Knowledge in Learning

Andreas Wichert

DEIC

(Página da cadeira: Fenix)

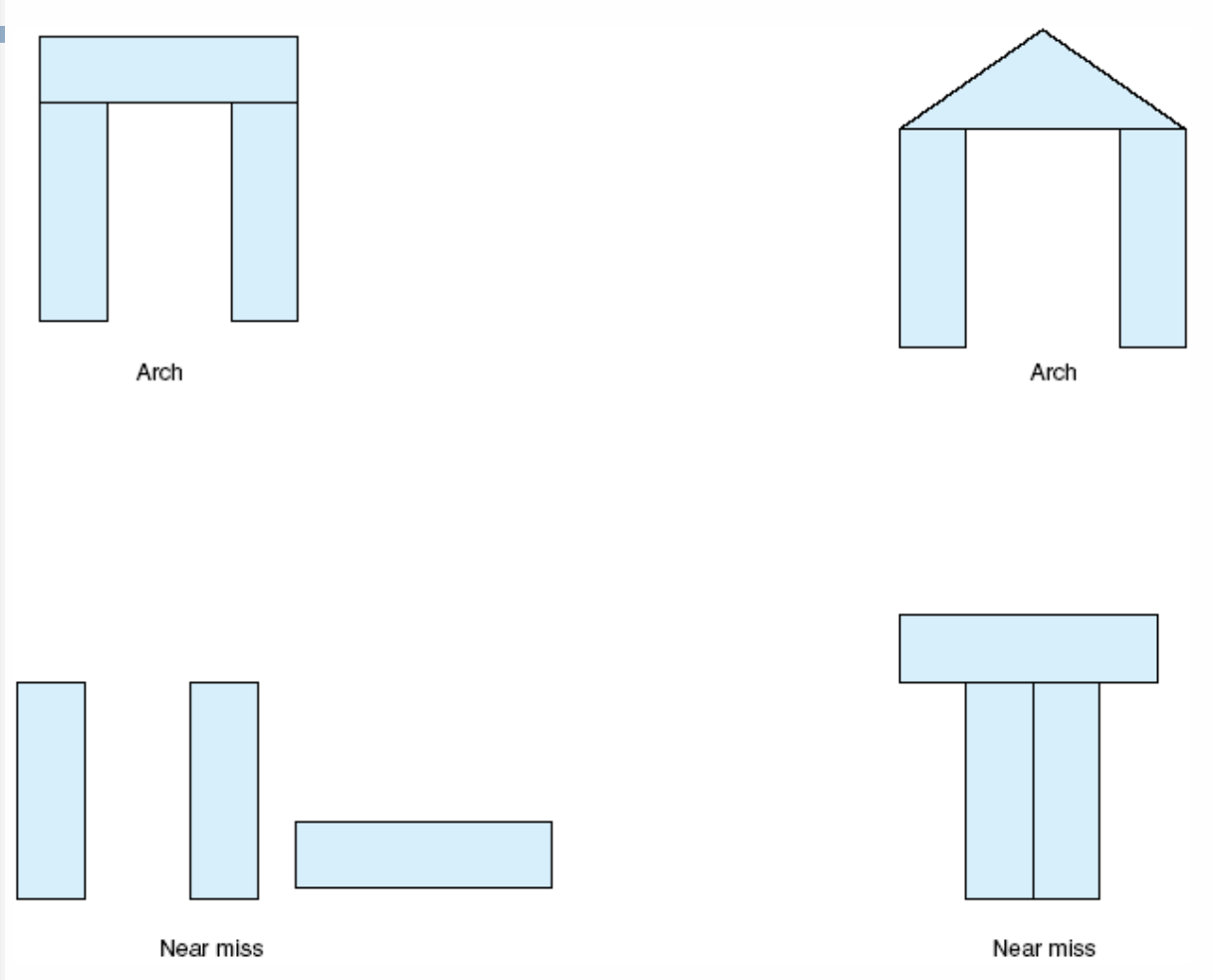
Machine Learning

- *"Statistical Learning"*
  - *Clustering, Self Organizing Maps (SOM)*
  - *Artificial Neural Networks, Kernel Machines*
  - Neural Networks (Biological Models)
  - *Bayesian Network*
- *Inductive Learning (ID3)*
- **Knowledge Learning**
- **Analogical Learning**
- **SOAR: Model of Cognition and Learning**

Knowledge and Learning

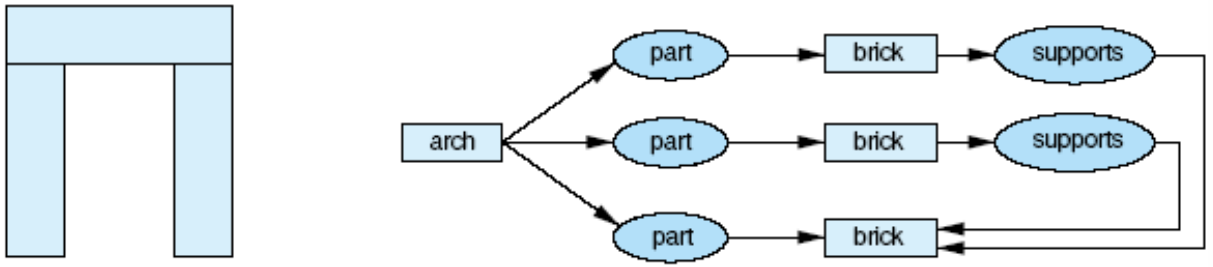
- Current best hypotheses (Differneces, Correct)
  - Version search space
- Last commitment search
  - EBL
  - RBL
  - ILP
- *Analogical Reasoning Case Based*

An Example (Patrick Winston-1975)

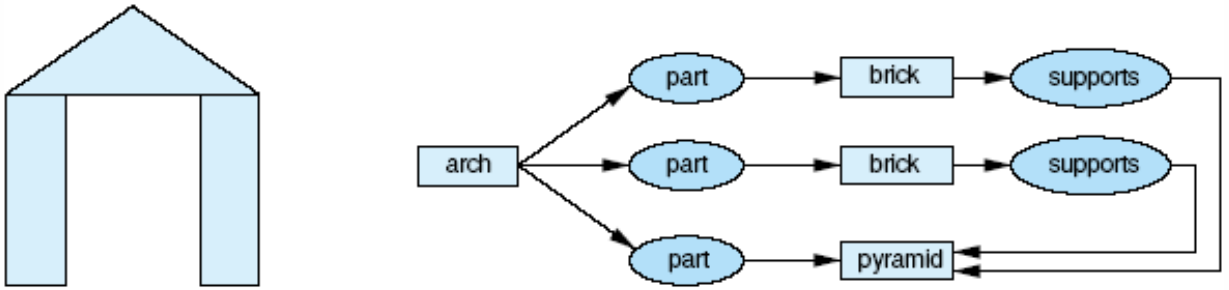


An Example (Patrick Winston-1975)

a. An example of an arch and its network description

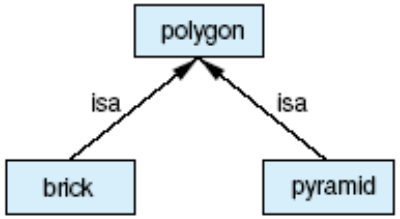


b. An example of another arch and its network description

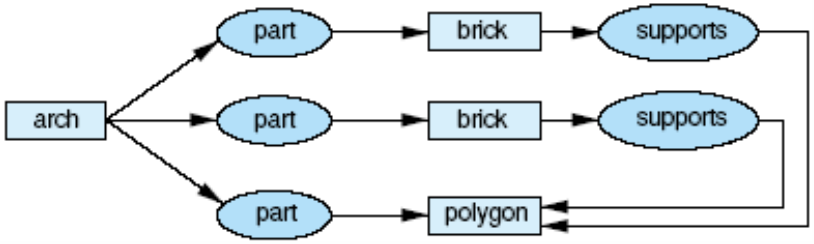


An Example (Patrick Winston-1975)

c. Given background knowledge that bricks and pyramids are both types of polygons

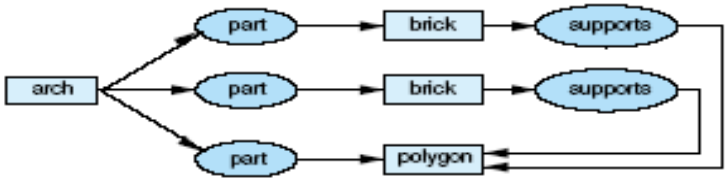


d. Generalization that includes both examples

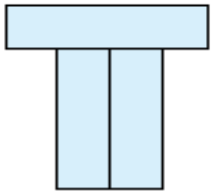


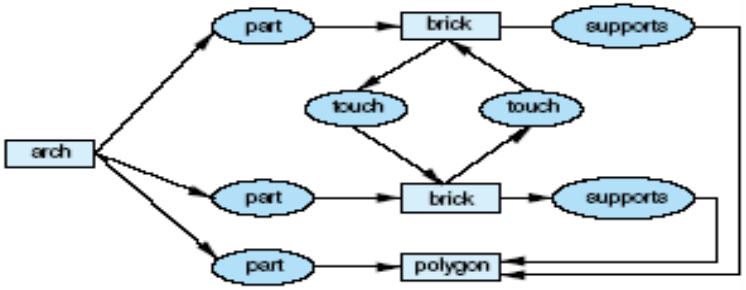
An Example (Patrick Winston-1975)

