CS 681 Fall 2008

Designing Expert Systems

6. Overview of Basic Machine Learning Strategies

Gheorghe Tecuci
tecuci@gmu.edu
http://lac.gmu.edu/

Learning Agents Center
and Computer Science Department
George Mason University

Overview

Machine Learning: Introduction

Inductive Learning

Deductive Learning

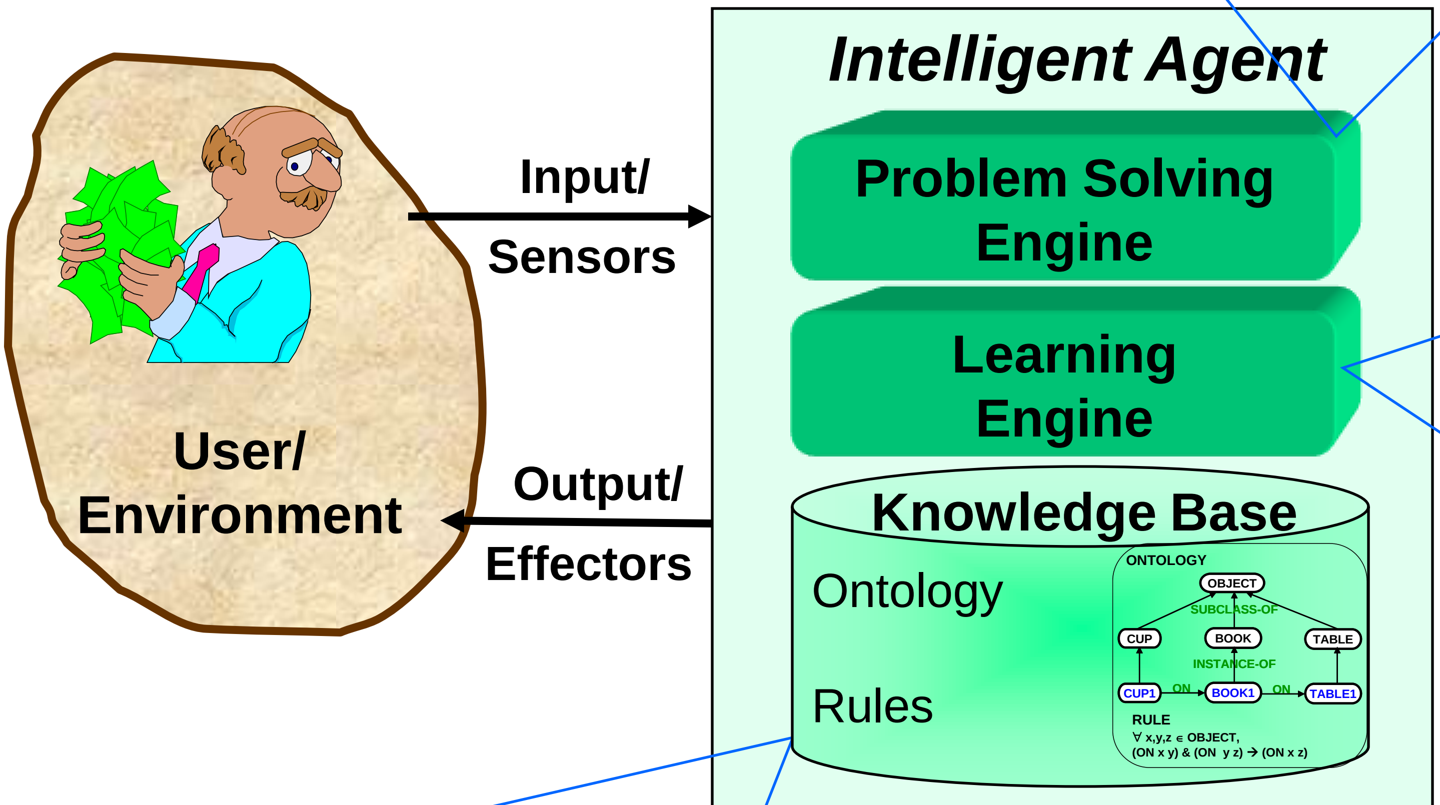
Abductive Learning

Analogical Learning

Multistrategy Learning

Overall Architecture of a Knowledge-Based Agent

Implements a general problem solving method that uses the knowledge from the knowledge base to interpret the input and provide an appropriate output.



Implements learning methods for extending and refining the knowledge in the knowledge base.

Data structures that represent the objects from the application domain, general laws governing them, actions that can be performed with them, etc.

What is Learning

Learning is a very general term denoting the way in which people and computers:

(1) Acquire, discover, and organize knowledge (by building, modifying and organizing internal representations of some external reality);

(2) Acquire skills (by gradually improving their motor or cognitive skills through repeated practice, sometimes involving little or no conscious thought).

Learning results in changes in the agent (or mind) that improve its competence and/or efficiency.

What is Machine Learning

Machine Learning is the domain of Artificial Intelligence which is concerned with building adaptive computer systems that are able to improve their performance through learning from input data, from a user, or from their own problem solving experience.

Two Complementary Dimensions of Learning

Competence

A system is improving its competence if it learns to solve a broader class of problems, and to make fewer mistakes in problem solving.

Efficiency

A system is improving its efficiency, if it learns to solve the problems from its area of competence faster or by using fewer resources.

Learning Strategies

A Learning Strategy is a basic form of learning characterized by the employment of a certain type of:

- *inference*
  (e.g. deduction, induction, abduction or analogy);
- *computational or representational mechanism*
  (e.g. rules, trees, neural networks, etc.);
- *learning goal*
  (e.g. learn a concept, discover a formula, acquire new facts, acquire new knowledge about an entity, refine an entity).

Representative Learning Strategies

Rote learning

Learning from instruction

Learning from examples

Explanation-based learning

Conceptual clustering

Quantitative discovery

Abductive learning

Learning by analogy

Instance-based learning

Reinforcement learning

Neural networks

Genetic algorithms and evolutionary computation

Bayesian learning

Multistrategy learning

Overview

Machine Learning: Introduction

Inductive Learning (from Examples)

Deductive Learning

Abductive Learning

Analogical Learning

Multistrategy Learning

The Learning Problem

Given
- a language of instances;
- a language of generalizations;
- a set of positive examples (E1, ..., En) of a concept
- a set of negative examples (C1, ... , Cm) of the same concept
- a learning bias
- other background knowledge

Determine
- a concept description which is a generalization of the positive examples that does not cover any of the negative examples

Purpose of concept learning

Predict if an instance is an example of the learned concept.

Generalization (and Specialization) Rules

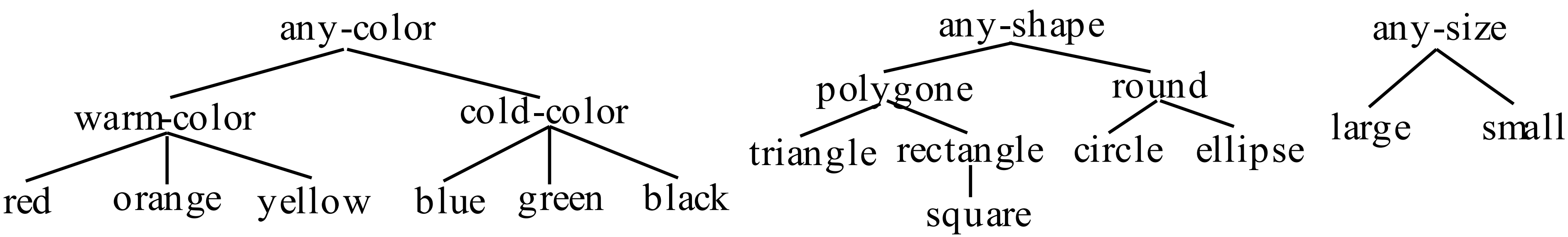
- Turning constants into variables
- Turning occurrences of a variable into variables
- Climbing the generalization hierarchy
- Dropping condition
- Extending intervals
- Extending ordered sets of intervals
- Extending discrete sets
- Using feature definitions
- Using inference rules

Problem

Language of instances: Objects with three attributes: color, shape, size.

