live By*. University of Chicago Press, Chicago and London, 1980.
- [10] Lowry, Michael, "Reformulation of Algorithms in Computational Geometry." Unpublished manuscript.
- [11] Minsky, Marvin, *The Society of Mind*. Unpublished manuscript.
- [12] McAllester, David Allen, "Symmetric Set Theory: A General Theory of Isomorphism, Abstraction, and Representation." MIT AI Memo 710, August, 1983.
- [13] Rich, Charles, *Inspection Methods in Programming*. MIT AI Technical Report 604, June, 1981.