a. Candidate description of an arch

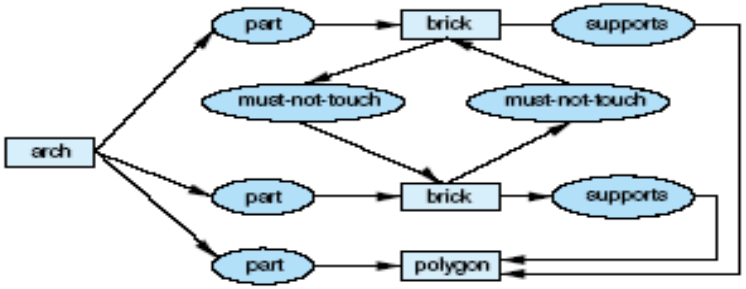


b. A near miss and its description

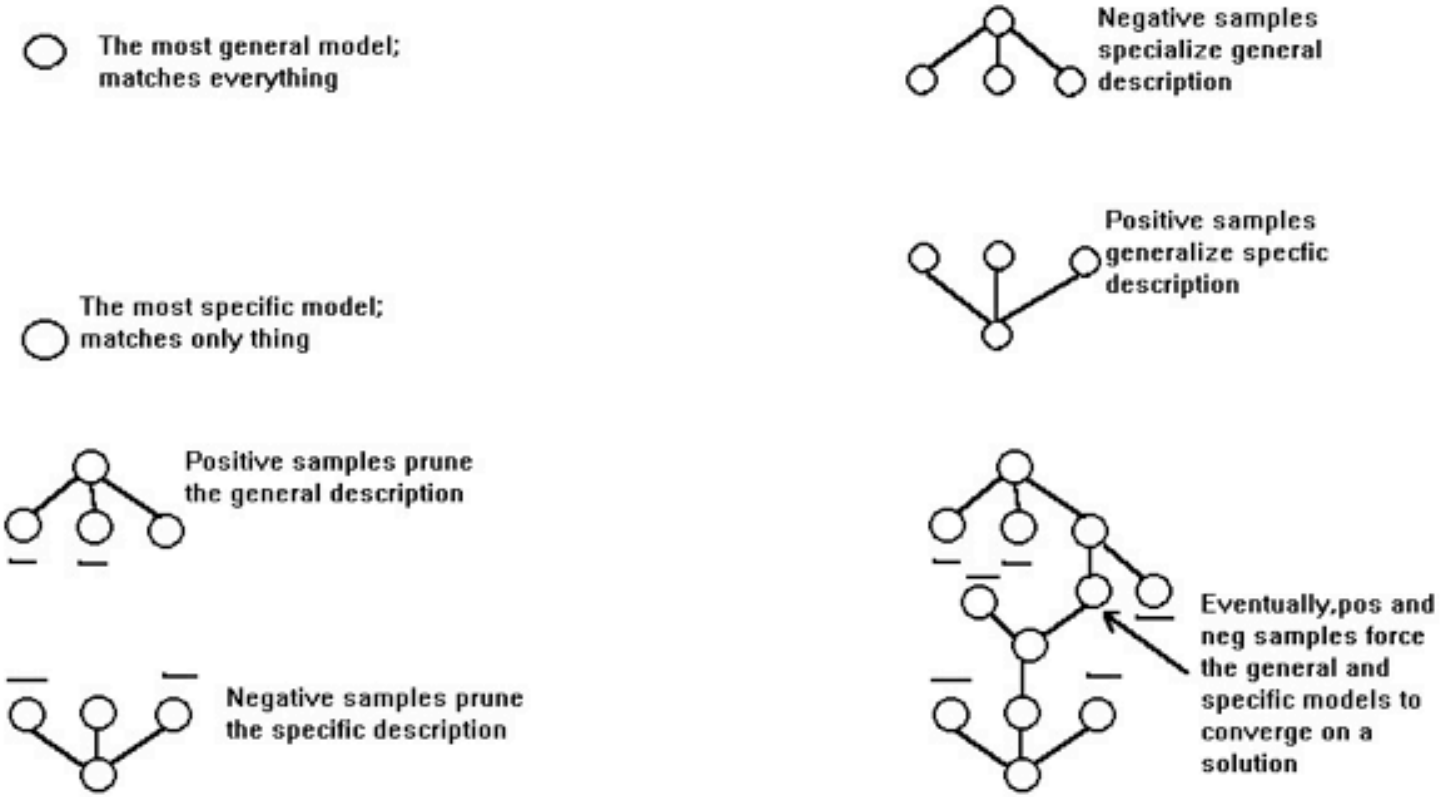




c. Arch description specialized to exclude the near miss

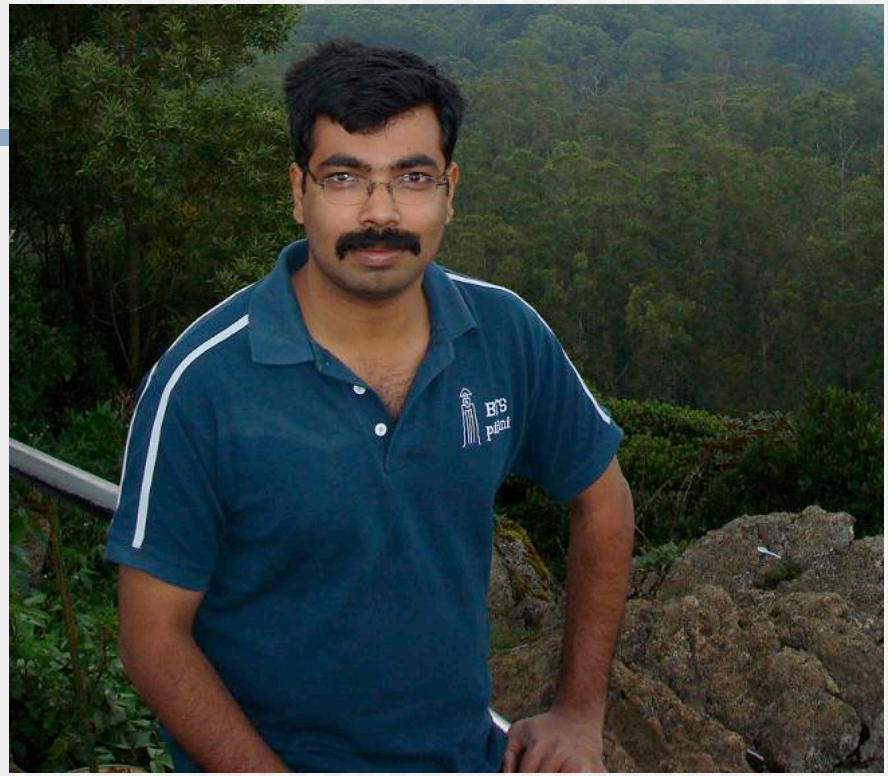


Version Spaces: Learning by managing multiple models



1. Each time a general model is specialized, that specialization must be a generalization of an existing specific model. Similarly each time a specific model is generalized, that generalization must be a specialization of an existing general model
2. Each time a general model is specialized, that specialization must not be a specialization of another general model

EBL



- S.P.Vimal
- CSIS Group
- BITS-Pilani

- http://discovery.bits-pilani.ac.in/discipline/csis/vimal/

Explanation Based Learning (EBL)

- ID3 generalize on the basis of training data
  - Similarity based algorithm
  - Generalization is a function of similarities across training examples
  - No assumptions about the semantics of the domain
- EBL uses domain (prior) knowledge to guide generalization

EBL

- Prior domain knowledge in learning
  - Similarity based algorithms relies large amount of training data …
  - A set of training examples can support an unlimited number of generalizations

EBL

- EBL uses explicitly represented domain theory to construct an explanation of a training example
  - It's a proof that the example logically follows from the theory
- Benefits
  - Filters noise
  - Selects relevant aspects of experience
  - Organizes training data into a systematic and coherent structure

EBL

Starts with…

- A target concept
  - The concept to which the definition is sought
- A training example
  - An instance of target
- A domain theory
  - Set of rules and facts that are used to explain how the training example is an instance of goal concept
- Operational Criteria
  - Some means of describing the form that the definition may take

Learn when an object is a cup

- Target concept:
  - premise (x)→ cup(x)

- Domain Theory:
  - liftable(x)∧holds_liquid(x)→cup(x)
  - part(z,w) ∧concave(w) ∧points_up(w)→holds_liquid(z)
  - light(y) ∧part(y,handle)→liftable(y)
  - small(a)→light(a)
  - made_of(a,feathers)→light(a)

EBL

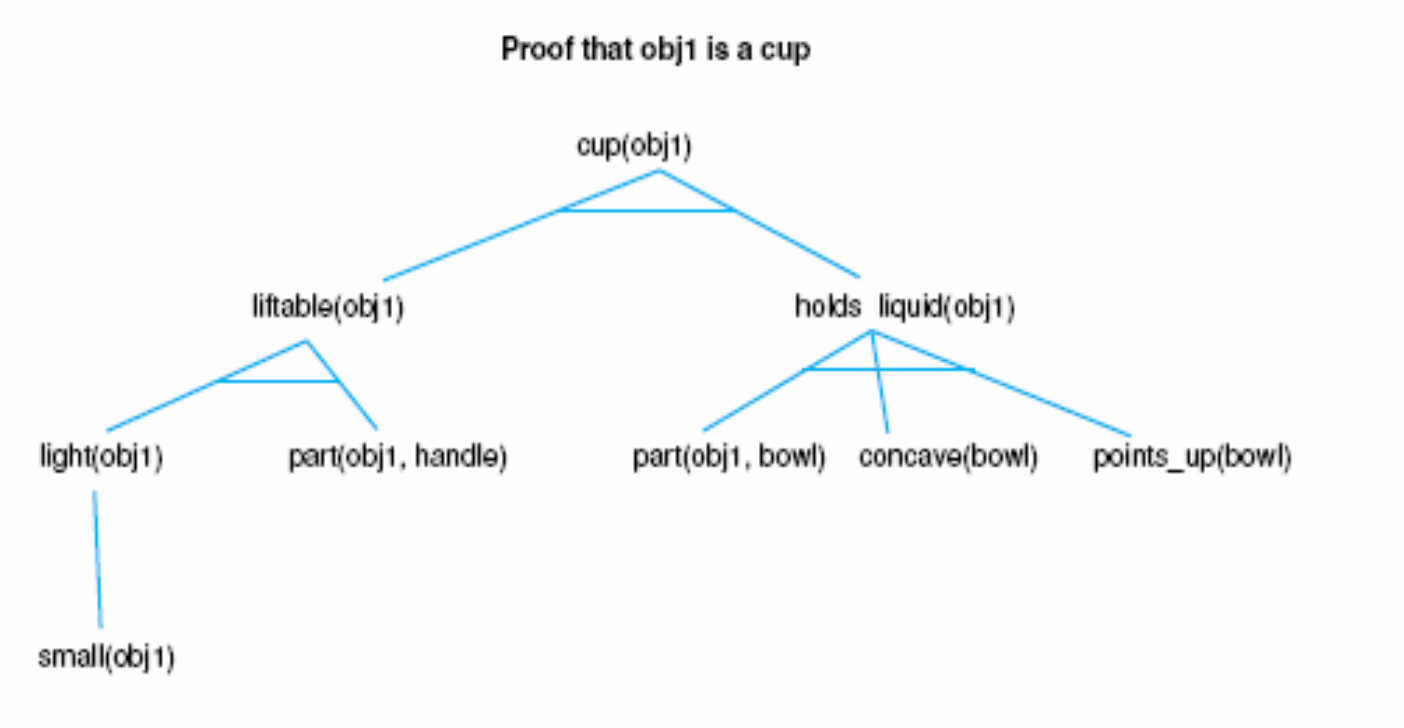
- Training Example:
  - cup(obj1)
  - small(obj1)
  - part(obj1,handle)
  - owns(bob,obj1)
  - part(obj1,bottom)
  - part(obj1,bowl)
  - points_up(bowl)
  - concave(bowl)
  - color(obj1,red)

EBL

- Operational criteria:
  - Target concept to be defined in terms of observable, structural properties of objects

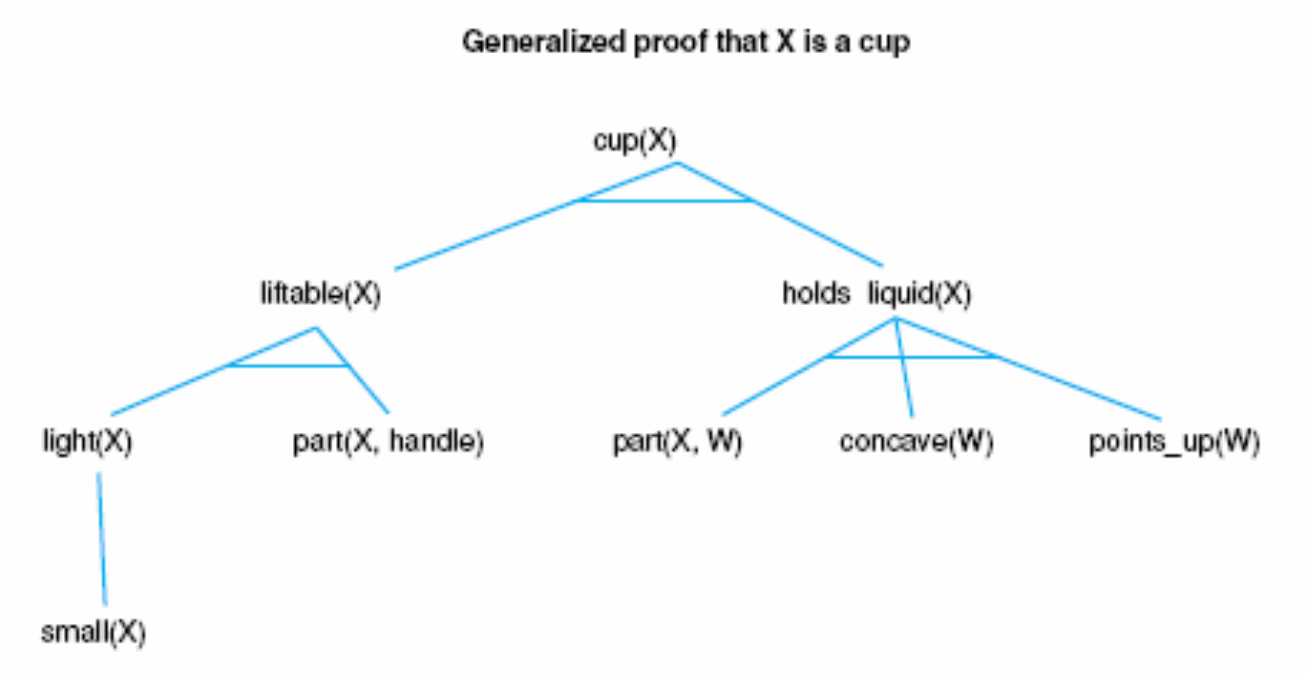
EBL

- Construct an explanation of why the example is an instance of the training concept



EBL

- Generalize the explanation to produce a concept definition that may be used to recognize other cups
- Substitute variables for those constants in the proof tree that depend solely on particular training instance



