April 1976

MSYS: A SYSTEM FOR REASONING ABOUT SCENES

by
Harry G. Barrow
Jay M. Tenenbaum

Artificial Intelligence Center
Technical Note 121

Projects 1187 and 4763

The research reported herein was supported by the Office of Naval Research under Contract N00014-71-C-0294. Additional support was furnished by the Advanced Research Projects Agency under Contracts DAHC04-72-C-0008 and DAHC04-75-C-0005.

ABSTRACT

MSYS is a system for reasoning with uncertain information and inexact rules of inference. Its major application, to date, has been to the interpretation of visual features (such as regions) in scene analysis. In this application, features are assigned sets of possible interpretations with associated likelihoods based on local attributes (e.g., color, size, and shape). Interpretations are related by rules of inference that adjust the likelihoods up or down in accordance with the interpretation likelihoods of related features. An asynchronous relaxation process repeatedly applies the rules until a consistent set of likelihood values is attained. At this point, several alternative interpretations still exist for each feature. One feature is chosen and the most likely of its alternatives is assumed. The rules are then used in this more precise context to determine likelihoods for the interpretations of remaining features by a further round of relaxation. The selection and relaxation steps are repeated until all features have been interpreted.

Scene interpretation typifies constraint optimization problems involving the assignment of values to a set of mutually constrained variables. For an interesting class of constraints, MSYS is guaranteed to find the optimal solution with less branching than conventional heuristic search methods.

MSYS is implemented as a network of asynchronous parallel processes. The implementation provides an effective way of using data driven systems with distributed control for optimal stochastic search.

I INTRODUCTION

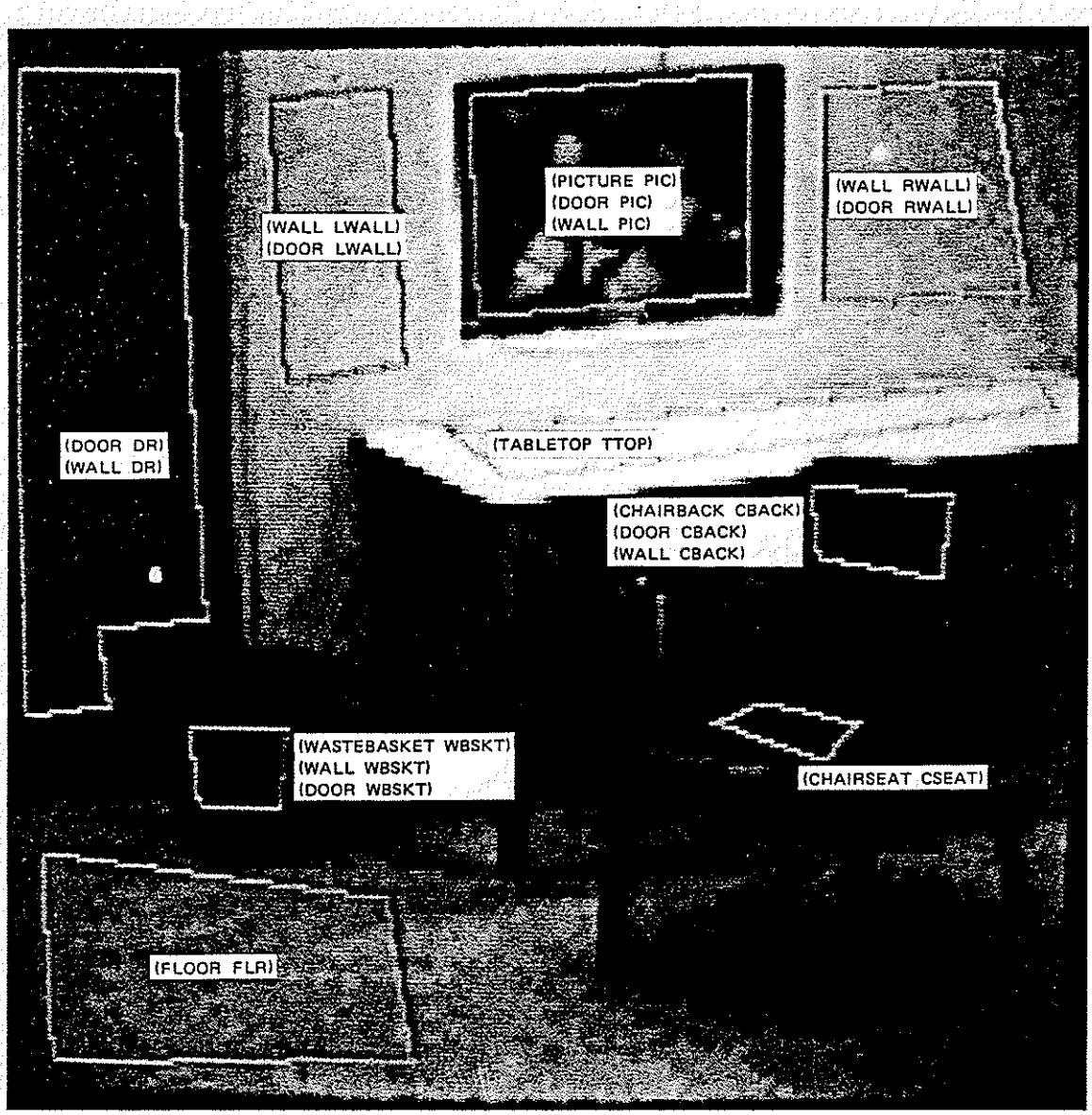
In scene analysis, it is frequently impossible to interpret parts of an image taken out of context. Different objects may have similar appearances, while objects belonging to the same functional class can have strikingly different appearances (e.g., chairs). Ambiguous local interpretations must be ruled out by using contextual constraints to achieve a meaningful, globally consistent interpretation of the whole scene.

We use an elementary example involving arbitrary constraints to illustrate the reasoning entailed in scene interpretation. A room scene is manually partitioned into regions, as shown in Figure 1. The labels in the figure indicate the locally possible, interpretations of each region. These interpretations are obtained by matching region attributes, such as height and orientation, against local constraints given in Figure 2.* Region DR, for example, must be either DOOR or WALL, since these are the only vertically oriented objects that can extend both below and above the allowed height ranges of other objects, such as PICTURES, CHAIRBACKS, and WASTEBASKETS. In this example, horizontally oriented regions all received unique interpretations determined by their height, but all vertical regions received at least two possible interpretations [DOOR, WALL, and when consistent with height extremes. WASTEBASKET, CHAIRBACK, or PICTURE]. It is assumed that regions do not span more than one object.

When the interpretation of a region cannot be uniquely determined from local attributes, it must be deduced from global relationships such as those in Figure 2. Deduction might proceed as follows:

---

*The scene analysis experiments reported herein were performed using coordinated arrays of color, intensity, and range data. The range data simulated the output of a developmental time-of-flight laser range finder, whose current accuracy is about an inch in ten feet. Region height and orientation were obtained from the range data using transformations described in Reference 1.[1] The local interpretations shown in Figure 1 were obtained automatically using the measured height and orientation of the regions.



SA-3805-45

FIGURE 1 POSSIBLE REGION INTERPRETATIONS OF A SIMPLE ROOM SCENE

INTERPRETATION	LOCAL CONSTRAINTS	RELATIONAL CONSTRAINTS
1. FLOOR	(HEIGHT FLOOR) < 0.1 feet (ORIENTATION FLOOR) ~ HORIZONTAL	(HOMOGENEOUS FLOOR)*
2. DOOR	0 < (HEIGHT DOOR) ≤ 7 feet (ORIENTATION DOOR) ~ VERTICAL	(HOMOGENEOUS DOOR) (NOT (ADJACENT DOOR PICTURE)) (ROOMPARTITION DOOR)**
3. WALL	0 < (HEIGHT WALL) ≤ 8 feet (ORIENTATION WALL) ~ VERTICAL	(HOMOGENEOUS WALL) (ROOMPARTITION WALL)
4. PICTURE	3 ≤ (HEIGHT PICTURE) ≤ 5.5 feet (ORIENTATION PICTURE) ~ VERTICAL	(NOT (ADJACENT PICTURE DOOR))
5. CHAIRBACK	1.5 ≤ (HEIGHT CHAIRBACK) < 3 feet (ORIENTATION CHAIRBACK) ~ VERTICAL	
6. CHAIRSEAT	1 ≤ (HEIGHT CHAIRSEAT) ≤ 2 feet (ORIENTATION CHAIRSEAT) ~ HORIZONTAL	
7. TABLETOP	2 ≤ (HEIGHT TABLETOP) ≤ 3 feet (ORIENTATION TABLETOP) ~ HORIZONTAL	
8. WASTEBASKET	0 < (HEIGHT WASTEBASKET) < 1.5 feet	

*The homogeneity constraint requires that all regions labeled with the constrained interpretation, in this case Floor, have approximately the same brightness and color. No particular brightness or color is, however, required.

**The roompartition constraint requires that all regions labeled with the constrained interpretation (Door or Wall) have approximately the same brightness found along the top edge of the image vertically above that region's center of mass. This constraint is based on the assumptions that Door and Wall are homogeneous (as defined above), and that in a standard eye level view of a normal sized room, Door and Wall will extend beyond the upper border of the image. This constraint is a special case of homogeneity and would not be required for analyzing exhaustively partitioned scenes.

FIGURE 2 SEMANTIC CONSTRAINTS FOR ROOM SCENE DEPICTED IN FIGURE 1 (PARTIAL LIST)

- Regions PIC, WBSKT, and CBACK cannot be WALL or DOOR, because their brightnesses are much less than that along the top edge of the image vertically above them, which violates the constraint ROOMPARTITION. Consequently, region PIC must be the PICTURE, WBSKT must be WASTEBASKET, and CBACK must be CHAIRBACK.
- Region LWALL and RWALL must then be WALL, since they are adjacent to region PIC, and DOOR cannot be adjacent to PICTURE.
- Region DR cannot be WALL because all regions labeled WALL are required to have the same brightness. Therefore, region DR must be DOOR.

Scene interpretation, as illustrated by the above example, is an attempt to explain observed sensory data in terms of prior knowledge about the depicted domain. The explanation can entail many types and levels of knowledge, some of which may be probablistic or inconsistent. It must also allow for the likelihood that the data is noisy. For these reasons, scene interpretation is not a purely deductive problem with a unique correct solution; it is a problem that requires a search for the best or optimum explanation. However, the quantities of data and knowledge that are involved appear to rule out the use of conventional search techniques.

This paper describes a working scene interpretation program, called MSYS, in which knowledge sources compete and cooperate until a consistent explanation of the scene emerges by consensus. The consensus is achieved by a network of processes (representing independent knowledge sources) that communicate via shared global variables. Each process attempts to explain a fragment of the data (a region or a few regions in a segmented scene) in terms of its own limited knowledge. The confidence of an explanation is communicated to other processes attempting to explain overlapping fragments, and may cause them to reevaluate their own hypothesis. The confidence adjustment cycle continues until equilibrium is achieved. The equilibrium confidence values establish a preference ordering for the alternative interpretations of each fragment, which is used to guide a heuristic search toward the best solution. We conjecture that, with enough knowledge (i.e., constraints), the equilibrium state will correspond directly to a solution (where one interpretation for each fragment is by far the best). A competent knowledge-based vision system would thus never actually need to resort to search.

In Section II of this report we develop a heuristic search algorithm, M^*, that employs the equilibrium process described above as an evaluation function. In Section III, we describe MSYS, which is an efficient serial implementation of the M^* algorithm using (simulated) asynchronous parallel processes. In Section IV we contrast MSYS with previous work on scene interpretation and constraint optimization and suggest applications of our work in both areas.

The work described in this report was motivated by the work of Duda at SRI, Barrow and Turner at Edinburgh, and Yakimovsky and Feldman at Stanford. Duda[20] was concerned with assigning interpretations to regions in a previously segmented scene. The interpretation process involved a tree search to determine the set of region interpretations having the highest joint likelihood. The inefficiencies of tree search limited this approach to simple scenes with few regions and few interpretations. Barrow and Turner[3] developed an elegant generalization of Waltz's filtering algorithm, that dramatically reduced the amount of search required to solve constraint satisfaction problems. Yakimovsky and Feldman[2] showed how segmentation and interpretation could be integrated by using the likelihoods of region interpretations to guide region merging. In this paper, we describe the combination of these ideas into a system that can efficiently determine optimal region interpretations in a segmented image. This work was begun in 1973; preliminary results were previously reported in Reference 4.

II THE M* ALGORITHM

M* is a heuristic search algorithm intended for multivariate optimization problems involving interacting nonlinear constraints. Problems are posed by providing (1) a set of possible assignments for each variable, with associated a priori likelihoods and (2) a set of constraints that determine the a posteriori likelihood of any variable assignment for a given instantiation of the remaining variables.

A solution of the problem--also called a terminal state--is any complete instantiation of the variables. A partial solution is a nonterminal state in which at least one variable still has a set of possible assignments. The objective is to find the solution in which the combined a posteriori likelihoods of the instantiated assignments is greatest. M* uses a relaxation method to solve simultaneously the set of constraint equations that determine the overall merit of a solution and also to bound the potential merits of partial solutions. These latter estimates are used to guide a conventional A* search algorithm toward the optimal solution.

The M* algorithm was initially formulated specifically for scene interpretation and is described here in those terms. However, it appears to have broad applicability as a general search algorithm, as is suggested later in this report.

The scene interpretation problem can be defined more formally as follows: Given a set of regions and corresponding region attributes, a generic set of possible region interpretations and a set of constraints on the generic interpretations determine the assignment of interpretations to regions that best satisfies the constraints (i.e., that assignment for which the combined a posteriori likelihoods of the interpretations is greatest). The search for this optimal solution starts from an initial state in which all regions have sets of possible interpretations. It then proceeds through a series of partially instantiated states to terminal states in which every region has been instantiated to a unique interpre-

tation. Region interpretations are instantiated by pinning the likelihoods of alternative interpretations of the region to zero.

A. State Evaluation

Each interpretation in a terminal state has a likelihood that is a heuristic function of the local region attributes and the likelihoods of the interpretations assigned to other regions. This likelihood tells how well the semantic constraints on the interpretation are satisfied in the current state. The precise form of a likelihood function need not concern us at this time, except to note that the likelihood associated with any particular region interpretation is, in general, a nonlinear combination of the likelihoods associated with all the other region interpretations. Hence, the determination of likelihoods in a terminal state may involve the simultaneous solution of a set of nonlinear equations. A relaxation method, described more fully in Section III.C, is used to obtain a consistent set of likelihood values.

A terminal state will be scored by summing the interpretation likelihoods over all regions. The optimization objective, then, is to find the highest scoring terminal state, hopefully without exhaustive enumeration.

To avoid enumeration, a heuristic search is desired. A heuristic search estimates the best terminal scores that could be ultimately achieved by further instantiating a given partially instantiated state. Search then proceeds by developing the state with the best potential terminal score. The best terminal score attainable from a given partially instantiated state is estimated by calculating, for each region interpretation remaining in that state, an upper limit on the a posteriori likelihood of that interpretation in any terminal state. The likelihoods of the most likely interpretation of all regions are then summed to provide an upper-bound on the best terminal score that could be obtained.

An upper limit on the likelihood of an interpretation is computed by ignoring interactions among the constraints governing that likelihood and assuming that each constraint can be optimized independently. In evaluating the contributions of each constraint, it is assumed that other regions are instantiated to the interpretation possibility that would maximize the

likelihood under consideration, if only that one constraint applied. An upperbound on the likelihood of the current interpretation is then computed on the basis of these individually optimized constraints. (For example, the likelihood estimate could be based on the strength of the least satisfied constraint.)

The likelihood limits of all interpretations are, of course, interdependent and should thus be computed simultaneously to obtain a tighter bound on the overall state score. In other words, the likelihood limit of each interpretation should be computed assuming all other interpretation likelihoods are at their upper limits. A consistent set of likelihood limits can be obtained with the same relaxation method used earlier to obtain consistent likelihoods in terminal states by suitably modifying the likelihood functions.

Not much can be said regarding either convergence of the relaxation process or the quality of the resulting likelihood estimates without knowing more about the functions that compute likelihood. Two important classes of likelihood functions are those that increase the likelihood of a constrained interpretation when the likelihood of compatible interpretations elsewhere in the image increase, and those that decrease likelihoods when the likelihood of incompatible interpretations increase. If we restrict ourselves to the former, then the partial derivatives of any interpretation likelihood, taken with respect to the likelihood of any other interpretation likelihood, will be positive. The positive partial derivatives guarantee that the relaxation process will be nonoscillatory: all likelihoods will monotonically increase (or decrease) until either a stable state is achieved, or the limit 0 (or 1) is reached. Consequently, the process is guaranteed to converge providing a consistent solution exists in that range. When a region is instantiated to some interpretation, the likelihoods of alternative interpretations of that region are reduced to zero. This can cause monotonic reductions in the likelihood estimates of other region interpretation, but no increases. Thus, the estimated score of a partially instantiated state is a strict upperbound on the score of any terminal state that can be reached along that branch of the search tree.

B. Heuristic Search

The above results suggest a heuristic search strategy patterned after the A* algorithm of Hart, Nilsson and Raphael.[5] The search proceeds, at each stage, by restoring the highest scoring partially instantiated state and then instantiating the highest likelihood interpretation of an uninstantiated region in a new copy of that state.

Following the instantiation, the relaxation process is repeated twice: first, to update the estimated likelihoods and potential score of the new instantiated state, and second, to update estimates in the original state with the instantiated interpretation removed as a possibility. (Effectively, the search is split into 2 disjoint branches.) Both states are added to a prioritized list of open states. Search then continues in the highest scoring state, terminating when the best state, is also a terminal state. In Appendix A it is proved that this algorithm, with the stated restrictions on the formal likelihood functions will terminate with the optimal set of interpretations for those constraints.

The complete M* algorithm, as actually implemented, is summarized in Figure 3. The heart of the algorithm is the use of relaxation methods in Steps 0 and 4 to estimate consistent likelihood limits of alternative interpretations of regions. These estimates are then used to improve the order of state selection and instantiation at Steps 1 and 2, respectively. Step 5 introduces an additional constraint on the solution, namely, that all regions have at least one reasonably likely interpretation; any state in which all possible interpretations of a region receive unacceptably low likelihoods is abandoned. Step 6 allows the algorithm to terminate early, whenever all regions in the highest scoring state have one clearly dominant interpretation (i.e., an interpretation at least ten times as likely as any alternative), whether or not the dominant interpretations have been formally instantiated.

While the basic search algorithm used by M* is inherently serial, the relaxation process that guides the search is conceptually parallel. All possible interpretations could, in principle, be represented by independent processes that interact to achieve equilibrium likelihoods. Considering the highly constrained nature of most scene interpretation problems, one might hope that the equilibrium likelihoods of "correct" region interpretations would dominate those of "incorrect" alternative interpretations, prior to any instantiation, and that this dominance would be further enhanced in the equilibrium states resulting from each subsequent correct

(0) Establish consistent likelihood limits for all interpretations based on the a priori likelihoods and constraints. Record all non-uniquely instantiated regions (on IQUEUE) and save the resultant state (on SQUEUE)*.

(1) Reinstate the current globally best state (from SQUEUE).

(2) Select a region interpretation for instantiation (from IQUEUE).

(3) Generate branches corresponding to acceptance and rejection of that instantiation hypothesis, setting up a new state for each.

(4) In each of the new states reevaluate all interpretation likelihoods affected by acceptance (or rejection) of the hypothesized interpretation; pursue the consequences of all reevaluations as far as possible short of further hypothesizing.

(5) Evaluate the global score of each state by summing the likelihoods of the best interpretation for each region. Any state in which all possible interpretations of some region are deleted (or receive a very low likelihood) is assigned a zero global likelihood.

(6) If all regions in either state are assigned unique interpretations, terminate and return that state as the best scene interpretation. If all regions in both states are uniquely interpreted, return the state with the highest score.

(7) Update the IQUEUE associated with each state and save both states (on SQUEUE) with priority determined by their respective scores.

(8) Go to (1).

*The roles of SQUEUE and IQUEUE are explained in section III-D.

FIGURE 3 BEST FIRST SEARCH ALGORITHM

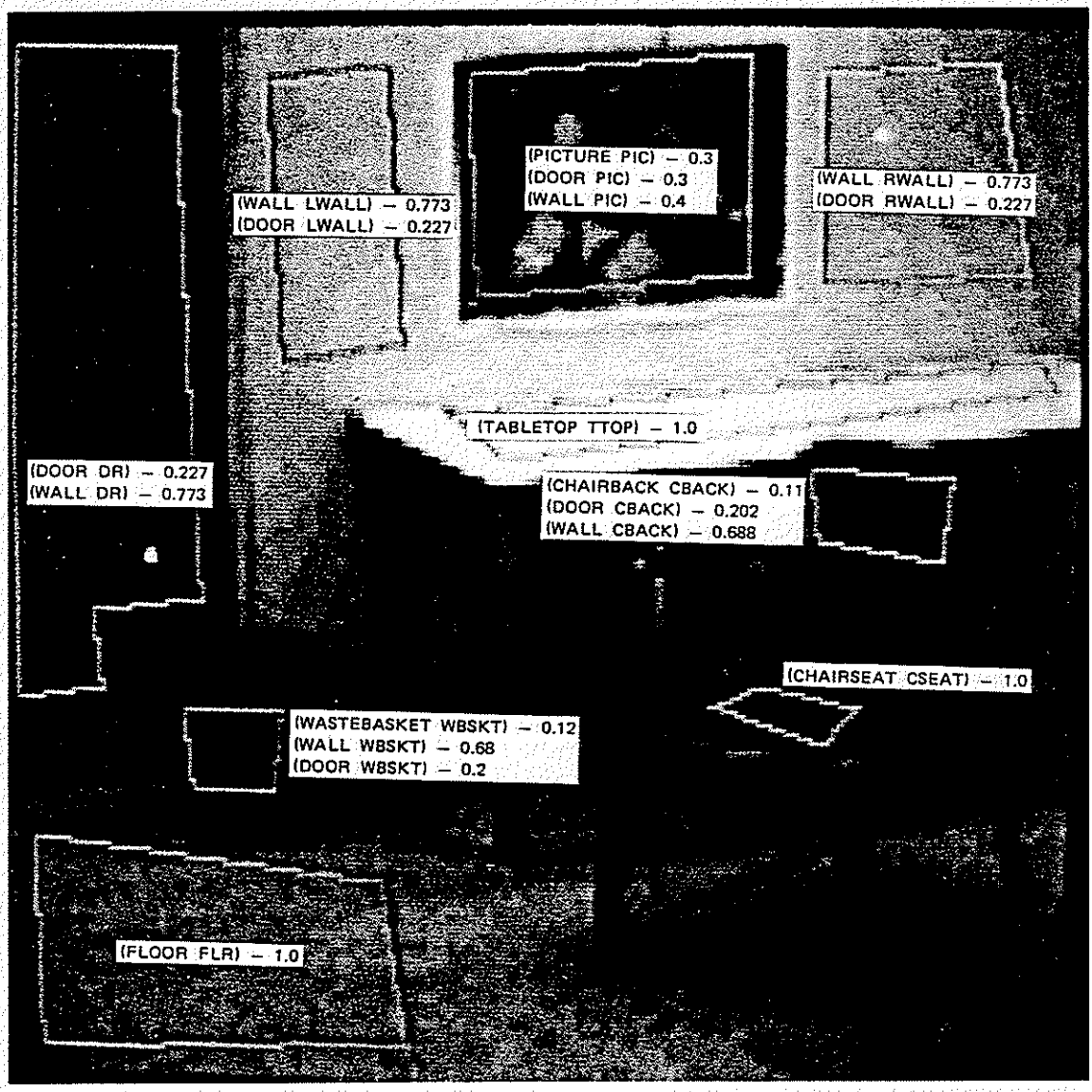
instantiation. This expectation has been substantially confirmed in experimentation with a variety of likelihood functions, including many that did not satisfy the restrictions required for formal optimality. Consequently, search proceeds toward the correct solution with little or no backup and instantiation becomes virtually a serial readout of a parallel search.

C. An Example

The M^* algorithm is illustrated with the same regions, interpretations, and constraints used in the introductory example (Figures 1 and 2), but with likelihoods attached to the interpretations. The a priori interpretation likelihoods for the example are shown in Figure 4. These values are based on the relative areas occupied by each of the alternative interpretations of a region in several training scenes. (The a priori likelihood of a region interpretation was computed by dividing the amount of area with that interpretation in the training images, by the sum of the corresponding areas over all the possible interpretations of that region. WALL was thus always a more likely a priori interpretation than DOOR for vertical regions.)

The first step of analysis entails the estimation of a consistent likelihood limit for each locally possible region interpretation, based both on the a priori (local) likelihood of that interpretation and on the likelihood limits estimated for other semantically constrained interpretations in the scene. The likelihood limit of each interpretation, or just likelihood for short, is computed by a function, hereafter known as a likelihood function. Likelihood functions typically consist of a combination (e.g., a product) of terms--one for each constraint applicable to that interpretation. Each term raises or lowers the likelihood of the interpretation, depending on the type of constraint and the likelihoods of the region interpretations that satisfy the constraint in the current scene. (In Section III.B, we discuss in detail the computation of likelihoods.)

Two basic relational constraints are used in our present example: adjacency and homogeneity. Both constraints reduce the likelihood of constrained interpretations due to the presence of incompatible interpretations elsewhere in the image.