Language of generalization: Object generalization characterized by a set of colors, shapes and sized, as defined by the following generalization hierarchies:

Background knowledge:



Problem: Learn the concept represented by the following examples:

color	shape	size	class	
orange	square	large	+	i1
blue	ellipse	small	-	i2
red	triangle	small	+	i3
green	rectangle	small	-	i4
yellow	circle	large	+	i5

Solution:

color	shape	size
warm-color	any-shape	any-size

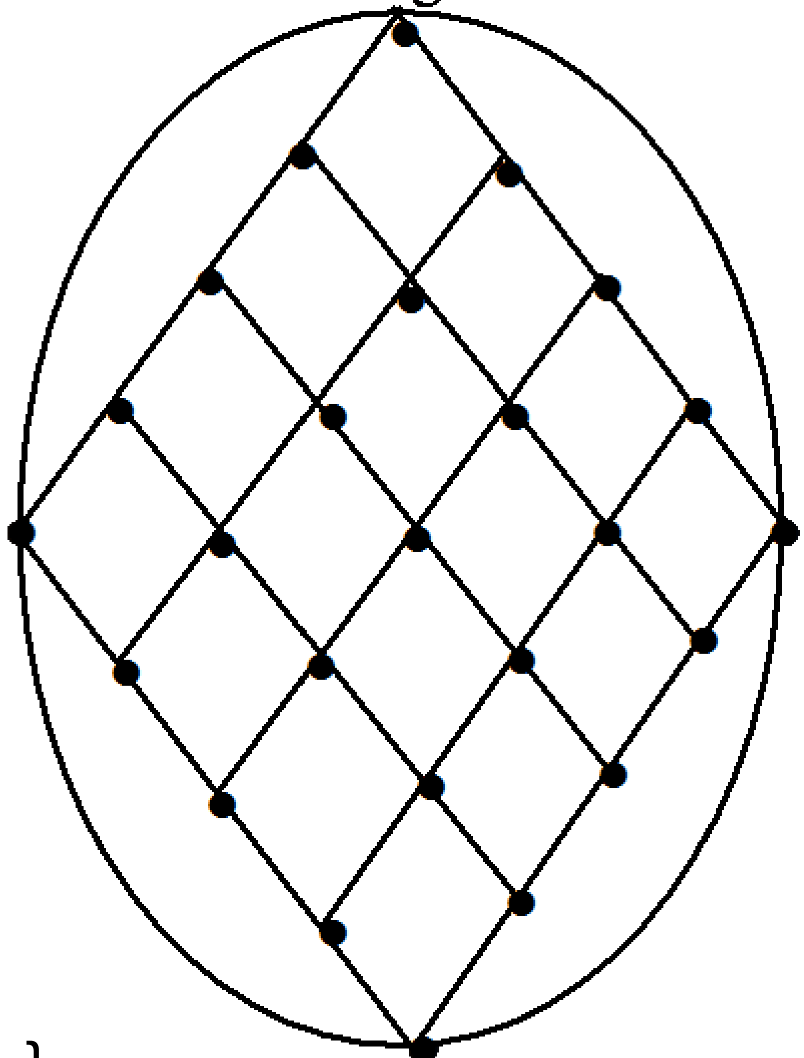
(C color warm-color
shape any-shape
size any-size)}

The Candidate Elimination Algorithm (Mitchell, 1978)

The most general generalization of e_1 $\mathbf{G}$: $\{e_g\}$

Let us suppose that we have an example e1 of a concept to be learned.

Then any sentence of the representation language which is more general than this example is a plausible hypothesis for the concept.



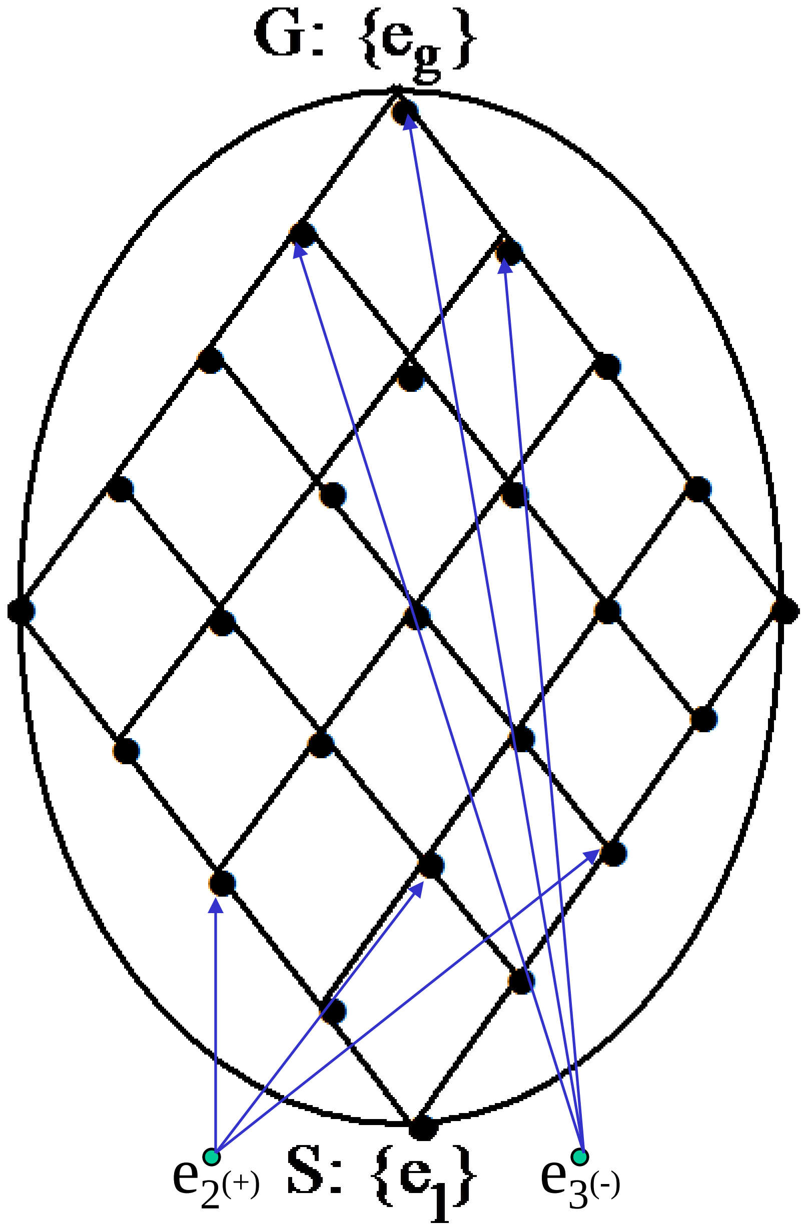
The version space is:

H = { h | h is more general than e_1 }

$\mathbf{S}$: $\{e_1\}$

The Candidate Elimination Algorithm

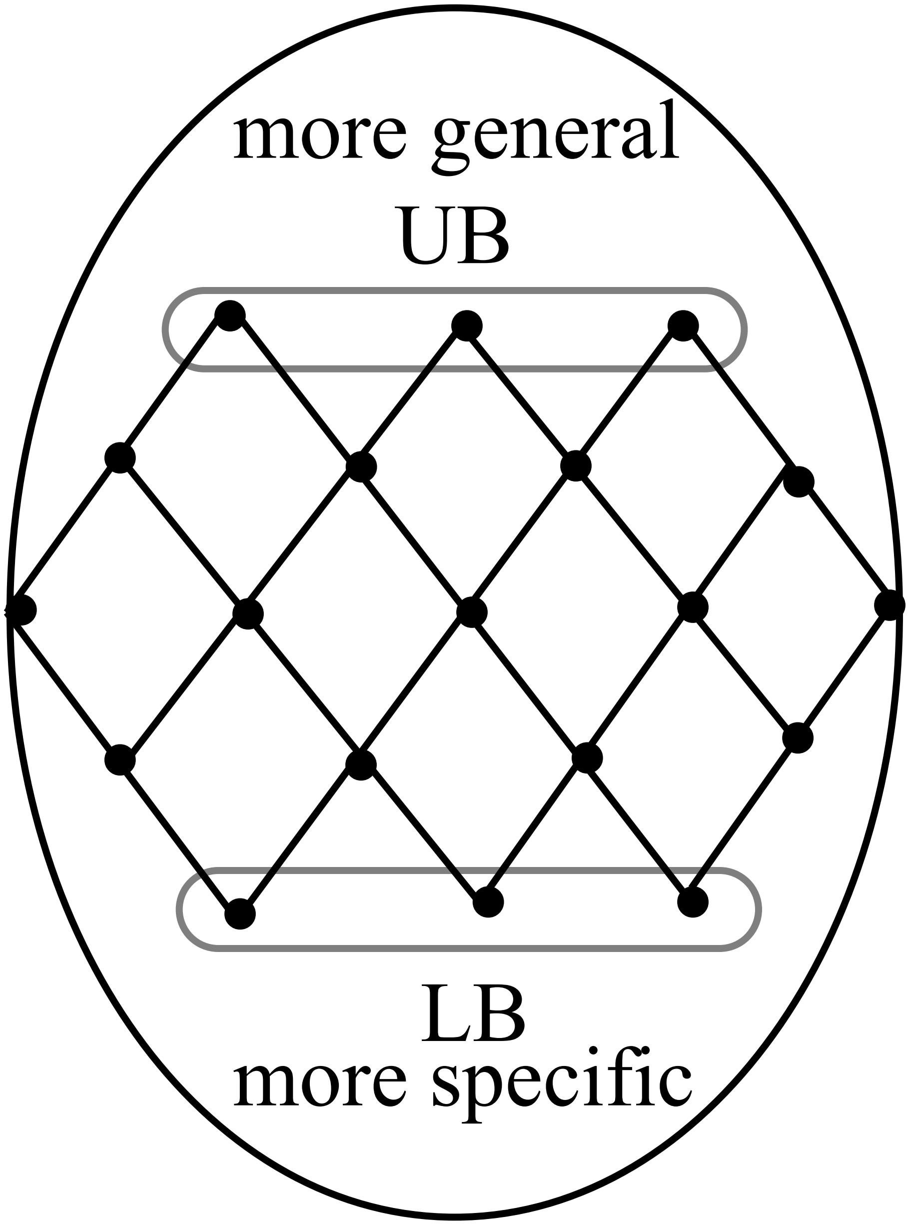
As new positive and negative examples are presented to the program, candidate concepts are eliminated from H.