EBL

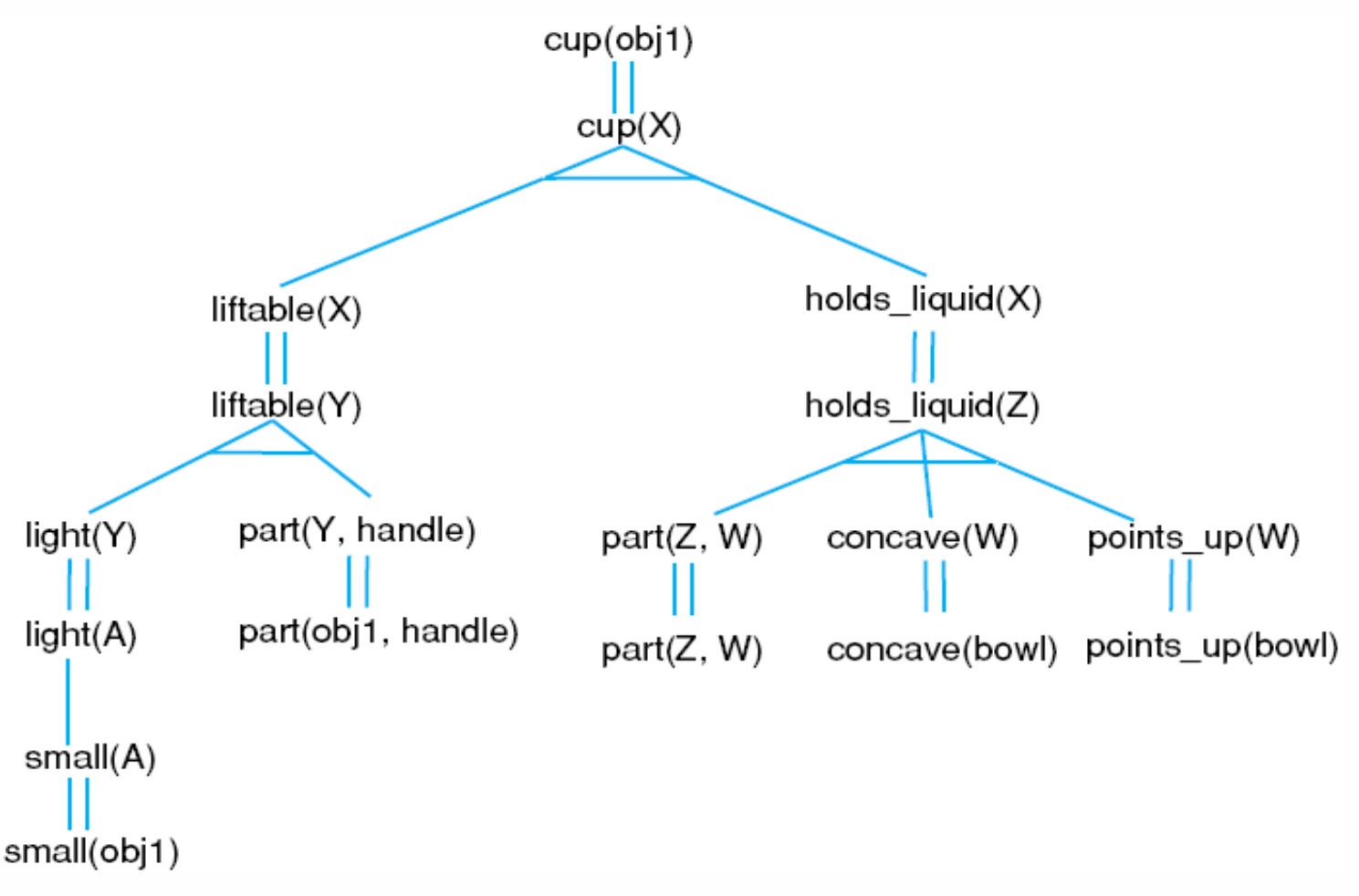
- Based on the generalized tree, EBL a new rule whose conclusion is the root of the tree and whose premise is the conjunction of leaves

  small(X)$\wedge$part(X,handle) $\wedge$part(X,W) $\wedge$concave(W)$\wedge$points_up(W) $\rightarrow$cup(X)

EBL- An example (contd…)

- Alternate way of constructing generalized proof tree:
  - Explanation structure (Dejong, Moone - 1986)
  - Maintains two parallel list of substitutions
    - To explain the training example
    - To explain the generalized goal

EBL (two parallel lists)



EBL, in short…

- From the training example, the EBL algorithm computes a **generalization** of the example that is **consistent** with **the goal concept** and that meets the operational criteria
  - a description of the appropriate form of the final concept)

EBL, An Issue…

- Require domain theory needs to be complete and consistent.
- *Complete* (knowledge):
  - An agent knows all possibly relevant information about its domain
  - *Closed world assumption*, under which any fact not known to the agent can be taken to be false
- *Consistent* (knowledge):
  - The environment can be assumed to change much slower (if at all) with respect to the speed of the reasoning and learning mechanisms

EBL, some more of it…

- Inferred rules could have been inferred from the KB without using training instance at all
- The function of a training instance is to focus the theorem prover on relevant aspects of the problem domain
  - Speedup learning / knowledge based reformulation
- Takes information implicit in a set of rules and makes it explicit

EBL some more ….

- **EBL** extracts general rules from single examples by explaining the examples and generalizing the explanation
- **RBL** uses prior knowledge to identify the relevant attributes
- **KBIL** finds inductive hypotheses that explain sets of observations with background knowledge
- **ILP** techniques perform KBIL using knowledge expressed in first-order logic

Analogical Reasoning

- Assumption
  - If two situations known to be similar in some respects, it is likely that they are similar in others
- Useful in applying existing knowledge to new situations
  - Ex: Teacher tells the student, that
    - Electricity is analogous to water
    - Voltage corresponding to pressure
    - Amperage corresponding to the amount of flow
    - Resistance to the capacity of pipe
    - ……..

Model for Analogical Reasoning

- Source of analogy
  - A theory that is relatively well understood
- Target of analogy
  - A theory that is not completely understood
- Analogy
  - Mapping between corresponding elements of the target and source

- Switch ⇔ Valve
- Amperage ⇔ Quantity of flow
- Voltage ⇔ Water Pressure
- _____ ⇔ Capacity of a pipe

Model for Analogical Reasoning

- Framework for computational model for analogical reasoning consists of stages
  - **Retrieval**
    - Identify those corresponding features in source & target
    - Index knowledge according to those features
  - **Elaboration**
    - Identify those additional features of the source
  - **Mapping and Inference**
    - Map corresponding features in source & target
  - **Learning**
    - Learn the target & store it for further learning

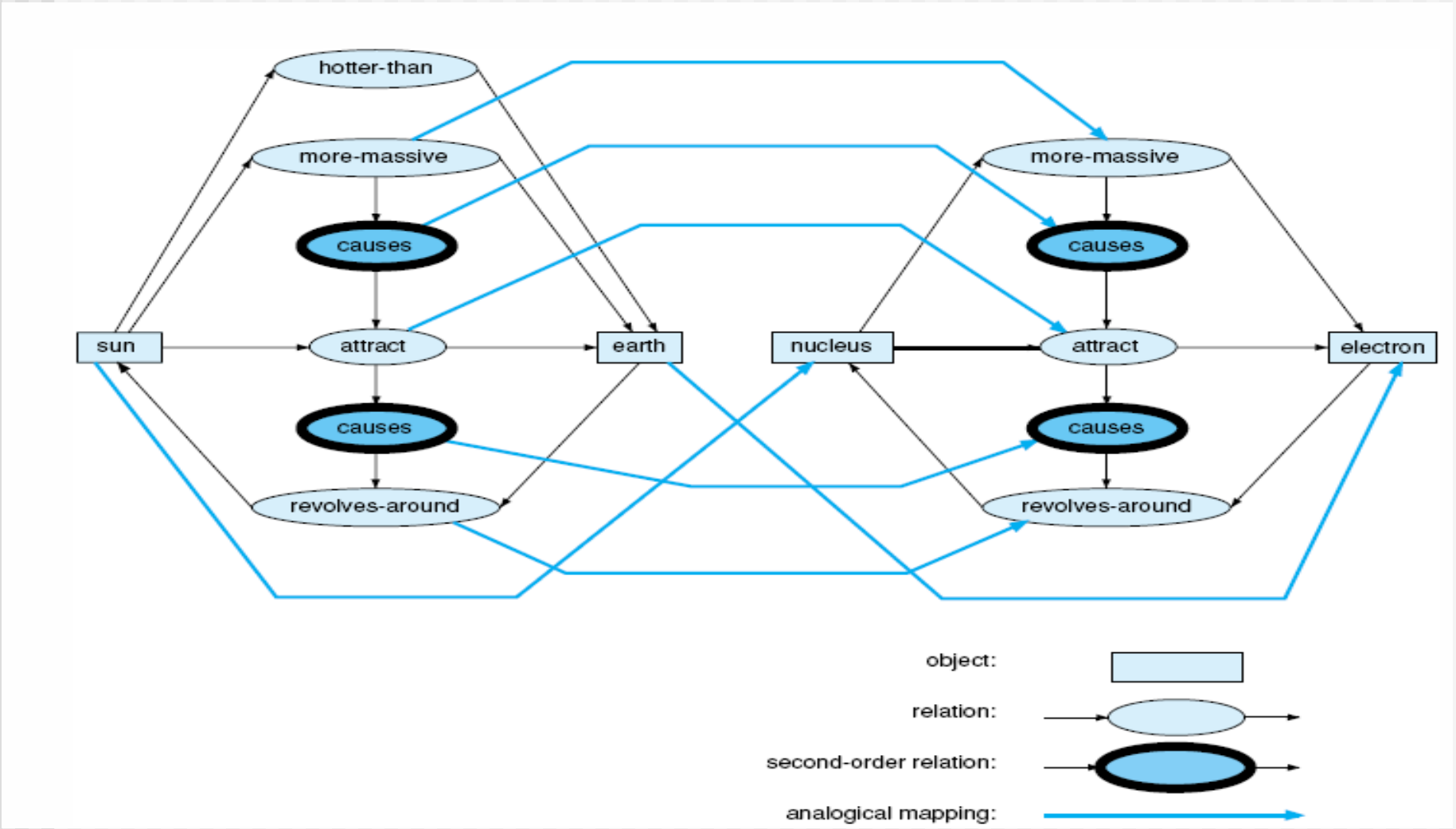
An Example

- **Systematicity**
  - Analogy should emphasize those systematic, structural features over a superficial analogies
    - ~~The atom is like solar system~~
    - The sunflower is like sun
- Source domain predicates
  - yellow(sun)
  - blue(earth)
  - hotter-than(sun,earth)
  - causes(more-massive(sun,earth), attract(sun,earth))
  - causes(attract(sun,earth), revolves-around(earth,sun))
- Target domain that the analogy is intended to explain
  - more-massive(nucleus, electron)
  - revolves-around(electron,nucleus)

An Example

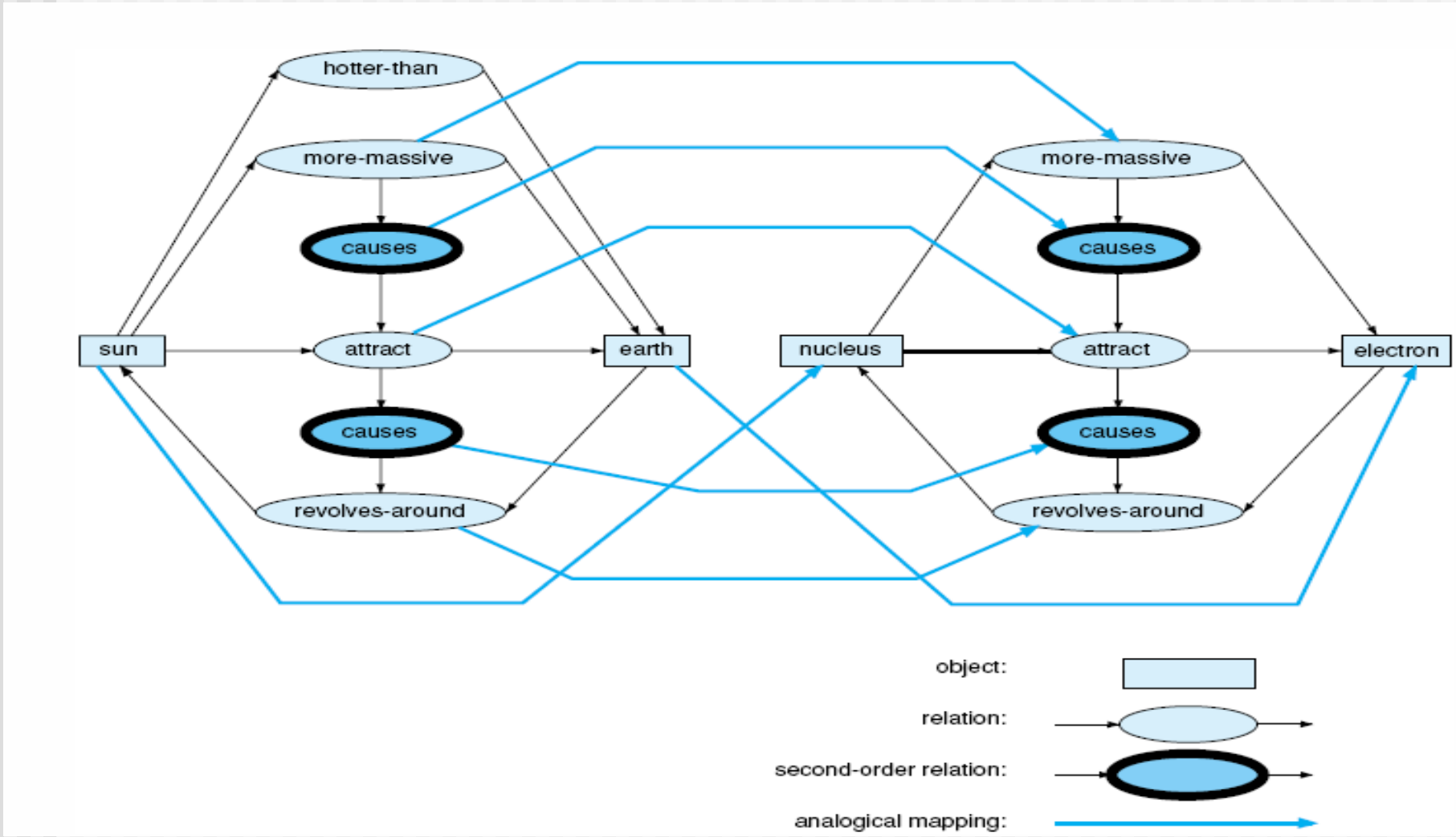
- Some constraints
  - Drop properties from source
  - Relations map unchanged from the source to target, arguments may differ
  - Focus higher order relations in mapping

