DECIDER-1: A SYSTEM THAT CHOOSES AMONG DIFFERENT TYPES OF ACTS'

Leonard Uhr
University of Wisconsin
Madison, Wisconsin

Abstract

This paper examines a programmed model (called DECIDER-1) that 1) recognizes scenes of things, among which are a) objects and b) words that form commands (or questions or other types of statements), 2) recognizes the import of these commands, 3) decides whether to obey one, and then 4) uses the command to guide the consequent actions, along with any necessary perceptual search.

It uses the same mechanisms to handle a) the perceptual processes involved with recognizing objects and describing scenes, b) the linguistic processes involved with parsing sentences and understanding their meaning, and c) the retrieval processes needed to access pertinent facts in pertinent memory. This is in sharp contrast to most of today's systems, which receive the *command* through one channel, to be "understood" by one special-purpose set of routines, and perceive their environments through *an* entirely different channel.

DECIDER-1 continues to characterize patterns, parse symbol strings, and access facts implied by input questions until an action is chosen, because it is sufficiently implied by this search through the memory net . Then it executes the implied action. Possible actions include Answering, Describing, Finding, Moving, and Naming.

Introduction

The basic purpose of this paper is to examine how word-things recognized in the external environment can help trigger actions that include recognition and manipulation of other, object-things, in the same environment. To do this we must examine several issues that have scarcely been touched upon in the research literature of psychology or computer science:

How does a system recognize that a perceived input is a symbol? How does it understand the import of a structure of symbols, e.g. that it is a *command,* or a question? How does it understand the meaning of that structure ? How does it decide what type of thing to do - whether to obey a command, and which command, or whether to continue to perceive, to respond to internal needs, or external presses from perceived objects ? How do the understanding of commands and other symbolic percepts and the recognition of perceived objects interact, helping one another; e.g. how does it recognize which objects are appropriate for carrying out a command? How does a system choose and execute the appropriate response, from variety of possible responses ?

This research has been partially supported by grants from the National Institute of Mental Health (MH-12266), the National Science Foundation (GJ-36312), NASA (NGR-50-002-160) and the University of Wisconsin Graduate School.

These are extremely complex and subtle matters, and the work presented here is only a first step. But there appear to be relatively simple yet powerful ways In which we can begin to handle them.

Description of the Problem, Background and Motivation

We will examine systems that a) input streams of information from the environment, b) attempt to recognize things, including symbols, and larger structures of these things, c) decide whether to continue in this perceptual process or to respond, and d) generate the appropriate response.

Let's look at recent research with chimpanzees and with robots for some examples.

Chimps that Obey Commands

Two chimps have been taught (by Gardner and Gardner, 1969[6],1971[5], and by Premack, I970[10], 197I[1J](to learn vocabularies of over 100 words, and to learn to use these words in simple sentences. One of the most interesting things Sarah (Premack's chimp) learned was to respond to a sentence like "SARAH PUT BANANA BOX" by going to the banana , grasping it, carrying it over to the box, and dropping it into the box. SARAH is in an experimental room with a number of objects, including a banana and a box, but also such things as a pail, an apple, a cracker, and other objects, and a board on which the experimenter places the words SARAH, PUT, BANANA, and BOX. (These "words" are colored nonsense shapes: all previous attempts to get chimps to talk failed because of their limited vocal apparatus.) Thus words, sentences and objects are perceived visually/ with words static objects much like Chinese ideograms.

Words are things that have symbolic significance. They come in through the same input channel as objects like the (image of the) banana. The chimp responds to the command, which is a structure of several word-things, by manipulating the objects to which the words refer: Sarah grasps the #BANANA, not the symbol BANANA. (I will use stars (*), as in *BANANA, *BOX, *APPLE, to indicate the actual object, as opposed to the word. Note that the star is not a built-in symbol for the computer program, but simply helps the reader distinguish between the word and the simplified representation of the object.)