SA-3805-44R

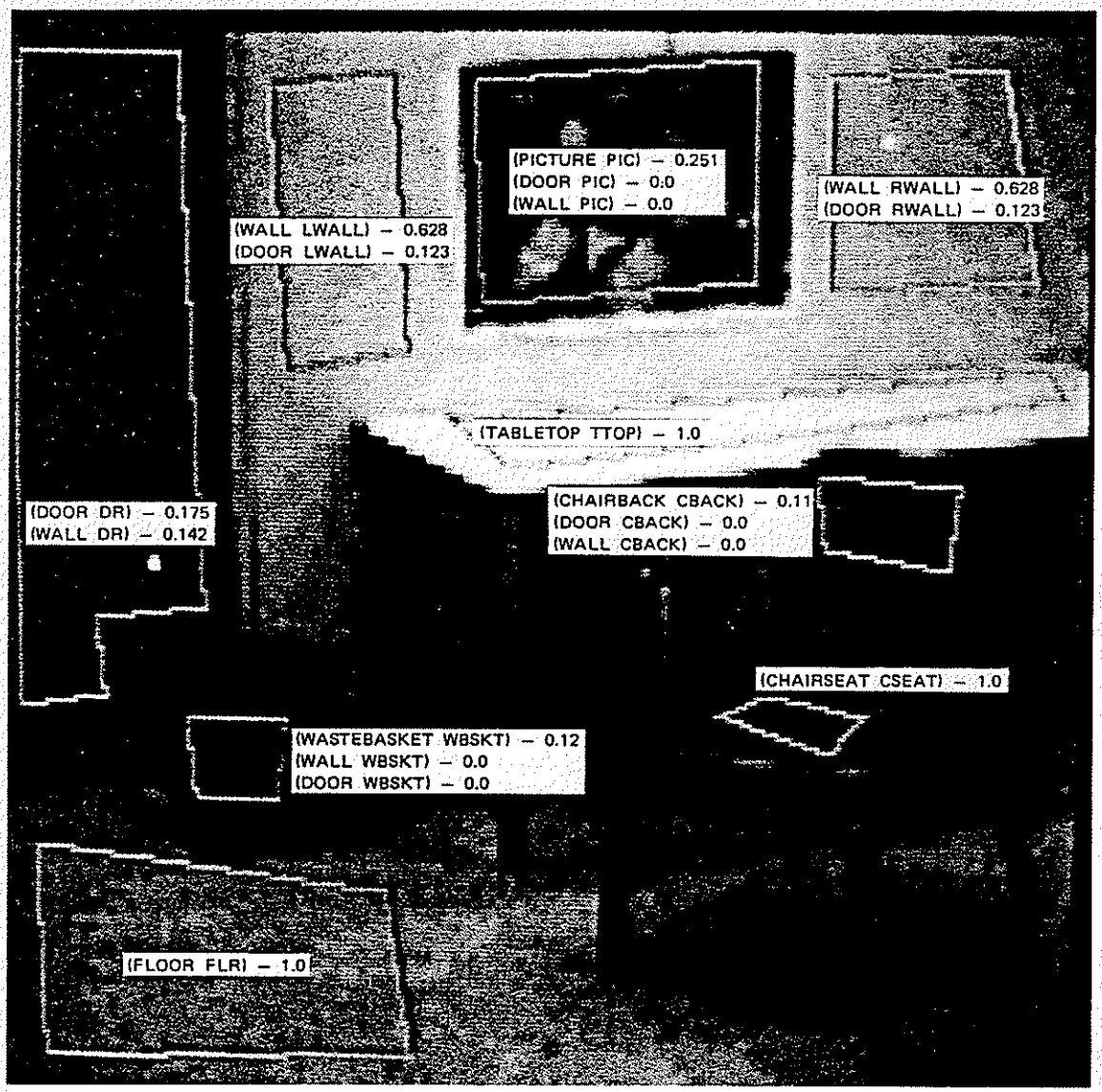
FIGURE 4 MANUALLY PARTITIONED ROOM SCENE WITH A PRIORI INTERPRETATION LIKELIHOODS BASED ON HEIGHT AND SURFACE ORIENTATION

The relational constraint (NOT*(ADJACENT PICTURE DOOR)) on Pictures (see Figure 2) introduces a term in the likelihood function of region interpretation (PICTURE PIC), reducing the likelihood that region PIC is PICTURE by an amount proportional to the likelihood that adjacent regions LWALL and RWALL are thought to be doors. Conversely, terms are included in the likelihood functions of the interpretations (DOOR LWALL) and (DOOR RWALL), reducing their likelihoods proportional to the likelihood that region PIC is PICTURE. A loose definition of adjacency has been adopted so this constraint could be used in a partially segmented scene. (Two regions are adjacent if the line connecting their centers does not pass through a third region.)

Homogeneity constraints require that all regions with a specified interpretation have approximately the same brightness. The likelihood of a constrained interpretation is therefore reduced in proportion to the maximum likelihood that any nonhomogeneous region also has that interpretation. In Figure 4, for example, the likelihood that a light colored region such as LWALL or RWALL is WALL must be reduced in proportion to the likelihood that any dark colored region, such as DR, PIC, or CBACK, is WALL, and vice versa. It does not matter whether WALL is light or dark--only that dark- and light-colored regions cannot with high likelihood simultaneously be interpreted as WALL.

ROOMPARTITION, as described above, is a special case of the homogeneity constraint. The brightnesses of regions admitting the interpretations WALL or DOOR (i.e., surfaces that "partition rooms") are required to be similar to the brightness at the top of the image vertically above the regions center of mass. Region interpretations that fail this test are rejected by reducing their a priori likelihoods to zero. The likelihoods of interpretations that pass are unaffected. This constraint had the effect of eliminating DOOR and WALL as possible interpretations of vertically oriented regions with inadequate vertical extent, specifically from the regions CBACK, PIC, and WBSKT.

The interpretation likelihoods resulting from an initial relaxation of the likelihood functions sketched above are shown in Figure 5. In that process, DOOR and WALL were eliminated by the constraint ROOMPARTITION,



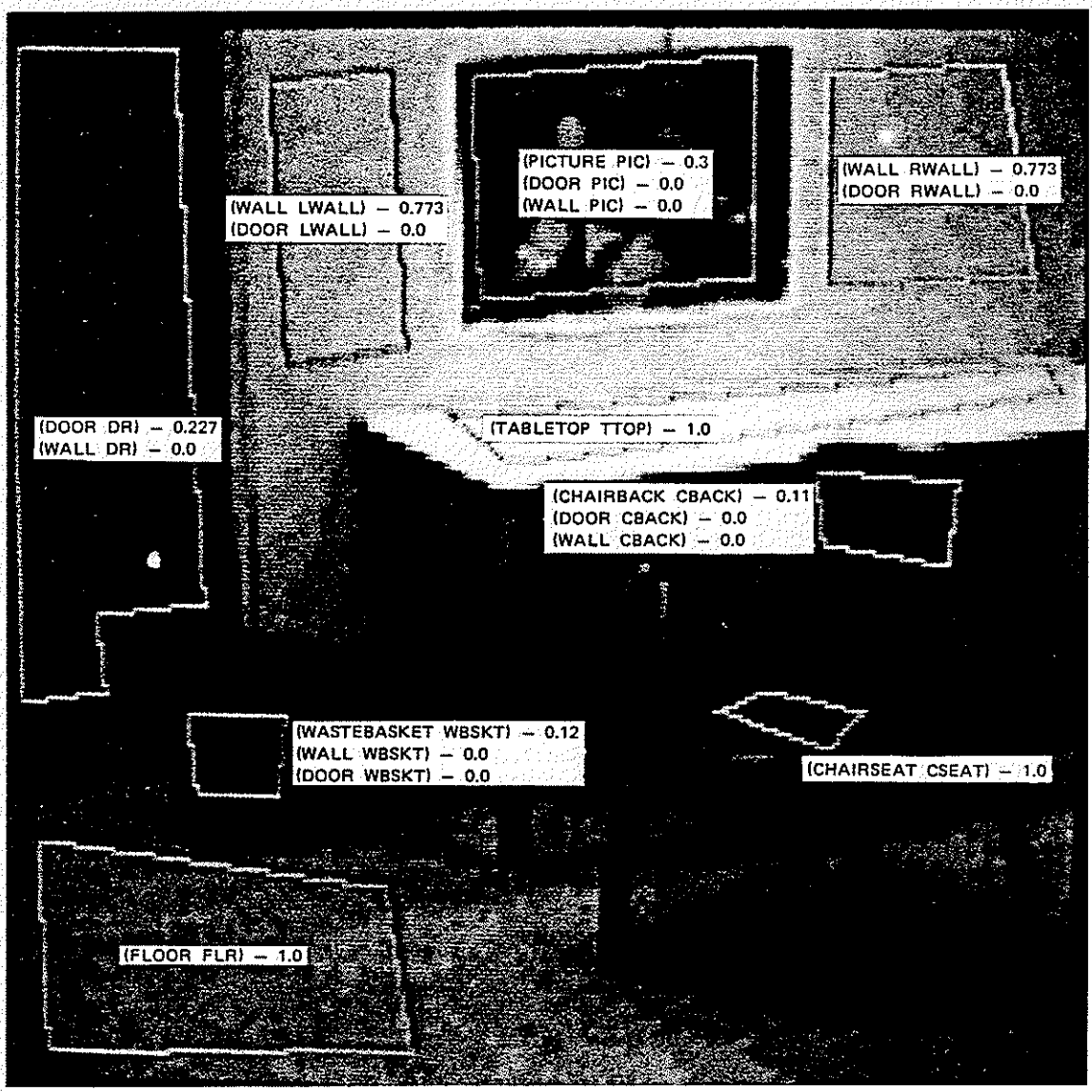
SA-3805-45R

FIGURE 5 INITIAL EQUILIBRIUM LIKELIHOODS BEFORE SEARCH

as possible interpretations of regions CBACK, PIC, and WBSKT, leaving region PIC with the unique interpretation of PICTURE. The latter result depressed the likelihood that adjacent regions LWALL or RWALL were DOOR, which in turn enhanced the likelihood that the dark region DR was a door. Significantly, the correct interpretation of every region has acquired a likelihood higher than that of any alternative interpretation for the region.

The final stage of analysis involves searching for a set of unique interpretations with the highest joint likelihood. The only remaining ambiguity involves the interpretation of regions DR, LWALL, and RWALL, all of which still admit both DOOR and WALL as possibilities. Homogeneity constraints force LWALL and RWALL, both light-colored regions, to take the same interpretation (either WALL or DOOR) and DR, a dark-colored region, to take the opposite interpretation. This basic ambiguity is resolved by the adjacency constraint on pictures, which leads to a contradiction when either LWALL or RWALL is instantiated to DOOR.

The search proceeded without need for backup, the relative likelihoods of correct interpretations increasing monotonically with each successive correct instantiation. The final interpretation likelihoods for the regions in Figure 4 are presented in Figure 6. A detailed trace of the reasoning showing all instantiations and resulting reevaluations appears in Appendix B.



SA-3805-46R

FIGURE 6 FINAL EQUILIBRIUM LIKELIHOODS FOLLOWING SEARCH

III MSYS--AN IMPRLEMENTATION OF M* FOR SCENE INTERPRETATION

MSYS is an operational system coded in INTERLISP that performs scene interpretation using the M* algorithm. The system consists of four major components:

- A facility for defining regions in a scene and for measuring their pictorial attributes.
- An initialization procedure that compiles, for each possible region interpretation, a function that estimates the maximum likelihood of that interpretation, based on the maximum estimated likelihoods of the other region interpretations.
- A relaxation mechanism for determining consistent likelihood estimates simultaneously for all the possible region interpretations.
- A mechanism for performing a backtrack search.

A. Region Definition and Description

Region definition and attribute measurement are done in MSYS by a previously developed interactive scene interpretation system known as ISIS.[1,28] Regions can be defined in ISIS manually, by outlining them on a display with a cursor; semiautomatically, by providing a crude outline which the system then refines; or fully automatically, by calling a region growing program similar to that employed by Yakimovsky.[2]

A variety of INTERLISP functions are available for assessing the attributes of defined regions. These attributes include statistics on brightness, hue, and saturation, as well as height and orientation when range data is available. Functions also exist for accessing the polygonal boundaries of regions in both (2-D) image and (3-D) world coordinates. The interactive features of ISIS proved useful in developing classification criteria for assigning local interpretations, and in devising procedures for testing spatial relations between regions.

B. Compilation of Likelihood Functions

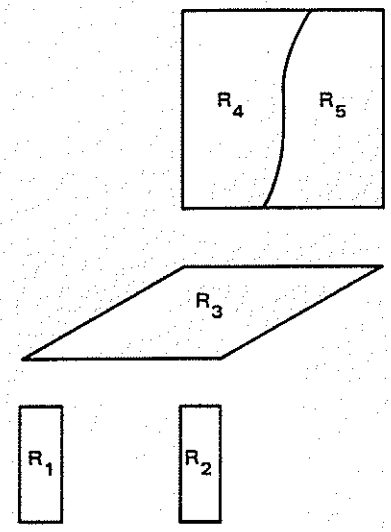
Before describing the compilation of likelihood functions, we digress briefly to describe the structure of these functions and their numerical evaluation.

1. Structure of Likelihood Functions

The likelihood of a region interpretation has been described as a heuristic function of local region attributes, region relations, and the likelihoods of other region interpretations. A prototypical likelihood function is illustrated in Figure 7. Here (CHAIRSEAT R3), the likelihood that Region R3 is CHAIRSEAT is represented as a conjunction of three independent terms. The first term is the a priori likelihood (0.8) that Region R3 is CHAIRSEAT, based on the attributes of that region, such as height and surface orientation. The other two terms express, respectively, the degree to which each of the generic constraints on the interpretation CHAIRSEAT--namely (ABOVE CHAIRSEAT CHAIRLEG) and (ABOVE CHAIRBACK CHAIRSEAT)--are satisfied by the interpretation possibilities of the other regions.

The likelihood expression in Figure 7 can be interpreted in the same way as a LISP function; each subexpression enclosed in parentheses is a function returning a real value, which is then used in evaluating the superexpression in which the subexpression appears. Terms representing region interpretations, such as (CHAIRLEG R2), evaluate to the the current likelihood of that interpretation, while terms representing region relations, e.g., (ABOVE R4 R3) express the degree to which two regions satisfy the specified relation, based on the relative coordinates of their respective boundary extremes. Both likelihoods and relations are defined the range (0,1). The functions AND* and OR* take real valued arguments on the range (0,1) and return values in the same interval. They should thus be considered as general functions for combining evidence rather than as conventional logical conjunctions and disjunctions. The nature of these evidence combining functions will be discussed in Section III.B.2. Functional definitions for some common region relations are given in Appendix C.

Terms expressing the satisfaction of a relational constraint such as (ABOVE CHAIRBACK CHAIRSEAT) follow a standard format. Each constraint is supported (i.e., satisfied) by a disjunction, OR*, of all the potential ways it can be satisfied in the image. A constraint is potentially satisfied by a region admitting the required interpretation (e.g., CHAIRBACK) and having the specified spatial relationship with the constrained region (e.g., ABOVE). The region interpretations (R4 CHAIRBACK) and (R5 CHAIRBACK),



SA-1530-59

POSSIBLE REGION INTERPRETATIONS

REGION	POSSIBLE INTERPRETATIONS AND A PRIORI LIKELIHOOD		MSYS REPRESENTATION
R1	CHAIRLEG	0.8	(CHAIRLEG R1)
	TABLELEG	0.2	(TABLELEG R1)
R2	CHAIRLEG	0.7	(CHAIRLEG R2)
	TABLELEG	0.3	(TABLELEG R2)
R3	CHAIRSEAT	0.8	(CHAIRSEAT R3)
	TABLETOP	0.2	(TABLETOP R3)
R4	CHAIRBACK	0.25	(CHAIRBACK R4)
	WALL	0.75	(WALL R4)
R5	CHAIRBACK	0.25	(CHAIRBACK R5)
	WALL	0.75	(WALL R5)

CONSTRAINTS ON CHAIRSEAT
(ABOVE CHAIRBACK CHAIRSEAT)
(ABOVE CHAIRSEAT CHAIRLEG)

LIKELIHOOD PROCEDURE FOR (CHAIRSEAT R3)

(AND* 0.8

(OR* (AND* (ABOVE R_3 R_2) (CHAIRLEG R_2))

(AND* (ABOVE R_3 R_1) (CHAIRLEG R_1)))

(OR* (AND* (ABOVE R_4 R_3) (CHAIRBACK R_4))

(AND* (ABOVE R_5 R_3) (CHAIRBACK R_5))))]

FIGURE 7 FORMATION OF A PROCEDURE FOR COMPUTING THE LIKELIHOOD OF A REGION INTERPRETATION

for instance, provide potential satisfaction of the constraint (ABOVE CHAIRBACK CHAIRSEAT) in Figure 7. The degree of satisfaction provided by each potential supporting region interpretation is represented by the conjunction (AND*) of the interpretation likelihood and the degree to which the region satisfies the stipulated relation with the constrained region. For instance, (AND* (ABOVE R4 R3) (CHAIRBACK R4)) expresses the degree to which the region interpretation (CHAIRBACK R4) satisfies the constraint (ABOVE CHAIRBACK CHAIRSEAT) imposed on region interpretation (CHAIRSEAT R3).

Negated relations such as (NOT (ADJACENT PICTURE DOOR)) are represented by first forming the disjunction of conjuncts that expresses support for the basic constituent relation, in this case (ADJACENT PICTURE DOOR). The resulting disjunction is then embedded in a negated clause of the form (NOT* (OR* - -)). In a likelihood function, this term has the desired effect of penalizing an interpretation to the extent that the forbidden relation is satisfied.

Constraints that do not fit the format of binary relations are represented within a likelihood function by support terms that are arbitrary functions of region attributes, relations, and interpretation likelihoods. For example, the support term representing the room scene constraint (HOMOGENEOUS DOOR) ensured that two regions of different brightness could not be simultaneously interpreted as DOOR with high likelihood.

The procedures in Figure 8 are the likelihood functions that were used in the room scene example. The listing omits functions of unconstrained interpretations, such as FLOOR, whose likelihoods always remain at their a priori values.

The likelihood function for region interpretation (DOOR LWALL), line 57 in Figure 8, illustrates both a negated relation and a nonrelational constraint. This likelihood function contains three terms, the first being the a priori likelihood 0.227. The second term, (NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT))), expresses support for the nonstandard constraint (HOMOGENEOUS DOOR). This term is the negation of a disjunction containing all region interpretations in which the interpretation DOOR is paired with a region whose brightness differs from that of the constrained region (LWALL) by more than 10%. Its effect is to reduce the likelihood

```
00001   VARIABLE: (WASTEBASKET WBSKT)
00002   VALUE:    .12
00003   PROCEDURE:
00004             (WASTEBASKET WBSKT)
00005   RELATIVES:
00006             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00007
00008
00009   VARIABLE: (DOOR WBSKT)
00010   VALUE:    0.0
00011   PROCEDURE:
00012             (DOOR WBSKT)
00013   RELATIVES:
00014             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00015             ((OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00016
00017
00018   VARIABLE: (WALL WBSKT)
00019   VALUE:    0.0
00020   PROCEDURE:
00021             (WALL WBSKT)
00022   RELATIVES:
00023             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00024             ((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))
00025
00026
00027   VARIABLE: (DOOR RWALL)
00028   VALUE:    .123
00029   PROCEDURE:
00030             [AND* .227 (AND* (NOT* (OR* (DOOR DR)
00031                                         (DOOR PIC)
00032                                         (DOOR CBACK)
00033                                         (DOOR WBSKT)))
00034                              (NOT* (AND* (ADJ RWALL PIC)
00035                                          (PICTURE PIC]
00036   RELATIVES:
00037             ((OPTION (DOOR RWALL) (WALL RWALL)))
00038             ((AND* (ADJ RWALL PIC) (DOOR RWALL)))
00039             ((OR* (DOOR LWALL) (DOOR RWALL)))
00040
00041
00042   VARIABLE: (WALL RWALL)
00043   VALUE:    .628
00044   PROCEDURE:
00045             [AND* .773 (NOT* (OR* (WALL DR)
00046                                   (WALL PIC)
00047                                   (WALL CBACK)
00048                                   (WALL WBSKT]
00049   RELATIVES:
00050             ((OPTION (DOOR RWALL) (WALL RWALL)))
00051             ((OR* (WALL LWALL) (WALL RWALL)))
```

FIGURE 8 DATABASE AT EQUILIBRIUM PRIOR TO SEARCH

```
00052
00053
00054   VARIABLE: (DOOR LWALL)
00055   VALUE:    .123
00056   PROCEDURE:
00057             [AND* .227 (AND* (NOT* (OR* (DOOR DR)
00058                                         (DOOR PIC)
00059                                         (DOOR CBACK)
00060                                         (DOOR WBSKT)))
00061                                (NOT* (AND* (ADJ LWALL PIC)
00062                                            (PICTURE PIC]
00063   RELATIVES:
00064             ((OPTION (DOOR LWALL) (WALL LWALL)))
00065             ((AND* (ADJ LWALL PIC) (DOOR LWALL)))
00066             ((OR* (DOOR LWALL) (DOOR RWALL)))
00067
00068
00069   VARIABLE: (WALL LWALL)
00070   VALUE:    .628
00071   PROCEDURE:
00072             [AND* .773 (NOT* (OR* (WALL DR)
00073                                   (WALL PIC)
00074                                   (WALL CBACK)
00075                                   (WALL WBSKT]
00076   RELATIVES:
00077             ((OPTION (DOOR LWALL) (WALL LWALL)))
00078             ((OR* (WALL LWALL) (WALL RWALL)))
00079
00080
00081   VARIABLE: (DOOR CBACK)
00082   VALUE:    0.0
00083   PROCEDURE:
00084             (DOOR CBACK)
00085   RELATIVES:
00086             ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00087             ((OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00088
00089
00090   VARIABLE: (CHAIRBACK CBACK)
00091   VALUE:    .11
00092   PROCEDURE:
00093             (CHAIRBACK CBACK)
00094   RELATIVES:
00095             ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00096
00097
00098   VARIABLE: (WALL CBACK)
00099   VALUE:    0.0
00100   PROCEDURE:
00101             (WALL CBACK)
00102   RELATIVES:
```

FIGURE 8 DATABASE AT EQUILIBRIUM PRIOR TO SEARCH (Continued)

```
00103              ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00104              ((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))
00105
00106
00107    VARIABLE: (PICTURE PIC)
00108    VALUE:      .251
00109    PROCEDURE:
00110              [AND* .3 (NOT* (OR* (AND* (ADJ RWALL PIC)
00111                                        (DOOR RWALL))
00112                                  (AND* (ADJ LWALL PIC)
00113                                        (DOOR LWALL]
00114    RELATIVES:
00115              ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00116              ((AND* (ADJ RWALL PIC) (PICTURE PIC)))
00117              ((AND* (ADJ LWALL PIC) (PICTURE PIC)))
00118
00119
00120    VARIABLE: (DOOR PIC)
00121    VALUE:      0.0
00122    PROCEDURE:
00123              (DOOR PIC)
00124    RELATIVES:
00125              ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00126              ((OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00127
00128
00129    VARIABLE: (WALL PIC)
00130    VALUE:      0.0
00131    PROCEDURE:
00132              (WALL PIC)
00133    RELATIVES:
00134              ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00135              ((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))
00136
00137
00138    VARIABLE: (DOOR DR)
00139    VALUE:      .175
00140    PROCEDURE:
00141              (AND* .227 (AND* (NOT* (OR* (DOOR LWALL)
00142                                          (DOOR RWALL)))
00143                               (NOT* 0.0)))
00144    RELATIVES:
00145              ((OPTION (DOOR DR) (WALL DR)))
00146              ((OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00147
00148
00149    VARIABLE: (WALL DR)
00150    VALUE:      .142
00151    PROCEDURE:
00152              [AND* .773 (NOT* (OR* (WALL LWALL)
00153                                    (WALL RWALL]
00154    RELATIVES:
00155              ((OPTION (DOOR DR) (WALL DR)))
00156              ((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))
```

FIGURE 8 DATABASE AT EQUILIBRIUM PRIOR TO SEARCH (Concluded)

that the bright region LWALL is door by an amount proportional to the likelihood that any of the dark regions DR, PIC, or CBACK are thought to be door. Incidentally, an identical negated disjuntion supports the constraint (HOMOGENEOUS DOOR) in the likelihood function of the region interpretation (DOOR RWALL), since regions RWALL and LWALL have similar brightnesses.

The third term (NOT* (AND* (ADJ LWALL PIC)(PICTURE PIC))), supports the constraint (NOT (ADJACENT PICTURE DOOR)). This term reduces the likelihood that region LWALL has the interpretation DOOR, proportional to the likelihood that region PIC has interpretation PICTURE and the degree to which regions PIC and LWALL are adjacent. The other likelihood functions shown in Figure 8 are composed of similar terms supporting applicable homogeneity or adjacency constraints.

2. Numerical Evaluation of Likelihoods

The numerical evaluation of likelihood functions requires that conventions for combining evidence be established. Two simple schemes were considered. The first scheme is a set theoretic formulation that treats interpretation likelihoods and region relation values as independent probabilities. A conjunction of likelihoods evaluates to the product of those likelihoods, the negation of a likelihood to one minus the likelihood, and a disjunction of likelihoods to one minus a product of the negations of the likelihoods.

The second scheme treats likelihood functions as definitions of fuzzy sets. Following Zadeh's conventions[6], a conjunction of likelihoods evaluates to the minimum likelihood, a disjunction of likelihoods evaluates to the maximum likelihood, and a negated likelihood evaluates to one minus the likelihood.

A few qualitative remarks can be made contrasting the evaluation of likelihood functions using the set theoretic and fuzzy set formulations. Since likelihood functions are composed of conjunctions of constraint terms, the likelihood of an interpretation is limited, in both formulations, by its least satisfied constraint. In particular, an interpretation can effectively be ruled out (i.e., its likelihood forced to zero) by a single badly violated constraint. The set theoretic approach penalizes the likelihood

of an interpretation that is supported by a large number of moderately satisfied constraints, while the fuzzy set approach does not. Each constraint term in a likelihood function is typically represented by a disjunction of terms expressing alternative ways of satisfying the constraint. The strength of a constraint in the fuzzy set formulation will thus equal the support provided by the individual interpretation that best satisfies the constraint. The set theoretic approach, on the other hand, yields a strength for constraints that is strictly greater than the support provided by any individual interpretation. This characteristic leads to overestimates of terminal likelihoods, which is consistent with requirements for optimal search stated earlier.

A number of more elaborate ways for summing evidence have appeared recently in the literatures of scene analysis, speech understanding, and diagnosis[7-12]. Yakimovsky's[7] Bayesian formulation of interpretation probabilities is theoretically pleasing but suffers from the difficulty of obtaining realistic conditional probabilities in complex scene domains. Barrow and Popplestone[8] describe an ad hoc method of evaluating conjunctions of constraints that is based on the number of constraints that are violated and the seriousness of the violation. Their evaluation gives preference to interpretations with many partially satisfied constraints over interpretations with fewer constraints that are more completely satisfied.

Shortliffe[9] described a quantification of inexact reasoning in medical diagnosis in terms of confirmation theory. His formulation has many desirable features as a basis for evaluating competing hypotheses in scene interpretation. Individual constraints make independent contributions to belief in a hypothesis and therefore can be acquired incrementally. Moreover, since evidence for and against a hypothesis is treated independently, hypotheses are not penalized for missing features (which may be optional or occluded) or for a large number of partially satisfied constraints in the absence of specific contradictory evidence. Duda[12] has recently devised a new Bayesian approach to the combination of evidence that appears to eliminate some potential discontinuities in Shortliffe's likelihood computations. A more sophisticated approach may be adopted if future experiments prove our simple set theoretic approach inadequate. So far, the choice of evaluation does not appear to be critical.