An Example


hotter-than
more-massive
causes
attract
causes
revolves-around
sun
earth
nucleus
more-massive
causes
attract
causes
revolves-around
electron
object:
relation:
second-order relation:
analogical mapping:

An Example

- Resultant mapping
  - sun→neucleus
  - earth→electron
- Extending the mapping leads to the inference
  - causes(more-massive(neucleus, electron), attract(neucleus, electron))
  - causes(attract(neucleus, electron), revolves-around(electron, neucleus))
- Structure mapping described is not a complete theory of analogy


hotter-than
more-massive
causes
sun
attract
earth
causes
revolves-around
more-massive
causes
nucleus
attract
electron
causes
revolves-around
object:
relation:
second-order relation:
analogical mapping:

Verbal Categories

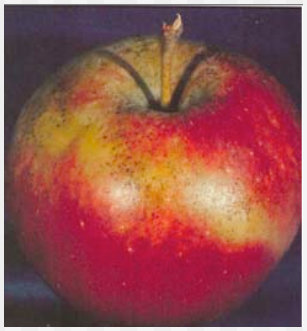
- Objects can be described by a set of discrete features, such as red, round and sweet
- The similarity between them can be defined as a function of the features they have in common
- An object is judged to belong to a verbal category to the extent that its features are predicted by the verbal category

- The similarity of a category *Ca* and of a feature set *B* is given by the following formula

$$Simc(Ca,B) = \frac{|Ca \cap B|}{|Ca|} - \frac{|(Ca - (Ca \cap B)|}{|Ca|} = \frac{2}{|Ca|} \cdot |Ca \cap B| - 1 \ \in [-1,1]$$

- Not symmetrical

- Psychological experiments suggest that unfamiliar categories are judged more equal to familiar then the other way around
  - For example, pomegranate is judged more similar to apple by most people than apple is to pomegranate



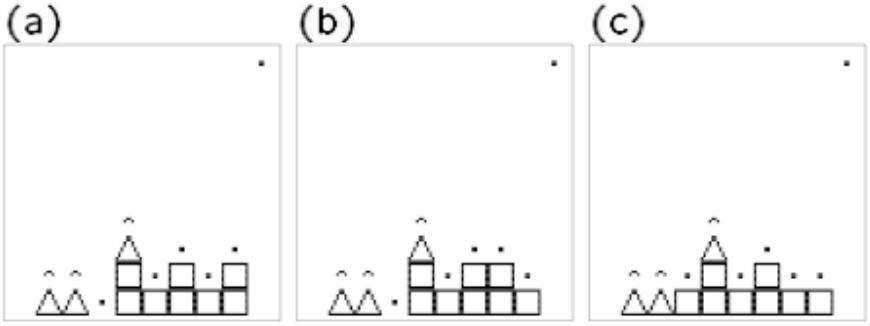
- The category *bird* is defined by the following features: flies, sings, lays eggs, nests in trees, eats insects
- The category *bat* is defined by the following features: flies, gives milk, eat insects The following features are present: flies and gives milk
- Following features are present: flies and gives milk

$$Simc(\mathbf{bird}, present\,features) = \frac{1}{5} - \frac{4}{5} = \frac{2}{5} \cdot 1 - 1 = -\frac{3}{5}$$

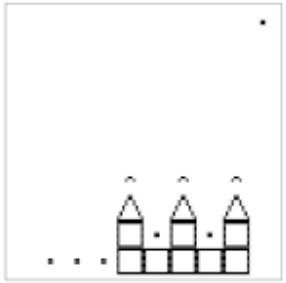
$$Simc(\mathbf{bat}, present\,features) = \frac{2}{3} - \frac{1}{3} = \frac{2}{3} \cdot 2 - 1 = \frac{1}{3}$$

$$Simc(\vec{C}, \vec{f}) = \frac{2}{c} \cdot \sum_{j=1}^{n} C_j f_j - 1$$

Visual categories



Similarity



of tower to (a) 0.81

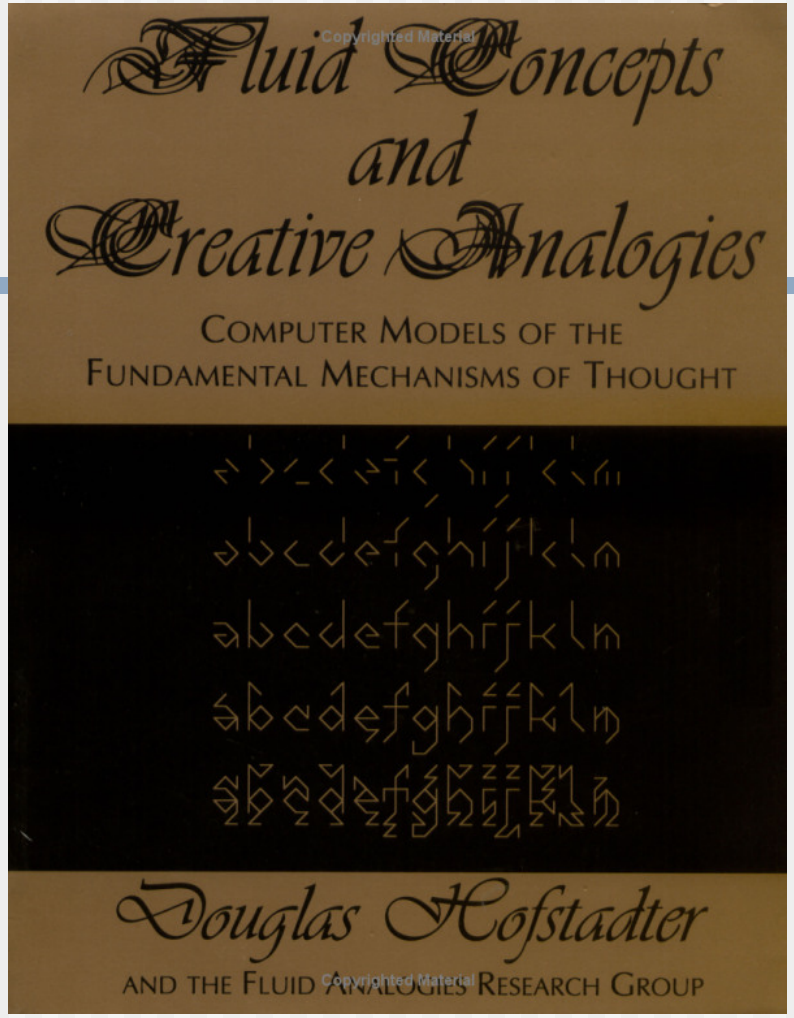
of tower to (b) 0.7

of tower to (c) 0.7

EBL vs. Similarity

Starts with…

- A target concept
  - The concept to which the definition is sought
- A training example
  - An instance of target
- **A domain theory** (Similarity, Geometry?)
  - **Set of rules and facts that are used to explain how the training example is an instance of goal concept**
- Operational Criteria (Similarity, Geometry?)
  - Some means of describing the form that the definition may take



- Hofstadter, Douglas. Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought. Basic Books. 1995

What is an „A“ ?

- What makes something similar to something else (specifically what makes, for example, an uppercase letter 'A' recognisable as such)

- Metamagical Themas, Douglas Hoffstader, Basic Books, 1985

- What is the essence of dogness or house-ness?
- What is the essence of 'A'-ness?
- What is the essence of a given person's face, that it will not be confused with other people's faces?
  - How to convey these things to computers, which seem to be best at dealing with hard-edged categories--categories having crystal-clear, perfectly sharp boundaries?

Analogy

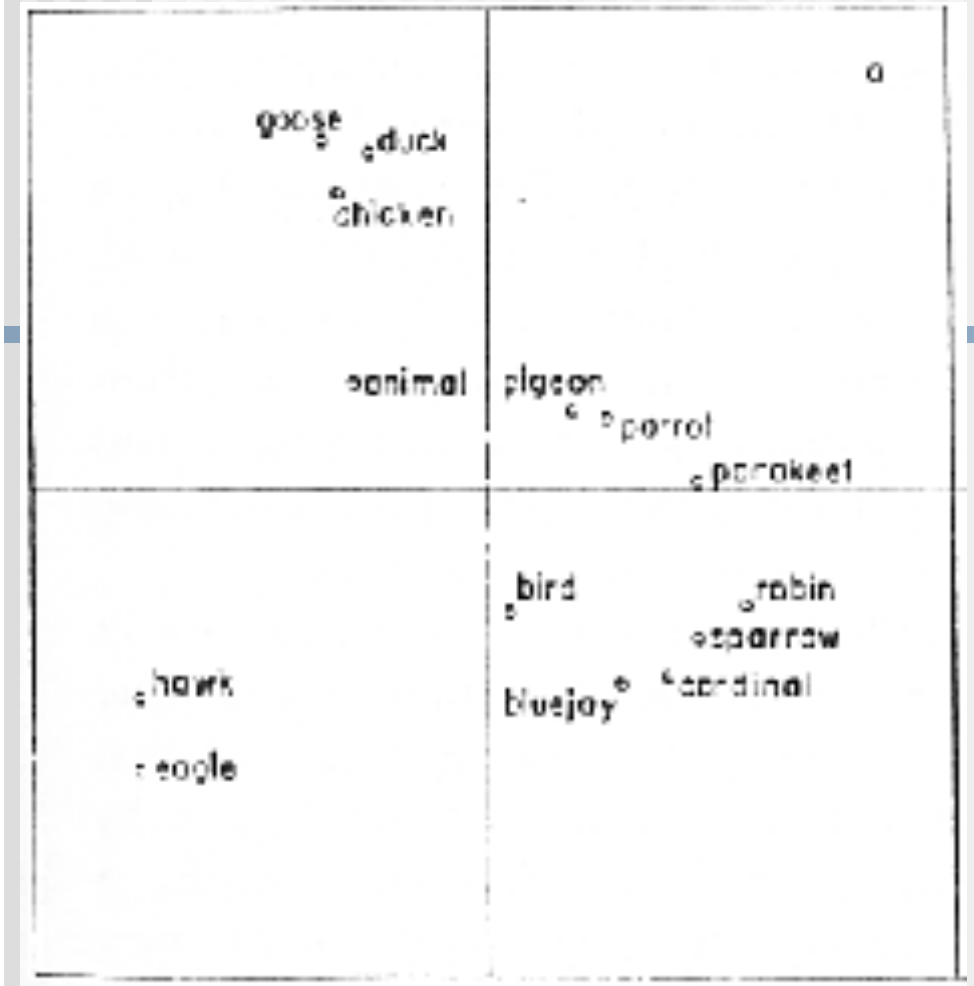
- Types of Comparison
  - Similarity: overall resemblance across all attributes
    - Alligator is like crocodile
    - Useful for generic, knowledge-light induction
  - Analogy: similarity of relational structure
    - The solar system is like an atom
    - Useful for powerful, knowledge-based induction

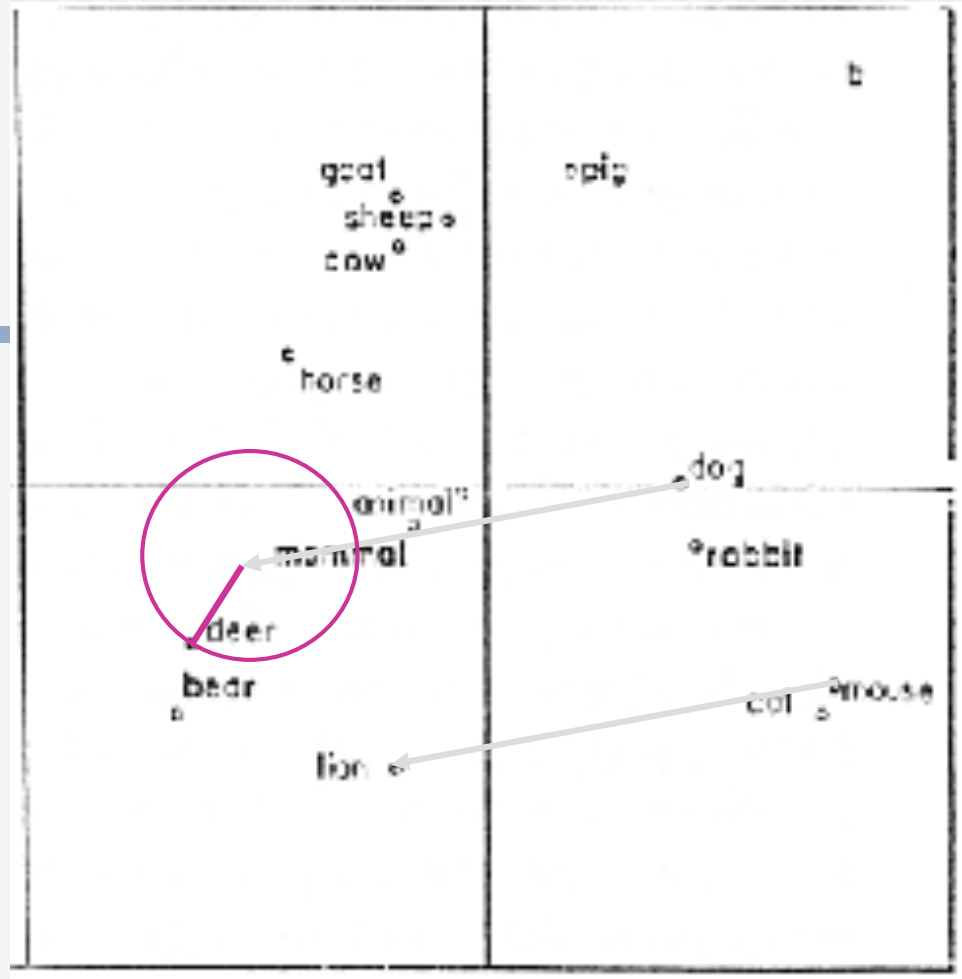
Stages of analogical reasoning

- Retrieval
  - Rutherford's description of an atom reminds one of the solar system
- Mapping
  - Electron -> Planet, Nucleus -> Sun
- Inferences
  - Rotation of earth around sun is caused by gravity, so maybe gravity causes electrons to rotate around nucleus of atom