Today's Pr6grams Cannot Model such Chimps

The above description may appear to belabor the obvious. But although any three year old human child, and now two chimps, can successfully sort out words and objects, recognize structures of words as commands, understand the import of the commands, and act appropriately, we have little idea how all this is done. Psychologists have given us no theoretical models, and even the most sophisticated of computer programs for pattern recognition or language manipulation are designed to handle only small pieces of this process,

A sufficiently powerful pattern recognizer (e.g. UhrandVossier, l96l[17]; Andrewset. aL, 1968[1]; Munson, 1968[8]; zobrist, 1971[24]) can correctly name many different examples of a pattern, all distorted in extremely complex, and unanticipated, non-linear ways. It can contend with very complex possible confusions between objects. In comparison, a *BANANA and an *APPLE are very different, and the symbols Premack has chosen for BANANA and for APPLE are also different enough so that the chimp has no perceptual difficulties. (This is not at all to say that the chimp and human are not sophisticated pattern recognizers, but simply to point out that the particular single-pattern problem we are examining, in which only a few very different patterns need be recognized, is simple compared to problems that programs, as well as chimps and people, can handle.) But today's pattern recognizers will not handle scenes of several things, much less compose things Into structures like sentences, or sort out the words and the objects, and their interrelationships.

A sufficiently powerful language processor (e.g. Winograd, 1971[23]; see Simmons, 1965[14],1970[15]) can handle more complex grammatical structures than the simple actor-act-object-indirect object of our example SARAH PUT BANANA BOX. But it will handle only clean, clear-cut sentences. It cannot content with sentences with misspelled words and noise that are embedded in larger scenes that include other objects.

At the very least, pattern recognizers must be extended to handle scenes of objects, and to parse word-phrases as well as characterize parts of objects (see Sauvain and Uhr, 1969[13]; Jordan, 1971[7];Uhr, 1972[18] 19?3c[21]).

Robots that are Pre-programmed to Obey Commands

Recent work with robots, in which they deduce paths to cluster boxes, and stack blocks (e.g. Nilsson, 1969 Raphael, 1968[12]; Feldman et. al., 1969[3]), has made some steps toward this interaction. But it has the flavor of interconnecting several separate black boxes, each performing a separate function, even though there often appears to be a great amount of overlap. Thus the reported robots input and handle commands and perceptual environment completely separately.

Today's robots use extremely complex systems of computer programs, almost always separated into four major sub-systems, to 1) understand the command, 2) recognize and build up a model of the objects in the environment, 3) deduce the set of actions appropriate to carrying out the command, and *4)* apply them to the environment. Each subsystem has its own subsystems:

1} The *command* is given through a teletype, and the system applies language "understanding" programs, to recognize words, parse the command statement, and derive from the statement a goal state that the robot must achieve, to obey the command.

2) Separate perceptual programs are used to handle the robot's perceived spatial environment, which almost *always* includes inputs from a tv *camera,* and occasionally is supplemented with inputs from range finders, touch sensors, photocells, or sonar. These programs input the sensed information, take a sequence of preprocessing steps (e.g. to eliminate noise, find gradients and edges, and begin to connect short edge fragments into lines), and try to build the salient features up into something that matches some internal description of some object (e.g. Brice and Fennema , 1970[2]). This results (if all goes well - which today *means* if objects are sufficiently simple, with enough background space between them, painted in colors that contrast sharply enough in the black-and-white tv image, and amply lighted) in the assignment of names to objects, and the assignment of these objects to their locations within the perceived environment.

3) A deductive problem-solving program (often a theorem prover) Is used to generate a sequence of actions that the robot might apply to the objects it has tentatively recognized in its perceived space, in order to achieve the desired goal state (e.g. Fikes and Nilsson, 1971[4]).

4) This entails binding the objects (including the robot itself) to the proposed actions, *so* that the robot can in the real world try out an action that it has deduced would make progress toward satisfying the overall command.

Perceiving the Import of Inputs

Real-World Commands Must first be Perceived

In the real world there is no separate input channel for a command, and there is no god-given a priori signal, known *to commander* and slave alike, that says, "this is a command," "this is a question," "name this object," "describe this scene," and so on. A question-answering program has built into its guts that inputs will be questions, and it is straightforwardly pre-programmed to answer them. A robot program similarly has built Into it that teletype inputs are commands, and that it is to carry them out, by manipulating the environment that it perceives through its sensory inputs.

But in the real world the command always comes in through sensory input channels. In fact the command is itself a complex structure of perceived objects. E.g. the command, "PUT THE BANANA IN THE BOX", is made up of a string of words that are further structured grammatically and, more importantly, semantically (in that they refer to things like bananas, relations between bananas and boxes, an understanding that bananas can go into certain boxes, and the implied action *of* the entity being commanded - that it. should put the banana in the box). And each word is made up of parts {letters if written, phonemes if spoken), each letter or phoneme is made up of parts, and so on.