3. Compilation Process

Every region interpretation in the image is represented by its own likelihood function. The process of compiling a likelihood function for a region interpretation begins by retrieving the applicable generic constraints. A support term is formuluted for each applicable constraint, expressing how well that constraint is satisfied by the other region interpretations in the image. These support terms are then combined with the a priori likelihood in a function that expresses an upper bound on the overall likelihood of the interpretation.

Formulation of the support term for a constraint involves searching the image to determine all region interpretations that could potentially satisfy the constraint in a terminal state. Details of the search vary, however, depending on the type of constraint. For standard relational constraints, all regions admitting the required interpretations are tested to determine whether they also obey the required region relation (e.g., above and adjacent) with respect to the constrained region. Region interpretations that pass* are represented in the support term for the constraint by a conjunction of the interpretation likelihood and the strength with which the region relations was satisfied.

Support terms for constraints that do not fit the standard relational format are compiled by special procedures that are provided for each such constraint. These support gathering procedures are called with the constrained region interpretation as an argument and return a support term that is inserted directly into the top-level conjunction of the likelihood function. Such a procedure was invoked, for example, in compiling the support term for the constraint (HOMOGENEOUS DOOR) in the likelihood function of the region interpretation (DOOR LWALL). This procedure first retrieves all region interpretations in the current image containing the constrained interpretation, in this example (DOOR LWALL), (DOOR DR),

---

*As an expedient, a region interpretation is only considered as potential support for a constraint if the strength of the corresponding region relation, which is static exceeds 0.1.

(DOOR PIC), and (DOOR RWALL). The procedure then forms a negated disjunction containing the subset of these whose regions differ in brightness from that of the constrained region (LWALL) by more than 10%.

The support term for the constraint (HOMOGENEOUS WALL) in the likelihood function of (WALL LWALL), line 72 in Figure 8, is compiled using the same procedure, which now operates by retrieving all region interpretations containing the constrained interpretation WALL.

Compound constraints, such as (NOT* (ADJACENT PICTURE WALL)) and (OR* (ADJACENT PICTURE WALL)(ADJACENT PICTURE FRAME)) are first parsed into their elementary constituent constraints. Support terms are obtained independently for each constituent. These support terms are then inserted back into the original compound constraint in place of the corresponding constituent. The resulting compound term expresses the support that exists for the original compound constraint. For example, consider the constraint (NOT* (ADJACENT PICTURE DOOR)) applied to the region interpretation (DOOR RWALL) in Figure 4. The term (AND* (ADJACENT PIC RWALL)(PICTURE PIC)) was first formulated to represent support for the constituent (ADJACENT PICTURE DOOR). (In this example, region interpretation (PICTURE PIC) was the sole source of support.) The support term then replaced the constituent in the original compound constraint, forming the compound expression (NOT* (AND* (ADJACENT PIC RWALL)(PICTURE PIC))). This expression appears in the likelihood function of region interpretation (DOOR RWALL) at line 34 of Figure 8.

If any constraint has no support among the other region interpretations, the likelihood of the constrained interpretation is pinned at zero. Alternatively, a compilation procedure could search the image for a new region with the needed interpretation, using techniques for goal directed search, such as those developed by Garvey[13].

C. Relaxation--The XDEMON System

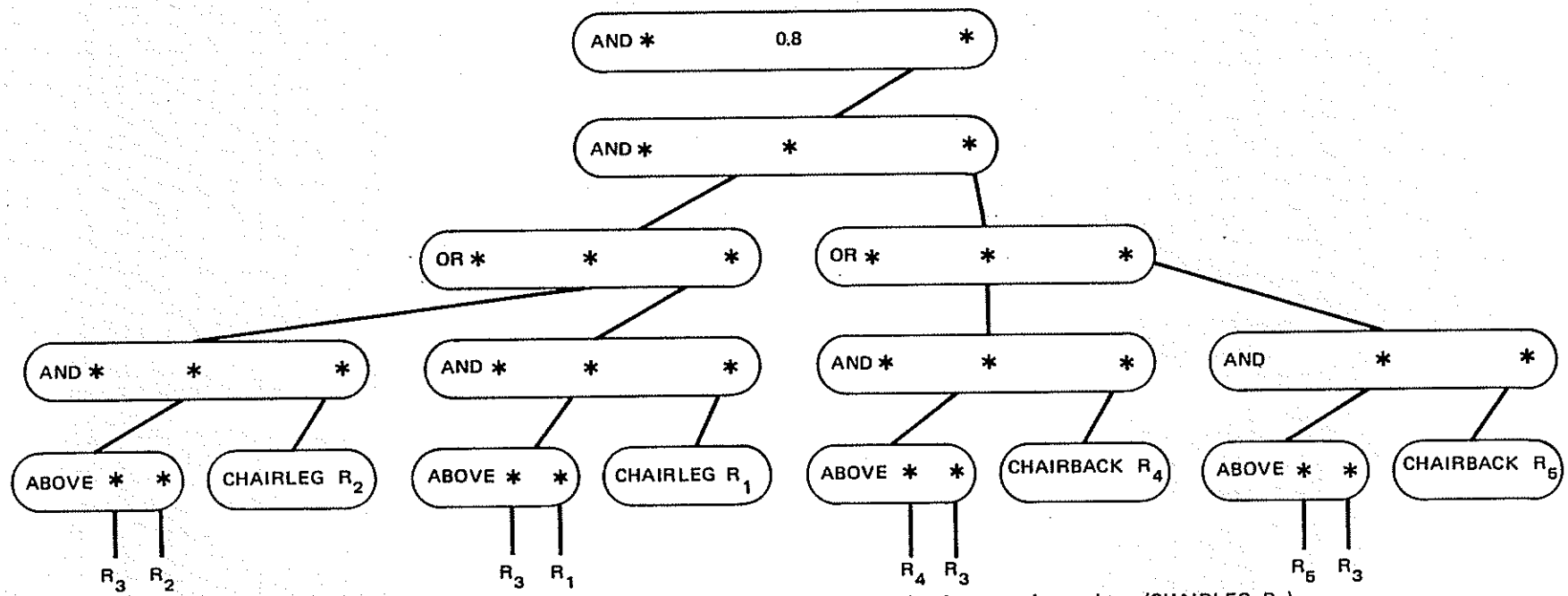
The likelihood functions must be evaluated simultaneously to determine consistent likelihood estimates for all interpretations. Previously, it was remarked that evaluation could proceed in a highly parallel manner, with each likelihood function represented by an independent process. This parallel approach has been efficiently simulated on a serial computer using asynchronous

parallel processes that interact through a global data base. A set of LISP function, known collectively as the system XDEMON, have been developed to facilitate creation of this data base for particular constraint problems. These functions are documented in Appendix D.

The global data base consists of variables, each of which has a value and an associated process that computes the value in terms of the current values of other variables. Each variable also has a list of related variables that use the present variable as input. When the cumulative change in the value of a variable exceeds a threshold, its related variables are reevaluated by adding their processes to a set of jobs to be run. Running a process can change the value of a related variable, causing additional processes to be activated. Execution terminates when the job set is empty.

For scene interpretation, each possible region interpretation is represented in the data base (1) by a variable of the form (CHAIRBACK R4) whose value is the current likelihood of that interpretation, and (2) by an associated process for computing that likelihood value based on the current likelihood values of other region interpretations. Likelihood evaluation is initiated by loading the job set with the processes of interpretations for which updated likelihoods are required. To obtain an initial set of consistent likelihoods the job set is loaded with the processes of every region interpretation. To pursue the consequences of a particular instantiation on likelihoods previously in equilibrium, the job set is loaded with the processes of the variables on the related variable list of the instantiated variable.

For efficiency, the processes that compute likelihoods are decomposed hierarchically into elementary s-expressions, each canonically represented by an XDEMON variable. Figure 9 illustrates this decomposition for the likelihood function described in Figure 7. Superexpression variables are placed on the related variable lists of variables representing subexpressions. At the lowest level, subexpression variables representing region relations (e.g., (ABOVE R1 R2)) and region interpretations (e.g., (CHAIRSEAT R1)) become relatives of atomic variables representing regions (e.g., R3) and interpretations (e.g., CHAIRSEAT). The value of an atomic region variable is the list of boundary coordinates of that region. If this value were



NOTE: The likelihood of (CHAIRSEAT R_3) is only recomputed when the likelihood of a supporting interpretation such as (CHAIRLEG R_2) changes or when a spatial relationship such as (ABOVE R_3 R_2) is altered by resegmentation.

SA-1530-60

FIGURE 9 HIERARCHICAL DECOMPOSITION OF PROCESS REPRESENTING (CHAIRSEAT R_3)

altered (e.g., by merging or splitting the region), then all region relations and region interpretation variables in which the altered region variable appeared would be reevaluated.

The above decomposition increases efficiency by terminating reevaluation at the lowest level subexpression whose value is unchanged by a triggering event. Suppose, for example, that the current strength of the region relation (ABOVE R3 R2) in Figure 9 was 0.4 and that the likelihood of region interpretation (CHAIRLEG R2) was 0.5. Assuming fuzzy logic, the superexpression (AND* (ABOVE R3 R2)(CHAIRLEG R2)) in the lower left corner of Figure 9 would then evaluate to 0.4 (i.e., the minimum of 0.4 and 0.5). A jump in the likelihood of (CHAIRLEG R2) from 0.5 to 0.6 would trigger reevaluation of the above superexpression. However, its value would be unchanged and the reevaluation process would immediately terminate. The canonical representation of subexpressions in the decomposition further minimizes redundant computation in cases where a subexpression is common to several superexpressions. In Figure 8, for example, the same support term for the constraint (HOMOGENEOUS WALL) appears at lines 45 and 72 in the likelihood functions of the interpretations (WALL LWALL) and (WALL RWALL), respectively. The same XDEMON variable represents this term in the decomposed likelihood functions of both interpretations. This variable is reevaluated only once when the likelihood of a supporting region interpretation e.g., (WALL DR) changes.

The alternative interpretations of a given region are associated by canonic variables of the form (OPTION (CHAIRBACK R4)(WALL R4)), which evaluates to the current number of interpretations of the region whose likelihood exceeds 0.1. Option variables are useful in situations where the competing interpretations must be manipulated as a set, such as constraints requiring that two regions take the same (unspecified) interpretation or normalizations requiring that the likelihoods of alternative interpretations sum to 1.0. Since option variables are reevaluated when any interpretation likelihood of their associated region is altered, many important administrative details of a search can be handled efficiently as side effects of the process (see Section III.D). The AND/OR/OPTION structure of the resulting global data base (Figure 8) is remarkably similar to that adopted independently for the HEARSAY-II Speech Understanding System.[14]

D. Search

Occasionally, the relaxation process will terminate with one highly likely interpretation for each region. More often, the interpretation of one or more regions remains ambiguous. A search is then needed to determine which of the possible interpretations maximize the combined likelihood of the whole scene.

The search algorithm outlined in Figure 3 is implemented using a general state-saving mechanism that allows a current computational context to be reinstated at a future time. The search context for M* consists of the complete network of variables described in Section III-C plus additional "state" variables that characterize each search state. These include a score, a list of previous region instantiations, and a priority queue of instantiations yet to be tried (IQUEUE). Search states in various stages of instantiation are inserted onto a priority queue of states (SQUEUE), ordered by score. A search proceeds by reinstating the highest scoring state on SQUEUE, selecting the best instantiation from the current IQUEUE, then reevaluating the network of variables in the new context created by that instantiation. An acceptable solution terminates the search. Otherwise, IQUEUE is updated and the resulting state is added to SQUEUE. The search then continues in the current highest scoring state.

IQUEUE contains the OPTION variables for all ambiguously interpreted regions remaining in a state, ordered by the likelihood of their most probable interpretation (highest first). An interpretation is hypothesized (Step 2 in Figure 3) by popping the top OPTION variable from IQUEUE and instantiating the corresponding region to its most probable interpretation. This best first instantiation policy was borrowed from conventional A* type search algorithms and proved adequate for our simple experiments. However, the choice of instantiation in a constraint satisfaction algorithm such as M* should also take into account the likelihood that the instantiation will lead to a quick contradiction, thereby allowing abandonment of that branch of the search tree. The likelihood of achieving a contradiction depends on factors, such as the number of alternative interpretations of the region, the number of interpretations directly constrained by the instantiated interpretation, the ambiguity of the regions associated with

those directly constrained interpretations, and the number of interpretations they in turn constrain. Clearly, finding the optimal instantiation is another major search problem, termed by Montanari "the secondary optimization" problem.[15]

Two search states are established in Step 3 to explore the consequences of both asserting and denying the interpretation hypothesis selected in Step 2. A region interpretation is denied by setting its local and current likelihoods to 0.0 and removing it from the relative lists of other variables.* An interpretation is asserted by denying the alternative interpretations for that region.+ The reason for hypothesizing a single interpretation, rather than, for example, splitting the set of possible interpretations of a region into approximately equal subsets, was to maximize the likelihood of forcing a contradiction.

The evaluation of each branch in the search (Steps 4 through 6) is handled as in the uninstantiated top level state, by executing a job list. The job list initially contains the set of variables directly related to variables whose likelihoods were altered in instantiating that branch. This initial set includes the OPTION variable of the instantiated region. Additional jobs and option variables are added dynamically as a consequence of reevaluating these initial variables. Processing terminates with an updated set of equilibrium likelihoods and updated values of the state variables SCORE and IQUEUE.

The termination tests in Step 5 and the updating of variables SCORE and IQUEUE are both accomplished as side effects of reevaluating OPTION variables. Specifically, the score is incremented by subtracting the previous best

---

*Setting local likelihood to zero permanently pins an interpretation's overall likelihood to zero. Removing that variable from relative lists thus avoids unnecessary reevaluations. Any variables thereby left with no relatives are themselves removed from relative lists of other variables so that variable reevaluations will occur only when a new value of the variable might be utilized. Note that the relative lists of region interpretation variables always include an OPTION variable and thus, such variables are not disconnected except when directly pinned to zero.

+In early experiments, the current likelihood of the asserted interpretation was boosted to 1.0 to excite the equilibrium process. This artificial stimulus proved unnecessary and was abandoned in later experiments.

interpretation likelihood of the region (stored on the property list of the OPTION variable) and adding the current best likelihood. If more than one interpretation of the region has a likelihood exceeding 0.1, a pointer to the OPTION variable is inserted onto IQUEUE in a position determined by the current likelihood of its best interpretation. If the likelihoods of all possible interpretations of a region drop below 0.1 the search state is assumed to be inconsistent and is abandoned. A solution (i.e., a consistent set of unique interpretations) is indicated if after evaluating all OPTION variables, no contradictions have been detected and the IQUEUE of that state is empty.

Updating state variables is an efficient process, because OPTION variables are added to the job list only when the likelihood of an associated region interpretation has actually been altered. Moreover, OPTION reevaluations, unlike other jobs, are added to the end of the job list so that evaluation occurs only once in each search state using the final equilibrium likelihoods.

The nature of the above search is determined by the functions used to update SQUEUE and IQUEUE and by the termination condition. Alternative search strategies are easily instituted. A depth first search, for example, is obtained by always adding new search states to the front of SQUEUE, while a breadth first search is realized by always adding them to the end.[16] Heuristic guidance can be introduced into the search by the function that updates IQUEUE. The termination condition can be chosen to select the highest scoring solution (i.e., completely instantiated terminal state), the first solution obtained (which, with arbitrary constraints, is not guaranteed optimal) or a complete enumeration of all consistent solutions.

E. Using MSYS

All experiments with MSYS have, to date, been performed on manually partitioned scenes. An interpretation problem is posed in the following way. First, the experimenter, using a trackball, circles, and names a set of test regions on the displayed image of a scene (see, for example, Figure 4). Next he enters the constraints to be used in the current experiment. He may also directly assert symbolic relationships among regions (e.g., that two regions be considered adjacent). This ability was useful for simulating unimplemented relational procedures. Interpretation is initiated by calling the function INTERPRET with a region file or a list

of regions as an argument. MSYS responds with a complete protocol of the interpretation process containing, first, a list of locally possible region interpretations and their initial a priori likelihoods, second, a trace of all jobs executed from the job list, and third, a final list of unique region interpretations with associated likelihoods (or else a message announcing failure to find a consistent set of interpretations.) Appendix B contains the complete output protocol for the room scene analysis discussed in Section II.C.

Constraints are entered in the format: (ADDCONST REL VLIST). REL is a simple relation such as (ABOVE CHAIRBACK CHAIRSEAT), a Boolean expression of simple relations, such as (OR (ADJACENT PICTURE WALL)(ADJACENT PICTURE FRAME)), or a functional constraint such as (HOMOGENEOUS DOOR). VLIST is a list of the interpretations to which the constraint applies. Thus, (ADDCONST (ADJACENT PICTURE WALL)(PICTURE)) requires all pictures to be adjacent to WALLS but puts no constraint on WALLS. If VLIST is omitted, MSYS assumes that the constraint applies mutually to all interpretations mentioned within it. Constraints on the same interpretation specified in different ADDCONST statements are embedded in an implicit conjunction. Figure 10 illustrates the actual format used for specifying the constraints in the example of Section II.C.

Symbolic region relations are asserted using the XDEMON SETVAL function (see Appendix D), which creates a corresponding data base variable and sets its likelihood to a desired value. For example, the fact that region R1 is adjacent to region R2 could be asserted with certainty by executing (SETVAL (ADJACENT R1 R2) 1.0).

F. Summary of Experimental Results in Room Scene Domain

The results shown in Section II.C were obtained with set theoretic logic. Identical final interpretations were obtained using fuzzy logic. With fuzzy logic, the relaxation process converged much more rapidly in every state because the values of disjunctions (conjunctions) could change only when the value of their strongest (weakest) supporting interpretation was altered. This advantage was offset by a reduction in quality of the resulting likelihood estimates, and necessitated backtracking during search. In particular, (WALL DR), an incorrect interpretation, had the highest overall

```
(ADDCONST (QUOTE (NOT* (ADJ DOOR PICTURE))))
(ADDCONST (QUOTE (FUNCTION ROOMPART))
          (QUOTE (DOOR WALL)))
(ADDCONST (QUOTE (FUNCTION HOMO))
          (QUOTE DOOR))
(ADDCONST (QUOTE (FUNCTION HOMO))
          (QUOTE WALL))
(ADDCONST (QUOTE (FUNCTION HOMO))
          (QUOTE CBACK))
```

FIGURE 10 SPECIFICATION OF RELATIONAL CONSTRAINTS GIVEN IN FIGURE 2

likelihood in the initial equilibrium state, and was, therefore, chosen as the first candidate for instantiation. Fortunately, the search was side-tracked only briefly because a higher global score was obtained in the competing context where (WALL DR) was denied. Thereafter, the search went directly to the solution obtained in Figure 6, and with significantly less propagation. A related experiment was performed using set theoretic logic with normalized likelihoods that summed to 1.0 over all possible interpretations of a region. This normalization affected neither the order of instantiation nor the final solution. However, it introduced many additional oscillations into the relaxation process, since the likelihoods of all interpretations of a region had to be readjusted whenever any was reevaluated. Normalization was thus rejected as unnecessary and inefficient. The overall conclusion was that any reasonable rules for combining evidence would probably suffice.

IV DISCUSSION

In this section, we examine our work in the general context of knowledge-based search and then suggest some applications.

A. M^* as a Heuristic Search Algorithm

M^* is basically a conventional heuristic search algorithm that employs a novel evaluation function to guide search. A relaxation method is invoked after each instantiation to obtain consistent likelihood estimates for the remaining interpretation possibilities of each region. The evaluation function forms an estimate of the highest scoring terminal state reachable from the current (partially instantiated) state, by summing the likelihoods of the highest scoring interpretation possibility of each region. The search then proceeds by returning to the state with the highest evaluation, and instantiating next, the highest likelihood interpretation of an uninstantiated region.

It is well known that the effectiveness of a generate-and-test search is improved, sometimes substantially, when problem constraints are used for guiding generation as well as for testing. With deterministic constraints, it may be possible to eliminate certain variable assignments from consideration and thereby reduce the branching of the search. For example, if it can be shown that a variable assignment is inconsistent with all possible assignments of another variable, then all cases involving that assignment need not be generated. In highly constrained problems, eliminations may propagate to reduce drastically the set of feasible solutions. Waltz's[17] filtering algorithm provides a well known and dramatic illustration of this phenomenon.

With probablistic constraints, variable assignments cannot be eliminated absolutely. However, they can be ordered preferentially so that a search will find the best solutions first, without having to enumerate cases involving unlikely assignments. The relaxation process in M^* can be viewed as the analog for probablistic constraints, of Waltz's filtering algorithm for deterministic constraints, where instead of eliminating inconsistent

assignments, their likelihoods are depressed. The advantage in both cases, derives from the fact that inconsistencies are detected once, before instantiation, rather than having to be discovered repeatedly on multiple branches of the search tree.

1. The Admissibility and Optimality of M^*

With one reasonable restriction, it can be formally proved that M^* is admissible (i.e., that it will achieve the highest scoring solution for a given set of constraints.)[+] This restriction requires that the likelihoods of interpretations be based solely on supportive constraints, e.g., (ADJACENT PICTURE WALL), as opposed to contraindicating constraints, e.g., (NOT (ADJACENT PICTURE DOOR)). This restriction guarantees that interpretation likelihoods, computed as per Section III.B, will be overestimates of their respective likelihoods in any terminal state. Hence, the evaluation function, which is formed by summing the overestimated likelihoods, will be an upper bound on the score of the best terminal state reachable from a given partially instantiated state. Moreover, the computed upper bound cannot be increased by additional instantiations. Under these conditions, M^* will perform the equivalent of an A^* search and can be formally proved to be admissible. Appendix A sketches the proof which is based on this analogy with the admissible A^* algorithm. Continuing the A^* analogy, M^* is also optimal in the sense that no other admissible algorithm, using an evaluation function that is a weaker upper bound, can reach the optimal solution through fewer partially instantiated states. We return to this point in Section IV.A.3, where M^* is compared with other search algorithms.

The restriction to constraints that are supportive rather than contradictory is not a serious one. A contraindicating constraint can, in principle, always be expressed by a set of supportive constraints, that explicitly enumerate the allowed possibilities. For example, the constraint (NOT (ADJACENT PICTURE DOOR)) is equivalent to a set of constraints of the form (ADJACENT PICTURE WALL), (ADJACENT PICTURE FRAME), and (ADJACENT PICTURE PICTURE) enumerating every region interpretation that can be legally

---

[+] Whether or not the highest scoring solution corresponds to the correct interpretation of the scene depends, of course, on the sufficiency of the constraints.

adjacent to PICTURE. If this set is too large to enumerate, a practical alternative, still satisfying the requirements of admissibility, would be a procedural constraint on the interpretation PICTURE, satisfied by the highest scoring interpretation, other than DOOR of each adjacent region. We therefore conclude that the restriction to supportive constraints is, at worst, a practical limitation affecting the parsimony of constraint expression.

The efficiency of a search (measured by number of instantiations) can sometimes actually be improved by sacrificing formal admissibility. Experience with strong contraindicative constraints has shown that erroneous instantiations usually produce drastic reductions in score and consequent early abandonment of false search paths. Thus, even with nonsupportive constraints, M^* is still an effective heuristic search algorithm. If a truly optimal solution is essential, the M^* algorithm can be modified so that solutions are enumerated exhaustively, in order of merit, subject to prior pruning by a branch and bound test.

2. Stability and Convergence of the Relaxation Process

The M^* algorithm presumes that the relaxation process will converge to a stable set of equilibrium likelihood values following each stage of instantiation. However, except in a few special cases, almost nothing can yet be said regarding formal criteria for guaranteeing this convergence. The constraint restrictions imposed for admissibility obviously preclude the relaxation process from oscillating (since likelihoods can only decrease) but they do not preclude monotonic decay toward a set of likelihoods, all of which are zero. Stability proofs have been formulated in related work for the special case of Boolean supportive constraints[17-18] and for the case where the likelihood of an interpretation is either a fuzzy function (composed of minimums and maximums) or a normalized sum of the interpretation likelihoods of other regions [19]. The substance of these proofs rests on establishing that the range of possible likelihood values for each interpretation--initially the interval (0,1)--can never diverge on any iteration.

The above convergence proofs concern constraints that were applied

homogeneously to all problem variables. Convergence behavior for the arbitrary, procedurally represented constraints allowed in M* can for now be established only empirically. In practice, we are not concerned with actual convergence (since relaxation can always be arbitrarily terminated), but only with the asymptotic behavior of the likelihoods after a reasonable number of iterations. As Rosenfeld [19] pointed out, the relaxation process can be thought of as a process for "enhancing" the likelihoods of correct interpretations. Our experiments involved constraints chosen without specific regard for convergence and in all cases the relaxation process had the desired effect of enhancing the likelihoods of the correct interpretations. These specific examples tend to support the intuition that in highly constrained problems, the correct interpretations will assert themselves via the relaxation process.

3. Relation to Other Search Algorithms

Many previous scene interpretation algorithms can be viewed as special cases of M*. Waltz's algorithm, for example, is equivalent to performing M* with deterministic supportive constraints, and a priori likelihoods of unity for all interpretation possibilities. With these restrictions (which, incidentally, satisfy the requirements for an admissible A* search), propagation is limited to cases where an interpretation's likelihood can be deduced to be zero. Moreover, since the likelihoods of possible interpretations remain equal (namely 1.0), they cannot be used in selecting which interpretations to instantiate.

The tree searching algorithms of Duda[20], Guzman[21], and Yakimovsky[7] can also be viewed as special cases of M*. Each is equivalent to performing M* with restrictions on the extent to which the consequences of likelihood reevaluations are propagated. With supportive constraints, these restrictions on propagation only raise likelihoods that are already overestimates. Therefore, invoking the A* analogy developed in Section V.A.1, these algorithms will perform searches that are less directed (as measured by the number of partially instantiated states examined) than will M*, given comparable constraints.

The algorithms of Duda and Yakimovsky illustrate the drawbacks resulting from restrictions on the use of constraints. Duda's algorithm,

performed an A* tree search, but, unlike M*, based its evaluation of search states and its selection of instantiations entirely on the unconstrained a priori likelihoods of interpretations. Deterministic constraints, which specified legally adjacent interpretations, were used only after instantiations to check whether the assigned interpretation was consistent with interpretations previously assigned to adjacent regions.

This limited use of constraints frequently necessitated redundant deductions. For example, a region could get instantiated to an interpretation that was inconsistent with all possible interpretations of an adjacent region and that inconsistency would not be discovered until each of those interpretations had been individually instantiated. Any intervening instantiations of a third region would simply be wasted work. M*, by contrast, would have avoided instantiating the inconsistent interpretation in the first place.

