Computer Science 4150
Artificial Intelligence
Winter, 2005

Instructor:
Nick Cercone - CSBldg 105
902-494-2832 - nick@cs.dal.ca

Winston's Learning Program

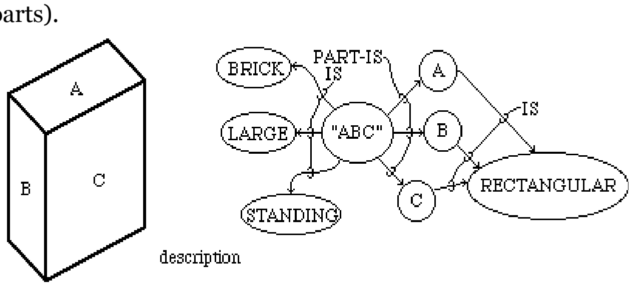
Winston's doctoral thesis at MIT entitled "Learning Structural Descriptions from Examples" (1970) was a major step towards a clarification of how concepts involving complex structural relationships might be learned.

His program is presented with line drawings of scenes containing children's toy blocks, such as bricks, cubes, pyramids, and wedges. The program forms descriptive networks for these scenes, which shows the properties and relationships of the objects appearing in them. Using these structural descriptions, the program can learn structural concepts such as "pedestal", "arch" or "arcade" on the basis of examples and counterexamples of the concepts.

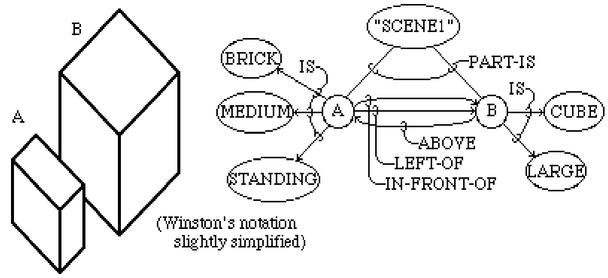
While the content of the scenes is severly restricted, the methods employed in the program seem quite general, or at least readily generalisible to more realistic concept-formation tasks (consider "chair", "table", "house", etc.) Certainly this type of structural learning is a long way off from learning by parameter adjustment, which has been the dominant paradigm in pattern recognition research (Uhr & Vossler's program and some others excepted).

Winston's program uses Guzman's algorithm to determine the bodies in a scene; it then determines which edges belong to which object and fills in partially occluded edges. Then it infers the types of objects (brick, wedge, etc.) from the shapes and adjacency relationships of the viable faces. The sizes and orientation are then readily available.

The description in the following example is in the form of a "semantic network". The nodes are particular things (such as the object "ABC", and its faces "A", "B", and "C") or general concepts (such as BRICK, LARGE, etc.) and the edges are relations between things and/or general concepts (e.g., PART-IS is a relation which holds between a thing and its parts).

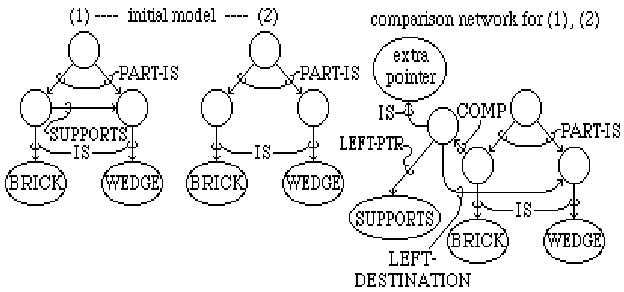


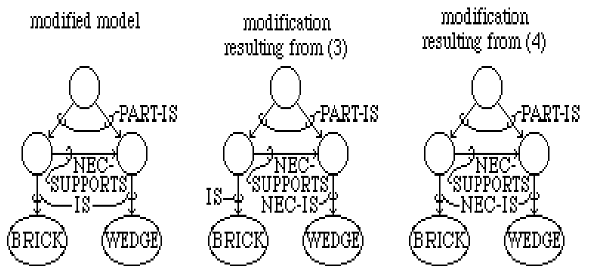
Next various heuristic routines are applied to the scene to obtain relationships *between* bodies, expecially *support* relationships and also relative position (above, behind, to the left of). The information thus obtained is again represented in descriptive network form. The final scene description includes these relations as well as the overall attributes of the objects determined earlier (brick, wedge, etc.) but not the finer details such as component faces and their shapes.