The command may sometimes come through a separate perceptual mode channel, as when it is spoken and refers to visually perceived objects. But this Is not the issue, for commands and objects often come in through the same channel, with the receiver hardly noticing. It is not the difference in channel that allows the human to infer "this is a command composed of symbols" and "that is *an* environment of objects". Rather, the hearer first recognizes the parts of the command as objects and only later as symbols, combines them up into larger and larger structures, and recognizes that these structures have symbolic Import, and are commands.

Deciding Whether to Obey Which Command

Closely related, the hearer has no built-in understanding that it will receive one command at a time, pointed to and surrounded by special symbols. Rather, since it must infer that parts of its perceived environment are commands, it always has the possibility of receiving more than one command. It must decide whether to obey a command, and which. And for any real-world receiver there will always be the issue of whether it should drop everything to carry out whatever command it has Just received and understood, or whether it should do something else - what it was already doing, or what will best satisfy some internal need (e.g. hunger) that is arising, or what will best respond to some interesting new object (e.g. a steak).

Thus a system should be able to decide which from among a set of alternative action-sequences it wants to carry out - whether to gather more information, or to obey this command or that, or to satisfy its own needs or goals.

Key Problems Being Handled

This paper focuses on the problem of handling fields of things, some of which are symbols, where these things combine into larger structures (e.g. eyes, nose, mouth, chin combine into face; SARAH PUT BANANA BOX combines into a command; B,0, X combine into BOX), some of which imply that the system should respond with an implied action upon other things to which reference is made.

We have discussed one example of such a situation, when the environment contains something like; ♦CRACKER SARAH PUT APPLE PAIL *BOX *BANANA *APPLE *PAIL (or) *BOX3 ROBOT PUSH BOX2 NEXT BOX1 *BOX4 *BOX1 *BOX5 *BOX2

Such a system can handle a number of other problems, for example an input like: ♦CRACKER ♦BOX DESCRIBE *APPLE THIS SCENE *BOX ♦BANANA *PAIL (In which case it must recognize the command DESCRIBE THIS SCENE and as a result output CRACKER BOX APPLE BOX BANANA PAIL).

Or it can be given an input like: ♦CRACKER ♦BOX FIND A BOX ♦APPLE ◦BANANA ♦PAIL (in which case It must recognize the command FIND BOX and as a result output something to Indicate the ♦BOX has been found.) Or it can be asked to Move an object, or to Answer a query.

A Brief Description of Decider's Behavior[1]

Overall Flow

DECIDER-1 inputs a SCENE that contains all physical objects and verbal utterances mixed together, *as* the above argument has shown is inevitably the case for real world intelligences. It applies two types of pattern recognition techniques to begin sensing this scene:

1) A set of primitive perceptual transforms is put onto the IDEAS list, thus initializing IDEAS to look for whatever the primitives suggest are the basics.

2) Delimiters (such as the edges, gradients and background spaces that often delineate objects and verbal utterances) are used to decompose the SCENE, giving a first tentative set of possible objects. Then DECIDER-1 searches Its memory for anything that these objects might imply, and, if it finds any, merges them onto the IDEAS list.

DECIDER-! applies the set of IDEAS, serially, to the input SCENE. Each idea that succeeds implies a variety of different objects, new ideas to apply, and acts. The single most highly implied idea is applied in turn to further characterize the SCENE - until it is a response act like Answer, Describe, Find, Move, or Name, *in* which case it is carried out, and the next SCENE is input. (Note that the particular primitives, perceptual characterizers, verbal rewrite rules, and action transforms depend upon what the programmer (or learning) has tabled into DECIDER-1, just *as* parsers or table-driven compilers depend upon the grammars given them.)

Perceiving Sensed Objects and Understanding Verbal Utterances

DECIDER-1 acts much like a typical pattern recognizer vis-a-vis sensory patterns to recognize and name, and much like a typical parser vis-a-vis verbal utterances. So it can handle mixed inputs of objects and words. It handles both the recognition characterizers and the parsing rewrite rules (let's call them both "transforms") in the same way: It gets the most highly implied transform from the IDEAS list, applies it to the input SCENE (which includes all previously-effected transforms), evaluates whether this transform succeeds and, if it *does,* merges the transforms that it implies onto the IDEAS list, and the objects that it implies onto the SCENE. (A transform looks for a configurational n-tuple, of which parsing rewrite rules are simple examples (see Uhr, 1971[16], 1973d[22]).)