Yakimovsky's algorithm performed an exhaustive depth first enumeration of possible region assignments; interpretations were assigned to regions in order of maximum likelihood and a branch and bound technique was used to prune unpromising branches. When an interpretation was instantiated, its likelihood was frozen and that value was used to update the likelihoods of interpretations that were directly constrained and not yet instantiated. Unlike M*, Yakimovsky's algorithm did not use uninstantiated interpretations to update a priori likelihoods, did not update the likelihoods of interpretations after instantiation, and did not propagate the consequences of a likelihood reevaluation beyond the interpretations directly constrained to an instantiated interpretation. For these reasons, Yakimovsky's likelihood estimates are less informed (i.e., greater upper-bounds) than those that M* would derive from the same knowledge. Moreover, because the likelihoods of interpretations were arbitrarily frozen at the time of instantiation, the terminal likelihoods in each completely instantiated state represent only an approximate solution to the constraints. Conceivably, the algorithm could thus converge on a false optimum.

4. Cost-Effectiveness

The number of instantiations is, of course, only one measure of search effectiveness. The computational effort expended in the relaxation

process must be considered in assessing whether the M^* algorithm actually achieves a cost-effective reduction in search.

The cost-effectiveness of M^* rests, intuitively, on many problem dependent factors, including the types of constraints, the representation chosen to express those constraints in MSYS, and the connectedness of the resulting constraint network. The types of constraints bear on the difficulty of constructing MSYS problem representations and on the cost-effectiveness of the constraint in reducing search. As an illustration of the trade-offs, it is easier to create an MSYS representation for a homogeneity constraint that applies only to adjacent regions admitting the constrained interpretation than for one that applies to every region in the scene admitting that interpretation. However, a local homogeneity constraint would be less effective in detecting global contradictions.

The choice of representation determines size of the search space, selectivity of constraint propagation, and cost of constraint execution. These criteria are often contradictory. For example, a representation based on sets of possible interpretations for each region naturally allows very efficient execution of constraints that operate on the entire set (e.g., constraints that intersect the possible interpretations of adjacent regions). On the other hand, a set representation hinders the selective propagation of constraints on individual interpretations. The connectedness of the constraint network affects how far the consequences of a likelihood reevaluation can propagate.

Although the above factors appear important, we have so far been unable to formulate crisp criteria for predicting whether M^* will prove cost-effective for a particular problem. The computational complexity of such a determination may be of the same order of difficulty as solving the original problem. Our guess is that in scene domains with sparse local constraints, the limited lookahead employed by Duda and Yakimovsky will be more cost-effective than the global approach of M^*. On the other hand, shallow lookahead is of little value in situations characterized by a dense network of highly interacting constraints. The line-drawing domain chosen by Waltz is a perfect example of this latter class wherein almost all of the problem reduction was accomplished not by search but by globally propagating the effects of local constraints.

5. Experimental Comparison

All of the algorithms described above can be emulated in MSYS by making minor modifications to the constraint propagation mechanism. Such emulation provides a fair basis for experimental comparisons of performance. Actual emulations of the Waltz and Duda algorithms have been performed and are described in Appendices E and F. These emulations showed that, at least for one particular set of constraints, the Waltz and M* algorithms were both more effective than Duda's algorithm. A systematic comparison of all the algorithms on significantly more complex scenes is planned for the future.

B. MSYS as a System Organization for Knowledge Based Search

MSYS, the implementation of M* in XDEMON, has a number of desirable attributes as a system organization for knowledge-based problem solving and perception. All knowledge resides in a global data base that is accessible to all parts of the system. Declarative and procedural knowledge as well as the alternative interpretation hypotheses are represented uniformly by XDEMON variables. Representing competing hypotheses explicitly in the data base where they are freely available has many advantages over hiding them in the internal variables of a backtracking program.

Processes representing individual elements of knowledge can be added or removed dynamically with incremental changes in system performance. Maintenance operations, such as likelihood updating and consistency checking, can be handled directly by activating related variables. This modularity is essential for assimilating new knowledge whether from human experimenters or an automatic learning module. It also allows a system to construct a working data base by drawing relevant constraints from a much larger store of general knowledge as the analysis evolves.

Control propagates throughout the data base in a highly efficient data-directed manner, following the currently most promising lines of deduction. Past decisions are reevaluated only when directly affected by subsequent ones. Strategy and demon processes can be included on the relative lists of interpretation variables to introduce goal direction. Goal direction can also be imposed by prioritizing the job set based on interest and expectation associated with each job.

XDEMON is a practical realization of many philosophical objectives expressed in recent artificial intelligence literature, notably uniformity of representation, modularity, and distributed control.[22-25] Similar system organizations have been previously reported in connection with work on speech understanding,[14] simulation,[26] and parallel computation.[27] Our major contribution was a demonstration of how a data driven system organization could be used effectively in performing an optimizing search.

The asynchronous interaction of knowledge processes in XDEMON suggests interesting possibilities for parallel implementation. In an extreme example, every XDEMON variable could be represented by an asynchronous microprocessor that computes new values whenever the value of subordinate variables change. The same restrictions that guarantee convergence of the M^* relaxation process, namely, that with supportive constraints, the range of likelihood values for an interpretation cannot diverge, also guarantees that this parallel implementation will be free of race hazards. The primary technical difficulty in such an implementation would be the reconfiguration of interconnections among the microprocessors needed to accommodate different constraint problems.

C. Applications to Scene Analysis

Scene analysis is the combined process of partitioning a scene into regions corresponding to meaningful entities and correctly interpreting those regions. Formally stated, the objective is to maximize the joint likelihood that region i has interpretation j over all partitions of the scene into regions and all assignments of interpretations to regions. Although partitioning logically precedes interpretation, these two processes must be tightly integrated in order to achieve the formal objective.

A recent paper by Tenenbaum and Barrow[28] presents a way of using an interpretation mechanism such as MSYS to guide segmentation. In this approach, knowledge from a variety of sources is used to make inferences about the interpretations of regions, and regions are merged in accordance with their possible interpretations. A scene is first partitioned into elementary regions consisting of individual pixels or, perhaps, groups of adjacent pixels with identical attributes. Beginning with this partition, the system first performs the most complete interpretation possible in the current partition. Based on this interpretation, it next merges a

pair of adjacent regions that are least likely to represent distinct objects. The process then iterates by revising the interpretation to fit the current partition and performing another merge. As the partition develops, region boundaries approach actual object boundaries, allowing interpretation to be refined.

In the present implementation, the deduction of region interpretations is performed using a limited but fast version of MSYS, coded in Fortran, which allows only deterministic constraints between adjacent regions. Constraints are expressed in tabular formats that specify, for each possible region interpretation, the allowed interpretations for an adjacent region in a given relationship (e.g., ABOVE, BESIDE, and INSIDE). Deduction proceeds as in Waltz's algorithm by eliminating possible region interpretations that are not consistent with any possible interpretation of an adjacent region. A second implementation is planned providing the full generality of MSYS, real-valued, procedurally represented constraints among arbitrary regions.

Virtually all of the scene interpretation research performed to date has involved narrow semantic domains with specific expectations (e.g., each region was known to be one of a dozen or so possible objects). While not arguing against the use of expectations when available, it is important to ask whether the MSYS approach can be generalized to work in real world domains containing millions of objects and millions of relations. Clearly, a more concise initial symbolic description at a general level is required, perhaps in terms of surface characteristics such as curvature (planar, convex, and concave), orientation (vertical, horizontal), texture and material (e.g., metal, plastic, and wood) and relations such as occlusion, joining, and support.

The accurate determination of these surface characteristics requires the same type of global reasoning used to deduce object interpretations but is based on general knowledge about shadows, illumination sources, relative depth, occlusion, surface orientation, texture gradients, and so forth, common to many domains. For example, the perceived color of a region depends primarily on its spectral reflectance characteristics but also upon the incident illumination, which in turn is affected by the

spectral characteristics and orientations of nearby surfaces. Consequently, the accurate color interpretation of one region may require the simultaneous global interpretation of color and orientation over all regions. Similarly, relative depth, in the absence of range data, must be inferred from the global consistency of partial depth orderings established by local cues, such as T-joints and texture gradients. Once the scene has been analyzed in general terms that analysis can be used to guide a more detailed specific analysis.

D. Applications to Problem Solving

The M^* algorithm and its implementation in XDEMON are useful in other areas of artificial intelligence. Scene interpretation typifies a broad class of problems in which values must be assigned to variables subject to constraints among the variables. This class encompasses symbolic constraint satisfaction problems, such as language parsing, line-drawing interpretation, and certain puzzles (e.g., Cryptarithmetic, Instant Insanity, Fifteen Puzzle) as well as constraint optimization problems such as diagnosis, data interpretation (e.g., mineral exploration) and many design tasks (e.g., architectural layout).

We, and others, have previously recognized the utility of global propagation techniques in simplifying symbolic constraint satisfaction problems.[4,29-31] These problems often confound conventional heuristic search algorithms because of large search spaces, which are not significantly reduced by individual instantiations. Moreover, there is usually little heuristic guidance available for choosing hypotheses. An excellent example of a problem with these characteristics is given in Appendix G along with a sketch of how MSYS was configured to solve it.

Our work extends the use of propagation techniques to constraint optimization problems opening up a wide range of real-world applications. These problems in common with perception, are characterized by the added complexities of noisy data, probablistic constraints, and multiple solutions of varying utility. M^* seems especially well suited for cooperative (man-machine) problem solving in tasks such as diagnosis; the equilibrium process can dynamically adjust the relative likelihoods of competing hypotheses to reflect new constraints, evidence, and hypotheses that may be interjected by the man at any time.

The related variables concept underlying XDEMON has additional applications in planning and problem solving besides constraint propagation. As one example, consider the problem of determining a cost-effective execution path through an AND-OR planning graph. The utilities of subgoals are interdependent and may vary with knowledge acquired during the course of execution making detailed elaboration of alternative execution sequences unwarranted. Garvey[13] describes an approach, first implemented in XDEMON, in which the utilities of interacting subgoals are relaxed to equilibrium to determine the next best think to do. If execution of the selected subgoal revises the utilities of remaining subgoals, the relaxation process can be repeated to update priorities before selecting the subsequent step. This incremental approach is particularly appropriate for information gathering strategies where incomplete knowledge is inherent at each step of the plan.

XDEMON can also serve as a simple event driven simulation language for establishing interacting cause and effect relationships in an analogue manner. J. S. Brown[32] has remarked that simulation is most appropriate as an inference technique in those situations where conventional "linear" deductive reasoning breaks down; i.e., where the consequences of an action lead to complex side effects including feedback type interactions that can alter the state on which a deduction was initiated. Such a situation motivated Brown's use of simulation to deduce the consequences of faults introduced into an electrical network. The interaction among circuit variables in a network resembles the interaction among interpretation likelihoods in a partitioned scene, which may account for resemblences among the relaxation process in MSYS and a simulation program.

The versatility of XDEMON suggests the inclusion of a similar subroutine package in artificial intelligence languages such as QLISP.[22]

E. Summary

Scene interpretation typifies a broad class of problem-solving tasks involving the assignment of values to variables that are mutually constrained. A general constraint optimization algorithm, M^* has been presented that for sufficiently constrained problems is more powerful than conventional heuristic search methods, many of which can be treated as special cases of this algorithm.

The algorithm is based on the notion of representing alternative hypotheses and constraints as (simulated) asynchronous parallel processes. These processes interact in a dynamic equilibrium that establishes the relative likelihoods of competing hypotheses. The equilibrium process serves as a new kind of global lookahead that improves the order of instantiation and context selection at each stage of a best first search.

The admissibility and stability of the M^* algorithm have been formally proved for a restricted class of constraints as has its optimality compared with conventional search algorithms, measured by number of instantiations. Open theoretical issues include a precise characterization of the class of "sufficiently constrained" problems for which the method is computationally cost-effective and explicit criteria for selecting the best problem representation, constraints, and instantiation order.

The algorithm has been applied successfully to both scene interpretation problems and constraint satisfaction puzzles. However, the scene interpretation experiments should be regarded as inconclusive, because of the simplicity of the test scene and the reliance on simulated range data for assigning the initial interpretations. Further experiments in more complex scenes are planned using an actual laser range finder. Additional experiments will be performed without range data, using constraints that infer height and orientation from pictorial cues such as image height and shadows.

Appendix A

THE ADMISSIBILITY AND OPTIMALITY OF M^* +

A. Introduction

In this appendix, we establish the admissibility and optimality of algorithm M^*, when used with supportive constraints. We will first establish conditions of admissibility and optimality for an abstract tree searching algorithm A^+ that encompasses M^* and then show that with supportive constraints, M^* fulfils those conditions.

B. Definitions

We consider the following search problem: Let Tr(s) be a finite tree with root node s. For each node n in Tr(s), let Tr(n) be the sub-tree with root node at n. Each tip node, r, of Tr(s) has a value v(r). Let r^*, with value v^*, be called a best tip node if no other tip node has a larger value. For non-tip nodes, n, let v(n) be the value of the best tip node in Tr(n). We note that for all n in Tr(s), $v(n) \leq v^*$.

We are concerned with search processes for finding a best tip node in Tr(s). This problem is analogous to that of finding a least costly path in a tree. For the latter problem there is a search algorithm, A^*, for which admissibility and optimality theorems have been proved.[5] Here we prove analogous theorems for an analogous algorithm, A^+, that finds best tip nodes.

C. Algorithm A^+

Let $\hat{v}$ be an estimating function for estimating the value of nodes in Tr(s). That is, $\hat{v}(n)$ is an estimate of the value, v(n), of the best tip node in Tr(n). Search algorithm A^+ is defined as follows:

1. Put node s on OPEN.
2. Select that node, n, on OPEN with the largest value of $\hat{v}$. Resolve ties arbitrarily but always in favor of tip nodes.
3. If node n is a tip node, terminate; else continue.
4. Expand node n by putting its successors on OPEN. Remove node n from OPEN.
5. Go to 2.

+The collaboration of Nils Nilsson in preparing this appendix is gratefully acknowledged.

With the assumption that $\hat{v}(n) \geq v(n)$ for all n in Tr(s), and that $\hat{v}(r) = v(r)$ for all tip nodes, r, we make the following observations:

A) At any time prior to termination of A^+, a best tip node in Tr(s) is in the Tr(n) for some node n on OPEN. For this node n, $\hat{v}(n) \geq v(n) = v^*$.

B) A^+ never re-opens nodes, because it is searching a tree.

C) Since Tr(s) is finite, A^+ must terminate. We note that it can only terminate by selecting a tip node in step 3.

D. Admissibility of A^+

We say that an algorithm is admissible if it terminates by finding a best tip node.

Theorem 1: If $\hat{v}(n) \geq v(n)$ for all nodes n, and $\hat{v}(r) = v(r)$ for tip nodes r, then A^+ is admissible.

Proof: Assume the contrary. That is, assume that A^+ terminates for some tip node r′ that is not best, i.e. for which $v(r') = \hat{v}(r') < v^*$. But then just before selecting r′, by observation A above, there was a node n on OPEN with $\hat{v}(n) \geq v^*$ contradicting our assumption that A^+ chose r′ instead of n.

E. Optimality of A^+

First we prove a lemma.

Lemma: If A^+ expands a non-tip node, n, and if $\hat{v}(n) \geq v(n)$ for all n, then $\hat{v}(n) \geq v^*$.

Proof: By observation A, at the time A^+ expanded n, there existed on OPEN a node n′ with $\hat{v}(n') \geq v^*$. If n′ = n, $\hat{v}(n) \geq v^*$. Otherwise, A^* chose n in preference to n′ and $\hat{v}(n) \geq \hat{v}(n') \geq v^*$.

Theorem 2: Let two versions of A^+, namely A_1^+ and A_2^+, search a tree, Tr(s), to termination using estimating functions $\hat{v}_1$ and $\hat{v}_2$, respectively. Then, if $\hat{v}_2(n) > \hat{v}_1(n) \geq v(n)$ for all non-tip nodes n in Tr(s), and if $\hat{v}_2(r) = \hat{v}_1(r) = v(r)$ for all tip nodes in Tr(s), A_2^+ will expand all the non-tip nodes expanded by A_1^+.

Proof: Suppose the contrary. Let $Tr_1(s)$ and $Tr_2(s)$ be the trees of nodes expanded by A_1^+ and A_2^+, respectively. If $Tr_1(s)$ contains a non-tip node not contained in $Tr_2(s)$, then (since both are rooted in s) there will

exist a non-tip node, n, in $Tr_1(s)$, not in $Tr_2(s)$, with a parent in both. That is, at the termination of A_2^+, node n is on the OPEN list of A_2^+.

But if A_2^+ didn't expand node n, it must have been because $v^* \geq \hat{v}_2(n)$. Also, since A_1^+ did expand, n we have by the Lemma that

$$\hat{v}_1(n) \geq v^*.$$

But these relations contradict our assumption that for non-tip nodes $\hat{v}_2(n) > \hat{v}_1(n)$.

F. Discussion

Theorem 1 states that as long as $\hat{v}$ is an upper bound, algorithm A^+ is admissible. Clearly, the search can be safely terminated when a terminal node is encountered whose actual score is higher than upper bounds on the scores of all terminal nodes that are reachable from any open node. Theorem 2 states that A^+ is optimal in the sense that it will never expand more nodes than any other admissible algorithm that relies on score estimates that are strictly larger than the upper bound estimates used by A^+. It now remains to be shown that algorithm M^* is an instance of the more general algorithm A^+ and that with supportive constraints, M^* fulfills the conditions required for proving Theorems 1 and 2.

M^* like A^+ is an algorithm for finding the highest scoring terminal node in a tree. In scene analysis, the nodes of the tree represent states of instantiation that are reached by pinning the likelihoods of particular region interpretations to zero. Nodes representing a state where every region has a unique interpretation are designated terminal nodes. Two arcs emanate from each non-terminal node representing the assertion and denial of the most likely interpretation of a previously uninstantiated region.

M^* selects for expansion the node with the highest valued score. This score is formed by summing the highest interpretation likelihood associated with each region. Since these likelihoods are in fact upper bound estimates, the score of a non-terminal node is an upper bound on the score of any terminal node reachable by further instantiation. The restriction to supportive constraints means that interpretation likelihoods will never

increase as a consequence of decreasing the likelihood of another interpretation. In particular, since instantiations are accomplished by setting the likelihood of alternative interpretations to zero, they cannot raise the likelihood of any uninstantiated interpretation. Consequently, the score of a node is a true upper bound on the combined interpretation likelihoods at any terminal node accessible from that node. This establishes the admissibility of M^*. The optimality of M^* follows from the fact that the score of a node is at least as tight a bound as one based on fewer supportive constraints or restricted propagation. Both of these factors can only raise individual likelihood estimates, thereby, loosening the upper bound on score. The tighter the bound on score, the more directed the search.

Appendix B

DETAILED TRACE OF AN INTERPRETATION

This appendix contains a detailed trace of the interpretation process for the vertical regions in Figure 4 as described in Sections II.C and III. Jobs executed from the job queue are preceded by the designations JOB or BG. BG stands for background jobs that were either placed on the queue to initialize processing (e.g., NETSETUP jobs) or added to the end of the queue to be processed last (e.g., OPTION variables). All other reevaluations are added dynamically to the front of the queue so that current lines of deduction are pursued first.

The real valued numbers following each job are updated likelihoods of the corresponding XDEMON variables produced by running the job.

A. Initialization Phase

Lines 1 through 83 of the trace (Figure B-1) constitute the initialization phase during which the constraint network is constructed and equilibrium likelihoods are obtained. Each NETSETUP job retrieves the constraints associated with an interpretation and then constructs and executes a global evaluation function that computes support for those constraints in the current scene partition.

The tag (0.0-setup) indicates an interpretation that was rejected outrightly because of an unsupported constraint. The likelihood of such interpretations are permanently set to zero and no evaluation function is compiled for them. In line 8, for example, the interpretation (DOOR CBACK) was rejected by the constraint ROOMPARTITION. The resultant drop in likelihood of (DOOR CBACK) triggered reevaluation of the likelihood functions for (DOOR LWALL) at line 13 and (DOOR RWALL) at line 16 which had been previously evaluated in lines 4 and 6, respectively, using the a priori likelihood of (DOOR CBACK). (In effect, decreasing the likelihood that a dark region was DOOR increased the likelihood that DOOR was a light region.) Similarly, the likelihoods of (WALL LWALL) and (WALL RWALL) are increased at lines 21 and 22 after rejecting (WALL CBACK) at line 18.

```
;  <TENENBAUM>BAYES.RPT;2   WED 4-FEB-76 1:53PM                    Page 1

00001    BG-(NETSETUP (QUOTE (WASTEBASKET WBSKT))): .12
00002    BG-(NETSETUP (QUOTE (DOOR WBSKT))): 0.0-SETUP
00003    BG-(NETSETUP (QUOTE (WALL WBSKT))): 0.0-SETUP
00004    BG-(NETSETUP (QUOTE (DOOR RWALL))): .0686
00005    BG-(NETSETUP (QUOTE (WALL RWALL))): .0328
00006    BG-(NETSETUP (QUOTE (DOOR LWALL))): .0686
00007    BG-(NETSETUP (QUOTE (WALL LWALL))): .0328
00008    BG-(NETSETUP (QUOTE (DOOR CBACK))): 0.0-SETUP
00009    JOB-((OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT))): .459
00010    JOB-((NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))): .541
00011    JOB-((AND* (NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00012    (NOT* (AND* (ADJ LWALL PIC) (PICTURE PIC))))): .379
00013    JOB-((DOOR LWALL)): .086
00014    JOB-((AND* (NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00015    (NOT* (AND* (ADJ RWALL PIC) (PICTURE PIC))))): .379
00016    JOB-((DOOR RWALL)): .086
00017    BG-(NETSETUP (QUOTE (CHAIRBACK CBACK))): .11
00018    BG-(NETSETUP (QUOTE (WALL CBACK))): 0.0-SETUP
00019    JOB-((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .864
00020    JOB-((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .136
00021    JOB-((WALL LWALL)): .105
00022    JOB-((WALL RWALL)): .105
00023    BG-(NETSETUP (QUOTE (PICTURE PIC))): .251
00024    BG-(NETSETUP (QUOTE (DOOR PIC))): 0.0-SETUP
00025    JOB-((OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT))): .227
00026    JOB-((NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))): .773
00027    JOB-((AND* (NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00028    (NOT* (AND* (ADJ LWALL PIC) (PICTURE PIC))))): .541
00029    JOB-((DOOR LWALL)): .123
00030    JOB-((AND* (ADJ LWALL PIC) (DOOR LWALL))): .123
00031    JOB-((AND* (NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00032    (NOT* (AND* (ADJ RWALL PIC) (PICTURE PIC))))): .541
00033    JOB-((DOOR RWALL)): .123
00034    JOB-((AND* (ADJ RWALL PIC) (DOOR RWALL))): .123
00035    BG-(NETSETUP (QUOTE (WALL PIC))): 0.0-SETUP
00036    JOB-((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .773
00037    JOB-((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .227
00038    JOB-((WALL LWALL)): .175
00039    JOB-((WALL RWALL)): .175
00040    BG-(NETSETUP (QUOTE (DOOR DR))): .175
00041    JOB-((OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT))): .175
00042    JOB-((NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))): .825
00043    JOB-((AND* (NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00044    (NOT* (AND* (ADJ LWALL PIC) (PICTURE PIC))))): .578
00045    JOB-((AND* (NOT* (OR* (DOOR DR) (DOOR PIC) (DOOR CBACK) (DOOR WBSKT)))
00046    (NOT* (AND* (ADJ RWALL PIC) (PICTURE PIC))))): .578
00047    BG-(NETSETUP (QUOTE (WALL DR))): .526
00048    JOB-((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .526
00049    JOB-((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .474
00050    JOB-((WALL LWALL)): .367
00051    JOB-((OR* (WALL LWALL) (WALL RWALL))): .478
```

FIGURE B-1 TRACE OF EXECUTION DURING ESTABLISHMENT OF INITIAL EQUILIBRIUM

```
;  <TENENBAUM>BAYES.RPT;2   WED 4-FEB-76 1:53PM                Page 1:1

00052   JOB=((NOT* (OR* (WALL LWALL) (WALL RWALL)))): .522
00053   JOB=((WALL DR)): .404
00054   JOB=((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .404
00055   JOB=((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .596
00056   JOB=((WALL LWALL)): .461
00057   JOB=((OR* (WALL LWALL) (WALL RWALL))): .556
00058   JOB=((NOT* (OR* (WALL LWALL) (WALL RWALL)))): .444
00059   JOB=((WALL DR)): .344
00060   JOB=((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .344
00061   JOB=((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .656
00062   JOB=((WALL LWALL)): .507
00063   JOB=((WALL RWALL)): .507
00064   JOB=((OR* (WALL LWALL) (WALL RWALL))): .757
00065   JOB=((NOT* (OR* (WALL LWALL) (WALL RWALL)))): .243
00066   JOB=((WALL DR)): .188
00067   JOB=((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .188
00068   JOB=((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .812
00069   JOB=((WALL LWALL)): .628
00070   JOB=((OR* (WALL LWALL) (WALL RWALL))): .817
00071   JOB=((NOT* (OR* (WALL LWALL) (WALL RWALL)))): .183
00072   JOB=((WALL DR)): .142
00073   JOB=((WALL RWALL)): .628
00074   JOB=((OR* (WALL LWALL) (WALL RWALL))): .862
00075   BG=((OPTION (DOOR DR) (WALL DR))): (DOOR DR)=.175
00076   BG=((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC))): (PICTURE PIC)=.251
00077   BG=((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK))): (CHAIRBACK
00078   CBACK)=.11
00079   BG=((OPTION (DOOR LWALL) (WALL LWALL))): (WALL LWALL)=.628
00080   BG=((OPTION (DOOR RWALL) (WALL RWALL))): (WALL RWALL)=.628
00081   BG=((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))): (
00082   WASTEBASKET WBSKT)=.12
00083   SAVESTATE: (TOPCNTXT), SCORE: 1.91, LENGTH(IQUEUE): 6
```

FIGURE B-1 TRACE OF EXECUTION DURING ESTABLISHMENT OF INITIAL EQUILIBRIUM (Concluded)

Actual computation of the equilibrium likelihoods shown in Figure 5 is completed by line 74 following execution of all NETSETUP jobs and consequent reevaluations. This phase is followed by the execution of OPTION variables in lines 75 through 82, which check for contradictions (i.e., a region left without a likely interpretation), increment the global score, and insert pointers to themselves in the priority queue (IQUEUE) governing instantiation. OPTION variables return as a value, the likelihood of the most probable interpretation for their associated region. The name of the interpretation is printed along with this likelihood for experimental convenience. The initialization process is terminated in line 83 by saving the resultant state and XDEMON variables before any instantiation. The search state is summarized by printing the summed likelihoods of the current best interpretation for every region (SCORE) and the number of instantiated regions remaining in that state and represented on IQUEUE.