The Candidate Elimination Algorithm (cont.)

As new positive and negative examples are presented to the program, candidate concepts are eliminated from H.

This is practically done by updating the set G (which is the set of the most general elements in H) and the set S (which is the set of the most specific elements in H).



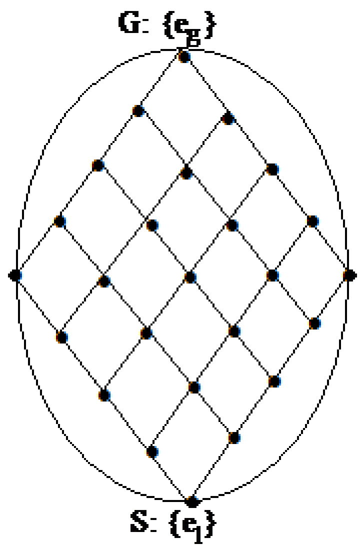
Explanation

Version spaces and the candidate elimination algorithm

This is a concept learning method based on exhaustive search. It was developed by Mitchell and his colleagues. Let us suppose that we have an example e1 of a concept to be learned. Then, any sentence of the representation language which is **more general than** this example, is a plausible hypothesis for the concept.

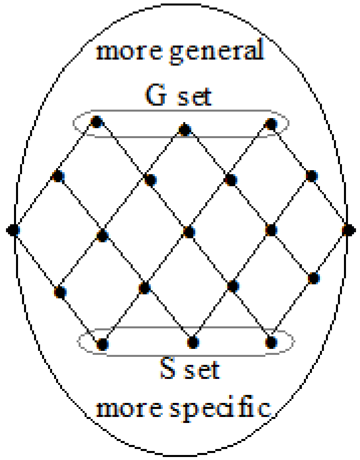
The set H of all the plausible hypotheses for the concept to be learned, is called the **version space**: H = { h | h is more general than e1 }
Let S be the set containing the example e1, and G be the set containing the most general description of the representation language which is more general than e1: S = { e1 }, G = { eg }

The following figure is an intuitive representation of the version space H (each hypothesis being represented as a point in the network):



H is the set of the concepts covering the example e1.

Because the **more general than** relation is a partial ordering relation, one may represent the version spaces H by its boundaries:
H = { h | h is more general than e1 and h is less general than eg }
or
H = {S, G}
As new examples and counterexamples are presented to the program, candidate concepts are eliminated from H. This is practically done by updating the set G (which is the set of the most general elements in H) and the set S (which is the set of the most specific elements in H):



Thus, the version space H is the set of all concept descriptions that are consistent with all the training instances seen so far. When the set H contains only one candidate concept, the desired concept has been found.

The Candidate Elimination Algorithm

1. Initialize S to the first positive example and G to its most general generalization

2. Accept a new training instance I

   If I is a positive example then

   - remove from G all the concepts that do not cover I;
   - generalize the elements in S *as little as possible* to cover I but remain less general than some concept in G;
   - keep in S the minimally general concepts.

   If I is a negative example then

   - remove from S all the concepts that cover I;
   - specialize the elements in G *as little as possible* to uncover I and be more general than at least one element from S;
   - keep in G the maximally general concepts.

3. Repeat 2 until G=S and they contain a single concept C (this is the learned concept)

The Candidate Elimination Algorithm: Illustration

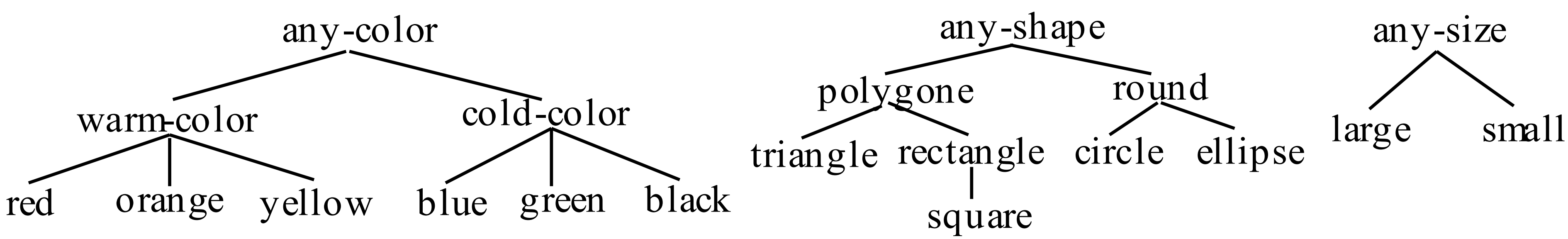
1. Initialize S to the first positive example and G to its most general generalization

(+i1 color orange
shape square
size large)

G: (C color any-color
shape any-shape
size any-size)

S: (C color orange
shape square
size large)

Background knowledge:



The Candidate Elimination Algorithm: Illustration

2. If the new training instance "I" is a negative example then:
   - Remove from S all the concepts that cover I;
   - Specialize the elements in G as little as possible to uncover I and be more general than at least one element from S. Keep in G the maximally general concepts.

```
G: {(C  color  any-color
        shape  any-shape
        size   any-size)
```

⇒

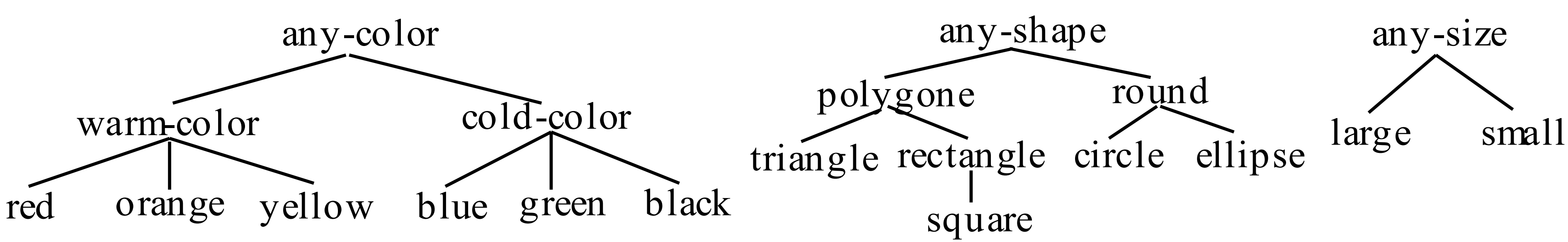
```
G: {(C  color  warm-color    (C  color  any-color    (C  color  any-color
        shape  any-shape         shape  polygon          shape  any-shape
        size   any-size)         size   any-size)        size   large)}
```

```
(-i2  color  blue
      shape  ellipse
      size   small)
```

```
S: {(C  color  orange
        shape  square
        size   large)}
```

⇒

```
S: {(C  color  orange
        shape  square
        size   large)}
```



The Candidate Elimination Algorithm: Illustration

2. If the new training instance "I" is a positive example then:
- Remove from G all the concepts that do not cover I;
- Generalize the elements in S as little as possible to cover I but remain less general than some concept in G. Keep in S the minimally general concepts.