The IDEAS list starts with primitive perceptual characterizers. As these transforms succeed, they will imply higher-level characterizers and, If DECIDER-1 recognizes verbal utterances, verbal rewrite rules. These will continue to imply perceptual

See Uhr, 1973d for more detailed descriptions, and the actual DECIDER-1 program.

Caps refer to major constructs in the DECIDER-1 program.

and verbal transforms, a) for still higher levels, b) to glance about at other parts of the scene in which what has so far been recognized suggests other things should be looked for, and c) to gather more information to confirm or deny the presence of tentatively implied objects (including words).

Deciding to Act

DECIDER-1'g decision to act is deceptively simple. Each transform on IDEAS has a type associated with it. As discussed up to now, at first the most highly implied transforms will be of type = P (to perceptually parse) - until the system has gathered enough information about the scene to imply some response action with a high enough weight to be chosen.

DECIDER-1 chooses the single most highly implied transform from IDEAS. At first these will be perceptual and verbal transforms. But they will begin to imply various possible acts, which will be merged back into IDEAS- Thus IDEAS serves as a vehicle for deciding among the various possible acts, at the same time that it serves to decide whether to continue looking and gathering information, or whether to respond.

When a response is chosen, as the most highly weighted transform on IDEAS, DECIDER-1 branches to effect that response - whether to Answer, Describe, Find, Move, or Name. Thus the response acts and the Information-gathering acts are all ordered on the single IDEAS list, from which DECIDER-1 continues to choose and execute the single most highly implied act, until a response act is chosen. This serves to order the execution of each type of act, and serves to choose when to decide to respond, as well as which response to make.

Types of Response Acts

DECIDER-1 can effect g variety of acts. Since our purpose has been to examine how a system can decide to act, the acts themselves are kept as simple as possible.

Answer outputs the piece of information (e.g., a fact, or a document name) that is stored in the Answer transform that was chosen from the IDEAS list.

Describe outputs the names of all objects that have been recognized and placed on the SCENE with a sufficiently high weight to exceed a CHOOSE parameter.

Find searches for and brackets *(in order* to indicate where it is) an object of the class specified by the particular Find transform chosen.

Move finds an object of the first class specified, and moves it from its original location in the SCENE, so that it Is next to an object of the second class specified.

Name gets and outputs the single most highly implied thing that is the name of an object in the SCENE.

The Flow of Processes is Multi-Determined

Any transform that succeeds can imply any potential act of any type. Characterizing transforms can be implied by what has been found so far about the input. A recognized and understood command can imply the act that obeys it. But an object can also imply an act (e.g. to Name, or Find it). And internal needs can imply acts (e.g. hunger can imply Find an object of the class = food).

Thus, e.g., when DECIDER-1 chooses to Name this can be because a) a number of nameable objects have been perceived, and each implies the act of naming with a low weight, b) a command like "NAME THIS" or a query like "WHAT IS THIS" has been recognized, as implying the act of Naming, and/or c) an internal need implies that the act of naming may lead to its satisfaction. I should emphasize again, because It Is a subtle point, that an act like Naming is not built in, but must be decided upon.

Some Examples *of* DECIDER-1 's Behavior[1]

Recognizing, Parsing, "Understanding" and Naming

DECIDER-1 is given input SCENEs that contain mixtures of words and objects. (These should be two-, or even three-dimensional scenes that extend over time. Then a written or spoken word like "APPLE" would extend in two dimensions Just as does the perceived apple. We reduce these to 1-dimensional strings, to keep DECIDER-1's characterizing processes from getting cumbersome. But see Uhr, 1973b, 1973c for the relatively straightforward extensions that allow a system to handle two-dimensional scenes that extend over a third time dimension.)

When given an input like:
NAME THIS ((along with the representation of an object))

DECIDER-1 will (If Ithas been given, or has learned, the needed transforms) output that object's name, e.g. "APPLE" or "TABLE". To do this it applies whatever perceptual transforms have been given it as primitives, and continues to apply linguistic rewrite transforms until the Name transform is triggered. By that time enough perceptual transforms have already been applied ♦o imply object names, and DECIDER-1 chooses and outputs the name of the most highly implied thing that belongs to the class of "OBJECT"s.