B. Search Phase

In this phase, MSYS performs a best first search for the set of unique region interpretations with the highest combined likelihood. (See Figure B-2.) The search begins at line 84 by reinstating the highest scoring state from SQUEUE which, by default, is TOPCNTEXT. The most likely interpretation of an uninstantiated region (WALL RWALL) is then instantiated in line 85. The instantiation sets to zero the likelihoods of alternative interpretations of the region RWALL. These adjustments trigger reevaluations (lines 86 through 92) which raise the likelihood of (DOOR DR).

State variables are then updated at line 93 by executing the OPTION variable for region RWALL, the only region experiencing a significant (greater than 0.05) change in an interpretation likelihood. Since interpretation ambiguities remain but no contradictions were detected, the resulting search state is saved on SQUEUE (line 94), labeled by its instantiation history. The alternative state based on denying (WALL RWALL) is evaluated in lines 96 to 112 and placed on SQUEUE (line 113) behind the higher scoring state, (((WALL RWALL) 1.0) TOPCNTXT). This concludes the first stage of search.

```
00084   REINSTATE: (TOPCNTXT)
00085   ASSERT - (WALL RWALL)
00086   JOB-((OR* (DOOR LWALL) (DOOR RWALL))): .123
00087   JOB-((NOT* (OR* (DOOR LWALL) (DOOR RWALL)))): .877
00088   JOB-((AND* (NOT* (OR* (DOOR LWALL) (DOOR RWALL))) (NOT* 0.0))): .877
00089   JOB-((DOOR DR)): .199
00090   JOB-((AND* (ADJ RWALL PIC) (DOOR RWALL))): 0.0
00091   JOB-((OR* (AND* (ADJ RWALL PIC) (DOOR RWALL)) (AND* (ADJ LWALL PIC)
00092   (DOOR LWALL)))): .123
00093   BG-((OPTION (DOOR RWALL) (WALL RWALL))): (WALL RWALL)-.628
00094   SAVESTATE: (((WALL RWALL) . .628) TOPCNTXT), SCORE: 1.91
00095   , LENGTH(IQUEUE): 5
00096   REINSTATE: (TOPCNTXT)
00097   DENY - (WALL RWALL)
00098   JOB-((OR* (WALL LWALL) (WALL RWALL))): .628
00099   JOB-((NOT* (OR* (WALL LWALL) (WALL RWALL)))): .372
00100   JOB-((WALL DR)): .288
00101   JOB-((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .288
00102   JOB-((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .712
00103   JOB-((WALL LWALL)): .551
00104   JOB-((OR* (WALL LWALL) (WALL RWALL))): .551
00105   JOB-((NOT* (OR* (WALL LWALL) (WALL RWALL)))): .449
00106   JOB-((WALL DR)): .347
00107   JOB-((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .347
00108   JOB-((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .653
00109   JOB-((WALL LWALL)): .505
00110   BG-((OPTION (DOOR RWALL) (WALL RWALL))): (DOOR RWALL)-.123
00111   BG-((OPTION (DOOR DR) (WALL DR))): (WALL DR)-.347
00112   BG-((OPTION (DOOR LWALL) (WALL LWALL))): (WALL LWALL)-.505
00113   SAVESTATE: (((WALL RWALL) . 0) TOPCNTXT), SCORE: 1.46, LENGTH(IQUEUE): 6
00114   REINSTATE: (((WALL RWALL) . .628) TOPCNTXT)
00115   ASSERT - (WALL LWALL)
00116   JOB-((OR* (DOOR LWALL) (DOOR RWALL))): 0.0
00117   JOB-((NOT* (OR* (DOOR LWALL) (DOOR RWALL)))): 1.0
00118   JOB-((AND* (NOT* (OR* (DOOR LWALL) (DOOR RWALL))) (NOT* 0.0))): 1.0
00119   JOB-((DOOR DR)): .227
00120   JOB-((AND* (ADJ LWALL PIC) (DOOR LWALL))): 0.0
00121   JOB-((OR* (AND* (ADJ RWALL PIC) (DOOR RWALL)) (AND* (ADJ LWALL PIC)
00122   (DOOR LWALL)))): 0.0
00123   JOB-((NOT* (OR* (AND* (ADJ RWALL PIC) (DOOR RWALL)) (AND* (ADJ LWALL
00124   PIC) (DOOR LWALL))))): 1.0
00125   JOB-((PICTURE PIC)): .3
00126   BG-((OPTION (DOOR LWALL) (WALL LWALL))): (WALL LWALL)-.628
00127   BG-((OPTION (DOOR DR) (WALL DR))): (DOOR DR)-.227
00128   SAVESTATE: (((WALL LWALL) . .628) ((WALL RWALL) . .628) TOPCNTXT)
00129   , SCORE: 1.96, LENGTH(IQUEUE): 4
00130   REINSTATE: (((WALL RWALL) . .628) TOPCNTXT)
00131   DENY - (WALL LWALL)
00132   JOB-((OR* (WALL LWALL) (WALL RWALL))): .628
00133   JOB-((NOT* (OR* (WALL LWALL) (WALL RWALL)))): .372
00134   JOB-((WALL DR)): .288
```

FIGURE B-2 CONTINUATION OF TRACE SHOWING EXECUTION OF SEARCH

```
00135   JOB-((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .288
00136   JOB-((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .712
00137   JOB-((WALL RWALL)): .551
00138   JOB-((OR* (WALL LWALL) (WALL RWALL))): .551
00139   JOB-((NOT* (OR* (WALL LWALL) (WALL RWALL)))): .449
00140   JOB-((WALL DR)): .347
00141   JOB-((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): .347
00142   JOB-((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): .653
00143   JOB-((WALL RWALL)): .505
00144   BG-((OPTION (DOOR LWALL) (WALL LWALL))): (DOOR LWALL)-.123
00145   BG-((OPTION (DOOR DR) (WALL DR))): (WALL DR)-.347
00146   BG-((OPTION (DOOR RWALL) (WALL RWALL))): (WALL RWALL)-.505
00147   SAVESTATE: (((WALL LWALL) . 0) ((WALL RWALL) . .628) TOPCNTXT), SCORE: 1.46
00148   , LENGTH(IQUEUE): 5
00149   REINSTATE: (((WALL LWALL) . .628) ((WALL RWALL) . .628) TOPCNTXT)
00150   ASSERT - (PICTURE PIC)
00151   SAVESTATE: (((PICTURE PIC) . .3) ((WALL LWALL) . .628) ((WALL RWALL)
00152   . .628) TOPCNTXT), SCORE: 1.96, LENGTH(IQUEUE): 3
00153   REINSTATE: (((WALL LWALL) . .628) ((WALL RWALL) . .628) TOPCNTXT)
00154   DENY - (PICTURE PIC)
00155   BG-((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC))):
00156   "CONTRADICTION DETECTED"
00157   REINSTATE: (((PICTURE PIC) . .3) ((WALL LWALL) . .628) ((WALL RWALL)
00158   . .628) TOPCNTXT)
00159   ASSERT - (DOOR DR)
00160   JOB-((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT))): 0.0
00161   JOB-((NOT* (OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))): 1.0
00162   JOB-((WALL LWALL)): .773
00163   JOB-((WALL RWALL)): .773
00164   BG-((OPTION (DOOR DR) (WALL DR))): (DOOR DR)-.227
00165   BG-((OPTION (DOOR LWALL) (WALL LWALL))): (WALL LWALL)-.773
00166   BG-((OPTION (DOOR RWALL) (WALL RWALL))): (WALL RWALL)-.773
00167   SAVESTATE: (((DOOR DR) . .227) ((PICTURE PIC) . .3) ((WALL LWALL)
00168   . .628) ((WALL RWALL) . .628) TOPCNTXT), SCORE: 2.25, LENGTH(IQUEUE): 2
00169   REINSTATE: (((PICTURE PIC) . .3) ((WALL LWALL) . .628) ((WALL RWALL)
00170   . .628) TOPCNTXT)
00171   DENY - (DOOR DR)
00172   BG-((OPTION (DOOR DR) (WALL DR))): (WALL DR)-.142
00173   SAVESTATE: (((DOOR DR) . 0) ((PICTURE PIC) . .3) ((WALL LWALL) . .628)
00174   ((WALL RWALL) . .628) TOPCNTXT), SCORE: 1.88, LENGTH(IQUEUE): 3
00175   REINSTATE: (((DOOR DR) . .227) ((PICTURE PIC) . .3) ((WALL LWALL)
00176   . .628) ((WALL RWALL) . .628) TOPCNTXT)
00177   ASSERT - (WASTEBASKET WBSKT)
00178   SAVESTATE: (((WASTEBASKET WBSKT) . .12) ((DOOR DR) . .227) ((PICTURE
00179   PIC) . .3) ((WALL LWALL) . .628) ((WALL RWALL) . .628) TOPCNTXT)
00180   , SCORE: 2.25, LENGTH(IQUEUE): 1
00181   REINSTATE: (((DOOR DR) . .227) ((PICTURE PIC) . .3) ((WALL LWALL)
00182   . .628) ((WALL RWALL) . .628) TOPCNTXT)
00183   DENY - (WASTEBASKET WBSKT)
00184   BG-((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))):
00185   "CONTRADICTION DETECTED"
00186   REINSTATE: (((WASTEBASKET WBSKT) . .12) ((DOOR DR) . .227) ((PICTURE
00187   PIC) . .3) ((WALL LWALL) . .628) ((WALL RWALL) . .628) TOPCNTXT)
00188   ASSERT - (CHAIRBACK CBACK)
00189   "SUCCESS"
```

FIGURE B-2 CONTINUATION OF TRACE SHOWING EXECUTION OF SEARCH (Concluded)

Subsequent stages of search proceed in an analogous manner, by reinstating the top scoring state from SQUEUE and then exploring the consequences of asserting and denying the highest likelihood interpretation of an uninstantiated region. The next step, for example, explores the consequences of asserting and denying the hypothesis (WALL LWALL) in the restored state (((WALL RWALL) 1.0) TOPCNTXT) (lines 114 to 148). In this example, the search proceeded directly to the desired global scene interpretation, guided by monotonically increasing state scores resulting from a sequence of correct instantiations.

Note the decreased amount of propagation following an instantiation and the detection of global inconsistency (lines 155 and 184) as the search becomes progressively more constrained. The search terminates successfully at line 189 with IQUEUE empty and no region without interpretation. Alternative sets of consistent interpretations, should any exist, could be developed in order of decreasing goodness by continuing to search remaining contexts until SQUEUE was emptied.

C. Final State of Data Base Following Search

The final state of the data base following search is shown in Figure B-3. It differs from the initial equilibrium state (Figure 8) in two ways: First, the likelihoods of incorrect interpretations have been reduced to zero. Second, the relative lists of some variables (e.g., (DOOR DR)) have been pruned by removing other variables whose likelihoods are pinned at zero or whose own relative lists have become empty.

```
00001   VARIABLE: (WASTEBASKET WBSKT)
00002   VALUE:    .12
00003   PROCEDURE:
00004             (WASTEBASKET WBSKT)
00005   RELATIVES:
00006             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00007
00008
00009   VARIABLE: (DOOR WBSKT)
00010   VALUE:    0.0
00011   PROCEDURE:
00012             (DOOR WBSKT)
00013   RELATIVES:
00014             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00015
00016
00017   VARIABLE: (WALL WBSKT)
00018   VALUE:    0.0
00019   PROCEDURE:
00020             (WALL WBSKT)
00021   RELATIVES:
00022             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00023             ((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))
00024
00025
00026   VARIABLE: (DOOR RWALL)
00027   VALUE:    0.0
00028   PROCEDURE:
00029             [AND* .227 (AND* (NOT* (OR* (DOOR DR)
00030                                         (DOOR PIC)
00031                                         (DOOR CBACK)
00032                                         (DOOR WBSKT)))
00033                              (NOT* (AND* (ADJ RWALL PIC)
00034                                          (PICTURE PIC]
00035   RELATIVES:
00036             ((OPTION (DOOR RWALL) (WALL RWALL)))
00037             ((AND* (ADJ RWALL PIC) (DOOR RWALL)))
00038             ((OR* (DOOR LWALL) (DOOR RWALL)))
00039
00040
00041   VARIABLE: (WALL RWALL)
00042   VALUE:    .773
00043   PROCEDURE:
00044             [AND* .773 (NOT* (OR* (WALL DR)
00045                                   (WALL PIC)
00046                                   (WALL CBACK)
00047                                   (WALL WBSKT]
00048   RELATIVES:
00049             ((OPTION (DOOR RWALL) (WALL RWALL)))
00050
00051
```

FIGURE B-3 DATABASE FOLLOWING SEARCH

```
00052   VARIABLE: (DOOR LWALL)
00053   VALUE:    0.0
00054   PROCEDURE:
00055             [AND* .227 (AND* (NOT* (OR* (DOOR DR)
00056                                         (DOOR PIC)
00057                                         (DOOR CBACK)
00058                                         (DOOR WBSKT)))
00059                                (NOT* (AND* (ADJ LWALL PIC)
00060                                            (PICTURE PIC]
00061   RELATIVES:
00062             ((OPTION (DOOR LWALL) (WALL LWALL)))
00063             ((AND* (ADJ LWALL PIC) (DOOR LWALL)))
00064             ((OR* (DOOR LWALL) (DOOR RWALL)))
00065
00066
00067   VARIABLE: (WALL LWALL)
00068   VALUE:    .773
00069   PROCEDURE:
00070             [AND* .773 (NOT* (OR* (WALL DR)
00071                                   (WALL PIC)
00072                                   (WALL CBACK)
00073                                   (WALL WBSKT]
00074   RELATIVES:
00075             ((OPTION (DOOR LWALL) (WALL LWALL)))
00076
00077
00078   VARIABLE: (DOOR CBACK)
00079   VALUE:    0.0
00080   PROCEDURE:
00081             (DOOR CBACK)
00082   RELATIVES:
00083             ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00084
00085
00086   VARIABLE: (CHAIRBACK CBACK)
00087   VALUE:    .11
00088   PROCEDURE:
00089             (CHAIRBACK CBACK)
00090   RELATIVES:
00091             ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00092
00093
00094   VARIABLE: (WALL CBACK)
00095   VALUE:    0.0
00096   PROCEDURE:
00097             (WALL CBACK)
00098   RELATIVES:
00099             ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00100             ((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))
00101
00102
```

FIGURE B-3 DATABASE FOLLOWING SEARCH (Continued)

```
;  <TENENBAUM>BAYES.RPT;2   WED 4-FEB-76 1:53PM                Page 4:2

00103   VARIABLE: (PICTURE PIC)
00104   VALUE:    .3
00105   PROCEDURE:
00106             [AND* .3 (NOT* (OR* (AND* (ADJ RWALL PIC)
00107                                       (DOOR RWALL))
00108                                 (AND* (ADJ LWALL PIC)
00109                                       (DOOR LWALL]
00110   RELATIVES:
00111             ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00112
00113
00114   VARIABLE: (DOOR PIC)
00115   VALUE:    0.0
00116   PROCEDURE:
00117             (DOOR PIC)
00118   RELATIVES:
00119             ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00120
00121
00122   VARIABLE: (WALL PIC)
00123   VALUE:    0.0
00124   PROCEDURE:
00125             (WALL PIC)
00126   RELATIVES:
00127             ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00128             ((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))
00129
00130
00131   VARIABLE: (DOOR DR)
00132   VALUE:    .227
00133   PROCEDURE:
00134             (AND* .227 (AND* (NOT* (OR* (DOOR LWALL)
00135                                         (DOOR RWALL)))
00136                              (NOT* 0.0)))
00137   RELATIVES:
00138             ((OPTION (DOOR DR) (WALL DR)))
00139
00140
00141   VARIABLE: (WALL DR)
00142   VALUE:    0.0
00143   PROCEDURE:
00144             [AND* .773 (NOT* (OR* (WALL LWALL)
00145                                   (WALL RWALL]
00146   RELATIVES:
00147             ((OPTION (DOOR DR) (WALL DR)))
00148             ((OR* (WALL DR) (WALL PIC) (WALL CBACK) (WALL WBSKT)))
00149
00150
00151   10088 conses
00152   128. seconds
```

FIGURE B-3 DATABASE FOLLOWING SEARCH (Concluded)

Appendix C

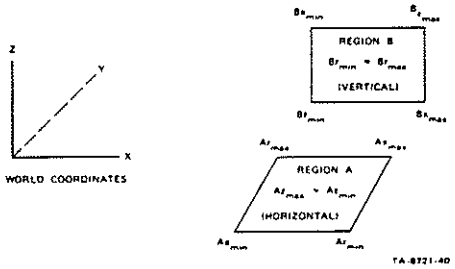
REPRESENTATIONS FOR SPATIAL RELATIONS

Spatial context is an important factor in resolving interpretation ambiguities. Procedural representations have been implemented for some common three dimensional spatial relationships between two regions, based on the relative world coordinates of vertices in their polygonal boundaries. These representations are described in Table C-1 and demonstrated in Table C-2 using the test regions in Figure C-1.

The above representations were originally developed for room scenes, assuming availability of range data. All relations except planarity can be reformulated in terms of two-dimensional image coordinates for standard eye level views.

Table C-1

REPRESENTATIONS FOR SPATIAL RELATIONS BETWEEN TWO PLANAR SURFACES
GIVEN 2 REGIONS--A (e.g., a Horizontal Chair Seat) and B (e.g., a Vertical Chair Back)



1. Left of/Right of

Let Axmin, Axmax = minimum and maximum X image coordinates of boundary points of Region A
Bxmin, Bxmax = minimum and maximum X image coordinates of boundary points of Region B
then

A. Region A is left of Region B

iff Bxmin + Bxmax > Axmin + Axmax
and Bxmin > Axmax or

$$\frac{\text{Max (Axmin, Bxmin)} - \text{Axmax} - 1}{\text{Min (Axmax - Axmin, Bxmax - Bxmin)} + 1} \leq .49$$

(The last condition provides a reasonable interpretation of the concept "left" in cases where Regions A and B partially overlap.)

B. Region A is right of Region B

iff Bxmin + Bxmax < Axmin + Axmax
and Axmin > Bxmax or

$$\frac{\text{Min (Axmax, Bxmax)} - \text{Axmin} + 1}{\text{Min (Axmax - Axmin, Bxmax - Bxmin)} + 1} \leq .49$$

2. Below/Above

Let Azmin = Azmax = height at maximum and minimum Y image coordinates of boundary points of Region A (horizontal surface)
Bzmin, Bzmax = height at maximum and minimum Y image coordinates of boundary points of Region B
then

A. Region A is below Region B

iff Bzmin + Bzmax > Azmin + Azmax
and Bzmin > Azmax or

$$\frac{\text{Max (Azmin, Bzmin)} - \text{Azmax} - 1}{\text{Min (Azmax - Azmin, Bzmax - Bzmin)} + 1} \leq .1$$

B. Region A is above Region B

iff Bzmin + Bzmax < Azmin + Azmax
and Azmin > Bzmax or

$$\frac{\text{Min (Azmax, Bzmax)} - \text{Azmin} + 1}{\text{Min (Azmax - Azmin, Bzmax - Bzmin)} + 1} \geq .1$$

(An additional relation, directly above/directly below may be defined by requiring that the regions involved not be to the right, left, in front, or in back of each other.)

3. Front/Back

Let Armin, Armax = maximum and minimum range of boundary points of Region A
Brmin, Brmax = maximum and minimum range of boundary points of Region B (vertical surface)
then

A. Region A is in front of Region B

iff Brmin + Brmax > Armin + Armax
and Brmin > Armax or

$$\frac{\text{Max (Armin, Brmin)} - \text{Armax} - 1}{\text{Min (Armax - Armin, Brmax - Brmin)} + 1} \geq .9$$

3. Front/Back (Concluded)

B. Region A is in back of Region B

iff Brmin + Brmax < Armin + Armax
and Armin > Brmax or

$$\frac{\text{Min (Armax, Brmax)} - \text{Armin} + 1}{\text{Min (Armax - Armin, Brmax - Brmin)} + 1} \leq .9$$

4. Coplanar

Let PLA, = least square planar surface fit to boundary points of Region A
PLB, = least square planar surface fit to boundary points of Region B
then

Region A and Region B are coplanar iff the following criteria hold:

(1) The surface normals of PLA and PLB must be parallel to within 10 degrees.

(2) Each local plane must intercept the same coordinate axis (X, Y, or Z) closest to the origin.

(3) These (most reliable) intercepts must agree within 10%.

(These above criteria compensate heuristically for uncertainties in range data.)

Table C-2

RELATIONS OF SURFACES IN FIGURE 9 USING REPRESENTATIONS IN TABLE 3.3a
(Table lists relations of object 1 to object 2)

where A = above F = in front L = Left
BL = below BK = in back R = right

	OBJECT 2						
OBJECT 1	Chairback	Chairseat	Door	Picture	Tabletop	Wall	Wastebasket
Chairback	---	A,BK	R	R,BL	F	R,BL	R,A
Chairseat	B,F	---	R	R,BL,F	BL,F	R,BL	R,A
Door	L	L	---	L	L	L	L,BK
Picture	L,A	L,A,BK	R	---	A,BK	R	R,A
Tabletop	BK	A,BK	R	BL,F	---	R,BL	R,A
Wall	L,A	L,A	R	L	L,A	---	R,A,BK
Wastebasket	L,BL	L,BL	R,F	L,BL	L,BL	L,BL,F	---

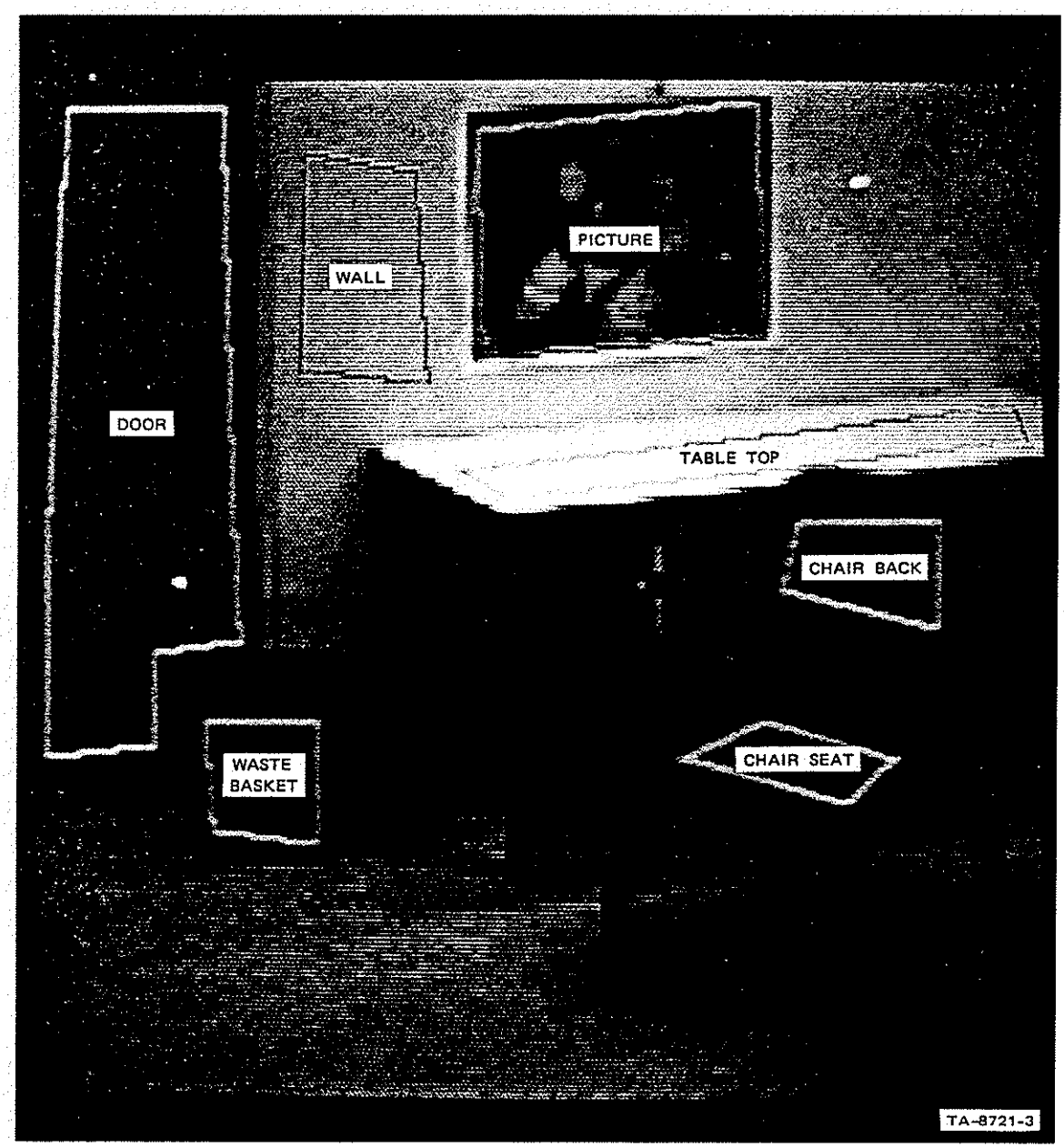


FIGURE C-1 REGIONS USED TO TEST SPATIAL RELATIONS (TABLE C-2)

Appendix D

XDEMON: A CONSTRAINT SATISFIER

A. Overview

A constraint satisfaction system has been implemented utilizing cooperating independent processes coupled through a global data base. The data base consists of variables representing constrained entities and constraints. Associated with each variable is a procedure for computing a value in terms of the current values of other variables. Each variable also has a list of related variables whose procedures utilize the present variable as input. When the value of a variable is changed, its related variables are activated by adding their procedures to a queue of jobs to be run. Thus, if running a process changes the value of its associated variable, additional processes may be activated. Execution terminates when the job queue is empty.

B. Details

Each variable in the data base contains four components:

- A FORM (procedure in the XDEMON program)
- A VALUE
- A set of RELATIVES (a list of related variables)
- VARPROPS (property list).

There are selecting and updating functions for each of the above components. The form of a variable is an internal representation of some s-expression. This internal representation, known as an H-expression, is composed of either a LISP atom, or a list of other variables. Each variable corresponds to a particular s-expression: in LISP a variable corresponds to a particular atom, and we have generalized this notion. For example, (FOO X Y) has a corresponding variable, as do FOO, X, and Y. So also does (FIE (FOO X Y) Z). As in LISP, character strings are normalized to yield a unique internal representation--the variable.