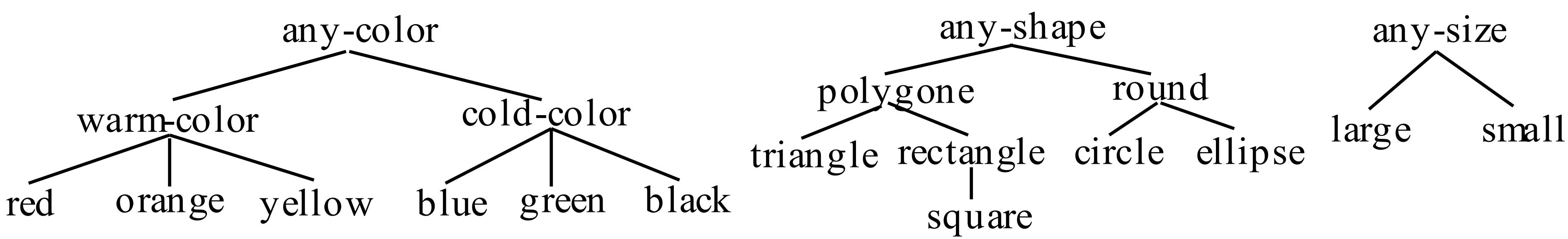
G: {(C color warm-color
shape any-shape
size any-size)
(C color any-color
shape polygon
size any-size)
~~(C color any-color
shape any-shape
size large)~~}

(+i3 color red
shape triangle
size small)

S: {(C color orange
shape square
size large)}

⇒

S: {(C color warm-color
shape polygon
size any-size)}



The Candidate Elimination Algorithm: Illustration

2. If the new training instance "I" is a negative example then:
   - Remove from S all the concepts that cover "I";
   - Specialize the elements in G as little as possible to uncover "I" and be more general than at least one element from S. Keep in G the maximally general concepts.

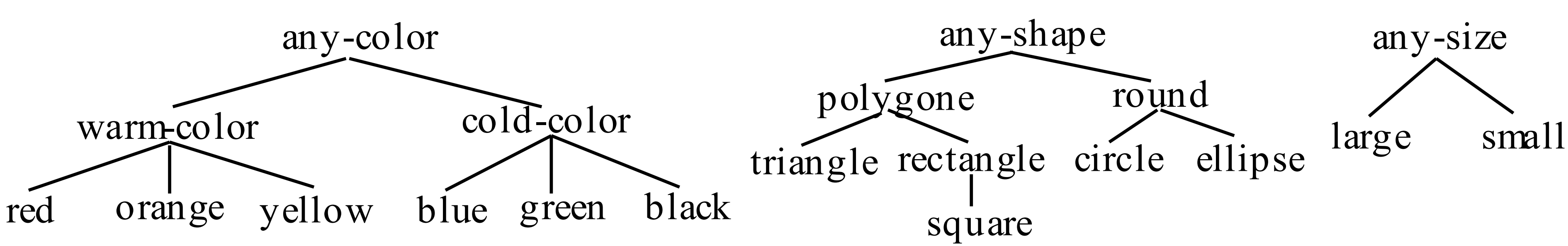
G: {(C color warm-color
shape any-shape
size any-size) (C color any-color
shape polygon
size any-size)}

G: {(C color warm-color
shape any-shape
size any-size) ~~(C color warm-color
shape polygon
size any-size)~~}

(-i4 color green
shape rectangle
size small)

S: {(C color warm-color
shape polygon
size any-size)}

S: {(C color warm-color
shape polygon
size any-size)}



The Candidate Elimination Algorithm: Illustration

2. If the new training instance “I” is a positive example then:
- Remove from G all the concepts that do not cover I;
- Generalize the elements in S as little as possible to cover I but remain less general than some concept in G. Keep in S the minimally general concepts.

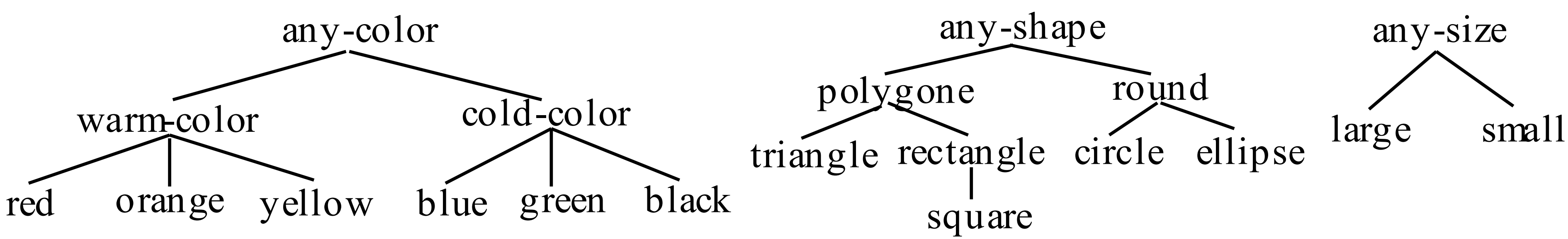
3. Repeat 2 until G=S and they contain a single concept C (this is the learned concept)

G: {(C color warm-color
shape any-shape
size any-size)}

(+i5 color yellow
shape circle
size large)

S: {(C color warm-color
shape polygon
size any-size)}

S = G: {(C color warm-color
shape any-shape
size any-size)}



Discussion

Does the order of the examples count?

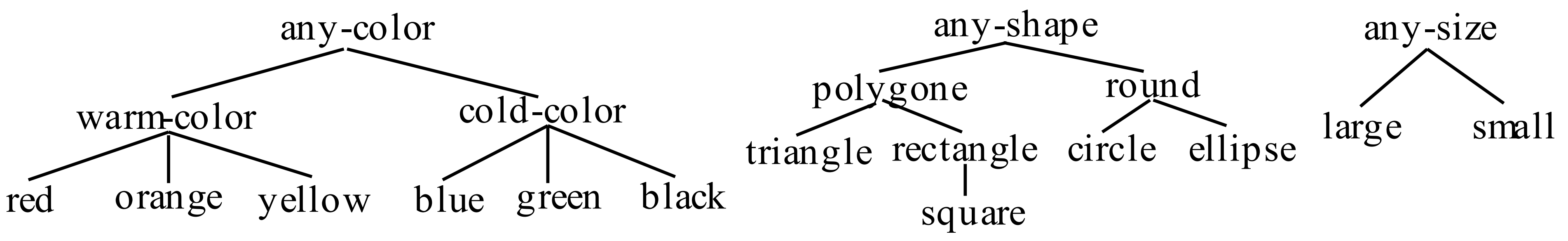
Why and how?

Discussion

Does the order of the examples count? Why and how?

Consider the following order:

color	shape	size	class	
orange	square	large	+	i1
red	triangle	small	+	i3
yellow	circle	large	+	i5
blue	ellipse	small	-	i2
green	rectangle	small	-	i4



Discussion

What happens if there are not enough examples for S and G to become identical?

Could we still learn something useful?

How could we classify a new instance?

When could we be sure that the classification is the same as the one made if the concept were completely learned?

Could we be sure that the classification is correct?

Discussion

What happens if there are not enough examples for S and G to become identical?

Let us assume that one learns only from the first 3 examples:

color	shape	size	class	
orange	square	large	+	i1
blue	ellipse	small	-	i2
red	triangle	small	+	i3

The final version space will be:

G: {(C color warm-color
shape any-shape
size any-size) (C color any-color
shape polygon
size any-size)}

S: {(C color warm-color
shape polygon
size any-size)}

Discussion

Assume that the final version space is:

G: {(C color warm-color
shape any-shape
size any-size) (C color any-color
shape polygon
size any-size)}

S: {(C color warm-color
shape polygon
size any-size)}

How could we classify the following examples, how certain we are about the classification, and why?

color	shape	size	class
blue	circle	large	–
orange	square	small	+
red	ellipse	large	don’t know
blue	polygon	small	don’t know

Discussion

Could the examples contain errors?

What kind of errors could be found in an example?

What will be the result of the learning algorithm if there are errors in examples?

Discussion

Could the examples contain errors?

What kind of errors could be found in an example?

- **Classification errors:**
  - **positive examples labeled as negative**
  - **negative examples labeled as positive**

- **Measurement errors**
  - **errors in the values of the attributes**

Discussion

What will be the result of the learning algorithm if there are errors in examples?

Let us assume that the 4th example is incorrectly classified:

color	shape	size	class	
orange	square	large	+	i1
blue	ellipse	small	-	i2
red	triangle	small	+	i3
green	rectangle	small	+	i4 (incorrect classification)
yellow	circle	large	+	i5

The version space after the first three examples is:

G: {(C color warm-color
shape any-shape
size any-size)
(C color any-color
shape polygon
size any-size)}

S: {(C color warm-color
shape polygon
size any-size)}

Continue learning

The Learning Bias

A bias is any basis for choosing one generalization over another, other than strict consistency with the observed training examples.

Types of bias:

- restricted hypothesis space bias;
- preference bias.