The objects might be represented at any level desired - from a linearized 2-dimenaional matrix of light and dark, e.g.:
000111000,001000100,010000010,001000100,
000111000,000010000, ((9 x 6 apple))
to a set of characteristics:
STEM APPLE-COLOR RED ROUND ((apple characteristics))
to a template-like representation:
OBJECT87 ((an encoded name for "apple" objects))

[1] See Uhr, 1973d for a more detailed development of a wider variety of examples, showing the transforms used and the ways these process Inputs.

Note that words can similarly be given at any of these levels.

Edges {e.g. 01, or 0011) of objects and spaces afterwords, characteristics, or templates, are used to delimit things. And the primitives that are initially put on the IDEAS list (e.g. the *letters of* the alphabet, and simple edges, angles, or other pattern recognition characterizers that are commonly useful) begin to recognize parts of the input. Thus the letters, and then words like "NAME" and "THIS", and the edges of objects, and then the objects themselves, will all be implied.

These In turn imply still other transforms that build compounds like "NAME THIS" or larger objects. These in turn Imply response acts. This means that characterizers will imply an apple *(and* probably some other objects, e.g. a pear and a face), and the implied *objects may themselves imply the act of* naming - *at* the same time that the verbal utterance NAME THIS also implies the act of naming, with *a* very high weight, so that it is very likely to be chosen.

Describing

An input of the sort;
DESCRIBE THIS {(representation of one or more objects))
will get DECIDER-*1* to output a simple description, of the names of all things belonging to the "OBJECT" class the objects implied above a CHOOSE level, e.g. "APPLE TABLE CHAIR BANANA". The recognition of objects is much as for Naming, but additional linguistic transforms will recognize the word "DESCRIBE" and the import of the phrase "DESCRIBE THIS" - as implying the Describing act.

Any number of variants *can be* handled, e.g.:
WHAT IS HERE {(objects))
SAY WHAT YOU SEE ((objects))
so long as the needed rewrite rules are In DECIDER-1's memory.

Finding Objects

If the verbal utterances within the scene {and/or internal needs) are recognized as Implying the act of Finding, DECIDER-1 will branch to the Find routine. This searches for an implied object that has been put into the transformed SCENE that is of the class designated by the particular Find transform being executed. If the object is found, success is indicated by the placement of brackets, as though pincers, around it.

Acts are Multi-Determined,, by Words, Things and Internal Needs

The verbal utterance can be a command, e.g.:
FIND FOOD ((a scene of objects)) ((or))
WHERE IS FOOD ((a scene of objects))
or a request:
IS ANY FOOD AROUND ((a scene of objects))
or any other kind of utterance that, when parsed and understood, implies that a food object be found.

But remember that an act like Finding can be implied not only by verbal utterances, but also by objects in the scene, and by Internal needs. Thus, e.g., the recognition of a valued object, HJce a banana, or the partial recognition of several food objects, like apples, bananas, and crackers, can serve to imply, and, possibly, lead to the choice of, the act of finding (that is, getting, or grasping) the banana, or the food object. Similarly, an internal need-state of hunger can imply the getting of a food object that will satisfy that hunger.

Moving and Manipulating Objects

An input like:
MOVE BOX1 TO BOX2 ({a scene of objects))
will lead to the recognition of the particular objects to be manipulated *and* the overall action desired. When Move Is chosen as response act, because it has become the most highly implied transform on *IDEAS,* DECIDER-1 will look for the specific objects or class members involved in the action, and actually change the SCENE, as specified.

DECIDER-1 assumes that the action will not be triggered until all the needed objects have been recognized. So the perceptual characterizers must be implied with high weights, so they are merged onto IDEAS with higher weights than are the response actions. But it would be quite easy to extend DECIDER-1 so that it continued to try to perceive objects not yet recognized, but needed to effect an act that it had decided to do {see Uhr, 1973c[21]).

Answering Queries

DECIDER-1 can output internally stored information in response to an input. Thus we see it a) talk about parts of the scene (Naming, Describing), b) manipulating parts of the scene (Finding, Moving), and c) Answering, by accessing information stored in its memory in response to questions, either direct or implied, in the scene. Thus, e.g.;
WHO IS THE PRESIDENT
might lead to the response "NIXON".

Now the scene is treated as a purely verbal utterance, simply because only verbal rewrite transforms succeed on it. But note how similar this Is to the situation in which a request like:
WHAT IS THIS {(scene of objects))
is intermingled with objects in a scene, in which case the act refers to information that has been extracted and recognized from that scene, rather than to information that has been stored In memory.