(NORMEXPR ⟨expression⟩) is a function that returns the variable corresponding to the given expression.

Variables are initialized to have the value Undef. Relatives and varprops are initialized to the value Nil.

A collection of variables may be linked to form a network. (CONNECT ⟨variable⟩) puts ⟨variable⟩ on the lists of relatives of all the variables in its form. (CONNECT! ⟨variable⟩) does the same thing recursively for the variables in the form as well. An example is given in Figure D-1. There are corresponding inverse functions DISCONNECT and DISCONNECT!.

Note that pointers to subexpressions are available via the form and pointers to superexpressions via the relatives.

The value of a variable is normally set by the function HSET. (HSET ⟨variable⟩ ⟨value⟩) returns ⟨value⟩ as its result. HSET is executed for its side-effects: if the new value is the same as the old value (under the equivalence HSETEQ, initially EQUAL), nothing happens: if the new value is different, then the variable's relatives are evaluated (or rather the evaluations are added to a list of jobs to be run). Note that other schemes could be used here; e.g., some relatives might be evaluated before the variable is reset.

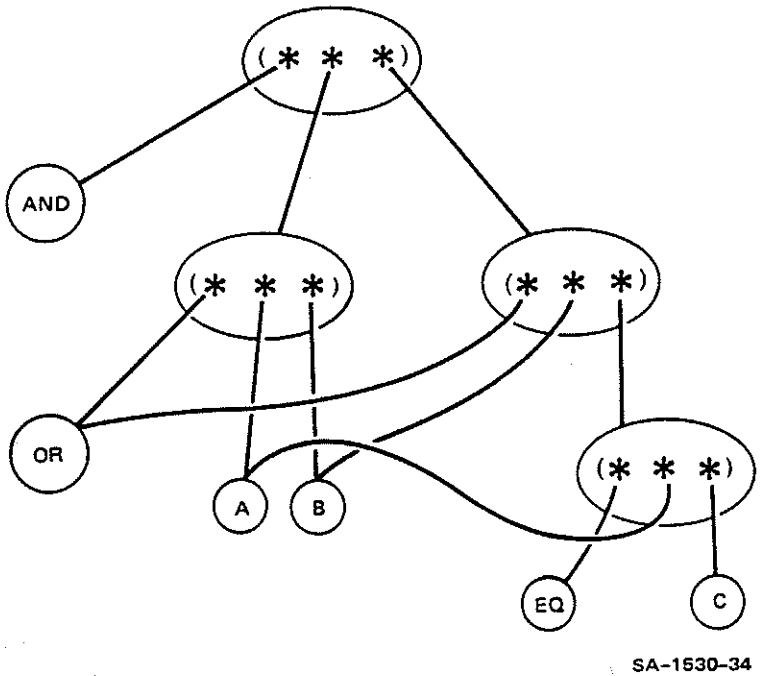


FIGURE D-1 REPRESENTATION FOR THE EXPRESSION

```
(CONNECT!
  (NORMEXPR
    (QUOTE
      (AND(OR A B)
        (OR B (EQ A C))))))
```

Variables are evaluated by (HEVAL ⟨variable⟩). HEVAL evaluates the form of the variable, and then HSETS the variable to the new value, perhaps causing its relatives to be evaluated. It returns the new value as its result. The value of an atomic H-expression is the value of the variable: otherwise, the value is the result of APPLYING the value of the CAR of the H-expression to the CDR.

Unlike LISP the definition of a function is kept in the value of a corresponding atomic variable, not in a special cell. Definitions are established be executing (HDEF ⟨function definition⟩). The ⟨function definition⟩ is a list of two elements, the name and the body (similar to LISP's DEFINE). The body can be the name of a LISP function, or a lambda expression. Note also that when a variable is HEVALuated, the evaluation is not recursive; the immediate value of the variables forming the H-expression are utilized without HEVALuating them.

Variables are only evaluated when the value of something to which they are relatives is changed, never just because a variable higher in the expression is evaluated. Values are thus remembered and not recomputed unnecessarily.

A list of outstanding jobs to be run is held in the global variable JOBLIST. Those jobs are executed by a call (RUNJOBS), which will successively execute and then delete the jobs on the list. RUNJOBS terminates when there are no jobs left (including those which have been added dynamically).

XDEMON listings are available from the authors.

Appendix E

AN MSYS EMULATION OF WALTZ FILTERING

MSYS can be easily modified to emulate a variety of search paradigms other than M*. This appendix describes an MSYS emulation of Waltz's filtering algorithm[17] and its application to region analysis.

Waltz analyzed line drawings by initially assigning all locally possible interpretations to each vertex and then eliminating any vertex interpretation that was inconsistent with all possible interpretations of a neighboring vertex along a common edge. Eliminating a possible vertex interpretation could result in the elimination of additional interpretations from adjacent vertices. This elimination process would often propagate until each vertex was left with a unique interpretation. A similar paradigm can be applied to region analysis by initially assigning all locally possible interpretations to each region and then eliminating interpretations inconsistent with those assigned to neighboring regions sharing a common boundary. Here, inconsistency is defined in terms of a list of legally adjacent interpretations. Such a paradigm was implemented in MSYS and used to analyze the scene partition of Figure 4.

The analysis utilized the adjacency constraints given in Figure E-1, where (LEGALADJ I1 (I2 ... IN)) specified a list (I2 ... IN) of legal interpretations for regions adjacent to a region with interpretation I1. The constraint (LEGALADJ DOOR (DOOR WALL FLOOR WASTEBASKET TABLETOP)), for example, required that regions labeled DOOR could be adjacent only to regions labeled DOOR, WALL, FLOOR, TABLETOP, or WASTEBASKET. (In a refinement on Waltz, it was further required that two door interpretations could be legally adjacent only if the regions involved had similar brightness.) WALL regions, similarly, could be adjacent only to regions labeled DOOR, WALL, PICTURE, WASTEBASKET, or TABLETOP. These adjacency constraints are somewhat contrived because region adjacency is an ill-defined concept in a partially partitioned scene. Fixtures, such as WALLS, DOORS, FLOOR, and BASEBOARD, have well-defined mutual adjacencies whereas moveable objects, such as WASTEBASKET and CHAIR, can appear in fairly arbitrary relationships with each other and with the fixtures. The analysis also used the constraint

```
(LEGALADJ DOOR (DOOR WALL FLOOR WASTEBASKET TABLETOP))
(LEGALADJ WALL (WALL DOOR PICTURE TABLETOP WASTEBASKET BASEBOARD))
(LEGALADJ PICTURE (WALL PICTURE TABLETOP))
(LEGALADJ WASTEBASKET (FLOOR WALL BASEBOARD DOOR TABLETOP CHAIRBACK))
(LEGALADJ CHAIRBACK (CHAIRSEAT TABLETOP WALL DOOR WASTEBASKET))
```

FIGURE E-1 ADJACENCY CONSTRAINTS FOR VERTICAL SURFACES OF ROOM SCENES

ROOMPARTITION to eliminate DOOR and WALL as possible interpretations of vertical regions with limited vertical extent.

Since Waltz dealt strictly with symbolic input, the classification routine was modified to return an a priori likelihood of 1.0 for all region interpretations that qualified as possibilities based on their height and surface orientation. This was the only actual modification to an MSYS routine required to emulate the Waltz filtering algorithm.

Figure E-2 contains a complete trace of the interpretation process encompassing network initialization (lines 1 to 27) and evaluation of the resulting solution (lines 28 to 35). A unique and consistent interpretation of the scene has been achieved without any instantiation and with considerably less propagation in the initialization phase than was required to achieve equilibrium likelihoods with nondeterministic constraints (see Appendix B). The final equilibrium likelihoods for all interpretation variables appear in Figure E-3, which presents the data base at equilibrium following initialization.

Figure E-3 also shows the procedures that were compiled from the adjacency constraints of Figure E-1 for computing interpretation likelihoods. These procedures contain support clauses that reduce the likelihood of an interpretation to zero whenever the likelihoods of all compatible interpretations in any adjacent region become zero. The evaluation function for (WALL LWALL), for example, contains clauses requiring that regions DR be DOOR, that region PIC be PICTURE or DOOR, that region WBSKT be DOOR or WASTEBASKET, and that region TTOP be TABLETOP. WALL was not an allowed interpretation for regions DR and PIC because their brightnesses were markedly different from that of region LWALL.

```
00001   BG-(NETSETUP (QUOTE (WASTEBASKET WBSKT))): 1.0
00002   BG-(NETSETUP (QUOTE (DOOR WBSKT))): 0.0-SETUP
00003   BG-(NETSETUP (QUOTE (WALL WBSKT))): 0.0-SETUP
00004   BG-(NETSETUP (QUOTE (DOOR RWALL))): 1.0
00005   BG-(NETSETUP (QUOTE (WALL RWALL))): 1.0
00006   BG-(NETSETUP (QUOTE (DOOR LWALL))): 1.0
00007   BG-(NETSETUP (QUOTE (WALL LWALL))): 1.0
00008   BG-(NETSETUP (QUOTE (DOOR CBACK))): 0.0-SETUP
00009   JOB-((OR* (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK))): 1.0
00010   BG-(NETSETUP (QUOTE (CHAIRBACK CBACK))): 1.0
00011   BG-(NETSETUP (QUOTE (WALL CBACK))): 0.0-SETUP
00012   JOB-((OR* (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK))): 1.0
00013   BG-(NETSETUP (QUOTE (PICTURE PIC))): 1.0
00014   BG-(NETSETUP (QUOTE (DOOR PIC))): 0.0-SETUP
00015   JOB-((OR* (DOOR PIC) (PICTURE PIC))): 1.0
00016   BG-(NETSETUP (QUOTE (WALL PIC))): 0.0-SETUP
00017   JOB-((AND* (OR* (WALL WBSKT) (WASTEBASKET WBSKT)) (TABLETOP TTOP)
00018   (WALL DR) (WALL PIC))): 0.0
00019   JOB-((DOOR LWALL)): 0.0
00020   JOB-((OR* (WALL LWALL) (DOOR LWALL))): 1.0
00021   JOB-((AND* (TABLETOP TTOP) (WALL PIC))): 0.0
00022   JOB-((DOOR RWALL)): 0.0
00023   BG-(NETSETUP (QUOTE (DOOR DR))): 1.0
00024   BG-(NETSETUP (QUOTE (WALL DR))): 0.0
00025   JOB-((AND* (OR* (WALL WBSKT) (WASTEBASKET WBSKT)) (TABLETOP TTOP)
00026   (WALL DR) (WALL PIC))): 0.0
00027   JOB-((OR* (WALL DR) (DOOR DR))): 1.0
00028   BG-((OPTION (DOOR DR) (WALL DR))): (DOOR DR)-1.0
00029   BG-((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC))): (PICTURE PIC)-1.0
00030   BG-((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK))): (CHAIRBACK
00031   CBACK)-1.0
00032   BG-((OPTION (DOOR LWALL) (WALL LWALL))): (WALL LWALL)-1.0
00033   BG-((OPTION (DOOR RWALL) (WALL RWALL))): (WALL RWALL)-1.0
00034   BG-((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))): (
00035   WASTEBASKET WBSKT)-1.0
00036   "SUCCESS"
```

FIGURE E-2 EXECUTION TRACE OF WALTZ EMULATION

```
00001   VARIABLE: (WASTEBASKET WBSKT)
00002   VALUE:    1.0
00003   PROCEDURE:
00004             (AND* 1.0 (AND* (TABLETOP TTOP)
00005                             (OR* (WALL CBACK)
00006                                  (CHAIRBACK CBACK)
00007                                  (DOOR CBACK))
00008                             (OR* (WALL DR)
00009                                  (DOOR DR))
00010                             (OR* (WALL LWALL)
00011                                  (DOOR LWALL))
00012                             (FLOOR FLR)))
00013   RELATIVES:
00014             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00015             ((OR* (WALL WBSKT) (WASTEBASKET WBSKT)))
00016             ((OR* (DOOR WBSKT) (WASTEBASKET WBSKT)))
00017             ((AND* (CHAIRSEAT CSEAT) (WASTEBASKET WBSKT) (TABLETOP
00018   TTOP)))
00019             ((OR* (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)))
00020
00021
00022   VARIABLE: (DOOR WBSKT)
00023   VALUE:    0.0
00024   PROCEDURE:
00025             (DOOR WBSKT)
00026   RELATIVES:
00027             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00028             ((OR* (DOOR WBSKT) (WASTEBASKET WBSKT)))
00029             ((OR* (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)))
00030
00031
00032   VARIABLE: (WALL WBSKT)
00033   VALUE:    0.0
00034   PROCEDURE:
00035             (WALL WBSKT)
00036   RELATIVES:
00037             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00038             ((OR* (WALL WBSKT) (WASTEBASKET WBSKT)))
00039             ((OR* (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)))
00040
00041
00042   VARIABLE: (DOOR RWALL)
00043   VALUE:    0.0
00044   PROCEDURE:
00045             (AND* 1.0 (AND* (TABLETOP TTOP)
00046                             (WALL PIC)))
00047   RELATIVES:
00048             ((OPTION (DOOR RWALL) (WALL RWALL)))
00049
00050
00051   VARIABLE: (WALL RWALL)
```

FIGURE E-3 DATABASE AT INITIAL EQUILIBRIUM

```
;  <TENENBAUM>WALTZ.RPT;5   THU 26-FEB-76 8:05PM                Page 2:1


00052   VALUE:      1.0
00053   PROCEDURE:
00054               [AND* 1.0 (AND* (TABLETOP TTOP)
00055                               (OR* (DOOR PIC)
00056                                    (PICTURE PIC)
00057   RELATIVES:
00058               ((OPTION (DOOR RWALL) (WALL RWALL)))
00059               ((AND* (TABLETOP TTOP) (WALL LWALL) (WALL RWALL)))
00060
00061
00062   VARIABLE: (DOOR LWALL)
00063   VALUE:      0.0
00064   PROCEDURE:
00065               (AND* 1.0 (AND* (OR* (WALL WBSKT)
00066                                    (WASTEBASKET WBSKT))
00067                               (TABLETOP TTOP)
00068                               (WALL DR)
00069                               (WALL PIC)))
00070   RELATIVES:
00071               ((OPTION (DOOR LWALL) (WALL LWALL)))
00072               ((OR* (WALL LWALL) (DOOR LWALL)))
00073               ((AND* (OR* (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT))
00074   (DOOR LWALL) (FLOOR FLR)))
00075
00076
00077   VARIABLE: (WALL LWALL)
00078   VALUE:      1.0
00079   PROCEDURE:
00080               [AND* 1.0 (AND* (OR* (DOOR WBSKT)
00081                                    (WASTEBASKET WBSKT))
00082                               (TABLETOP TTOP)
00083                               (DOOR DR)
00084                               (OR* (DOOR PIC)
00085                                    (PICTURE PIC)
00086   RELATIVES:
00087               ((OPTION (DOOR LWALL) (WALL LWALL)))
00088               ((OR* (WALL LWALL) (DOOR LWALL)))
00089               ((AND* (TABLETOP TTOP) (WALL LWALL) (WALL RWALL)))
00090               ((AND* (OR* (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT))
00091   (WALL LWALL) (FLOOR FLR)))
00092
00093
00094   VARIABLE: (DOOR CBACK)
00095   VALUE:      0.0
00096   PROCEDURE:
00097               (DOOR CBACK)
00098   RELATIVES:
00099               ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00100               ((OR* (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK)))
00101
00102
```

FIGURE E-3 DATABASE AT INITIAL EQUILIBRIUM (Continued)

```
00103   VARIABLE: (CHAIRBACK CBACK)
00104   VALUE:    1.0
00105   PROCEDURE:
00106             (AND* 1.0 (AND* (CHAIRSEAT CSEAT)
00107                             (WASTEBASKET WBSKT)
00108                             (TABLETOP TTOP)))
00109   RELATIVES:
00110             ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00111             ((OR* (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK)))
00112
00113
00114   VARIABLE: (WALL CBACK)
00115   VALUE:    0.0
00116   PROCEDURE:
00117             (WALL CBACK)
00118   RELATIVES:
00119             ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00120             ((OR* (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK)))
00121
00122
00123   VARIABLE: (PICTURE PIC)
00124   VALUE:    1.0
00125   PROCEDURE:
00126             (AND* 1.0 (AND* (TABLETOP TTOP)
00127                             (WALL LWALL)
00128                             (WALL RWALL)))
00129   RELATIVES:
00130             ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00131             ((OR* (DOOR PIC) (PICTURE PIC)))
00132
00133
00134   VARIABLE: (DOOR PIC)
00135   VALUE:    0.0
00136   PROCEDURE:
00137             (DOOR PIC)
00138   RELATIVES:
00139             ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00140             ((OR* (DOOR PIC) (PICTURE PIC)))
00141
00142
00143   VARIABLE: (WALL PIC)
00144   VALUE:    0.0
00145   PPOCEDURE:
00146             (WALL PIC)
00147   RELATIVES:
00148             ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00149             ((AND* (TABLETOP TTOP) (WALL PIC)))
00150             ((AND* (OR* (WALL WBSKT) (WASTEBASKET WBSKT)) (TABLETOP
00151   TTOP) (WALL DR) (WALL PIC)))
00152
00153
```

FIGURE E-3 DATABASE AT INITIAL EQUILIBRIUM (Continued)

```
00154   VARIABLE: (DOOR DR)
00155   VALUE:    1.0
00156   PROCEDURE:
00157               (AND* 1.0 (AND* (OR* (WALL WBSKT)
00158                                    (DOOR WBSKT)
00159                                    (WASTEBASKET WBSKT))
00160                               (WALL LWALL)
00161                               (FLOOR FLR)))
00162   RELATIVES:
00163               ((OPTION (DOOR DR) (WALL DR)))
00164               ((OR* (WALL DR) (DOOR DR)))
00165               ((AND* (OR* (DOOR WBSKT) (WASTEBASKET WBSKT)) (TABLETOP
00166   TTOP) (DOOR DR) (OR* (DOOR PIC) (PICTURE PIC))))
00167
00168
00169   VARIABLE: (WALL DR)
00170   VALUE:    0.0
00171   PROCEDURE:
00172               (AND* 1.0 (AND* (OR* (WALL WBSKT)
00173                                    (DOOR WBSKT)
00174                                    (WASTEBASKET WBSKT))
00175                               (DOOR LWALL)
00176                               (FLOOR FLR)))
00177   RELATIVES:
00178               ((OPTION (DOOR DR) (WALL DR)))
00179               ((OR* (WALL DR) (DOOR DR)))
00180               ((AND* (OR* (WALL WBSKT) (WASTEBASKET WBSKT)) (TABLETOP
00181   TTOP) (WALL DR) (WALL PIC)))
00182
00183
00184   11532 conses
00185   144. seconds
```

FIGURE E-3 DATABASE AT INITIAL EQUILIBRIUM (Concluded)

Appendix F

AN MSYS EMULATION OF DUDA'S ALGORITHM

Duda[20] formulated scene interpretation as a tree searching problem. Pictorial regions were each represented by a node of the tree and the branches emanating from a node corresponded to the possible interpretations for that region. The first region selected for labeling was designated as the "start node." A path through the tree from the start node to a terminal node represented a unique labeling of the scene.

Every region interpretation had a likelihood based on the attributes (e.g., color and size) of its associated region and every node had a score representing the sum of interpretation likelihoods along the path from the start node to that node. Legal interpretations for adjacent regions were constrained deterministically, as in the Waltz analysis (see Figure E-1). An A* search was used to find the highest scoring path through the tree that satisfied these constraints.

The A* search proceeded at each stage by expanding the open node with the greatest score. Initially, only nodes emanating from the start node, representing possible interpretations of the first region, are open. To expand a node, a region was selected that was not previously considered on the path to that node. The expansion node was removed from the list of open nodes and new open nodes were added for each interpretation of the selected region that was not incompatible with any previously assigned region interpretation on the path. Node expansion was repeated until a terminal node was selected for further expansion; the path to that terminal node represented the highest scoring legal labeling of the scene.

Duda's algorithm differs from M* in two significant ways. First, the selection of open nodes was based solely on the a priori likelihoods of previously instantiated interpretations, rather than on globally refined a posteriori estimates of all interpretation likelihoods, as in M*. As in the Waltz analysis, constraints were strictly Boolean and thus could not be used to adjust likelihoods, except to zero. Second, Duda's algorithm invoked constraints only when a node was expanded, and for the sole purpose

of determining which interpretations of the selected region were compatible with the other interpretations on the path to that node. Waltz and M*, by contrast, do not require instantiation as a prerequisite for invoking a constraint. Region interpretations are eliminated whenever they are inconsistent with all possible interpretations of any other region. This policy allows all subtrees containing the eliminated interpretation to be pruned from the search and avoids redundant discovery of the same inconsistency on distinct branches of the search. Moreover, eliminations can propagate, allowing additional inconsistencies to be discovered.

Figures F-1 through F-4 document the interpretation by Duda's algorithm of the scene partition depicted in Figure 4, using the constraints given in Figure E-1 (the same problem previously analyzed by Waltz's algorithm in Appendix E). Figure F-1 documents the initialization phase. Unlike M* (Appendix B) and Waltz (Appendix E), no likelihood reevaluation occurs because constraints do not apply until interpretations are instantiated. Figure F-2, a snapshot of the data base following initialization, shows all interpretations still carrying their a priori likelihoods.

In the search phase (Figure F-3), instantiations are proposed on the basis of the a priori interpretation likelihoods. As a consequence, the region interpretation (WALL DR) was chosen as the third instantiation (Figure F-3, line 78) and the search was forced to backtrack. Note that reevaluations are propagated only when interpretations are asserted and not when they are denied because only in the case of a unique instantiation are new constraints activated. In the terminal state (Figure F-4), the likelihoods of correct interpretations remain at their a priori values while the likelihoods of all other interpretations have been reduced to zero.

The emulation of Duda's algorithm in MSYS was accomplished by modifying the procedures associated with region interpretation variables and OPTION variables. The OPTION procedure was modified so that only the likelihoods of instantiated interpretation variables contributed to the score of a search state. The modified OPTION procedure returned a Boolean value, 1.0, if the associated region was uninstantiated, and 0.0, if it was instantiated. The procedures that computed interpretation likelihoods were then modified to always return the a priori likelihood of the interpretation or else zero. The likelihood of an interpretation was zero only if that interpretation had been instantiated, and an adjacent region had previously been instantiated to an incompatible interpretation.

A comparison of the likelihood procedures in Figures E-3 and F-2 shows how the standard M* likelihood procedures used in the Waltz emulation were modified to suppress propagation in the Duda emulation. Consider, in particular, the procedures associated with the interpretation variable (PICTURE PIC) at line 125 in Figure E-3 and line 188 in Figure F-2. First, each supporting interpretation of a constraint was replaced by a disjunction (OR**) of that interpretation and its associated OPTION variable. All support clauses were then enclosed in a grand disjunction with the OPTION variable of the constrained interpretation (PICTURE PIC). AND** and OR** were threshold versions of AND* and OR*, respectively, that evaluated to zero or one depending on whether or not the corresponding unthresholded function evaluated to less than 0.1. Since OPTION variables of uninstantiated regions have the value 1.0, the likelihood of (PICTURE PIC) is pinned at the a priori likelihood 0.12, as long as region PIC is uninstantiated, or all of the following conditions apply: Region TTOP is uninstantiated or instantiated to TABLETOP, region LWALL is uninstantiated or instantiated to WALL, and region RWALL is uninstantiated or instantiated to WALL. Otherwise, the likelihood of (PICTURE PIC) drops to zero.

```
;  <TENENBAUM>DUDA.RPT;2   WED 4-FEB-76 2:16PM                    Page 1

00001    BG-(NETSETUP (QUOTE (WASTEBASKET WBSKT))): .12
00002    BG-(NETSETUP (QUOTE (DOOR WBSKT))): 0.0-SETUP
00003    BG-(NETSETUP (QUOTE (WALL WBSKT))): 0.0-SETUP
00004    BG-(NETSETUP (QUOTE (DOOR RWALL))): .227
00005    BG-(NETSETUP (QUOTE (WALL RWALL))): .773
00006    BG-(NETSETUP (QUOTE (DOOR LWALL))): .227
00007    BG-(NETSETUP (QUOTE (WALL LWALL))): .773
00008    BG-(NETSETUP (QUOTE (DOOR CBACK))): 0.0-SETUP
00009    JOB-((OR** (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)) (WALL
00010    CBACK) (CHAIRBACK CBACK) (DOOR CBACK))): 1.0
00011    BG-(NETSETUP (QUOTE (CHAIRBACK CBACK))): .11
00012    BG-(NETSETUP (QUOTE (WALL CBACK))): 0.0-SETUP
00013    JOB-((OR** (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)) (WALL
00014    CBACK) (CHAIRBACK CBACK) (DOOR CBACK))): 1.0
00015    BG-(NETSETUP (QUOTE (PICTURE PIC))): .3
00016    BG-(NETSETUP (QUOTE (DOOR PIC))): 0.0-SETUP
00017    JOB-((OR** (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (DOOR PIC)
00018    (PICTURE PIC))): 1.0
00019    BG-(NETSETUP (QUOTE (WALL PIC))): 0.0-SETUP
00020    JOB-((OR** (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (WALL PIC))): 1.0
00021    BG-(NETSETUP (QUOTE (DOOR DR))): .227
00022    BG-(NETSETUP (QUOTE (WALL DR))): .773
00023    BG-((OPTION (DOOR DR) (WALL DR))): 1.0
00024    BG-((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC))): 1.0
00025    BG-((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK))): 1.0
00026    BG-((OPTION (DOOR LWALL) (WALL LWALL))): 1.0
00027    BG-((OPTION (DOOR RWALL) (WALL RWALL))): 1.0
00028    BG-((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))): 1.0
00029    SAVESTATE: (TOPCNTXT), SCORE: 0, LENGTH(IQUEUE): 6
```