Analogy with Multi-dimensional Scaling

- Parallelogram rule
  - Create MDS space with many terms
  - Problem: A:B: :C:? (e.g. person:skin::tree:?)
- Find vector from A to B in MDS space
- Find item that produces the most similar vector (angle and length) going from C to the A->B vector





MDS can find geometric representations for abstract data sets

Mouse:Lion::Dog:?

Answer: Deer

Multidimensional Scaling (MDS)

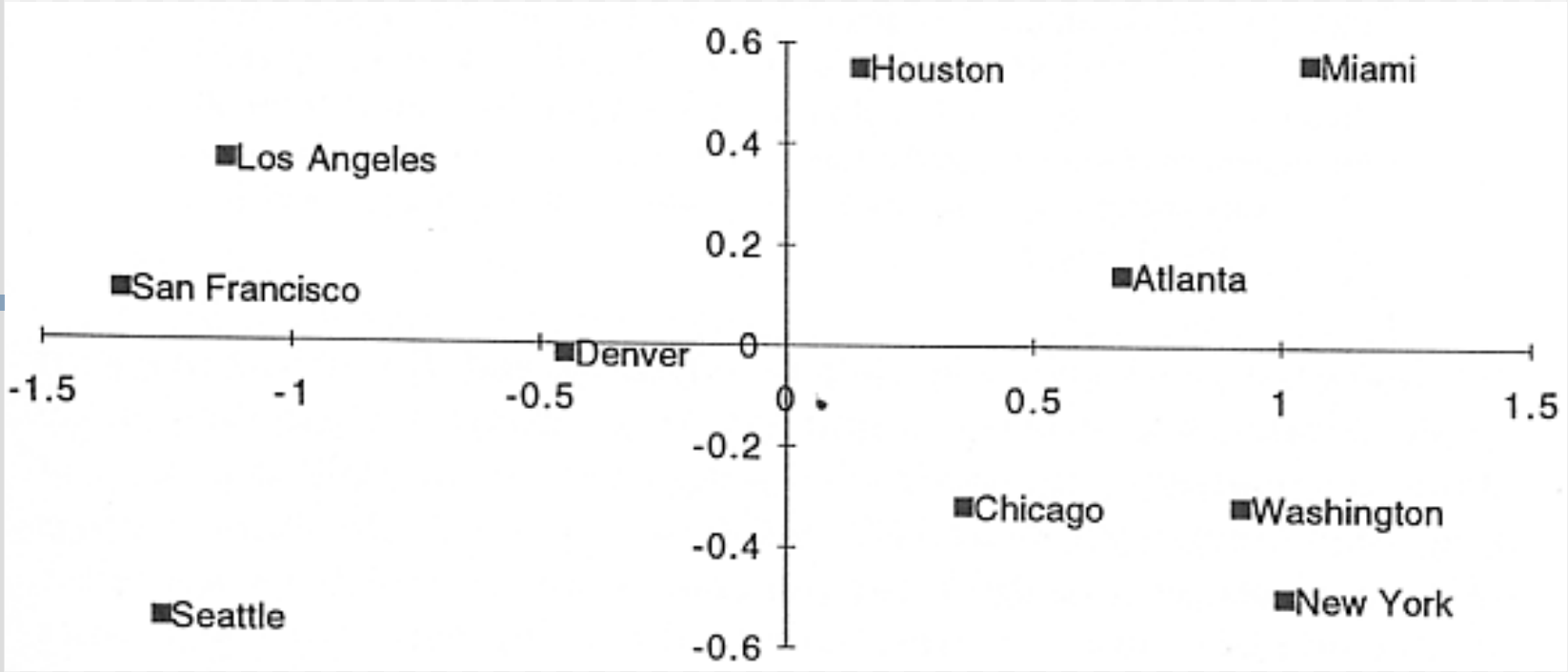
- Input: a distance or proximity matrix that describes how close every object in a set is to every other object
  - N objects are represented by N(N-1)/2 numbers (distances)
- Output: a geometric representation where every objects is represented as a point in D-dimensional space
  - N objects are represented by ND numbers (coordinates)
  - Each object is represented as a point in space

Purposes of MDS

- Information compression if D<N
  - Efficient codes that capture most of data
- Reveal the dimensionality of a data set
- Give psychological interpretations to the dimensions
- Automatically create numeric representations for any kind of object

TABLE 2.1
Flying Distances Between 10 Cities in the United States

	Atlanta	*Chicago*	*Denver*	*Houston*	*Los Angeles*	*Miami*	*New York*	*San Francisco*	*Seattle*	*Washington, DC*
Atlanta										
Chicago	587									
Denver	1,212	920								
Houston	701	940	879							
Los Angeles	1,936	1,745	831	1,374						
Miami	604	1,188	1,726	968	2,339					
New York	748	713	1,631	1,420	2,451	1,092				
San Francisco	2,139	1,858	949	1,645	347	2,594	2,571			
Seattle	2,182	1,737	1,021	1,891	959	2,734	2,408	678		
Washington, DC	543	597	1,494	1,220	2,300	923	205	2,442	2,329	



MDS recovers absolute original locations for the objects (subject to rotation and reflection) from the distances

Structure Mapping Engine

- Input: a propositional description for each of two situations
- Output: an alignment between the two descriptions
- Stages
  - 1) Create all possible correspondences between scenarios, dropping out attributes, and requiring corresponding relations to be identical
  - 2) combine local correspondences into consistent, connected clusters

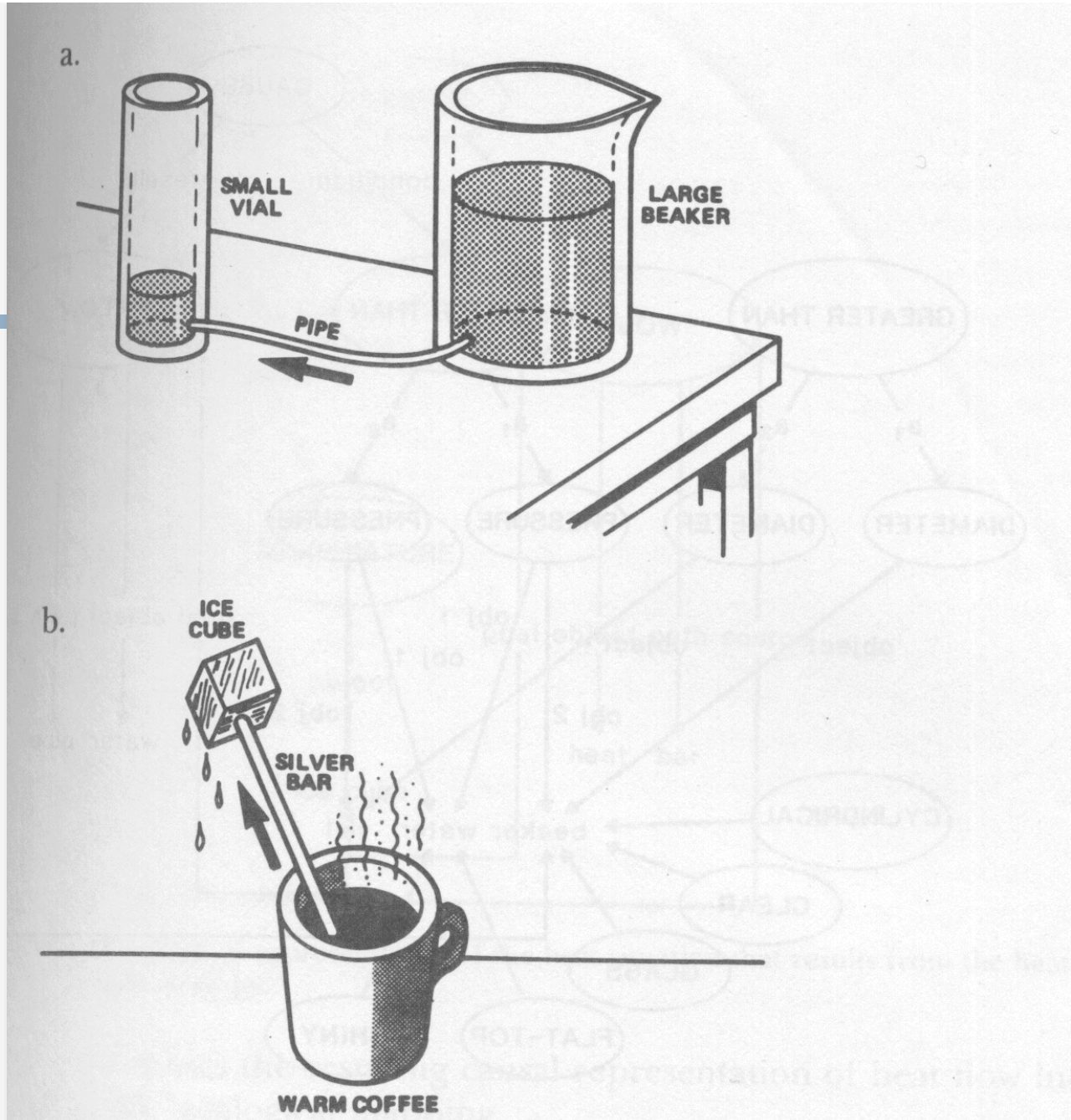


Figure 7.1. Examples of physical situations involving (*a*) water flow and (*b*) heat flow (adapted from Buckley, 1979, pp. 15–25).