Further Examples and Future Extensions

DECIDER-l is described more fully, with a number of examples worked out in detail, in Uhr, 1973d[22], The actual program is presented, explained, and discussed. It is coded to EASEy, an English-like programming language (a variant of SNOBOL) designed to be easy to understand (Uhr, 1973a[19]).

DECIDER-1 is presently being extended to handle scenes that continue over time. In this extension, perceiving, responding, and problem-solving all go on In parallel. The system decides when to respond. But to respond it may need to perceive further objects appropriate to its response. Or it *may need to solve* problems, to deduce a sequence of actions appropriate to the chosen response. Or it may output a request for needed objects, or for help. Thus perceiving, problem-solving, and acting are all intermingled. Each may call upon, and depend upon the other.

References

1. Andrews, D. R., Atrubin, A. J., and Hu, K. The IBM 1975 optical page reader: Part III: Recognition and logic development, IBM J. Res, and Devel., 1968, 12, 364-372.
2. Brice, C. R. and Fennema, C. L. Scene analysis using regions . Artificial Intelligence, 1970,2./ 205-226.
3. Feldman, J. A. et al., The Stanford hand-eye project. Proc. 1st Joint Int. Conf. on Artificial Intelligence, Washington, D. C, 1969, 521-526.
4. Fikes, R. E. and Nilsson, N. J. STRIPS: A new approach to the application of theorem proving to problem solving. Proc. 2d Joint Int. Conf. on Artificial Intel!., London, 1971, 608-620.
5. Gardner, Beatrice T. and Gardner, R. A. Two-way communication with an *Infant* chimpanzee. In; Behavior of Nonhuman Primates, Vol. 4 (A. Schrier and F. Stollnitz, Eds.) New York: Academic Press, 1971, 117-184.
6. Gardner, R. A. and Gardner, Beatrice T. Teaching sign language to a chimpanzee. Science, 1969, 165, 664-672.
7. Jordan, Sara R. Learning to Use Contextual Patterns In Language Processing. Unpubl. Ph.D. Diss., University of Wisconsin, Madison, 1971.
8. Munson, J. H. Experiments in the recognition of hand-printed text. AFIPS Proc. , 1968, *33, 1125-* 1138.
9. Nilsson, N. J. A mobile automaton: an application of artificial intelligence techniques. Proc. 1st Tolnt Int. Conf. on Artificial Intelligence, Washington, D. C, 1969, 509-520.
10. Premack, D. The education of Sarah. Psychology Today, 1970, ±, September, 54-58.
11. Premack, D. Language in Chimpanzee? Science, 1971, 172,< 808-822,
12. Raphael, B. Programming a robot. Proc. IFIP Congress 68. Edinburgh, 1968, H135-H14D.
13. Sauvain, R. and Uhr, L. A teachable pattern describing and recognizing program. Pattern Recognition, 1969, I, 219-232.
14. Simmons, R. F. Answering English questions by computer: a survey, Comm. Asspc. Comput. Mach., 1965, 8, 53-70.
15. Simmons, R. F. Natural language question-answering systems: 1969, Comm. Assoc. Comput. Mach., 1970, 13, 15-30.
16. Uhr, L. Flexible linguistic pattern recognition. Pattern Recognition, 1971, 3, 363-384.
17. Uhr, L. and Vossler, C. A pattern recognition program that generates, evaluates and adjusts its own operators. AFIPS WJCC Conf. Proc., 1961, 19, 555-569.
18. Uhr, L. Layered "recognition cone" networks that preprocess, classify and describe. JEEE. Trans . Comput. , 1972, 2]_, 758-768.
19. Uhr, L. EASEy-2 *-an English-like* Programming Language. Computer Sciences Dept. Tech. Report 178, Univ. of Wisconsin, Madison, 1973. a
20. Uhr, L. Pattern Recognition, Learning and Thought. Englewood-Cliffs: Prentice-Hall, 1973. b
21. Uhr, L., The Description of Scenes Over Time and Space, Proceedings First National Computer Conference, New York, June, 1973. c
22. uhr, L. Recognizing, "understanding," deciding whether to obey, and executing commands. Computer Sciences Dept. Tech. Report 173, Univ. of Wisconsin, 1973. d
23. Winograd, T. Procedures as a Representation for Data In a Computer *Program* for Understanding Natural Language. Unpubl. Ph.D. Diss., MIT, 1971.
24. Zobrist, A, L. The organization of extracted features for pattern recognition. Pattern Recognition, 1971, 3, 23-30.