The initial model formed by the system is simply a description of the first true instance of the concept. The model is generalised for subsequent examples so that it will accept any new instances and reject non-instances (near misses). So that the system doesn't have too difficult a time determining which features of a non-instance disqualify it, the non-instances are required to be fairly close to true instances. Each modification of the model is made by generalising a *comparison network* for the current model and the given instance or non-instance.This comparison network describes the similarities nd differences between the model and the new example. The descriptions of scenes (1) and (2), their comparison network, and subsequent modifications of the model are shown for the "house" sequence below.

Note the absence of the SUPPORTS pointer in the second network. Basically this is the difference described in the comparison network, which says that there is an extra pointer in the "left' network, labelled SUPPORTS and with destination "the node for the wedge". The modified model says there is *necessarily* a support relation between the two parts of the scene. This modified model will reject (2) as an instance of a house. Similarly (3) causes a reinforcement of the "wedge" property of the supported object by the necessaryy operator, and (4) causes a reinforcement of the "brick" property of the supporting object. Just as certain features of the model can be reinforced through counterexamples, others can be relaxed when true instances are presented which differ from the model. For example, if a sample house were presented with a pyramidal rather than a "wedge" shaped roof, the NEC IS WEDGE requirement for the roof would be relaxed and replaced by the pair of alternatives MAY BE WEDGE and MAY BE PYRAMID. Now if the counterexample with a brick shaped roof were again presented, the requirement NEC-IS-NOT-BRICK would be added to the existing alternatives.





The program can be faulted on the many ad-hoc decisions it embodies, both in analysing scenes and in the learning process. Also there are logical inadequacies in the network formalism (disjunction and quantification can not be represented). Nevertheless, the program incorporates theoretically interesting ideas (the network formalism and representation and the algorithm for comparison networks) and exhibits nontrivial learning behaviour.

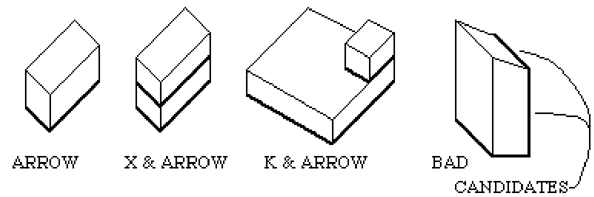
Method for Determining Object Properties and Relations

Object Properties

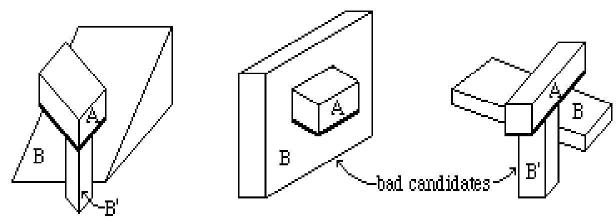
(a) edges belonging to an object
   1.- exterior edges belong to object if bounded by object
   2.- at object-object boundaries, T verticies are used to assign edges to objects
   3.- bottom collinear line segments are connected
(b) shape of faces - triangle, quadrilateral, hexagon, octagon, etc.
(c) size of an object - tiny (.5% of visual area), small, medium (1.5-5% of visual area), large, huge (35-100% of visual area)
(d) orientation - standing, lying (applicable to bricks only)

Object Relations

(a) left-of, right-of: A is to the left-of B is centre of area of A is left-of centre of area of B and the rightmost part of A is left-of the rightmost part of B.
(b) above, below (similar to (a))
(c) supports, supported-by: first find base line of objects: if the lower end and an interior line of an object lies at an exterior vertex, the exterior edges radiating from that vertex are candidate base lines. Example



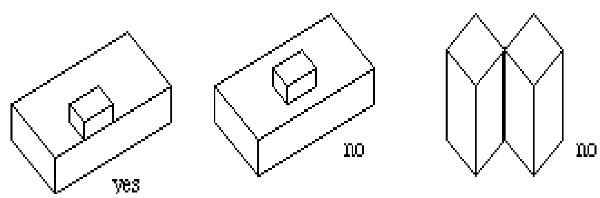
Eliminate both candidates if one candidate is more vertical than the interior line concerned. There are various refinements, e.g., extension of base lines through matched T's. Now if any object B is unbounded by a base line of an object A, B is a possible support of A. Examples



Eliminate "vertical slides" (using vertical edges in supposed supporting surface) and eliminate mismatched-height cases.

(d) in front-of, behind: an object bounded by a line belonging to another object and not a base line is behind that other object; there are many refinements.

(e) marries - common face common edge. Examples



X, K, and T verticies common to 2 or more objects are used to determine the "marries" property.