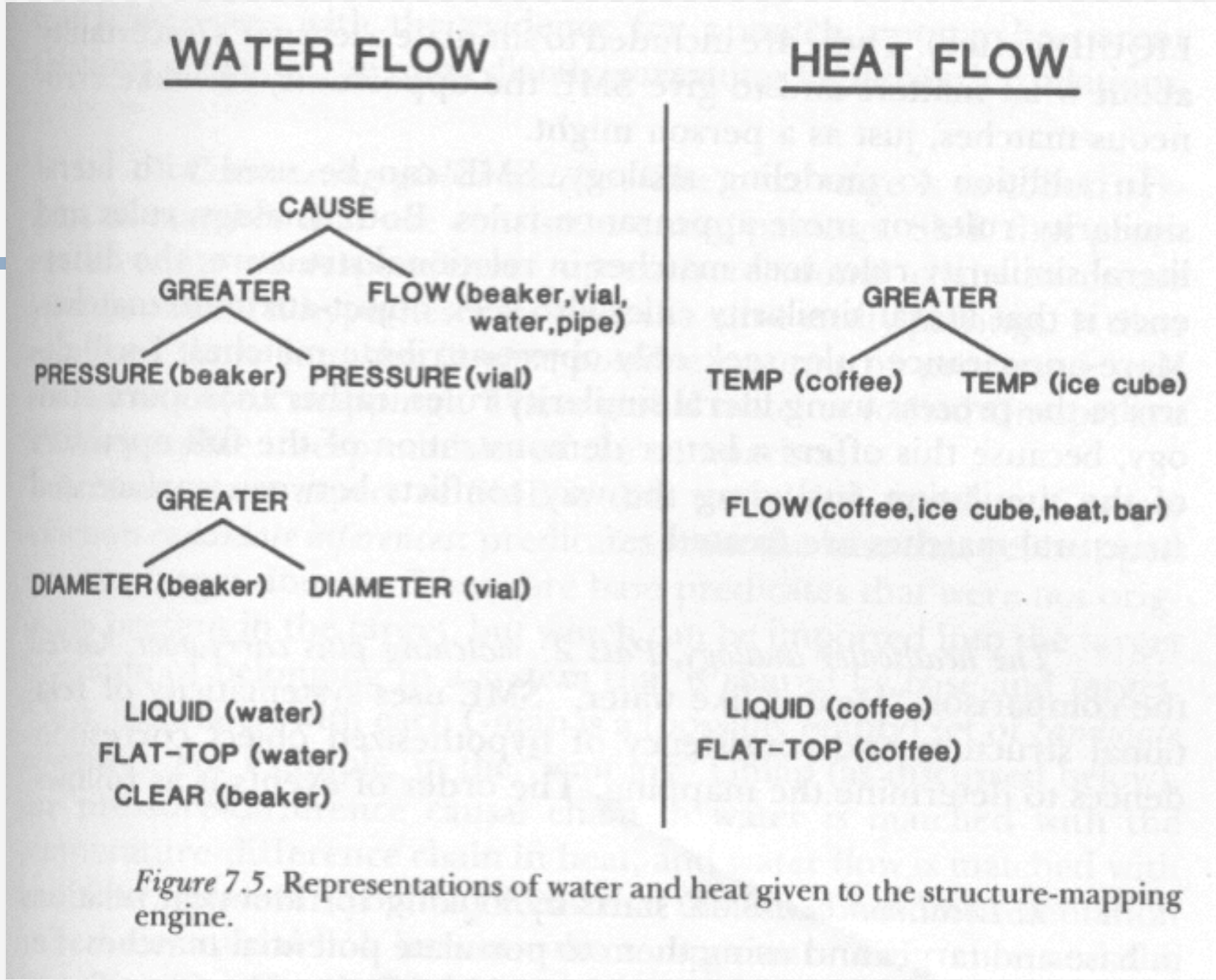
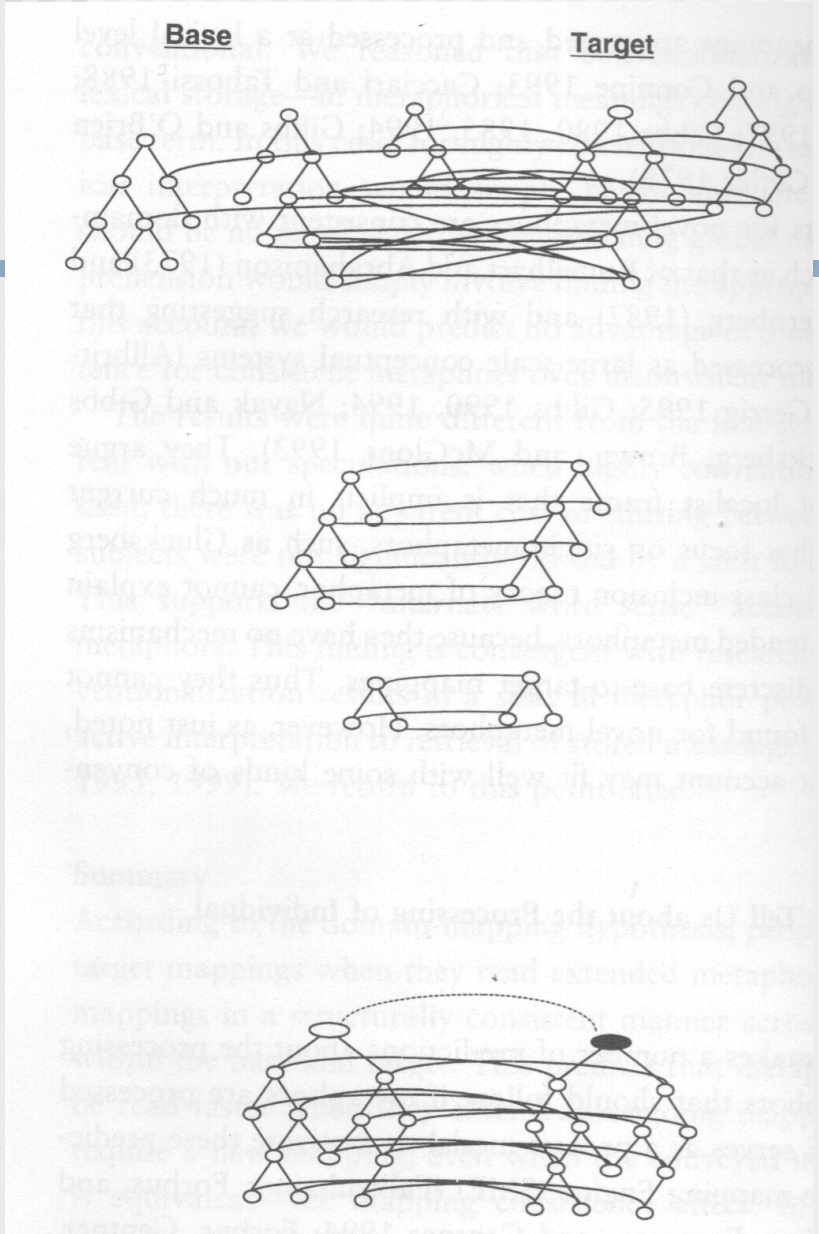


Figure 7.5. Representations of water and heat given to the structure-mapping engine.

Stages



Create local matches

Structural coalescence into consistent mappings

Small structures combine into maximal interpretation; candidate inferences

Copycat

- Central insight: create representations for situations at the same time that one is comparing the situations
  - Much of the action in real analogizing is finding the correct representation
- Letter analogies
  - Simple micro-world rather than “real” analogies taken from science

Architectural elements

- Slipnet
  - Semantic network showing activation of different concepts
    - Examples: predecessor, successor, opposite, group, length
- Distances between concepts changes as processing continues

The Slipnet

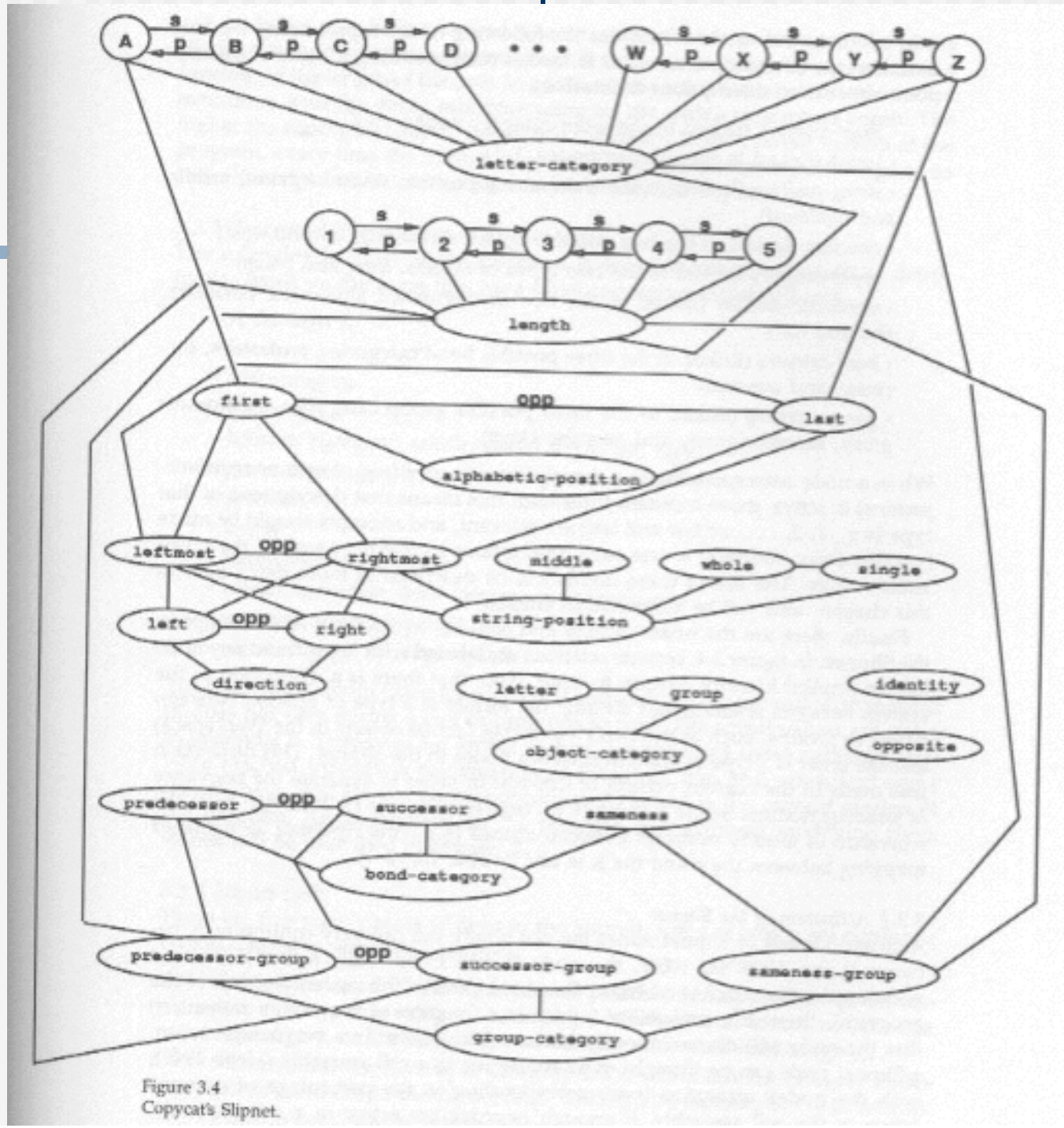
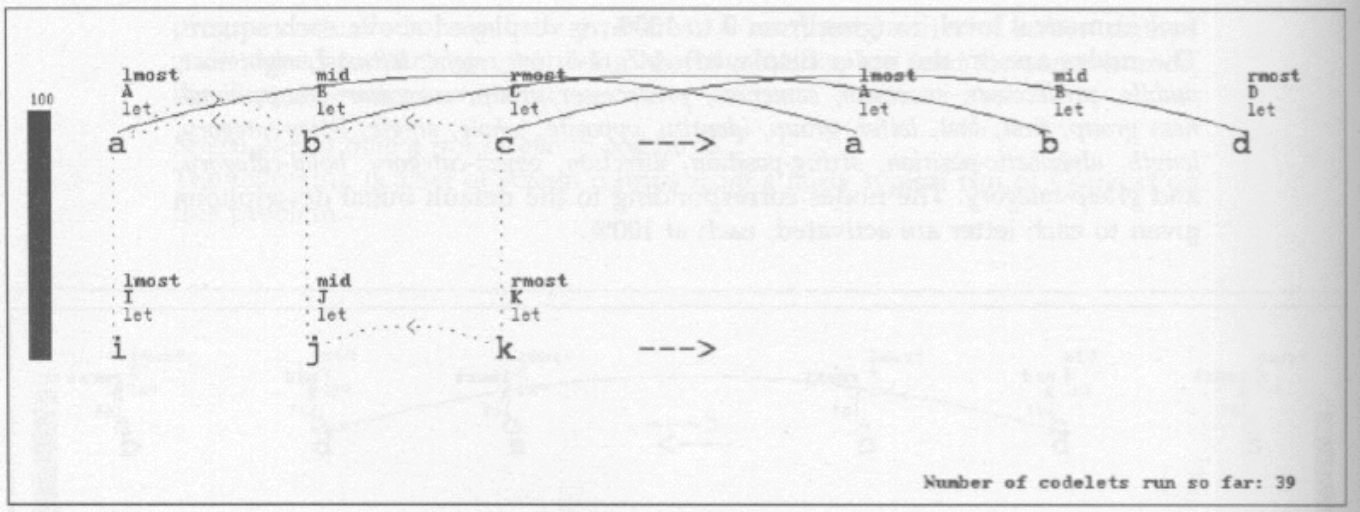


Figure 3.4
Copycat's Slipnet.

Architectural elements

- Codelets
  - Each does a small task like create or dissolve a group or notice a relation
  - Parallel terraced scan - pursue the most promising route given the current structure found
    - Several paths pursued simultaneously, with speed of pursuit based on likelihood of success



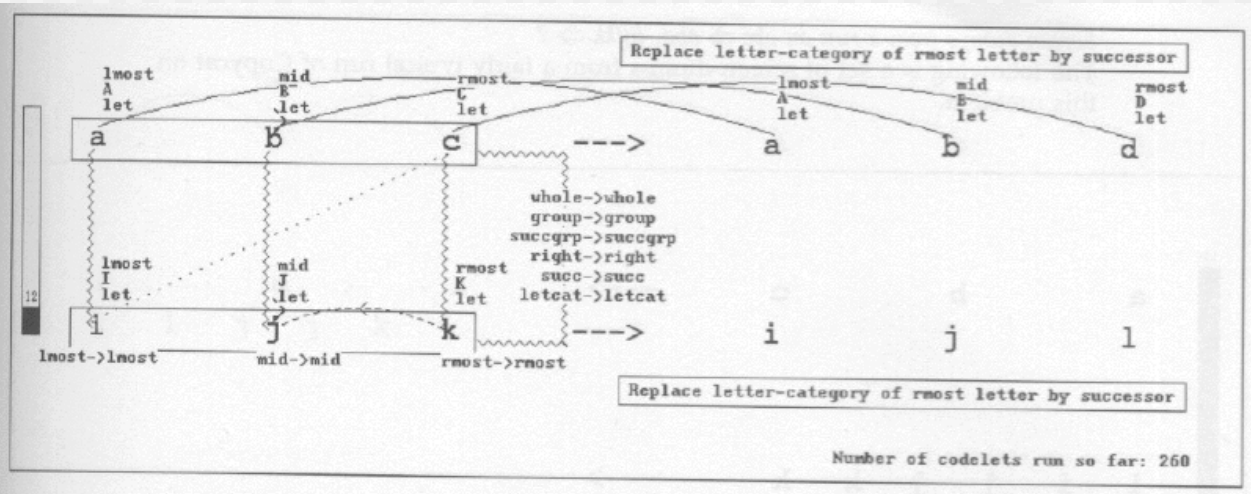
3. Workspace: Now 39 codelets have run. All three replacement arcs between letters in **abc** and **abd** have been built. (Since determining these is trivial, they are almost always constructed quite early on.) Several possible bonds and correspondences are being considered, but since none has been built yet the temperature remains at 100.