FIGURE F-1 TRACE OF EXECUTION DURING ESTABLISHMENT OF INITIAL EQUILIBRIUM

```
00001   VARIABLE: (WASTEBASKET WBSKT)
00002   VALUE:      .12
00003   PROCEDURE:
00004               (AND* .12 (OR** (OPTION (WASTEBASKET WBSKT)
00005                                       (DOOR WBSKT)
00006                                       (WALL WBSKT))
00007                               (AND** (OR** (OPTION (TABLETOP TTOP))
00008                                            (TABLETOP TTOP))
00009                                      (OR** (OPTION (DOOR CBACK)
00010                                                    (CHAIRBACK CBACK)
00011                                                    (WALL CBACK))
00012                                            (WALL CBACK)
00013                                            (CHAIRBACK CBACK)
00014                                            (DOOR CBACK))
00015                                      (OR** (OPTION (DOOR DR)
00016                                                    (WALL DR))
00017                                            (WALL DR)
00018                                            (DOOR DR))
00019                                      (OR** (OPTION (DOOR LWALL)
00020                                                    (WALL LWALL))
00021                                            (WALL LWALL)
00022                                            (DOOR LWALL))
00023                                      (OR** (OPTION (FLOOR FLR))
00024                                            (FLOOR FLR)
00025   RELATIVES:
00026               ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00027               ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00028   (WALL WBSKT) (WASTEBASKET WBSKT)))
00029               ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00030   (DOOR WBSKT) (WASTEBASKET WBSKT)))
00031               ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00032   (WASTEBASKET WBSKT)))
00033               ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00034   (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)))
00035
00036
00037   VARIABLE: (DOOR WBSKT)
00038   VALUE:      0.0
00039   PROCEDURE:
00040               (DOOR WBSKT)
00041   RELATIVES:
00042               ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00043               ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00044   (DOOR WBSKT) (WASTEBASKET WBSKT)))
00045               ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00046   (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)))
00047
00048
00049   VARIABLE: (WALL WBSKT)
00050   VALUE:      0.0
00051   PROCEDURE:
```

FIGURE F-2 DATABASE PRIOR TO SEARCH

```
00052                (WALL WBSKT)
00053    RELATIVES:
00054                ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00055                ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00056    (WALL WBSKT) (WASTEBASKET WBSKT)))
00057                ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00058    (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)))
00059
00060
00061    VARIABLE: (DOOR RWALL)
00062    VALUE:      .227
00063    PROCEDURE:
00064                (AND* .227 (OR** (OPTION (DOOR RWALL)
00065                                         (WALL RWALL))
00066                                 (AND** (OR** (OPTION (TABLETOP TTOP))
00067                                              (TABLETOP TTOP))
00068                                        (OR** (OPTION (PICTURE PIC)
00069                                                      (DOOR PIC)
00070                                                      (WALL PIC))
00071                                              (WALL PIC)
00072    RELATIVES:
00073                ((OPTION (DOOR RWALL) (WALL RWALL)))
00074
00075
00076    VARIABLE: (WALL RWALL)
00077    VALUE:      .773
00078    PROCEDURE:
00079                (AND* .773 (OR** (OPTION (DOOR RWALL)
00080                                         (WALL RWALL))
00081                                 (AND** (OR** (OPTION (TABLETOP TTOP))
00082                                              (TABLETOP TTOP))
00083                                        (OR** (OPTION (PICTURE PIC)
00084                                                      (DOOR PIC)
00085                                                      (WALL PIC))
00086                                              (DOOR PIC)
00087                                              (PICTURE PIC)
00088    RELATIVES:
00089                ((OPTION (DOOR RWALL) (WALL RWALL)))
00090                ((OR** (OPTION (DOOR RWALL) (WALL RWALL)) (WALL RWALL)))
00091
00092
00093    VARIABLE: (DOOR LWALL)
00094    VALUE:      .227
00095    PROCEDURE:
00096                (AND* .227 (OR** (OPTION (DOOR LWALL)
00097                                         (WALL LWALL))
00098                                 (AND** (OR** (OPTION (WASTEBASKET WBSKT)
00099                                                      (DOOR WBSKT)
00100                                                      (WALL WBSKT))
00101                                              (WALL WBSKT)
00102                                              (WASTEBASKET WBSKT))
```

FIGURE F-2 DATABASE PRIOR TO SEARCH (Continued)

```
00103                                        (OR** (OPTION (TABLETOP TTOP))
00104                                              (TABLETOP TTOP))
00105                                        (OR** (OPTION (DOOR DR)
00106                                                    (WALL DR))
00107                                              (WALL DR))
00108                                        (OR** (OPTION (PICTURE PIC)
00109                                                    (DOOR PIC)
00110                                                    (WALL PIC))
00111                                              (WALL PIC]
00112   RELATIVES:
00113              ((OPTION (DOOR LWALL) (WALL LWALL)))
00114              ((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL)
00115   (DOOR LWALL)))
00116              ((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (DOOR LWALL)))
00117
00118
00119   VARIABLE: (WALL LWALL)
00120   VALUE:    .773
00121   PROCEDURE:
00122              [AND* .773 (OR** (OPTION (DOOR LWALL)
00123                                       (WALL LWALL))
00124                               (AND** (OR** (OPTION (WASTEBASKET WBSKT)
00125                                                    (DOOR WBSKT)
00126                                                    (WALL WBSKT))
00127                                              (DOOR WBSKT)
00128                                              (WASTEBASKET WBSKT))
00129                                        (OR** (OPTION (TABLETOP TTOP))
00130                                              (TABLETOP TTOP))
00131                                        (OR** (OPTION (DOOR DR)
00132                                                    (WALL DR))
00133                                              (DOOR DR))
00134                                        (OR** (OPTION (PICTURE PIC)
00135                                                    (DOOR PIC)
00136                                                    (WALL PIC))
00137                                              (DOOR PIC)
00138                                              (PICTURE PIC]
00139   RELATIVES:
00140              ((OPTION (DOOR LWALL) (WALL LWALL)))
00141              ((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL)
00142   (DOOR LWALL)))
00143              ((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL)))
00144
00145
00146   VARIABLE: (DOOR CBACK)
00147   VALUE:    0.0
00148   PROCEDURE:
00149              (DOOR CBACK)
00150   RELATIVES:
00151              ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00152              ((OR** (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK))
00153   (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK)))
```

FIGURE F-2 DATABASE PRIOR TO SEARCH (Continued)

```
00154
00155
00156   VARIABLE: (CHAIRBACK CBACK)
00157   VALUE:    .11
00158   PROCEDURE:
00159             [AND* .11 (OR** (OPTION (DOOR CBACK)
00160                                     (CHAIRBACK CBACK)
00161                                     (WALL CBACK))
00162                             (AND** (OR** (OPTION (CHAIRSEAT CSEAT))
00163                                          (CHAIRSEAT CSEAT))
00164                                    (OR** (OPTION (WASTEBASKET WBSKT)
00165                                                  (DOOR WBSKT)
00166                                                  (WALL WBSKT))
00167                                          (WASTEBASKET WBSKT))
00168                                    (OR** (OPTION (TABLETOP TTOP))
00169                                          (TABLETOP TTOP)
00170   RELATIVES:
00171             ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00172             ((OR** (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK))
00173   (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK)))
00174
00175
00176   VARIABLE: (WALL CBACK)
00177   VALUE:    0.0
00178   PROCEDURE:
00179             (WALL CBACK)
00180   RELATIVES:
00181             ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00182             ((OR** (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK))
00183   (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK)))
00184
00185
00186   VARIABLE: (PICTURE PIC)
00187   VALUE:    .3
00188   PROCEDURE:
00189             [AND* .3 (OR** (OPTION (PICTURE PIC)
00190                                    (DOOR PIC)
00191                                    (WALL PIC))
00192                            (AND** (OR** (OPTION (TABLETOP TTOP))
00193                                         (TABLETOP TTOP))
00194                                   (OR** (OPTION (DOOR LWALL)
00195                                                 (WALL LWALL))
00196                                         (WALL LWALL))
00197                                   (OR** (OPTION (DOOR RWALL)
00198                                                 (WALL RWALL))
00199                                         (WALL RWALL)]
00200   RELATIVES:
00201             ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00202             ((OR** (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (DOOR
00203   PIC) (PICTURE PIC)))
00204
```

FIGURE F-2 DATABASE PRIOR TO SEARCH (Continued)

```
00205
00206   VARIABLE: (DOOR PIC)
00207   VALUE:    0.0
00208   PROCEDURE:
00209             (DOOR PIC)
00210   RELATIVES:
00211             ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00212             ((OR** (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (DOOR
00213   PIC) (PICTURE PIC)))
00214
00215
00216   VARIABLE: (WALL PIC)
00217   VALUE:    0.0
00218   PPOCEDURE:
00219             (WALL PIC)
00220   RELATIVES:
00221             ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00222             ((OR** (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (WALL
00223   PIC)))
00224
00225
00226   VARIABLE: (DOOR DR)
00227   VALUE:    .227
00228   PROCEDURE:
00229             (AND* .227 (OR** (OPTION (DOOR DR)
00230                                      (WALL DR))
00231                              (AND** (OR** (OPTION (WASTEBASKET WBSKT)
00232                                                   (DOOR WBSKT)
00233                                                   (WALL WBSKT))
00234                                           (WALL WBSKT)
00235                                           (DOOR WBSKT)
00236                                           (WASTEBASKET WBSKT))
00237                                     (OR** (OPTION (DOOR LWALL)
00238                                                   (WALL LWALL))
00239                                           (WALL LWALL))
00240                                     (OR** (OPTION (FLOOR FLR))
00241                                           (FLOOR FLR)
00242   RELATIVES:
00243             ((OPTION (DOOR DR) (WALL DR)))
00244             ((OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR) (DOOR DR)))
00245             ((OR** (OPTION (DOOR DR) (WALL DR)) (DOOR DR)))
00246
00247
00248   VARIABLE: (WALL DR)
00249   VALUE:    .773
00250   PROCEDURE:
00251             (AND* .773 (OR** (OPTION (DOOR DR)
00252                                      (WALL DR))
00253                              (AND** (OR** (OPTION (WASTEBASKET WBSKT)
00254                                                   (DOOR WBSKT)
00255                                                   (WALL WBSKT))
```

FIGURE F-2 DATABASE PRIOR TO SEARCH (Continued)

```
;  <TENENBAUM>DUDA.RPT:2   WED 4-FEB-76 2:16PM                    Page 2:5

00256                                                  (WALL WBSKT)
00257                                                  (DOOR WBSKT)
00258                                                  (WASTEBASKET WBSKT))
00259                                          (OR** (OPTION (DOOR LWALL)
00260                                                        (WALL LWALL))
00261                                                  (DOOR LWALL))
00262                                          (OR** (OPTION (FLOOR FLR))
00263                                                  (FLOOR FLR)
00264   RELATIVES:
00265              ((OPTION (DOOR DR) (WALL DR)))
00266              ((OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR) (DOOR DR)))
00267              ((OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR)))
```

FIGURE F-2 DATABASE PRIOR TO SEARCH (Concluded)

```
00030   REINSTATE: (TOPCNTXT)
00031   ASSERT = (WALL RWALL)
00032   RG=((OPTION (DOOR RWALL) (WALL RWALL))): 0.0
00033   JOB=((OR** (OPTION (DOOR RWALL) (WALL RWALL)) (WALL RWALL))): 1.0
00034   JOB=((OR** (OPTION (DOOR RWALL) (WALL RWALL)) (AND** (OR** (OPTION
00035   (TABLETOP TTOP)) (TABLETOP TTOP)) (OR** (OPTION (PICTURE PIC) (DOOR
00036   PIC) (WALL PIC)) (DOOR PIC) (PICTURE PIC))))): 1.0
00037   SAVESTATE: (((WALL RWALL) . .773) TOPCNTXT), SCORE: .773
00038   , LENGTH(IQUEUE): 5
00039   REINSTATE: (TOPCNTXT)
00040   DENY = (WALL RWALL)
00041   JOB=((OR** (OPTION (DOOR RWALL) (WALL RWALL)) (WALL RWALL))): 1.0
00042   RG=((OPTION (DOOR RWALL) (WALL RWALL))): 1.0
00043   SAVESTATE: (((WALL RWALL) . 0) TOPCNTXT), SCORE: 0, LENGTH(IQUEUE): 6
00044   REINSTATE: (((WALL RWALL) . .773) TOPCNTXT)
00045   ASSERT = (WALL LWALL)
00046   JOB=((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (DOOR LWALL))): 1.0
00047   JOB=((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL) (DOOR LWALL)))
00048   : 1.0
00049   RG=((OPTION (DOOR LWALL) (WALL LWALL))): 0.0
00050   JOB=((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (DOOR LWALL))): 0.0
00051   JOB=((AND** (OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00052   (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)) (OR** (OPTION (DOOR
00053   LWALL) (WALL LWALL)) (DOOR LWALL)) (OR** (OPTION (FLOOR FLR)) (FLOOR
00054   FLR)))): 0.0
00055   JOB=((OR** (OPTION (DOOR DR) (WALL DR)) (AND** (OR** (OPTION (
00056   WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)) (WALL WBSKT) (DOOR WBSKT)
00057   (WASTEBASKET WBSKT)) (OR** (OPTION (DOOR LWALL) (WALL LWALL)) (DOOR
00058   LWALL)) (OR** (OPTION (FLOOR FLR)) (FLOOR FLR))))): 1.0
00059   JOB=((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL))): 1.0
00060   JOB=((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (AND** (OR** (OPTION
00061   (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)) (DOOR WBSKT) (WASTEBASKET
00062   WBSKT)) (OR** (OPTION (TABLETOP TTOP)) (TABLETOP TTOP)) (OR** (OPTION
00063   (DOOR DR) (WALL DR)) (DOOR DR)) (OR** (OPTION (PICTURE PIC) (DOOR
00064   PIC) (WALL PIC)) (DOOR PIC) (PICTURE PIC))))): 1.0
00065   JOB=((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL) (DOOR LWALL)))
00066   : 1.0
00067   SAVESTATE: (((WALL LWALL) . .773) ((WALL RWALL) . .773) TOPCNTXT)
00068   , SCORE: 1.55, LENGTH(IQUEUE): 4
00069   REINSTATE: (((WALL RWALL) . .773) TOPCNTXT)
00070   DENY = (WALL LWALL)
00071   JOB=((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL))): 1.0
00072   JOB=((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL) (DOOR LWALL)))
00073   : 1.0
00074   RG=((OPTION (DOOR LWALL) (WALL LWALL))): 1.0
00075   SAVESTATE: (((WALL LWALL) . 0) ((WALL RWALL) . .773) TOPCNTXT), SCORE: .773
00076   , LENGTH(IQUEUE): 5
00077   REINSTATE: (((WALL LWALL) . .773) ((WALL RWALL) . .773) TOPCNTXT)
00078   ASSERT = (WALL DR)
00079   JOB=((OR** (OPTION (DOOR DR) (WALL DR)) (DOOR DR))): 1.0
00080   JOB=((OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR) (DOOR DR))): 1.0
```

FIGURE F-3 CONTINUATION OF TRACE SHOWING EXECUTION OF SEARCH

```
00081   BG-((OPTION (DOOR DR) (WALL DR))): 0.0
00082   JOB-((OR** (OPTION (DOOR DR) (WALL DR)) (AND** (OR** (OPTION (
00083   WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)) (WALL WBSKT) (DOOR WBSKT)
00084   (WASTEBASKET WBSKT)) (OR** (OPTION (DOOR LWALL) (WALL LWALL)) (DOOR
00085   LWALL)) (OR** (OPTION (FLOOR FLR)) (FLOOR FLR))))): 0.0
00086   JOB-((WALL DR)): 0.0
00087   JOB-((OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR) (DOOR DR))): 0.0
00088   JOB-((AND** (OR** (OPTION (TABLETOP TTOP)) (TABLETOP TTOP)) (OR**
00089   (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)) (WALL CBACK)
00090   (CHAIRBACK CBACK) (DOOR CBACK)) (OR** (OPTION (DOOR DR) (WALL DR))
00091   (WALL DR) (DOOR DR)) (OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL
00092   LWALL) (DOOR LWALL)) (OR** (OPTION (FLOOR FLR)) (FLOOR FLR)))): 0.0
00093   JOB-((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00094   (AND** (OR** (OPTION (TABLETOP TTOP)) (TABLETOP TTOP)) (OR** (OPTION
00095   (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)) (WALL CBACK) (CHAIRBACK
00096   CBACK) (DOOR CBACK)) (OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR)
00097   (DOOR DR)) (OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL) (DOOR
00098   LWALL)) (OR** (OPTION (FLOOR FLR)) (FLOOR FLR))))): 1.0
00099   JOB-((OR** (OPTION (DOOR DR) (WALL DR)) (DOOR DR))): 0.0
00100   JOB-((AND** (OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00101   (DOOR WBSKT) (WASTEBASKET WBSKT)) (OR** (OPTION (TABLETOP TTOP)) (
00102   TABLETOP TTOP)) (OR** (OPTION (DOOR DR) (WALL DR)) (DOOR DR)) (OR**
00103   (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (DOOR PIC) (PICTURE PIC))))
00104   : 0.0
00105   JOB-((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (AND** (OR** (OPTION
00106   (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)) (DOOR WBSKT) (WASTEBASKET
00107   WBSKT)) (OR** (OPTION (TABLETOP TTOP)) (TABLETOP TTOP)) (OR** (OPTION
00108   (DOOR DR) (WALL DR)) (DOOR DR)) (OR** (OPTION (PICTURE PIC) (DOOR
00109   PIC) (WALL PIC)) (DOOR PIC) (PICTURE PIC))))): 0.0
00110   JOB-((WALL LWALL)): 0.0
00111   JOB-((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL))): 0.0
00112   JOB-((AND** (OR** (OPTION (TABLETOP TTOP)) (TABLETOP TTOP)) (OR**
00113   (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL)) (OR** (OPTION (DOOR
00114   RWALL) (WALL RWALL)) (WALL RWALL)))): 0.0
00115   JOB-((OR** (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (AND** (OR**
00116   (OPTION (TABLETOP TTOP)) (TABLETOP TTOP)) (OR** (OPTION (DOOR LWALL)
00117   (WALL LWALL)) (WALL LWALL)) (OR** (OPTION (DOOR RWALL) (WALL RWALL))
00118   (WALL RWALL))))): 1.0
00119   JOB-((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL) (DOOR LWALL)))
00120   : 0.0
00121   JOB-((AND** (OR** (OPTION (TABLETOP TTOP)) (TABLETOP TTOP)) (OR**
00122   (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)) (WALL CBACK)
00123   (CHAIRBACK CBACK) (DOOR CBACK)) (OR** (OPTION (DOOR DR) (WALL DR))
00124   (WALL DR) (DOOR DR)) (OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL
00125   LWALL) (DOOR LWALL)) (OR** (OPTION (FLOOR FLR)) (FLOOR FLR)))): 0.0
00126   BG-((OPTION (DOOR DR) (WALL DR))): "CONTRADICTION DETECTED"
00127   REINSTATE: (((WALL LWALL) . .773) ((WALL RWALL) . .773) TOPCNTXT)
00128   DENY - (WALL DR)
00129   JOB-((OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR) (DOOR DR))): 1.0
00130   BG-((OPTION (DOOR DR) (WALL DR))): 1.0
00131   SAVESTATE: (((WALL DR) . 0) ((WALL LWALL) . .773) ((WALL RWALL) .
```

FIGURE F-3 CONTINUATION OF TRACE SHOWING EXECUTION OF SEARCH (Continued)

```
00132   .773) TOPCNTXT), SCORE: 1.55, LENGTH(IQUEUE): 4
00133   REINSTATE: (((WALL DR) . 0) ((WALL LWALL) . .773) ((WALL RWALL) .
00134   .773) TOPCNTXT)
00135   ASSERT - (PICTURE PIC)
00136   SAVESTATE: (((PICTURE PIC) . .3) ((WALL DR) . 0) ((WALL LWALL) . .773)
00137   ((WALL RWALL) . .773) TOPCNTXT), SCORE: 1.55, LENGTH(IQUEUE): 3
00138   REINSTATE: (((WALL DR) . 0) ((WALL LWALL) . .773) ((WALL RWALL) .
00139   .773) TOPCNTXT)
00140   DENY - (PICTURE PIC)
00141   JOB-((OR** (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (DOOR PIC)
00142   (PICTURE PIC))): 1.0
00143   BG-((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC))):
00144   "CONTRADICTION DETECTED"
00145   REINSTATE: (((PICTURE PIC) . .3) ((WALL DR) . 0) ((WALL LWALL) . .773)
00146   ((WALL RWALL) . .773) TOPCNTXT)
00147   ASSERT - (DOOR DR)
00148   SAVESTATE: (((DOOR DR) . .227) ((PICTURE PIC) . .3) ((WALL DR) . 0)
00149   ((WALL LWALL) . .773) ((WALL RWALL) . .773) TOPCNTXT), SCORE: 1.55
00150   , LENGTH(IQUEUE): 2
00151   REINSTATE: (((PICTURE PIC) . .3) ((WALL DR) . 0) ((WALL LWALL) . .773)
00152   ((WALL RWALL) . .773) TOPCNTXT)
00153   DENY - (DOOR DR)
00154   JOB-((OR** (OPTION (DOOR DR) (WALL DR)) (DOOR DR))): 1.0
00155   JOB-((OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR) (DOOR DR))): 1.0
00156   BG-((OPTION (DOOR DR) (WALL DR))): "CONTRADICTION DETECTED"
00157   REINSTATE: (((DOOR DR) . .227) ((PICTURE PIC) . .3) ((WALL DR) . 0)
00158   ((WALL LWALL) . .773) ((WALL RWALL) . .773) TOPCNTXT)
00159   ASSERT - (WASTEBASKET WBSKT)
00160   SAVESTATE: (((WASTEBASKET WBSKT) . .12) ((DOOR DR) . .227) ((PICTURE
00161   PIC) . .3) ((WALL DR) . 0) ((WALL LWALL) . .773) ((WALL RWALL) . .773)
00162   TOPCNTXT), SCORE: 1.55, LENGTH(IQUEUE): 1
00163   REINSTATE: (((DOOR DR) . .227) ((PICTURE PIC) . .3) ((WALL DR) . 0)
00164   ((WALL LWALL) . .773) ((WALL RWALL) . .773) TOPCNTXT)
00165   DENY - (WASTEBASKET WBSKT)
00166   JOB-((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00167   (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT))): 1.0
00168   JOB-((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00169   (WASTEBASKET WBSKT))): 1.0
00170   JOB-((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00171   (DOOR WBSKT) (WASTEBASKET WBSKT))): 1.0
00172   BG-((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))):
00173   "CONTRADICTION DETECTED"
00174   REINSTATE: (((WASTEBASKET WBSKT) . .12) ((DOOR DR) . .227) ((PICTURE
00175   PIC) . .3) ((WALL DR) . 0) ((WALL LWALL) . .773) ((WALL RWALL) . .773)
00176   TOPCNTXT)
00177   ASSERT - (CHAIRBACK CBACK)
00178   "SUCCESS"
```

FIGURE F-3 CONTINUATION OF TRACE SHOWING EXECUTION OF SEARCH (Concluded)

```
00001   VARIABLE: (WASTEBASKET WBSKT)
00002   VALUE:    .12
00003   PROCEDURE:
00004             (AND* .12 (OR** (OPTION (WASTEBASKET WBSKT)
00005                                     (DOOR WBSKT)
00006                                     (WALL WBSKT))
00007                             (AND** (OR** (OPTION (TABLETOP TTOP))
00008                                          (TABLETOP TTOP))
00009                                    (OR** (OPTION (DOOR CBACK)
00010                                                  (CHAIRBACK CBACK)
00011                                                  (WALL CBACK))
00012                                          (WALL CBACK)
00013                                          (CHAIRBACK CBACK)
00014                                          (DOOR CBACK))
00015                                    (OR** (OPTION (DOOR DR)
00016                                                  (WALL DR))
00017                                          (WALL DR)
00018                                          (DOOR DR))
00019                                    (OR** (OPTION (DOOR LWALL)
00020                                                  (WALL LWALL))
00021                                          (WALL LWALL)
00022                                          (DOOR LWALL))
00023                                    (OR** (OPTION (FLOOR FLR))
00024                                          (FLOOR FLR)
00025   RELATIVES:
00026             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00027             ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00028   (DOOR WBSKT) (WASTEBASKET WBSKT)))
00029             ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00030   (WASTEBASKET WBSKT)))
00031             ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00032   (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)))
00033
00034
00035   VARIABLE: (DOOR WBSKT)
00036   VALUE:    0.0
00037   PROCEDURE:
00038             (DOOR WBSKT)
00039   RELATIVES:
00040             ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00041             ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00042   (DOOR WBSKT) (WASTEBASKET WBSKT)))
00043             ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00044   (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)))
00045
00046
00047   VARIABLE: (WALL WBSKT)
00048   VALUE:    0.0
00049   PROCEDURE:
00050             (WALL WBSKT)
00051   RELATIVES:
```

FIGURE F-4 DATABASE FOLLOWING SEARCH

```
00052              ((OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT)))
00053              ((OR** (OPTION (WASTEBASKET WBSKT) (DOOR WBSKT) (WALL WBSKT))
00054    (WALL WBSKT) (DOOR WBSKT) (WASTEBASKET WBSKT)))
00055
00056
00057    VARIABLE: (DOOR RWALL)
00058    VALUE:    0.0
00059    PROCEDURE:
00060              (AND* .227 (OR** (OPTION (DOOR RWALL)
00061                                       (WALL RWALL))
00062                               (AND** (OR** (OPTION (TABLETOP TTOP))
00063                                            (TABLETOP TTOP))
00064                                      (OR** (OPTION (PICTURE PIC)
00065                                                    (DOOR PIC)
00066                                                    (WALL PIC))
00067                                            (WALL PIC)
00068    RELATIVES:
00069              ((OPTION (DOOR RWALL) (WALL RWALL)))
00070
00071
00072    VARIABLE: (WALL RWALL)
00073    VALUE:    .773
00074    PROCEDURE:
00075              (AND* .773 (OR** (OPTION (DOOR RWALL)
00076                                       (WALL RWALL))
00077                               (AND** (OR** (OPTION (TABLETOP TTOP))
00078                                            (TABLETOP TTOP))
00079                                      (OR** (OPTION (PICTURE PIC)
00080                                                    (DOOR PIC)
00081                                                    (WALL PIC))
00082                                            (DOOR PIC)
00083                                            (PICTURE PIC)
00084    RELATIVES:
00085              ((OPTION (DOOR RWALL) (WALL RWALL)))
00086              ((OR** (OPTION (DOOR RWALL) (WALL RWALL)) (WALL RWALL)))
00087
00088
00089    VARIABLE: (DOOR LWALL)
00090    VALUE:    0.0
00091    PROCEDURE:
00092              (AND* .227 (OR** (OPTION (DOOR LWALL)
00093                                       (WALL LWALL))
00094                               (AND** (OR** (OPTION (WASTEBASKET WBSKT)
00095                                                    (DOOR WBSKT)
00096                                                    (WALL WBSKT))
00097                                            (WALL WBSKT)
00098                                            (WASTEBASKET WBSKT))
00099                                      (OR** (OPTION (TABLETOP TTOP))
00100                                            (TABLETOP TTOP))
00101                                      (OR** (OPTION (DOOR DR)
00102                                                    (WALL DR))
```

