SPATIAL PLANNING, GEOMETRIC MODELLING AND FUZZY PRODUCTION RULES IN ROBOTIC SYSTEMS

David R. Dodds
BELL-NORTHERN RESEARCH
OTTAWA CANADA

ABSTRACT

This paper is intended for those interested in a means of obstacle avoidance, and generalized object and plan understanding in a robotic system. A spatial planning technique is examined whereby the robot system plans in a general space which greatly enhances the flexibility of manipulator operation by leaving resolution of trajectory and placement minutae to "motion execution time". This permits the robot planning system to deal with elements in the working environment which are in continuous or semi-continuous motion, as well as those normally considered to be at rest. A multi-modal representation and Knowledge Base are used, in part, to achieve this goal. Some examples from a simulated robot system are given to aid the explanation of the ideas.

KEYWORDS: SPATIAL METAPHOR, SPATIAL PLANNING, FUZZY SETS, SCRIPTS, PRODUCTION RULES

I. INTRODUCTION

In order to present some of the ideas more clearly a simplified environment, or "micro-world", is simulated and presented graphically. Since it is the planning aspects that are of most importance and not the quality of the graphics presentation, the graphics have been kept to a low resolution and deal with simplified spatial situations, for the present. The system is evolving and presently resides on an IBM Personal Computer, another reason not to unnecessarily require overhead cycles as fancy graphics would.

The use of simulation and graphical presentation allows testing of hypotheses in the form of "stills" and short "animated" sequences, through operation of the programs. At this juncture the graphics are rather primitive because a refined graphics composition system tool was not available for use on the pc, and the resolution of the pc graphics screen is not particularly high.

The graphics are presented within rectangular views, depicting either a plan view, or a pseudo-three-dimensional view. This will be made clearer at a later point in the presentation.

The simulated robot architecture has been purposely simplified for the time being in order that concentration on the planning may be made. To this end the present robot may be conceived loosely as a device somewhat like an xy plotter on fixed stilts where the "plotter head" is replaced with a shaft free to move in the vertical or z direction. At one end of this shaft there is an appliance which may be thought of as a gripper, for grasping simulated objects. A multi-axis simulated robot is under consideration for future implementation, or perhaps a multi-axis robot simulation will be aquired from a third party.

There are several principle points that are to be conveyed in this paper. The most important concept is that of spatial metaphor. Also of substantial importance is the use of Fuzzy Set Theory. Three concepts from the field of Artificial Intelligence figure into this approach as well, they are : (a) PLANNING, (b) SCRIPTS, (c) PRODUCTION RULES. The standard concepts associated with Geometric Modelling are further necessary.

Not all people who delve into things robotic are aquainted with developments in the field of Artificial Intelligence and so a brief recounting of the meaning of these points will be covered.

PLANNING is the process of developing a sequence of actions to achieve a goal, deciding on a course of action before acting. It is a kind of problem solving. A plan is a representation of a course of action. Usually the plan consists of a list or sequence of actions because we want to achieve certain goals first before we tackle other remaining goals.

When we are putting on our shoe in the morning we look around to find our sock, put it on, then the shoe. Of course some plans are unordered, implicit plans, such as a grocery list, are most often not in any particular order. Sometimes a step in a plan is vague and require further specification. In the overall daily plan of our lives the sub-plan "eat-lunch" is left unspecified. At the point of this step, or most any time prior to it, we may decide to elaborate the details for where we will eat, what we will eat there, and when we will leave. Oftentimes plans have a rich subplan structure; each goal in a plan can be replaced by a more detailed subplan to achieve it. A completed plan is a linear or partial ordering of problem-solving operators {17}.

SCRIPTS are frame-like structures specifically designed for representing sequences of events. Both scripts and frames refer to means of organizing the knowledge representation in a way that directs attention and facilitates recall and inference. By far the most frequent example of scripts and frames is the scenario of the trip to the restaurant. When we visit a restaurant where we have never been before, we have a vast array of expectations based on experience in other restaurants about things we will find, such as menus, tables, waiters and so on. We have expectations about the objects in a typical restaurant, and we also have strong expectations about the sequences of events that are likely to take place. Frames and scripts are tools used in Artificial Intelligence knowledge-representation schemes and the focus has been representing knowledge about the objects and events typical to specific situations.