100 A	100 B	100 C	47 D	3 E	3 F	3 G	47 H	53 I	100 J	100 K	47 L	3 M
3 N	3 O	3 P	3 Q	3 R	3 S	3 T	3 U	3 V	3 W	3 X	3 Y	3 Z
1	2	3	4	5	4 left	14 right	100 lmost	100 rmost	100 middle	pred	succ	same
pred group	succ group	same group	15 first	last	4 letter	8 group	iden	opp	whole	single	100 letter cat	8 length
alpha pos	100 string pos	direction	27 object cat	bond cat	group cat							

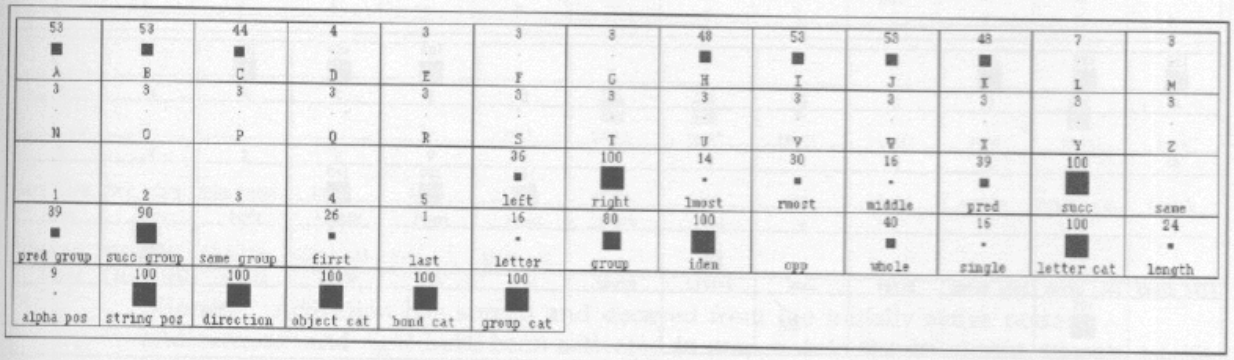
Slipnet: The initial activations have decayed and spread in various ways (e.g., the activation of the node *letter* has decayed, each of the 26 letter-category nodes has received a tiny bit of activation from the node *letter-category*, and *letter-category* has also spread some activation to *object-category* and *length*), and additional activation has come from codelet actions in the Workspace (e.g., the letter categories involved in the proposed bonds were reactivated by the codelets proposing the bonds).

Bolland's Copykitten

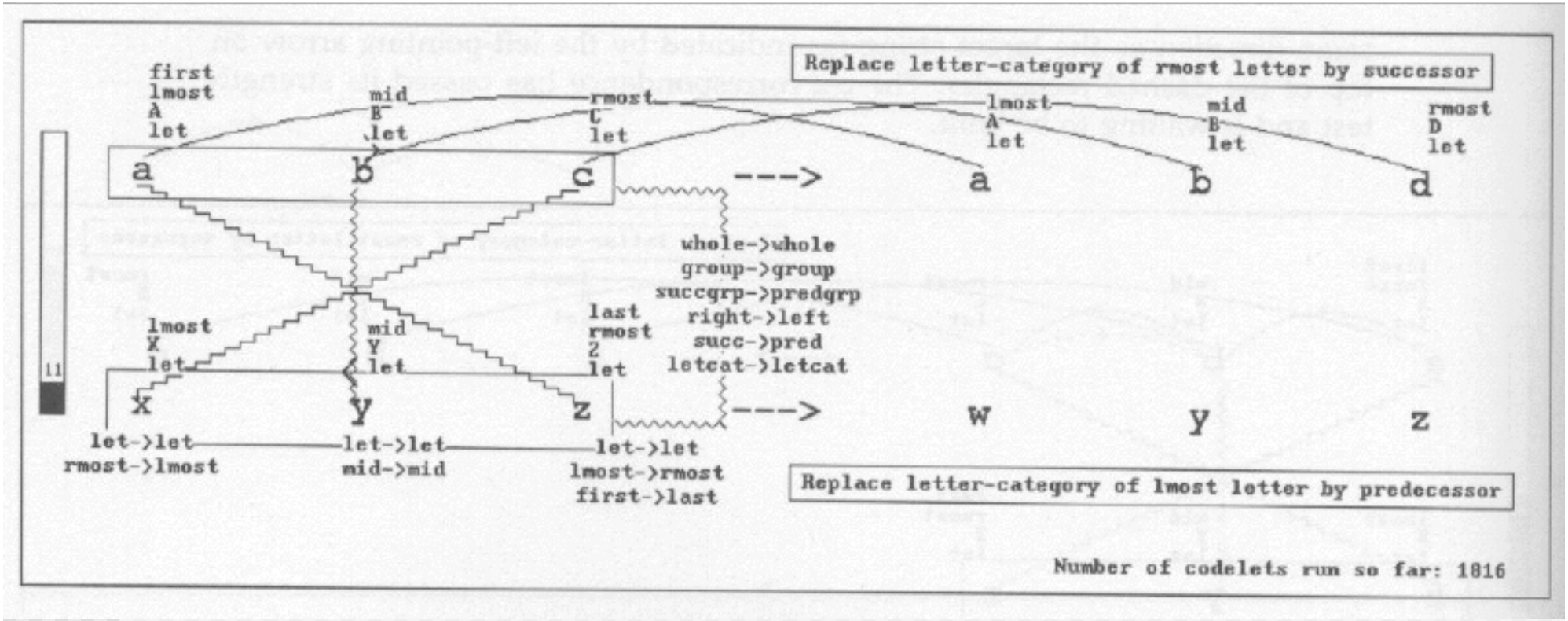
- Temperature
  - Decreases with increasing cohesiveness to solution
  - Higher temperature leads to more randomness in the choice of codelets



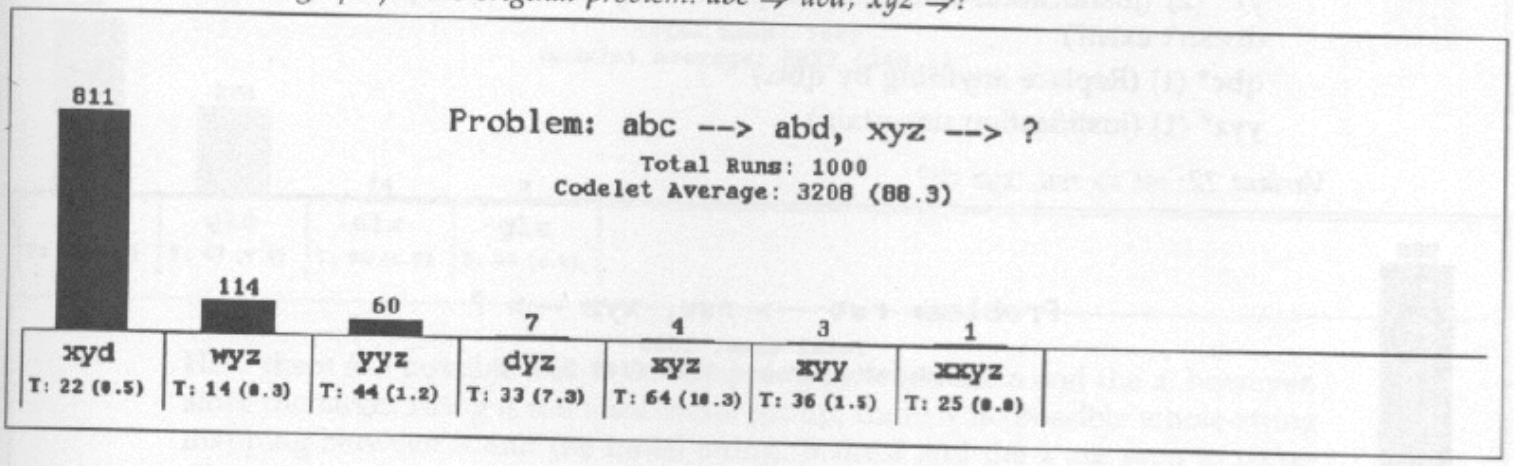
12. Workspace: The rule has been "translated" (although since all the concept mappings are identities, no changes were needed), and the answer **ijl** has been constructed according to the translated rule (the answer, with the translated rule beneath it, appears at the right-hand side of the screen). The low final temperature of 12 indicates that the program is very satisfied with this answer. This run consisted of 260 codelets (the average number of codelet steps for this problem is about 290).



Slipnet: The final configuration of the Slipnet indicates what concepts were found to be relevant in this problem: the individual letter categories' activations have decayed, and the notions of *right*, *successor*, *successor-group*, *group*, and *identity* are activated, along with nodes corresponding to various categories (e.g., *bond-category*).



The bar graph for the original problem: abc ⇒ abd, xyz ⇒?