Restricted Hypothesis Space Bias

The hypothesis space H (i.e. the space containing all the possible concept descriptions) is defined by the generalization language. This language may not be capable of expressing all possible classes of instances. Consequently, the hypothesis space in which the concept description is searched is restricted.

Some of the restricted spaces investigated:

- logical conjunctions (i.e. the learning system will look for a concept description in the form of a conjunction);
- decision tree;
- three-layer neural networks with a fixed number of hidden units.

Restricted Hypothesis Space Bias: Example

The language of instances consists of triples of bits as, for example: (0, 1, 1), (1, 0, 1).

How many concepts are in this space?
The total number of subsets of instances is $2^8 = 256$.

The language of generalizations consists of triples of 0, 1, and *, where * means any bit, for example: (0, *, 1), (*, 0, 1).

How many concepts could be represented in this language?

This hypothesis space consists of 3x3x3 = 27 elements.

Preference Bias

A preference bias places a preference ordering over the hypotheses in the hypothesis space H. The learning algorithm can then choose the most preferred hypothesis f in H that is consistent with the training examples, and produce this hypothesis as its output.

Most preference biases attempt to minimize some measure of syntactic complexity of the hypothesis representation (e.g. shortest logical expression, smallest decision tree).

These are variants of Occam's Razor, which is the bias first defined by William of Occam (1300-1349):

Given two explanations of data, all other things being equal, the simpler explanation is preferable.

Preference Bias: Representation

How could the preference bias be represented?

In general, the preference bias may be implemented as an order relationship **'better(f1, f2)'** over the hypothesis space H.
Then, the system will choose the "best" hypothesis f, according to the "better" relationship.

An example of such a relationship:
"less-general-than" which produces the least general expression consistent with the data.

Features of the Version Space Method

- In its original form learns only conjunctive descriptions.
- However, it could be applied successively to learn disjunctive descriptions.
- Requires an exhaustive set of examples.
- Conducts an exhaustive bi-directional breadth-first search.
- The sets S and G can be very large for complex problems.
- It is very important from a theoretical point of view, clarifying the process of inductive concept learning from examples.
- Has very limited practical applicability because of the combinatorial explosion of the S and G sets.
- It is at the basis of the powerful Disciple multistrategy learning method which has practical applications.

Exercise

The instance space for a concept learning problem is a set of objects, each object having two features - shape and size. The shape of an object can be ball, brick, cube or star. The size of an object can be small, medium or large. An instance is represented by a feature vector with two features. For example, (ball, large) represents a large ball. There is no other background knowledge. Each concept is also represented by a feature vector with two features, shape and size, except that there are two additional values for these features, any-shape and any-size.

Consider the following positive and negative examples of a concept to be learned:
+ (ball, large), - (brick, small), - (cube, large), + (ball, small).

Learn the concept represented by the above examples by applying the candidate elimination algorithm.

Which will be the results of learning if only the first three examples are available?

Exercise

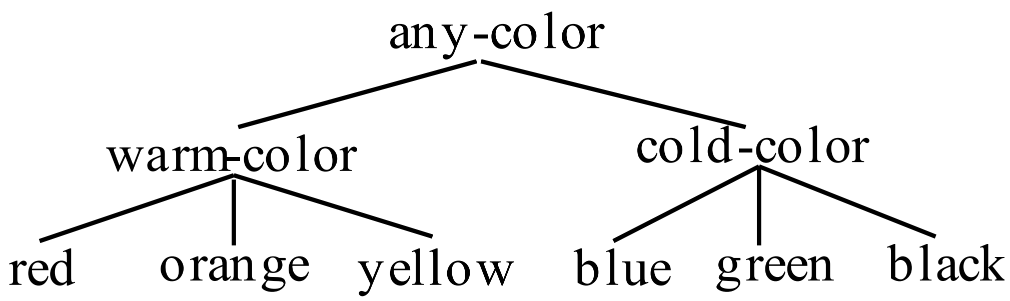
Consider the following:

Instance language
color {red, orange, yellow, blue, green, black}

Generalization language
color {red, orange, yellow, blue, green, black, warm-color, cold-color, any-color}

sequence of positive and negative examples of a concept, and the background knowledge represented by the following hierarchy:

example1(+): orange
example2(-): blue
example3(+): red



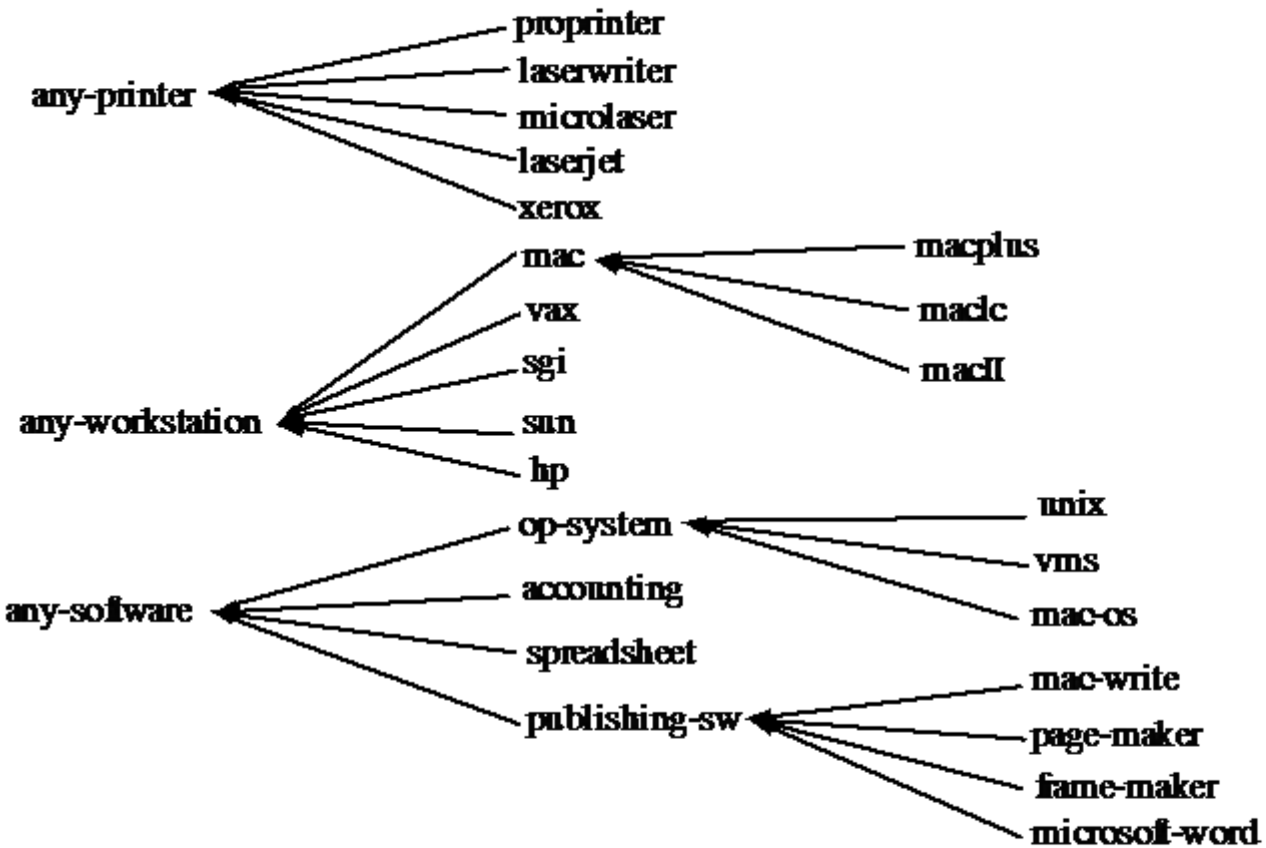
Apply the candidate elimination algorithm to learn the concept represented by the above examples.

Exercise

Consider the following positive and negative examples of a concept to be learned using the candidate elimination algorithm:

workstation	software	printer	class	
macII	microsoft-word	proprinter	+	e1
sgi	spreadsheet	laserwriter	-	c1

and the following background knowledge



Exercise (cont.)

a. Which are the sets S and G corresponding to the first example e1?

b. Which are the new sets S and G after learning from the negative example c1?

c. Assume that after learning from another example, the sets S and G are the following:

S: {[(workstation = mac) & (software = publishing-sw) & (printer = any-printer)]}
G: {[(workstation = mac) & (software = any-software) & (printer = any-printer)],
[(workstation = any-workstation) & (software = publishing-sw) & (printer = any-printer)]}

What will be the new sets S and G after learning from the following example:

workstation	software	printer	class	
sun	frame-maker	laserwriter	+	e3

Reading

Tecuci G., These Lecture Notes (required).