PRODUCTION RULES or just productions, exist in Artificial Intelligence in a variety of fashions, but they all are based on one very general, underlying idea--the notion of condition-action pairs. Simply stated, a production rule is a statement in the form "IF this condition holds, THEN this action is appropriate. The condition part states the conditions that must be present for the production to be applicable, and the action part is the appropriate action to take. Generally, production rules exist in a production system which also includes an interpreter and a buffer-like data structure sometimes called a context. At first glance they may seem to be nothing more than IF-THEN conditionals found in almst any computer language, however, the activation sequence of a set of production rules does not necessarily follow that of a group of IF-THENs.

```
{label1:} (fuzzy).trigger.pattern        LHS
|| resulting.(fuzzy).output {label2}     RHS
```

Figure 1-1 Production Rule in Canonical form.

Production rule chaining is often used to speed up a production system. This speedup is achieved by including the identity of the next production rule to be examined as part of the right-hand-side or action part of a production rule. Thus the production system examines a select set (as though a "line" or path) of rules rather than the entire system of rules. There is, of course, the possible penalty of "overlooking" a pertinent rule if the chain is not developed carefully.

A slight departure from the usual production rule chaining, wherein there is only one forward chain address, is the addition of a facility wherein the forward chaining "id" or "address" is governed by the context at any given time. The simplest way of conceiving this is to imagine that each rule has a right-hand-side component like the computed transfer statement of most algebraic languages, where the index is tied to the system's present context.

Context is nothing more than a representation of the situation that the system is constrained by at any one time. Context is usually an indication of the present macro-level activity that the robot is undertaking. Bateson's Context Marker is used to derive the indication of the context in the robot simulation discussed here.

Bateson's Context Marker is a cognitive conceptualization where events in one's environment are recognized as being members of one or more particular (cognitive) sets. For example, when a dog hears the leash rattle on the hallway coathook, when brushed by his master say, the dog gets all excited and runs to that place in the hall. This is not simply a conditioned reflex, it is the cognitive association in the dog's mind that the leash is a major element in the going out for a walk context. If the dog is not interested in going for a walk he will not respond to the rattle. If it were simply a conditioned reflex he would always run to the rattling of the leash. A style of Bateson's Context Marker is employed in the robot system as a means of activating scripts. The context so indicated may also be used to (de)activate selective production rule chaining.

It is assumed that everyone is more or less familiar with the notions in Geometric Modelling (GM). Often there are two kinds of representation representation in GM (a) wire-frame and, (b) boundary representation. Wire-frame is simply an explicit visual metaphor, the graphic depiction of objects once physically modelled by building small represenative polygons with wire. Boundary representation carries the graphics aspect farther by explictly including the surface between each set of wire "edges", or often doing away with the edges altogether via continuous smooth, shaded/highlighted explicit surfaces.

In the future Mandelbrot's Fractal Geometry will be investigated for means of better representing/defining irregular patterns and fragmented shapes. Combined with fuzzy systems approaches, Fractal Geometry is expected to permit increased flexibility in the recognition of objects and the definition of more complex behaviour.

A FUZZY SET is a class in which there may be a continuum of grades of membership as, say, in the class of long objects. Such sets underlie much of our ability to summarize, communicate, and make decisions under uncertainty or partial information. Fuzzy sets appear to play an essential role in human cognition, especially in relation to concept formation, pattern classification, and logical reasoning. The grade continuum is usually expressed as a real number between 0:1, 1 being the highest grade.

Boolean Operation	Fuzzy Definition
x AND y	min(x,y)
x OR y	max(x,y)
NOT y	1-y
x IMPLIES y	max(1-x,y)

Figure 1-2 Fuzzy definition equivalents of Boolean Operations.

SPATIAL METAPHOR, sometimes also known as spatial analogy, is the practice of using objects observed in the visual environment as sources of concept primitives. In the case of human vision we witness such events as grains of sand falling in an hour-glass and coin phrases such as "the sands of time". An everyday example of spatial metaphor is the man on the street's (Newtonian) notion of what time is. "Time" is the motion of the hands of his watch, or the little numbers changing in their little window. Neither device "measures time", they are simply convenient systems for displaying regularity, to which depiction we have added numbers in order to conveniently distinguish one regular period or duration from another.

One should keep in mind that a multi-modal representation methodology is prefered to that of a unimodal representation system. This is one of the ways that humans exert such flexible control in varying and complex action environments.