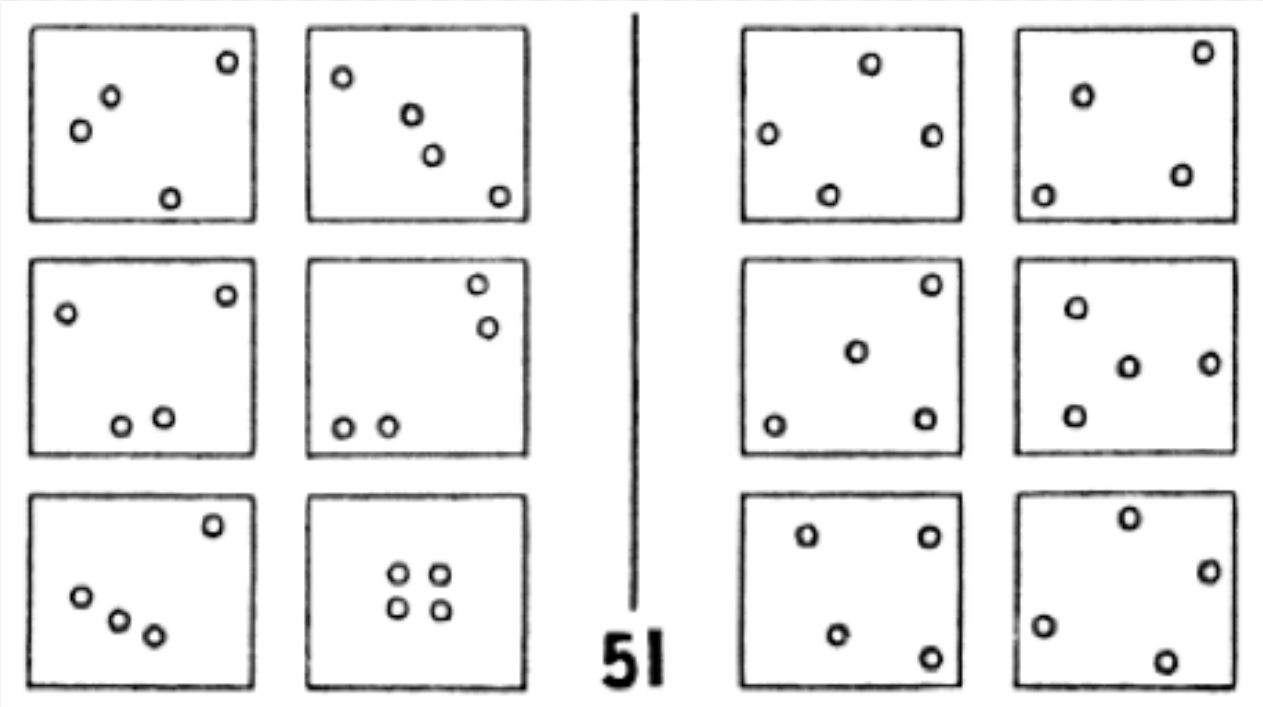


PR and AI

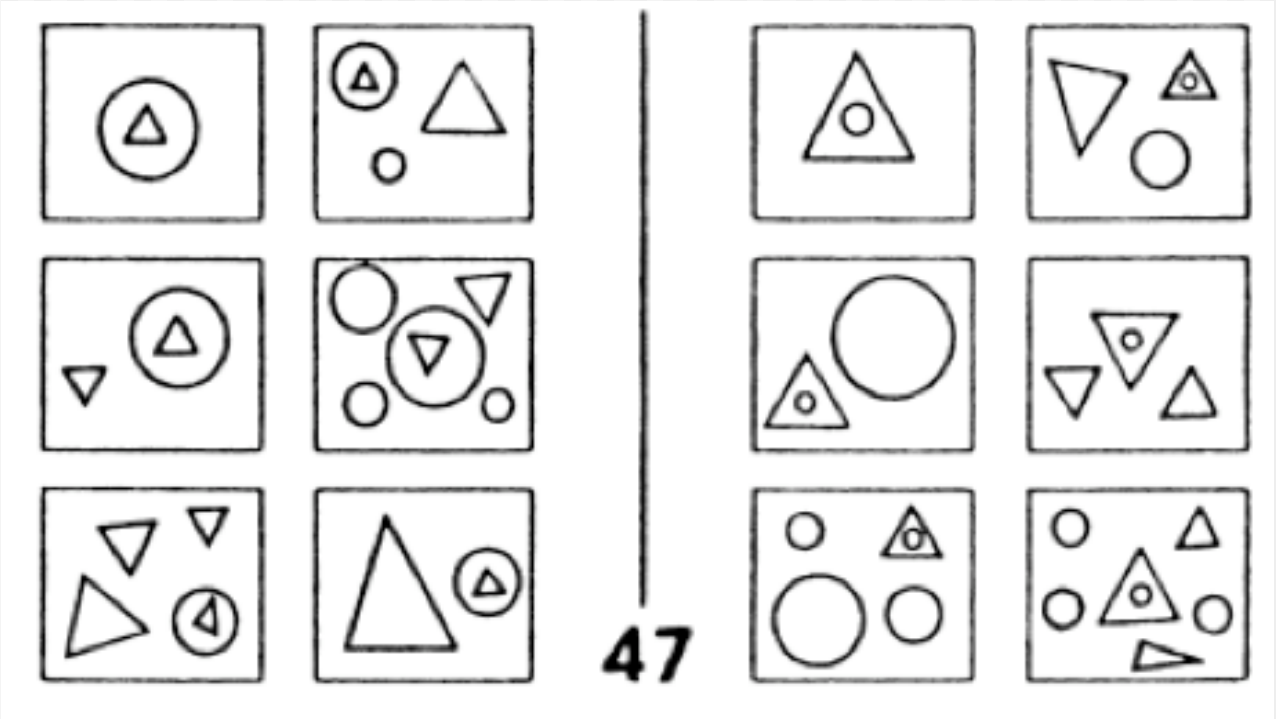
- Researchers in PR were concerned with getting machines to do such things as read handwriting or typewritten text, visually recognize objects in photographs, and understand spoken language
- What is the essence of 'A'-ness?

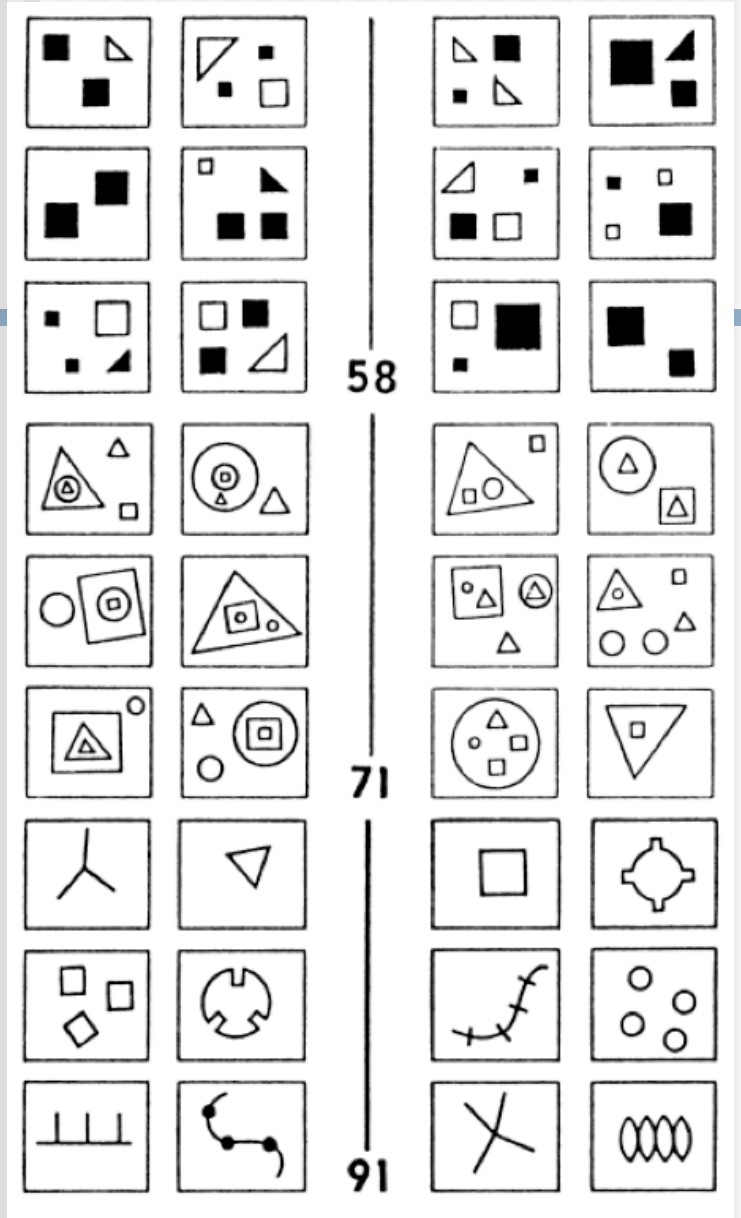
Bongard problem

- What was the criterion for separating the twelve into these two sets?



Criterion that distinguishes Category 1 from Category 2

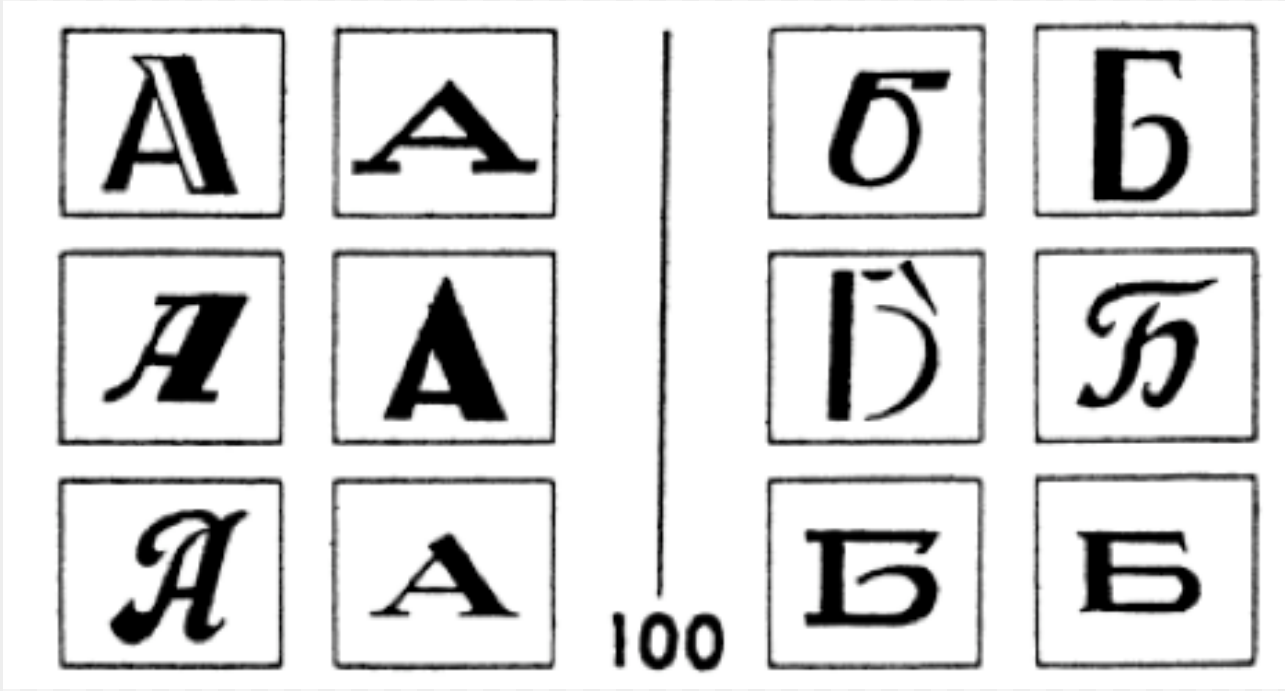


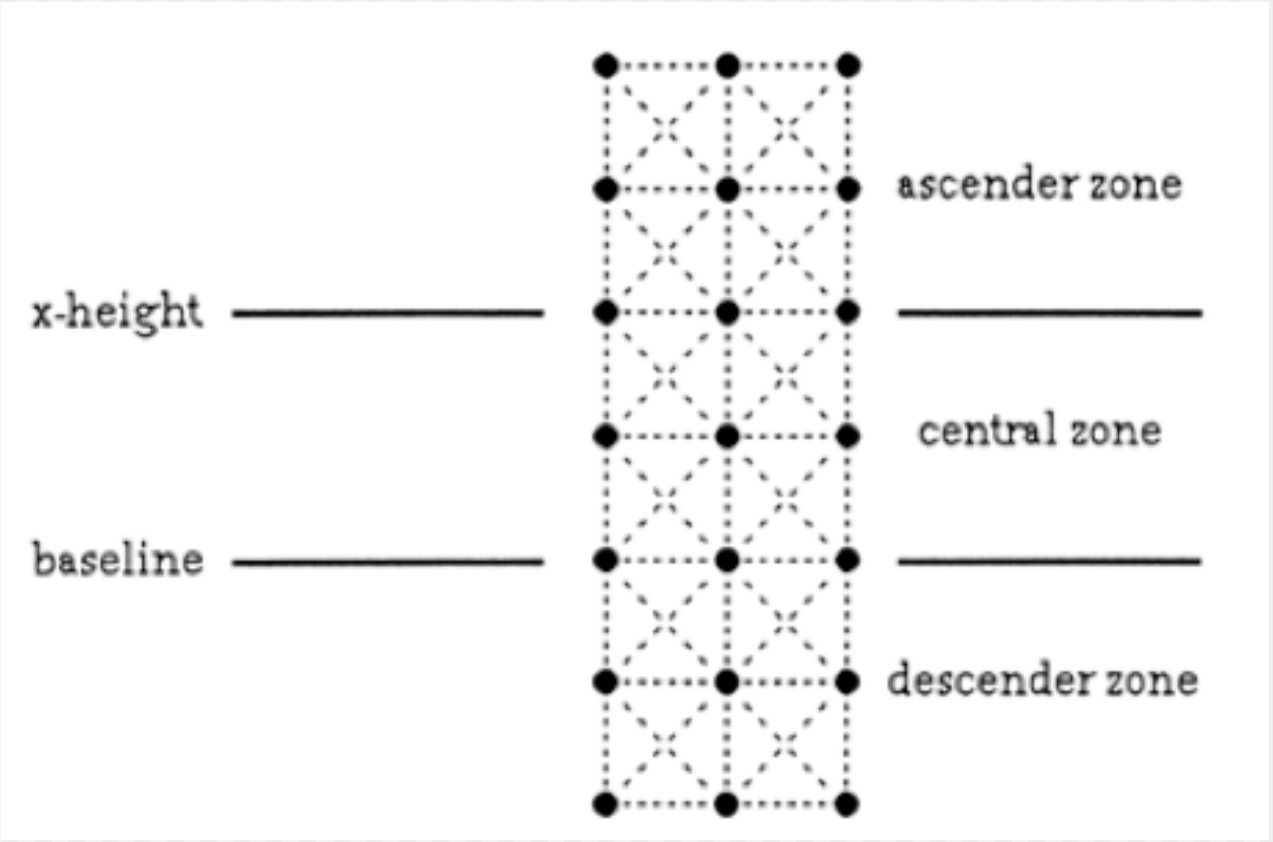


- Bongard problems involve highly abstract conceptual properties

- People usually have a very good intuitive sense, given a Bongard problem, for which types of features will wind up mattering and which are mere distractors
  - Its core seems to be analogy- making--that is, the activity of **abstracting out important features** of complex situations (thus filtering out what one takes to be superficial aspects) and **finding resemblances** and differences between situations at that high level of description.

- "Letter Spirit," and it is concerned with the visual forms of the letters of the roman alphabet
- Goal is to build a computer program that can design all 26 lowercase letters, "a" through "z," in any number of artistically consistent styles


100


ascender zone
x-height
central zone
baseline
descender zone