Mitchell T.M., *Machine Learning,* Chapter 2: Concept learning and the general to specific ordering, pp. 20-51, McGraw Hill, 1997 (recommended).

Mitchell, T.M., Utgoff P.E., Banerji R., Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics, in Readings in Machine Learning (recommended).

Russell S., and Norvig P., Artificial Intelligence: A Modern Approach, Prentice Hall, Second edition, pp. 649 – 653, 678 – 686 (recommended).

Overview

Machine Learning: Introduction

Inductive Learning

Deductive (Explanation-based) Learning

Abductive Learning

Analogical Learning

Multistrategy Learning

The Explanation-Based Learning Problem

Given

A training example

A positive example of a concept to be learned.

Learning goal

A specification of the desirable features of the concept to be learned from the training example.

Background knowledge

Prior knowledge that allows proving (explaining) that the training example is indeed a positive example of the concept.

Determine

A concept definition representing a deductive generalization of the training example that satisfies the learning goal.

Purpose of learning

Improve the problem solving efficiency of the agent.

Explanation-Based Learning Problem: Illustration

Given

Training Example - The description of a particular cup:
OWNER(OBJ1, EDGAR) & COLOR(OBJ1, RED) & IS(OBJ1, LIGHT) & PART-OF(CONCAVITY1, OBJ1) & ISA(CONCAVITY1, CONCAVITY) & IS(CONCAVITY1, UPWARD-POINTING) & PART-OF(BOTTOM1, OBJ1) & ISA(BOTTOM1, BOTTOM) & IS(BOTTOM1, FLAT) & PART-OF(BODY1, OBJ1) & ISA(BODY1, BODY) & IS(BODY1, SMALL) & PART-OF(HANDLE1, OBJ1) & ISA(HANDLE1, HANDLE) & LENGTH(HANDLE1, 5)

Learning goal
Find a sufficient concept definition for CUP, expressed in terms of the features used in the training example (LIGHT, HANDLE, FLAT, etc.)

Background Knowledge
∀x, LIFTABLE(x) & STABLE(x) & OPEN-VESSEL(x) → CUP(x)
∀x ∀y, IS(x, LIGHT) & PART-OF(y, x) & ISA(y, HANDLE) → LIFTABLE(x)
∀x ∀y, PART-OF(y, x) & ISA(y, BOTTOM) & IS(y, FLAT) → STABLE(x)
∀x ∀y, PART-OF(y,x) & ISA(y, CONCAVITY) & IS(y, UPWARD-POINTING) → OPEN-VESSEL(x)

Determine
A deductive generalization of the training example that satisfies the learning goal
∀x∀y1∀y2∀y3, [PART-OF(y1, x) & ISA(y1, CONCAVITY) & IS(y1, UPWARD-POINTING) &
PART-OF(y2, x) & ISA(y2, BOTTOM) & IS(y2, FLAT) & IS(x, LIGHT) &
PART-OF(y3, x) & ISA(y3, HANDLE) => CUP(x)]

The Explanation-Based Learning Method

Explain
Construct an explanation that proves that the training example is an example of the concept to be learned.

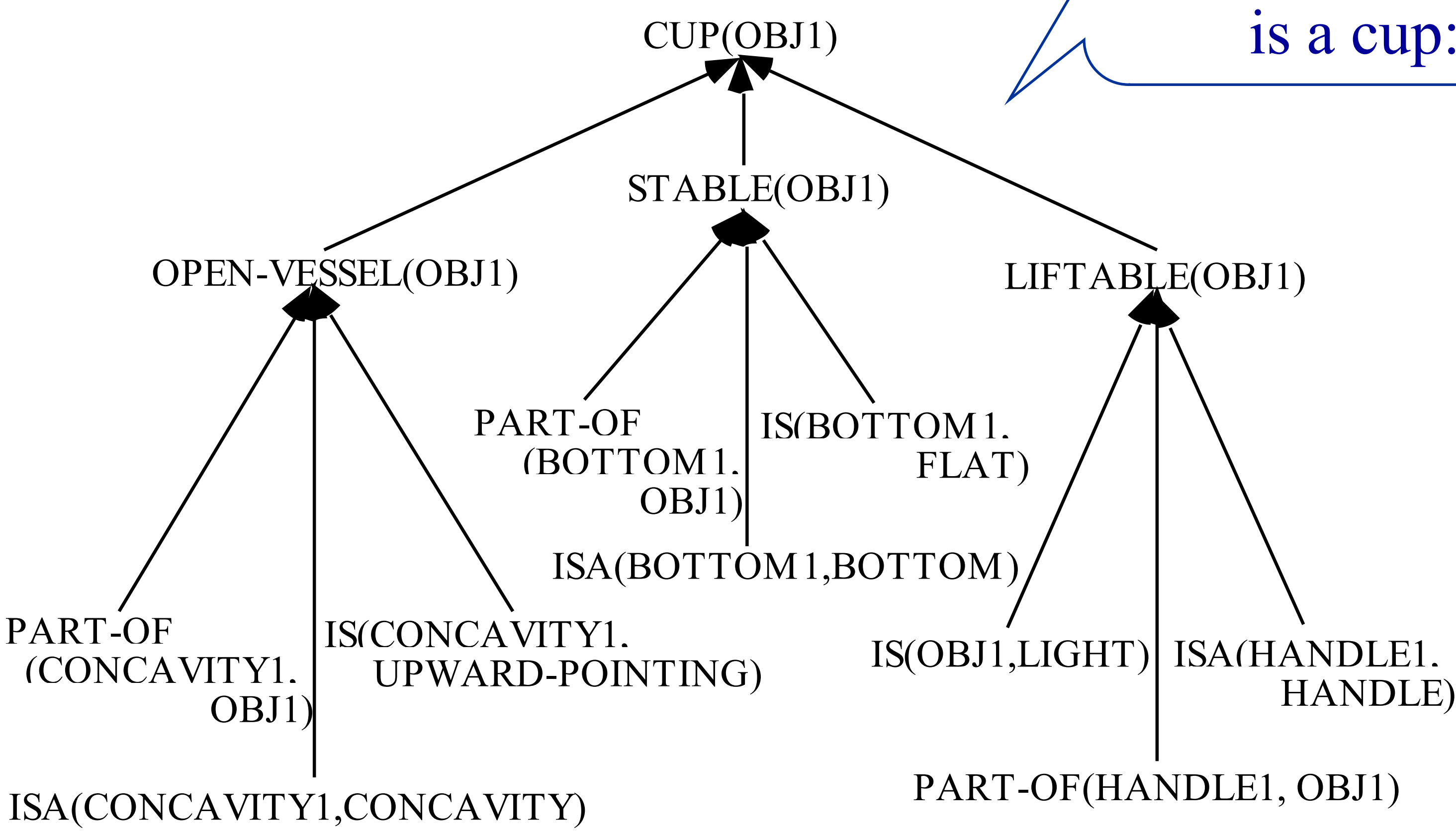
Generalize
Generalize the found explanation as much as possible so that the proof still holds, and extract from it a concept definition that satisfies the learning goal.