Figure 3-1 depicts the visualization space or planning space which is displayed to signify the actions taken by the planning software. It shows a single low-resolution view of a block depicted as a wire-frame model.

Figure 3-2 depicts the same block with FUZZY LINGUISTIC VARIABLE labels which indicate to humans where the labelled places are in the pseudo-three-dimensional illustration. The same labels are used by the robot system to refer to FUNCTIONS which map 3D co-ordinates into grade of membership magnitudes.

Figure 3-3 The block in Figure 3-2 rotated slightly about its long axis.

Figure 3-4 depicts the path of motion used by the simulated robot system when using its stock FUZZY SCRIPT, called MOVEIT. In order to conserve space on the page the psuedo-three-dimensional depiction of the block involved was collapsed to 2D and scaled down in size. The starting position of the block is at the location having the block outlined with the value zero. The robot (not shown in the illustration) moves the block through the path shown by the numbered blocks. Think of the illustration as showing the position of the block photographed by means of a strobe light, where the number represents the sequential position of the block.

Notice that the script MOVEIT does not display any "imagination", it simply moves the "arm" over the block, reaches down to it, grasps it, lifts it straight up, moves to the side a bit once it reaches the top and then descends straight down, ungrasps it at the bottom. This is the standard mode of MOVEIT, the default if you will.

Figure 3-5 depicts a slight change in the path used to move the block from the "zero" position to the "seven" position. Instead of unimaginatively lifting straight up moving sideways and then going straight down again, this time the robot descended using a curved path, resulting in placing the block not only more to the left than before, but also slightly back.

The standard MOVEIT could have lifted the block as always and placed the block in the same position as in Figure 3-5 simply by moving more to the side in Figure 3-5 than it did in Figure 3-4. While this is true if MOVEIT always executed exactly the same manouever as in Figure 3-4 it would too often place more than one object in the same place, which is usually undesirable. By adding some COMMON SENSE to MOVEIT we see that the path shown in Figure 3-5 is one of many paths the robot might have chosen. The common sense we speak of at this point merely has the script examine the "landing spot" of the current action of moving the block. If that spot is explicitly occupied, in whole or in part, as by the dotted block in the figure, then another landing spot is needed and this script calls on a different script to choose a new spot. This landing checking is performed before the robot actually executes any motion regarding the block itself.

What the robot does is to execute the move in PLANNING SPACE according to the dictates of the proposed mode of (standard, in this case) MOVEIT. By modelling the plan (overall action resulting from the execution of all scripts involved) in planning space the system can determine where conflicts will arise. If a conflict arises during the modelling of the proposed plan the nature of the conflict is used to critique the plan and ultimately to alter the plan so as to reduce or eliminate any conflicts (such as multiply occupied space).

The robot planning system, through its spatial modelling of its working environment (a kind of simplified World Model), having detected occupation of the originally planned landing spot, knows by virtue of the script contents, to select a new landing site. The new site need only meet the constraint that it not be (already) occupied. In Figure 3-5 this means that almost any site will do. Presently the "algorithm" used to select a new spot is to use the old spot as a center and compute a single gaussian distributed dither. The location of the dither is used as the centroid for the new landing spot. This proposed landing site is then checked itself to determine if it is in conflict with any current state of the environment.

When non-stationary objects are in the environment the system treats most motion as though the object moves instanteously occupying all the positions it assumes by virtue of its motion. The information base is updated as objects assume different locations, whether through manipulation by the robot or by another agency. (The robot knows about gravity and the need for support of objects.) Consider the information base containing the equivalent of a time-lapse image of a regularly moving, such as rotating or reciprocating, object. This prevents the placement of any object in a location which may be only temporarily unoccupied. (The difference between a high speed snaphot view and a time exposure view. For example, a fan blade seems a blur of nearly continuous material to our vision while our brain, by virtue of our memory, tells our mind that the fan blade is actually several distinct parts with gaps in between.)

What are the commands which operate the robot?

MOVETO(X,Y,Z)

move the end-effector to co-ordinates x,y,z

PLACE($,AT)

place the named "$" at the location "AT"

CRASP

UNGRASP

WHERE($,X,Y,Z)

tell the system to define object "$" at co-ords x,y,z

DEFINE($,COL,VAL)

enter the attribute "VAL" into the DB at "COL" for the object called "$"

SIZE($,X,Y,Z)

a quick way of giving the (centroid) "expansion" values for object "$"

These are the basic primitives for the robot simulation system. They may be used either by a person operating the robot in manual mode or by any of the scripts, for their needs.