FIGURE F-4 DATABASE FOLLOWING SEARCH (Continued)

```
00103                                                (WALL DR))
00104                                          (OR** (OPTION (PICTURE PIC)
00105                                                        (DOOR PIC)
00106                                                        (WALL PIC))
00107                                                (WALL PIC]
00108   RELATIVES:
00109               ((OPTION (DOOR LWALL) (WALL LWALL)))
00110               ((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL)
00111   (DOOR LWALL)))
00112
00113
00114   VARIABLE: (WALL LWALL)
00115   VALUE:      .773
00116   PROCEDURE:
00117               (AND* .773 (OR** (OPTION (DOOR LWALL)
00118                                        (WALL LWALL))
00119                                (AND** (OR** (OPTION (WASTEBASKET WBSKT)
00120                                                     (DOOR WBSKT)
00121                                                     (WALL WBSKT))
00122                                             (DOOR WBSKT)
00123                                             (WASTEBASKET WBSKT))
00124                                       (OR** (OPTION (TABLETOP TTOP))
00125                                             (TABLETOP TTOP))
00126                                       (OR** (OPTION (DOOR DR)
00127                                                     (WALL DR))
00128                                             (DOOR DR))
00129                                       (OR** (OPTION (PICTURE PIC)
00130                                                     (DOOR PIC)
00131                                                     (WALL PIC))
00132                                             (DOOR PIC)
00133                                             (PICTURE PIC]
00134   RELATIVES:
00135               ((OPTION (DOOR LWALL) (WALL LWALL)))
00136               ((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL)
00137   (DOOR LWALL)))
00138               ((OR** (OPTION (DOOR LWALL) (WALL LWALL)) (WALL LWALL)))
00139
00140
00141   VARIABLE: (DOOR CBACK)
00142   VALUE:      0.0
00143   PROCEDURE:
00144               (DOOR CBACK)
00145   RELATIVES:
00146               ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00147               ((OR** (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK))
00148   (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK)))
00149
00150
00151   VARIABLE: (CHAIRBACK CBACK)
00152   VALUE:      .11
00153   PROCEDURE:
```

FIGURE F-4 DATABASE FOLLOWING SEARCH (Continued)

```
00154              (AND* .11 (OR** (OPTION (DOOR CBACK)
00155                                      (CHAIRBACK CBACK)
00156                                      (WALL CBACK))
00157                                (AND** (OR** (OPTION (CHAIRSEAT CSEAT))
00158                                             (CHAIRSEAT CSEAT))
00159                                       (OR** (OPTION (WASTEBASKET WBSKT)
00160                                                     (DOOR WBSKT)
00161                                                     (WALL WBSKT))
00162                                             (WASTEBASKET WBSKT))
00163                                       (OR** (OPTION (TABLETOP TTOP))
00164                                             (TABLETOP TTOP]
00165   RELATIVES:
00166              ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00167              ((OR** (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK))
00168   (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK)))
00169
00170
00171   VARIABLE: (WALL CBACK)
00172   VALUE:    0.0
00173   PROCEDURE:
00174              (WALL CBACK)
00175   RELATIVES:
00176              ((OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK)))
00177              ((OR** (OPTION (DOOR CBACK) (CHAIRBACK CBACK) (WALL CBACK))
00178   (WALL CBACK) (CHAIRBACK CBACK) (DOOR CBACK)))
00179
00180
00181   VARIABLE: (PICTURE PIC)
00182   VALUE:    .3
00183   PROCEDURE:
00184              [AND* .3 (OR** (OPTION (PICTURE PIC)
00185                                     (DOOR PIC)
00186                                     (WALL PIC))
00187                               (AND** (OR** (OPTION (TABLETOP TTOP))
00188                                            (TABLETOP TTOP))
00189                                      (OR** (OPTION (DOOR LWALL)
00190                                                    (WALL LWALL))
00191                                            (WALL LWALL))
00192                                      (OR** (OPTION (DOOR RWALL)
00193                                                    (WALL RWALL))
00194                                            (WALL RWALL]
00195   RELATIVES:
00196              ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00197              ((OR** (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (DOOR
00198   PIC) (PICTURE PIC)))
00199
00200
00201   VARIABLE: (DOOR PIC)
00202   VALUE:    0.0
00203   PROCEDURE:
00204              (DOOR PIC)
```

FIGURE F-4 DATABASE FOLLOWING SEARCH (Continued)

```
00205   RELATIVES:
00206               ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00207               ((OR** (OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)) (DOOR
00208   PIC) (PICTURE PIC)))
00209
00210
00211   VARIABLE: (WALL PIC)
00212   VALUE:      0.0
00213   PROCEDURE:
00214               (WALL PIC)
00215   RELATIVES:
00216               ((OPTION (PICTURE PIC) (DOOR PIC) (WALL PIC)))
00217
00218
00219   VARIABLE: (DOOR DR)
00220   VALUE:      .227
00221   PROCEDURE:
00222               [AND* .227 (OR** (OPTION (DOOR DR)
00223                                        (WALL DR))
00224                                (AND** (OR** (OPTION (WASTEBASKET WBSKT)
00225                                                     (DOOR WBSKT)
00226                                                     (WALL WBSKT))
00227                                             (WALL WBSKT)
00228                                             (DOOR WBSKT)
00229                                             (WASTEBASKET WBSKT))
00230                                       (OR** (OPTION (DOOR LWALL)
00231                                                     (WALL LWALL))
00232                                             (WALL LWALL))
00233                                       (OR** (OPTION (FLOOR FLR))
00234                                             (FLOOR FLR]
00235   RELATIVES:
00236               ((OPTION (DOOR DR) (WALL DR)))
00237               ((OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR) (DOOR DR)))
00238               ((OR** (OPTION (DOOR DR) (WALL DR)) (DOOR DR)))
00239
00240
00241   VARIABLE: (WALL DR)
00242   VALUE:      0.0
00243   PROCEDURE:
00244               [AND* .773 (OR** (OPTION (DOOR DR)
00245                                        (WALL DR))
00246                                (AND** (OR** (OPTION (WASTEBASKET WBSKT)
00247                                                     (DOOR WBSKT)
00248                                                     (WALL WBSKT))
00249                                             (WALL WBSKT)
00250                                             (DOOR WBSKT)
00251                                             (WASTEBASKET WBSKT))
00252                                       (OR** (OPTION (DOOR LWALL)
00253                                                     (WALL LWALL))
00254                                             (DOOR LWALL))
00255                                       (OR** (OPTION (FLOOR FLR))
```

FIGURE F-4 DATABASE FOLLOWING SEARCH (Continued)

```
;  <TENENBAUM>DUDA.RPT;2   WED 4-FEB-76 2:16PM                    Page 4:5

00256                                                  (FLOOR FLR)
00257    RELATIVES:
00258                ((OPTION (DOOR DR) (WALL DR)))
00259                ((OR** (OPTION (DOOR DR) (WALL DR)) (WALL DR) (DOOR DR)))
00260
00261
00262    16078 conses
00263    179. seconds
```

FIGURE F-4 DATABASE FOLLOWING SEARCH (Concluded)

Appendix G

SOLVING CONSTRAINT SATISFACTION PUZZLES WITH A VISION SYSTEM

The puzzle, defined in Figure G-1, exemplifies a class of constraint satisfaction problems that are very difficult to solve by conventional heuristic search methods. These problems have state spaces that are far too large to search by exhaustive enumeration and usually, there are no obvious heuristics for selecting instantiations. Moreover, backtracking is of limited utility because many of the constraints cannot be tested until several problem variables have been instantiated.

A reasonable state space representation for the problem defined in Figure G-1 consists of 5^6 (approximately 15,000) sextuples, each containing instantiations of the variables nationality, house position, house color, drink, cigarette, and pet (e.g., (ENGLISHMAN, MIDDLE HOUSE, RED, TEA, KOOLS, and DOG)). The problem is solved by finding five sextuples, having unique instantiations for each variable, that satisfy all 15 constraints.

An exhaustive search of this space, choosing sextuples five at a time, requires examination of $5^5 \times 4^5 \times 3^5 \times 2^5 \times 1^5 = 2.5 \times 10^{10}$ sets of sextuples. (There are five choices of house position, house color, pet, drink, and cigarette available in the sextuple corresponding to the first nationality, leaving four choices in each category free for the second nationality, three choices each for the third nationality, and so forth.) Even if these sets of sextuples could be tested against the constraints at a rate of 100/second, it would still take almost eight years to complete the exhaustive search.

Fortunately, astute puzzle solvers have discovered that the requirements for search can be sharply reduced or even eliminated by using information in the constraints directly to eliminate inconsistent elements from the ranges of problem variables.[30, 33] Constraints 1 and 5, for example, allow immediate deletion from the original set of problem states of all sextuples containing the nationality Ukranian and a drink other than tea, as well as all those containing the drink tea and a nationality other than Ukranian. Constraints 4, 9, 6, and 13, respectively, eliminate additional sextuples in which the nationality Ukranian is paired with the green house, the middle house, the house to the right of the irovy house, and the house where Luckys are smoked. The elimination process has a cumulative effect

1. There are five houses, each of a different color and inhabited by men of different nationalities, with different pets, drinks, and cigarettes.
2. The Englishman lives in the red house.
3. The Spaniard owns the dog.
4. Coffee is drunk in the green house.
5. The Ukranian drinks tea.
6. The green house is immediately to the right (your right) of the ivory house.
7. The Old Gold smoker owns snails.
8. Kools are smoked in the yellow house.
9. Milk is drunk in the middle house.
10. The Norwegian lives in the first house on the left.
11. The man who smokes Chesterfields lives in the house next to the man with the fox.
12. Kools are smoked in the house next to the house where the horse is kept.
13. The Lucky Strike smoker drinks orange juice.
14. The Japanese smokes Parliaments.
15. The Norwegian lives next to the blue house.

WHO DRINKS WATER? WHO OWNS THE ZEBRA?

FIGURE G1 A CONSTRAINT SATISFACTION PUZZLE

since deletions can combine with other constraints to trigger further deletions. Thus, from constraint 6, whenever ivory is eliminated as a possible color for one of the houses, green can be immediately eliminated as a possible color for the house immediately to the right. Moreover, by constraint 1, eliminating all but one of the possible colors for one house allows the elimination of all tuples in which that color is paired with a different house.

The constraint satisfaction mechanisms in MSYS provide an ideal way to efficiently propagate such deletions.

A. Solving the Problem in MSYS

The state space for the puzzle in Figure G-1 was represented in MSYS as canonic variables representing all the possible pairwise associations between different problem variables. The sextuple (H1 English RED KOOLS TEA DOG), for example, was partitioned into the following 15 MSYS variables.*

(H1 ENGLISH)	(H1 RED)	(H1 KOOLS)	(H1 TEA)	(H1 DOG)
	(ENGLISH RED)	(ENGLISH KOOLS)	(ENGLISH TEA)	(ENGLISH DOG)
		(RED KOOLS)	(RED TEA)	(RED DOG)
			(KOOLS TEA)	(KOOLS DOG)
				(TEA DOG)

A total of 375 tuples were required to represent the complete state space.

An initialization procedure was written to generate these variables and set their initial likelihoods to 1.0. The likelihood procedure associated with each tuple was a conjunction containing other tuples and disjunctions of tuples whose validity provided support for applicable constraints. Deletions were propagated by reevaluating the procedures of tuples in which a deleted tuple appeared.

Sets of tuples representing alternative associations for the same problem variables are again linked by OPTION variables. Each tuple appears in two OPTION variables, one for each element of the pair. For example, the tuple (H1 RED) is linked to all possible colors for house

*The five houses in left-right order are designated as H1----H5.

H1 by the variable (OPTION (H1 RED) (H1 YELLOW)----(H1 BLUE)) and to all possible houses associated with the color red, by the variable (OPTION (H1 RED) (H2 RED)----(H5 RED)). OPTION procedures again perform a variety of functions, such as checking for contradictions (where all alternatives have been eliminated) and updating the instantiation queue. They also play a crucial role in propagating deletions resulting from the elimination of all but one of their alternatives, as explained below. OPTION variables evaluate to the number of currently valid alternatives they encompass, initially 5.

MSYS proceeds towards a solution by deleting all tuples (i.e., setting their likelihood to 0.0) shown to be inconsistent with the given constraints. Constraints 2 through 15 were represented in MSYS by the procedural constraints ASSERT, RIGHTOF, and NEXTTO, as shown in Figure G-2. ASSERT operates by deleting all remaining alternatives in both OPTION variables associated with the asserted pair, For example, ASSERT(UKRANIAN TEA) deletes all tuples of the form (UKRANIAN $\neg$TEA) where $\neg$TEA designates any beverage other than TEA, as well as all those of the form ($\neg$UKRANIAN TEA).

The constraint RIGHTOF(IVORY GREEN) (specific arguments are used for clarity) states that a house cannot be IVORY if the house immediately to its right cannot be green. Similarly, a house cannot be colored GREEN unless the house adjacent on its left can be colored IVORY. This constraint operates by deleting all tuples of the form (HI GREEN), $1 < I < 5$ for which (HI-1 IVORY) is already eliminated, as well as all tuples of the form (HI IVORY) for which (HI+1 GREEN) is false. (H1 GREEN) and (H5 IVORY) are always eliminated. Any remaining, mutually supporting tuples of the form (HI IVORY), (HI+1 GREEN) are then added to each others likelihood procedures so that the subsequent deletion of either one will automatically trigger deletion of the other.

The constraint NEXTTO(KOOLS HORSE) operates in a similar fashion by deleting all variables of the form (HI KOOLS) when neither (HI-1 HORSE) or (HI+1 HORSE) are still valid. Similarly, all variables of the form (HI HORSE) unsupported by either (HI-1 KOOLS) or (HI+1 KOOLS) get deleted. Support clauses are then formulated for surviving instances of (HI KOOLS) and (HI HORSE) and added to their likelihood procedures. These clauses are expressed as disjunctions of the form (OR (HI-1 HORSE)(HI+1 HORSE)) and (OR (HI-1 KOOLS)(HI+1 KOOLS)) for instances of KOOLS and HORSE,

```
(ASSERT (@ (ENGLISH RED)))
(ASSERT (@ (SPANISH DOG)))
(ASSERT (@ (GREEN COFFEE)))
(ASSERT (@ (UKRAN TEA)))
(ASSERT (@ (OLDGOLD SNAILS)))
(ASSERT (@ (YELLOW KOOLS)))
(ASSERT (@ (H3 MILK)))
(ASSERT (@ (H1 NORWEG)))
(ASSERT (@ (LUCKYS OJ)))
(ASSERT (@ (JAPAN PARLIAMENT)))
(RIGHTOF (IVORY GREEN))
(NEXTTO (CHESTERFIELDS FOX))
NEXTTO (KOOLS HORSE))
(NEXTTO (NORWEG BLUE))
```

FIGURE G2 MSYS REPRESENTATION OF CONSTRAINTS

respectively. The disjunctions prevent deletion of the supported variable unless the required supporting interpretation has been eliminated at both adjacent houses.

Relatively few variables are directly eliminated by constraints 2 through 15. Far more inconsistencies are deduced by a process of elimination based on the uniqueness requirements expressed in constraint 1. MSYS capitalizes on uniqueness in two ways. First, whenever an OPTION procedure observes that all but one alternative has been eliminated, it eliminates variables associated with the surviving alternative through that survivor's other OPTION variable. For example, eliminating (H1 YELLOW), (H1 GREEN), (H1 IVORY), and (H1 BLUE) causes the OPTION variable associated with the color of H1 to delete all variables that are inconsistent with the surviving alternative (H1 RED); namely, (H2 RED) (H3 RED) (H4 RED) and (H5 RED).

Second, a tuple can be eliminated when both of its components are uniquely associated with different values of a third variable. Thus, the assertion of (ENGLISHMAN RED) and (COFFEE GREEN) requires elimination of (ENGLISHMAN COFFEE). The mechanism for propagating this kind of deletion is set up by the ASSERT procedure. A set of functions are created that compare pairs of OPTION variables, linking the two elements of the assertion with alternative bindings of a common third variable. For example, the ASSERTION (ENGLISHMAN RED) creates a function

```
(MATCH [OPTION (COFFEE ENGLISHMAN)----(WATER ENGLISHMAN)]
       [OPTION (COFFEE RED)----(WATER RED)])
```

for comparing the OPTION variables associating ENGLISHMAN and RED with alternative beverages. Similar expressions are created linking ENGLISHMAN and RED to alternative house numbers, pets, and cigarettes. The function MATCH compares corresponding alternatives (left to right) in a pair of OPTION variables and deletes any alternative whose opposite number is invalid. Evaulation of this function is automatically triggered whenever the value of either OPTION variable is altered by deletion of one of its alternatives. In particular, the elimination of (COFFEE RED) following the assertion of (COFFEE GREEN) triggers execution of the MATCH variable illustrated above, resulting in the elimination of (ENGLISHMAN COFFEE).

Execution of the constraints shown in Figure G-2 and propagation of their consequences, immediately eliminates 193 of the 375 original variables. Further state-space reductions must then be accomplished by instantiation. The search proceeds as in scene interpretation by popping from IQUEUE the OPTION variable with fewest remaining alternatives and then instantiating one of those alternatives.*

The system was fortunate in selecting a correct hypothesis (UKRANIAN BLUE) from the OPTION variable (OPTION (NORWEGIAN BLUE) (UKRANIAN BLUE)----(JAPAN BLUE)), for its first instantiation. Deductions propagated from this single assertion led directly to a successful termination consisting of the 75 surviving variables shown in Figure G-3. These variables provide unique associations between people, beverages, houses, house colors, pets, and cigarettes summarized in Figure G-4 that satisfy all problem constraints. (The reader might wish to try his own skill before peeking at the nationalities associated with Water and Zebra.) Altogether, the solution required almost 30 minutes of PDP-10 CPU time running in interpreted LISP. About one-half of this time was spent in setting up the data base and in the initial phase of constraint satisfaction prior to instantiation. The power of the constraint satisfaction approach utilized by MSYS compared with an exhaustive search is clear.

The OPTION variable used in the above instantiation was one of 35 candidates with three eliminated alternatives, all of equal priority. A less fortuitous instantiation might have eliminated fewer variables or even led to a contradiction. In either case, MSYS would then have explored in a separate context the consequences of denying the hypothesis. The search would then have continued, if necessary, in whichever context had the largest total number of eliminated variables. A few experiments were performed wherein incorrect hypotheses were purposefully asserted and correct ones denied. These errors led to immediate contradictions.

---

*In the absence of real valued Boolean likelihoods, the priority of OPTION variables on IQUEUE was based on the number of their alternatives already eliminated. OPTION variables, reduced to a unique alternative, were removed from the queue. The instantiation priority of alternatives within an OPTION variable was assigned arbitrarily based on their position in the OPTION expression.

(COFFEE ZEBRA) (OJ DOG) (MILK SNAILS) (TEA HORSE) (WATER FOX) (PARLIAMENT ZEBRA) (LUCKS DOG) (OLDGOLD SNAILS) (CHESTERFIELDS HORSE) (KOOLS FOX) (PARLIAMENT COFFEE) (LUCKYS OJ) (OLDGOLD MILK) (CHESTERFIELDS TEA) (KOOLS WATER) (GREEN ZEBRA) (IVORY DOG) (RED SNAILS) (BLUE HORSE) (YELLOW FOX) (GREEN COFFEE) (IVORY OJ) (RED MILK) (BLUE TEA) (YELLOW WATER) (GREEN PARLIAMENT) (IVORY LUCKYS) (RED OLDGOLD) (BLUE CHESTERFIELDS) (YELLOW KOOLS) (JAPAN ZEBRA) (SPANISH DOG) (ENGLISH SNAILS) (UKRAN HORSE) (NORWEG FOX) (JAPAN COFFEE) (SPANISH OJ) (ENGLISH MILK) (UKRAN TEA) (NORWEG WATER) (JAPAN PARLIAMENT) (SPANISH LUCKYS) (ENGLISH OLDGOLD) (UKRAN CHESTERFIELDS) (NORWEG KOOLS) (JAPAN GREEN) (SPANISH IVORY) (ENGLISH RED) (UKRAN BLUE) (NORWEG YELLOW) (H5 ZEBRA) (H4 DOG) (H3 SNAILS) (H2 HORSE) (H1 FOX) (H5 COFFEE) (H4 OJ) (H3 MILK) (H2 TEA) (H1 WATER) (H5 PARLIAMENT) (H4 LUCKYS) (H3 OLDGOLD) (H2 CHESTERFIELDS) (H1 KOOLS) (H5 GREEN) (H4 IVORY) (H3 RED) (H2 BLUE) (H1 YELLOW) (H5 JAPAN) (H4 SPANISH) (H3 ENGLISH) (H2 UKRAN) (H1 NORWEG)

FIGURE G-3 FINAL STATE OF DATA BASE

	HOUSES				
	H1	H2	H3	H4	H5
NATIONALITY	NORWEGIAN	UKRANIAN	ENGLISH	SPANISH	JAPANESE
COLOR	YELLOW	BLUE	RED	IVORY	GREEN
PET	FOX	HORSE	SNAILS	DOG	ZEBRA
CIGARETTE	KOOLS	CHESTERFIELD	OLD GOLD	LUCKYS	PARLIAMENT
DRINK	WATER	TEA	MILK	OJ	COFFEE

FIGURE G4 SUMMARY OF ASSOCIATIONS BETWEEN PROBLEM VARIABLES PROVIDED BY VARIABLES IN FIGURE G3

REFERENCES

1. J. M. Tenenbaum et al., "An Interactive Facility for Scene Analysis Research," SRI Artificial Intelligence Center, Technical Note 87, Stanford Research Institute, Menlo Park, California (December 1973).

2. Y. Yakimovsky and J. Feldman, "A Semantics-Based Decision Theoretic Region Analyzer," Proc. 3IJCAI, pp. 580-588 (1973).

3. K. J. Turner, "Computer Perception of Curved Objects Using a Television Camera," Ph.D. Dissertation, Department of Machine Intelligence, School of Artificial Intelligence, University of Edinburgh (1974).

4. N. Nilsson, "Artificial Intelligence--Research and Applications," pp. 82-127, Contract DAHC04-72-C-0008, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (April 1974).

5. P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions, Sys. Sci. and Cybernetics, pp. 100-107, Vol. SSC-4, No. 2 (July 1968).

6. L. Zadeh, "Fuzzy Logic and Approximate Reasoning," Memorandum No. ERL-M479, Electronics Research Laboratory, University of California, Berkeley, California (November 1974).

7. Y. Y. Yakimovsky, "On the Recognition of Complex Structures: Computer Software Using AI Applied to Pattern Recognition," Proceedings International Joint Conference on Pattern Recognition, pp. 345-353, Copenhagen, Denmark (August 1974).

8. H. Barrow and R. Popplestone, "Relational Descriptions in Picture Processing," Machine Intelligence, Vol. 6 (Edinburgh University Press, 1971).

9. E. H. Shortliffe and B. G. Buchanan, "A Model of Inexact Reasoning in Medicine," Mathematical Biosciences, pp. 351-379, Vol 23 (1975).

10. P. H. Winston, "Learning Structural Descriptions from Examples," MAC-TR-76, Massachusetts Institute of Technology, Cambridge, Massachusetts (September 1974).

11. W. H. Paxton, "A Best-First Parser," Proceedings IEEE Symposium on Speech Recognition, Carnegie-Mellon University, Pittsburgh, Pennsylvania, p. 218 (April 1974). Technical Note 92, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (May 1974).

12. R. Duda, P. Hart, and N. Nilsson, "Subjective Bayesian Methods for Rule-Based Inference Systems," Technical Note 124, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (January 1976).

13. T. D. Garvey and J. M. Tenenbaum, "On the Automatic Generation of Programs for Locating Objects in Office Scenes," Proceedings Second International Joint Conference on Pattern Recognition, Copenhagen, Denmark (August 1974).

14. L. Erman and V. Lesser, "A Multilevel Organization for Problem Solving Using Many Diverse, Cooperating Sources of Knowledge," Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania (March 1975).

15. U. Montanari, "Optimization Methods in Image Processing," Proceedings IFIP-74, Stockholm, Sweden (August 1974).

16. N. Nilsson, Problem Solving Methods in Artificial Intelligence, Chapter 3 (McGraw Hill Book Company, New York, 1971).

17. D. G. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," AI TR-271, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (August 1972).

18. U. Montanari, "Networks of Constraints: Fundamental Properties and Applications to Picture Processing," Information Sciences, 7, pp. 95-132 (1974).

19. A. Rosenfeld, A. Hummel, and S. Zucker, "Scene Labeling by Relaxation Operations," TR-379, Computer Science Department, University of Maryland (1975).

20. R. O. Duda, "Some Current Techniques for Scene Analysis," Technical Note 46, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (October 1970).

21. A. Guzman, "Analysis of Curved Line Drawings Using Context and Global Information," Machine Intelligence, Vol. 6, B. Meltzer and D. Michie, eds., pp. 325-376 (Edinburgh University Press, Edinburgh, Scotland, 1971).

22. E. Sacerdoti et al, "QLISP: A Language for the Interactive Development of Complex Systems," Technical Note 120, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (March 1976).

23. C. Hewitt, "PLANNER: A Language for Manipulating Models and Proving Theorems in a Robot," AI Memorandum 158, Massachusetts Institute of Technology, Cambridge, Massachusetts (August 1970).

24. P. H. Winston, "The MIT Robot," Machine Intelligence, Vol. 7, pp. 431-463, B. Meltzer and D. Michie, eds., (Edinburgh University Press, Edinburgh and American Elsevier Publishing Company, 1972).

25. M. Minsky, "A Framework for Representing Knowledge," The Psychology of Computer Vision, P. H. Winston, ed. (McGraw Hill Book Company, New York, 1973).

26. J. D. Ichbiah and S. P. Morse, "General Concepts of the SIMULA 67 Programming Language," Companie Internationale pour le Informatique, Paris, France, September, 1971.

27. D. Chamberlain, "The Single Assignment Approach to Parallel Processing," Proceedings Fall Joint Computer Conference (AFIPS Press, 1971).

28. J. M. Tenenbaum and H. G. Barrow, "Experiments in Interpretation Guided Segmentation," Technical Note 123, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (March 1976).

29. John Gaschnig, "A Constraint Satisfaction Method for Inference Making," Proceedings of Twelfth Allerton Conference on Circuit and Systems Theory, University of Illinois (October 1974).

30. R. E. Fikes, "REF-ARF: A system for Solving Problems Stated as Procedures, "Artificial Intelligence, Vol. 1(1), Technical Note 14, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (September 1969).

31. P. H. Winston, The Psychology of Computer Vision, p. 8 (McGraw Hill Book Company, New York, 1975).

32. J. S. Brown, "Sophie--A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting," Report No. 2790, Bolt Beranek and Newman, Inc. (March 1974).

33. W. A. Wickelgren, How to Solve Problems, (W. H. Freeman and Company, 1974).