Explain

Background Knowledge

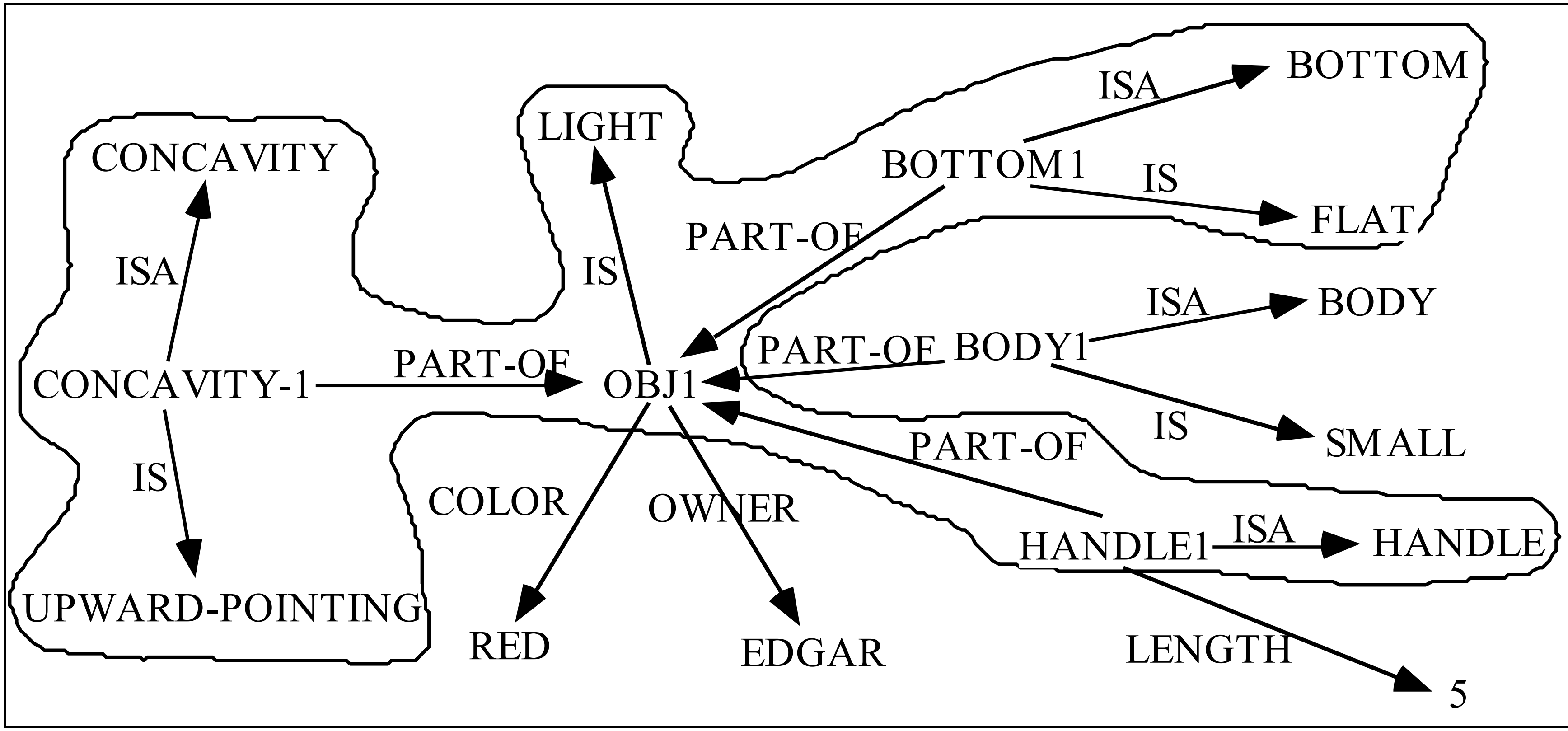
∀x, LIFTABLE(x) & STABLE(x) & OPEN-VESSEL(x) → CUP(x)
∀x ∀y, IS(x, LIGHT) & PART-OF(y, x) & ISA(y, HANDLE) → LIFTABLE(x)
∀x ∀y, PART-OF(y, x) & ISA(y, BOTTOM) & IS(y, FLAT) → STABLE(x)
∀x ∀y, PART-OF(y,x) & ISA(y, CONCAVITY) & IS(y, UPWARD-POINTING) → OPEN-VESSEL(x)

Prove that the training example is a cup:



The leaves of the proof tree are those features of the training example that allows one to recognize it as a cup. By building the proof one isolates the relevant features of the training example.

Discovery of the Relevant Features

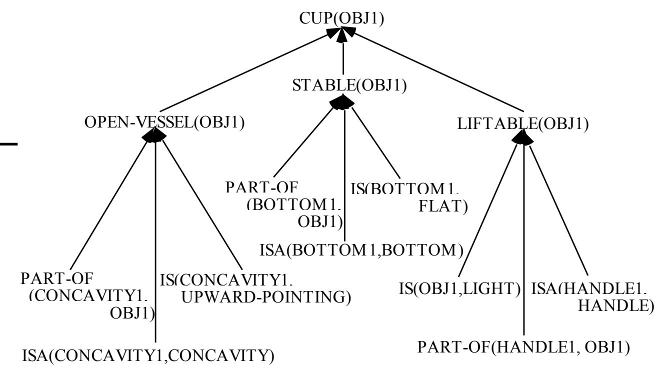


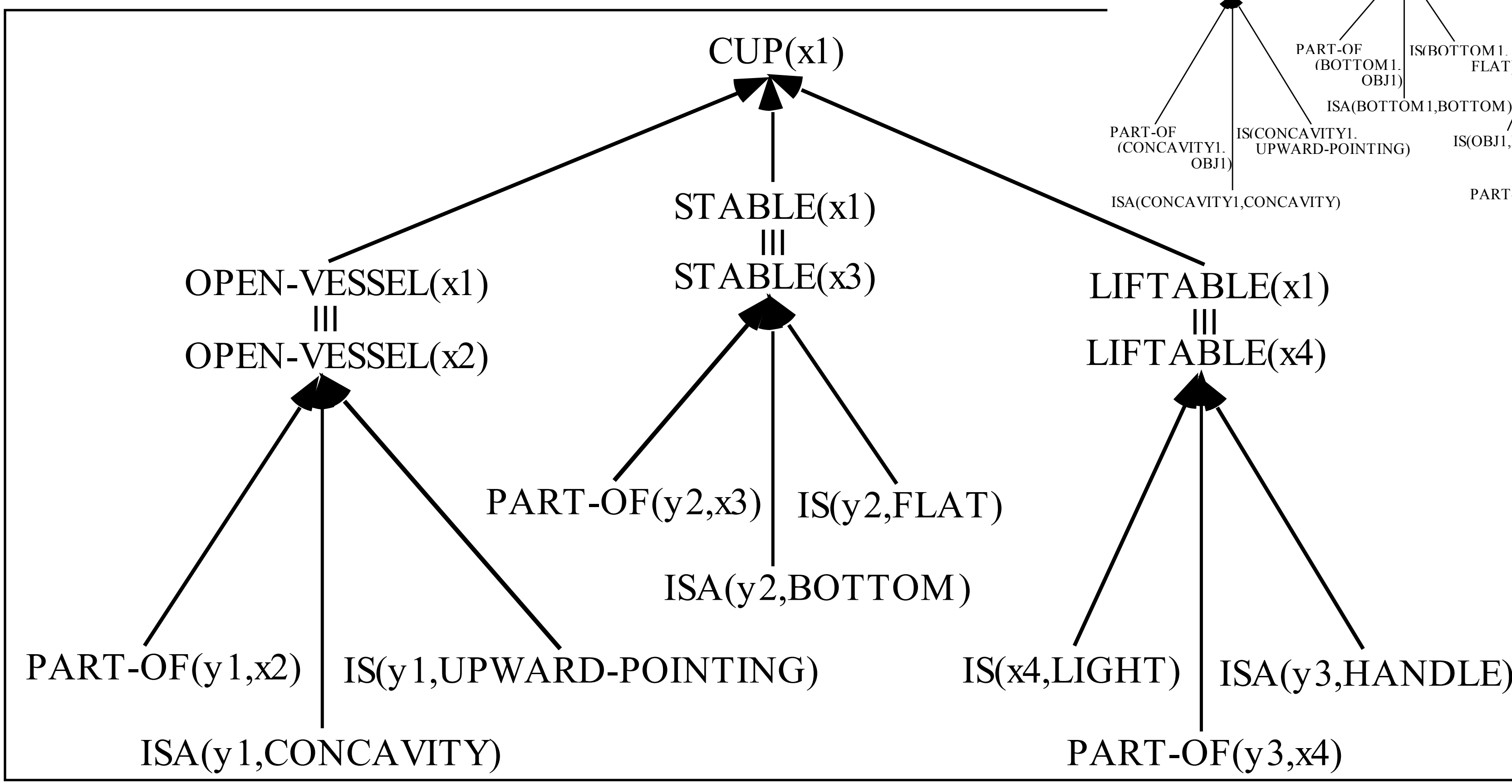
The ontological representation of the cup example.
The enclosed features are the relevant ones.