Figure 3-6 Depicts a "multiple-exposure view" of a CLEARTOP script. The action that such a script takes is to take an object (block 0) which is ONTOPOF another object (block *) and put it (block 0) someplace else, clearing the top of the second object (block *). This comes in handy when there is a need to manipulate an object that is in part of a HEAP. (CLEARTOP is one of the major pieces of knowledge that the robot knows for generalized HEAP BUSTING. Notice that in this scene there are now three other objects, blocks A,B,C. The system uses script MOVEIT, which itself uses script CLEARTOP, to locate block 0 from its origin atop block *, planning to place it at 7.

Figure 3-7 Result of the CLEARTOP action, note blocks A and B had moved. Block B has moved into the planned path, necessitating a change in the PLAN. The script MOVEIT still plans to put the block as close as possible to where it had decided before. Through other scripts and its own features MOVEIT performs spatial planning which provides collision avoidance via centering a NEAR mapper around the centroid of the "offender" ("B" in Figure 3-7) and then calculating the standard single gaussian distributed dither to obtain co-ordinates for the landing site, leaving the obscuring object (block 0 in Figure 3-6) at the "somewhere else" site ("X" in Figure 3-7). Finally, the robot executes the moving of the original object of interest (block * in Figure 3-6), now that its top has been cleared in preparation for this action, to its own "somewhere else" site ("*" in Figure 3-7). The location of "somewhere else" sites is by means of a constrained random selection.

A process which I have dubbed, "gyration", for lack of better term, is used to run through PLANned actions in the PLANNING SPACE before operating the robot. Gyration does not place objects in the same place, by rote nor is the location selected entirely randomly. The macro behaviour BUST HEAP, for example, uses CHANGE DEMONS to keep the planning space uptodate vis-a-vis the centroid expansions of all objects in the space. All such volumes are then rejected by the PLANner as volumes through which the robot can move and as candidates for landing sites, etc. By centering a fuzzy space mapping function, such as NEAR $, on some object $ in the environment, the robot can then calcultate all those spaces which are NEAR, rather than AT, by means of attending to the grade of membership values. Next a single dither can be calculated in this sub-space. This gives a totally random site in this space, which prevents collisions or interference in the future. In order to make pickup of objects more efficient from this general sub-space in the future it is functional to use a guassian distribution about the NEAR centroid in use to cause a many object tendency to cluster nearer the centered object/site than farther away from it as a uniform distribution randomness would cause. The many object tendency to cluster reduces the amount of future motion the robot would have to go through in order to grasp these objects once again.

Figure 3-8 Simplified map of grade of membership in three dimensions. It depicts how the basic grade of membership mapping function (Figure 3-9b) is employed in representing the concept NEAR. Locations in the planning space may be FUZZILY located relative to objects in the space. For example, NEAR * is shown to be locations in the space having larger grades of membership for the location relative to the location of the object *, (WHERE * 38 7 7).

Visualize the unimodal function, in Figure 3-9b, being centered about the location of * in Figure 3-8. Further visualize the function being "rotated" about *, changing the 2D graph into a 3D ("solid") graph. The base of the graph would be placed at Z = 7. For locations less than Z = 7, that is elevations between Z = 0 and Z = 7, the solid graph would be flipped (to "point downward").

The lines of numbers labelled X, Y, Z; radiating from * in Figure 3-8; are the magnitudes of the grades of membership (times .1). Simply stated in English, the farther one goes from the location * in Figure 3-8, the lower the grade of membership value is.

FAR *, therefore, is easily seen to be simply unity (1.00) minus the grade of membership for NEAR *, (i.e. 1.00 - NEAR *). A position of NEAR * being grade of membership equal to nine (9), for example, the equivalent for FAR * (for the same position *) is 1.00 - 0.900 = 0.100. NEAR $ is thus a complementary relationship with FAR $. The fuzzy locations BELOW, DOWN, ABOVE, UP, (BE)SIDE are similarly related, according to axis direction (x,y,z) and axis sign (+,-).

Another example from Figure 3-8. BACK may be represented (along Y) as the graph in Figure 3-9a. That is to say, BACK is any location (along Y axis) with a larger grade of membership. FRONT is the fuzzy location also based on Y axis locations. Imagine the converse of Figure 3-9b, i.e. the peak at zero and the bottom of the curve tending away from X's zero. FRONT is any location with a large grade of membership, along Y.

Imagine Figure 3-9a being labelled GRAMS along the X axis, grade of membership along Y. This being so, HEAVY would be depicted by this graph, and so on for ON, COLOUR, LOCATION, INERTIA, SHAPE, etc.

The context marker can be used after: IF x LIKE y THEN context = A. X is an ordered list of attributes, as is y. LIKE, in its simplest form then is simply the value of the product moment correlation of x and y, with a threshold applied. (Y has context A, if ? LIKE Y, then the context is similar. Things similar to the same thing are similar to each other.)

IF X sufficiently LIKE Y THEN context = A-like. Sufficiently is a HEDGE, computed by having the threshold for LIKE map using similar function to Figure 3-9a. The value of context LIKEness is another mapping of the correlation, using a similar function. (I.E. "If x is somewhat like y then contexts are similar" can, thusly, be computed.)

So what? If you know how to perform action x (say move a cube out of the way) then you also know how to move anything else not too uncube-like, also!

Due to the eight (8) page limitation imposed on the length of papers in this Proceedings it was not possible to include many of the details of interest. An addendum may be available which explicates these, for those who are interested. (Write to the author.)

```
OBJECT: BLOCK          (any object which
                        has the attribute
   IS  MOVEABLE         BLOCK, automatic-
   HAS LOCATION         ally INHERITS
   HAS INERTIA          these properties,
   HAS COLOUR           and thus it is for
   HAS SHAPE            all other kinds of
   HAS IDNAME           objects.)
```

```
OBJECT: BLOCK

   IS  MOVEABLE         DEFINITELY
   HAS LOCATION         AT BLOCK0 PLACE0
   HAS INERTIA          SOME
   HAS COLOUR           GREEN
   HAS SHAPE            RECTANGULAR SOLID
   HAS IDNAME           BLOCK0, NO ALIASES
   HAS OWNER            NONE
```

```
BLOCK frame

 Specialization-of:  PHYSICALOBJECT
 Geometry-type:      RECTANGULARSOLID
 Orthogonality:      an integer (DEF=3)
 Colour:             GREEN, RED, or BLUE
 Inertia:            SOME, MEDIUM, HEAVY,
                             or MASSIVE
```

```
BLOCK0 frame

 Specialization-of:  BLOCK
 Geometry-type:      RECTANGULARSOLID
 Orthogonality:      3
 Colour:             GREEN
 Inertia:            SOME
```

```
MOVEIT Script

 Objects:   (Blocks, Robot, Surface)
 Roles:     (Statics, Movers, Recipients
             Actor)
 Viewpoint: Actor
 OccurTime: (ActorChangeEnvironment)
 OccurPlace (LocationOfRecipient)
```

```
Event-Sequence:

 first: Locate-Recipient Script
 then1: if (CLEAR-PATH)
          then (MOVETO $Recipient)
 then2: if (CLEARTOP $Recipient)
          then (GRASP, MOVETO $?,
                UNGRASP, HOME)
         FAIL (BUSTHEAP $Recipient)
           then first
 finally: Change-Demon-Script,
           Leave-Plan-Script
```

Figure 3-1 A Low-Resolution Three Dimensional Planning Space

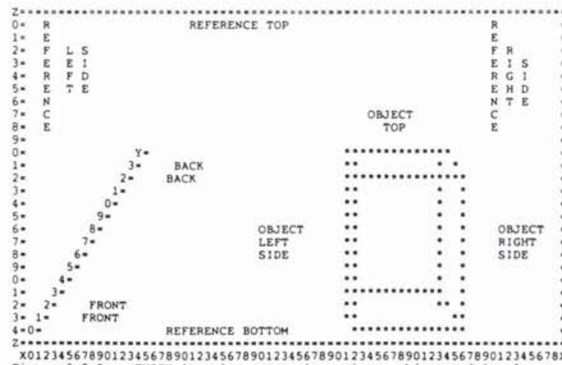

Figure 3-2 Some FUZZY location-names vis-a-vis an object and its frame.

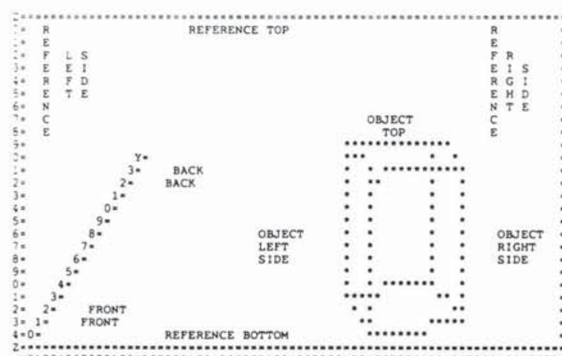

Figure 3-3 The block in Figure 3-2 rotated slightly about its long axis.

Figure 3-4 Motion of block based on a simple FUZZY SCRIPT.

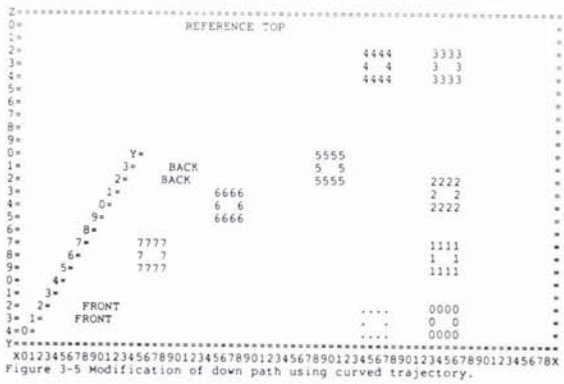

Figure 3-5 Modification of down path using curved trajectory.

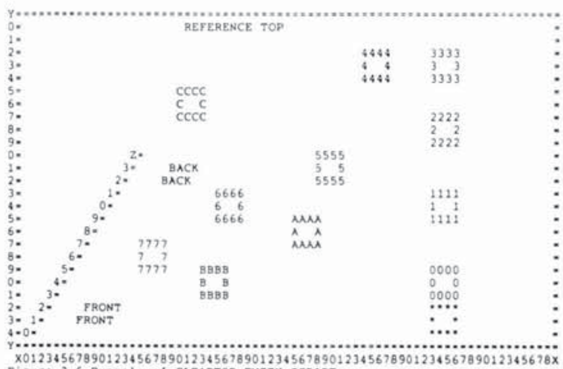

Figure 3-6 Example of CLEARTOP FUZZY SCRIPT.

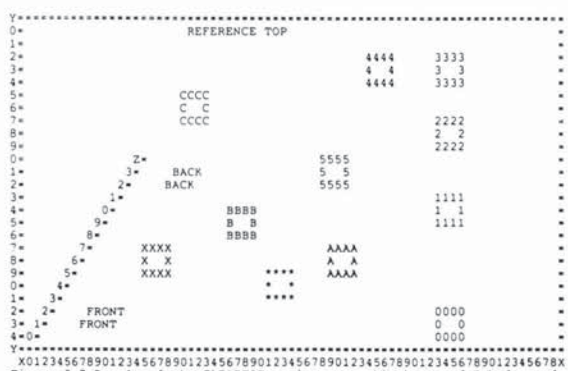

Figure 3-7 Result of the CLEARTOP action, note blocks A and B had moved.

Figure 3-8 Simplified map of grade of membership in three dimensions.

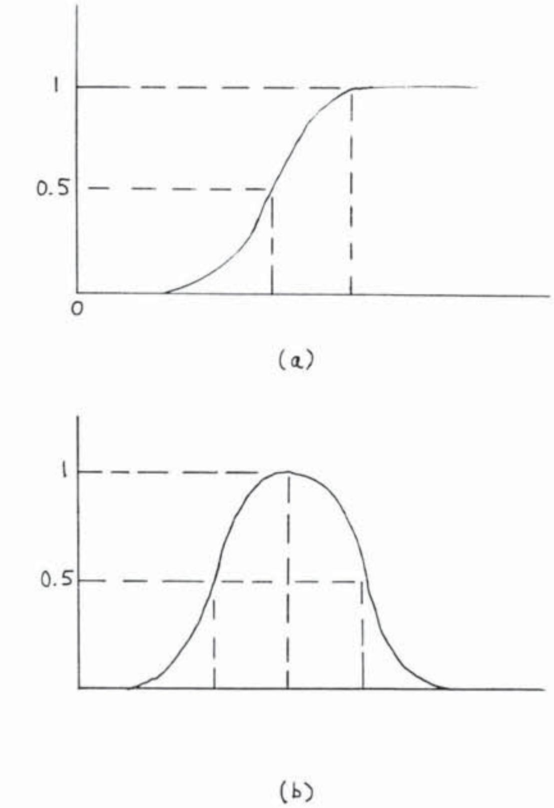

Figure 3-9

References

1. D. R. Dodds, "Use of Spatial Metaphor in World Model Representation, Spatial Planning and Robotic System Task Definition," Proceedings of Robotic Intelligence and Productivity Conference, Detroit, November 1983.

2. P. Cohen and M. Grinberg, "A Theory of Heuristic Reasoning About Uncertainty", The AI Magazine, American Association for Artificial Intelligence, Vol. 4, No. 2, Summer 1983, pp. 17-24.

3. J. Larkin et al., "Expert and Novice Performance in Solving Physics Problems," Science, Vol. 208, 20 June 1980, pp. 1335-1342.

4. "VAL Primer", Unimation Robotics Inc., May 1980

5. "User's Guide to VAL: A Robot Programming and Control System", Unimation Robotics Inc., June 1980.

6. N. Sondheimer, "Spatial Reference and Semantic Nets," American Journal of Computational Linguistics AJCL71.

7. B. Colby, "Culture Grammars," Science, Vol. 187, 14 March 1975, Number 4180, pp. 913-919.

8. A. Kandel and S. Lee, "Fuzzy Switching and Automata: Theory and Applications," Crane Russak, New York, 1979.

9. J. Weiss, M. Donnell, "A General Purpose Policy Capturing Device Using Fuzzy Production Rules", in Advances in Fuzzy Set Theory and Applications, North-Holland, pp. 589-604.

10. R. Jain, "Application of Fuzzy Sets for the Analysis of Complex Scenes", in Advances in Fuzzy Set Theory and Applications, North-Holland North-Holland, pp. 577-588.

11. M. Nowakowska, "Fuzzy Concepts: Their Structure and Problems of Measurement", in Advances in Fuzzy Set Theory and Applications, North-Holland, pp. 361-387.

12. M. Kokawa et al., "Fuzziness and Catastrophe in Estimation and Decision Processes", in Advances in Fuzzy Set Theory and Applications, North-Holland, pp. 463-480.

13. D. McDermott, "Contexts and Data Dependencies: A Synthesis", I.E.E.E. Trans. on Pattern Analysis and Machine Intelligence Vol. PAMI-5, No. 3, May 1983, pp. 237-246.

14. D. R. Dodds, "Fuzzy Logic Computer Implementation of Metaphor from Ordinary Language," American Association for the Advancement of Science, Annual Meeting, Toronto, Contributed Paper 239, January 1981.

15. ---, "Socialization in the Future Machine," Canadian Information Processing Society, CIPS '76, Montreal, 1976.

16. Data Driven Automation, an I.E.E.E. Spectrum Special Issue, May 1983.

17. P. Miller, "ATTENDING: Critiquing a Physician's Management Plan", I.E.E.E. Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 5, September 1983, pp. 449-461.

18. R. Jarvis, "A Laser Time-of-Flight Range Scanner for Robotic Vision", I.E.E.E. Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 5, September 1983, pp. 505-512.

19. C. Rieger, "Understanding By Conceptual Inference", American Journal of Computational Linguistics AJCL13.

20. H-C. Lee and K-S. Fu, "Generating Object Descriptions for Model Retrieval", I.E.E.E. Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 5, September 1983, pp. 462-471.

21. G. Bateson, "Steps To An Ecology of Mind," Chandler Pub. Co., 1972.

22. P. Winston, "Learning and Reasoning by Analogy: The Details", Artificial Intelligence Memorandum AIM 520, Massachussets Institute of Technology Artificial Intelligence Laboratory, May 1980.

23. ---, "Learning by Augmenting Rules and Accumulating Censors", Artificial Intelligence Memorandum AIM 678, Massachussets Institute of Technology Artificial Intelligence Laboratory, May 1982.

24. S. Vere, "Planning in Time: Windows and Durations for Activities and Goals", I.E.E.E. Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 3, May 1983, pp. 246-267.