Generalize

Generalize the proof tree as much as possible so that the proof still holds:

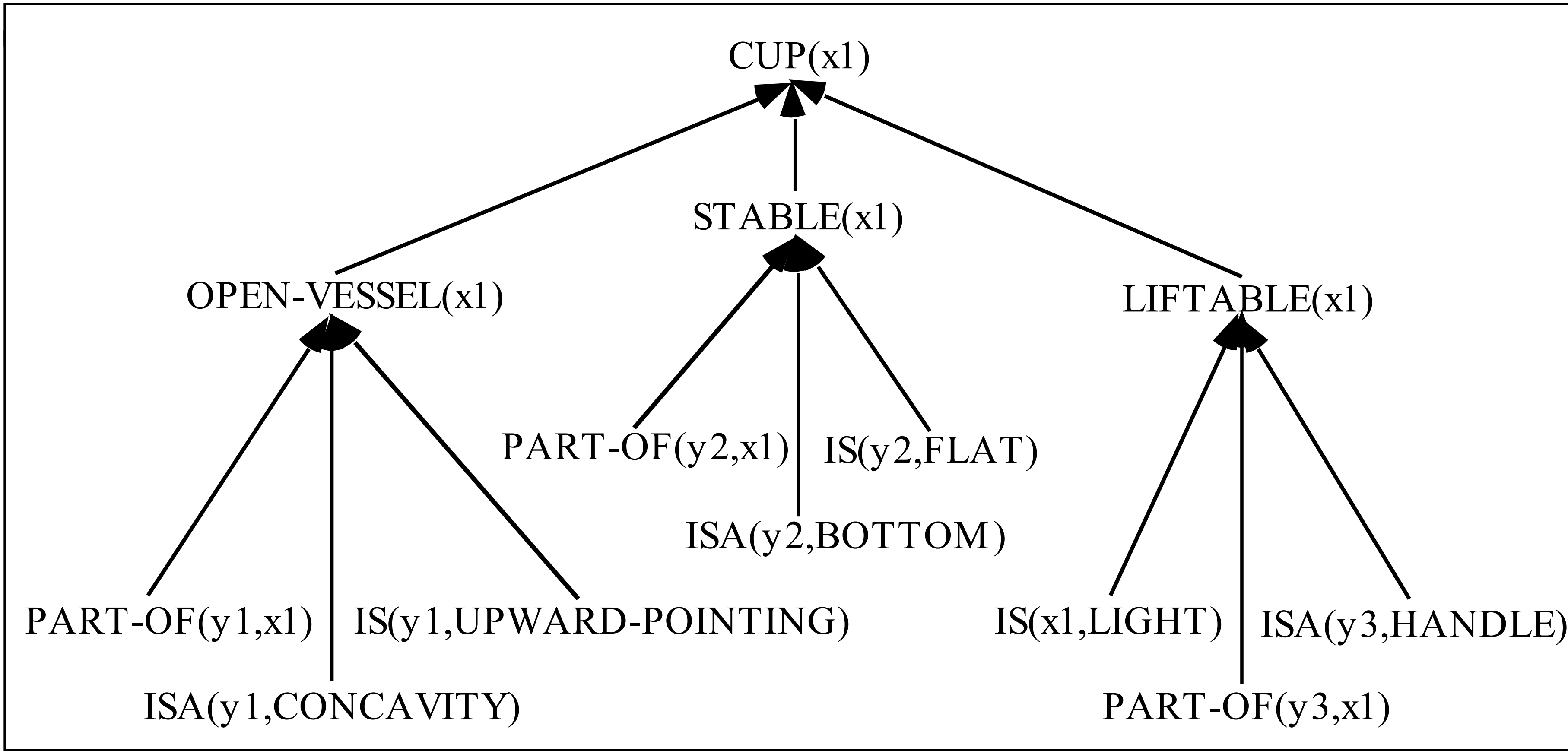
- replace each rule instance with its general pattern;
- find the most general unification of these patterns.





OPEN-VESSEL (x1)	STABLE (x1)	LIFTABLE (x1)
|||	|||	|||
OPEN-VESSEL (x2)	STABLE (x3)	LIFTABLE (x4)

Therefore x1=x2=x3=x4=x



The leaves of this generalized proof tree represent an operational definition of the concept CUP:

∀x1∀y1∀y2∀y3, [PART-OF(y1, x1) & ISA(y1, CONCAVITY) &
IS(y1, UPWARD-POINTING) & PART-OF(y2, x1) &
ISA(y2, BOTTOM) & IS(y2, FLAT) & IS(x1, LIGHT) &
PART-OF(y3, x1) & ISA(y3, HANDLE) => CUP(x1)]

Exercise

Given

• *A training Example*

The following example of "supports":

[book(book1) & material(book1, rigid) & cup(cup1) & material(cup1, rigid) & above(cup1, book1) & touches(cup1, book1)] => supports(book1, cup1)

• *Learning goal*

Find a sufficient concept definition for "supports", expressed in terms of the features used in the training example.

• *Background Knowledge*

$$\forall x\ \forall y\ [\text{on-top-of}(y, x)\ \&\ \text{material}(x, \text{rigid}) \rightarrow \text{supports}(x, y)]$$
$$\forall x\ \forall y\ [\text{above}(x, y)\ \&\ \text{touches}(x, y) \rightarrow \text{on-top-of}(x, y)]$$
$$\forall x\ \forall y\ \forall z\ [\text{above}(x, y)\ \&\ \text{above}(y, z) \rightarrow \text{above}(x, z)]$$

Determine

A deductive generalization of the training example that satisfies the learning goal.

Solution

- *Training Example:* [book(book1) & material(book1, rigid) & cup(cup1) & material(cup1, rigid) & above(cup1, book1) & touches(cup1, book1)] => supports(book1, cup1)

- *Background Knowledge:* $\forall x\ \forall y$ [on-top-of(y, x) & material(x, rigid) $\rightarrow$ supports(x, y)]
  $\forall x\ \forall y$ [above(x, y) & touches(x, y) $\rightarrow$ on-top-of(x, y)]
  $\forall x\ \forall y\ \forall z$ [above(x, y) & above(y, z) $\rightarrow$ above(x, z)]

Discussion

How does this learning method improve the efficiency of the problem solving process?

Discussion

Does the learner need a training example to learn an operational definition of the concept?

Background Knowledge

$\forall$x, LIFTABLE(x) & STABLE(x) & OPEN-VESSEL(x) $\rightarrow$ CUP(x)

$\forall$x $\forall$y, IS(x, LIGHT) & PART-OF(y, x) & ISA(y, HANDLE) $\rightarrow$ LIFTABLE(x)

$\forall$x $\forall$y, PART-OF(y, x) & ISA(y, BOTTOM) & IS(y, FLAT) $\rightarrow$ STABLE(x)

$\forall$x $\forall$y, PART-OF(y,x) & ISA(y, CONCAVITY) & IS(y, UPWARD-POINTING) $\rightarrow$ OPEN-VESSEL(x)

Discussion

Does the learner need a training example to learn an operational definition of the concept?

Answer:
The learner does not need a training example. It can simply build proof trees from top-down, starting with an abstract definition of the concept and growing the tree until the leaves are operational features.

Then why do we use a training example?

Without a training example the learner will learn many operational definitions. The training example focuses the learner on the most typical example(s).

General Features of Explanation-Based Learning

- Needs only one example
- Requires complete knowledge about the concept (which makes this learning strategy, in its pure form, impractical)
- Improves agent's efficiency in problem solving
- Shows the importance of explanations in learning

Exercise

Given

• *Training Example*

An example of the concept "LIKES(x, y)":

HUMAN(John) & HAPPY(John) & AGE(John, 32) => LIKES(John, John)

• *Learning goal*

Find a sufficient concept definition for "LIKES", expressed only in terms of the features used in the training example (i.e. HUMAN, HAPPY, AGE)

• *Background Knowledge*

$\forall$x $\forall$y KNOWS(x, y) & PERSON-TYPE(y, nice) $\rightarrow$ LIKES(x, y)

$\forall$z ANIMATE(z) $\rightarrow$ KNOWS(z, z)

$\forall$u HUMAN(u) $\rightarrow$ ANIMATE(u)

$\forall$v FRIENDLY(v) $\rightarrow$ PERSON-TYPE(v, nice)

$\forall$w HAPPY(w) $\rightarrow$ PERSON-TYPE(w, nice)

Determine

A deductive generalization of the training example that satisfies the learning goal

Reading

Tecuci G., These Lecture Notes (required).

Russell S., and Norvig P., Artificial Intelligence: A Modern Approach, Prentice Hall, Second edition, pp. 690 – 694 (recommended).

Mitchell T.M., *Machine Learning,* Chapter 11: Analytical Learning, pp. 307 - 333, McGraw Hill, 1997 (recommended).

Mitchell T.M., Keller R.M., Kedar-Cabelli S.T., Explanation-Based Generalization: A Unifying View, *Machine Learning* 1, pp. 47-80, 1986. Also in *Readings in Machine Learning*, J.W.Shavlik, T.G.Dietterich (eds), Morgan Kaufmann, 1990 (recommended).

DeJong G., Mooney R., Explanation-Based Learning: An Alternative View, *Machine Learning* 2, 1986. Also in *Readings in Machine Learning*, J.W.Shavlik, T.G.Dietterich (eds), Morgan Kaufmann, 1990 (recommended).

Tecuci G. & Kodratoff Y., Apprenticeship Learning in Imperfect Domain Theories, in Kodratoff Y. & Michalski R. (eds), *Machine Learning*, vol 3, Morgan Kaufmann, 1990 (recommended).

Overview

Machine Learning: Introduction

Inductive Learning

Deductive Learning

Abductive Learning

Analogical Learning

Multistrategy Learning

Abduction

Abduction is the operation of adopting an explanatory hypothesis that would account for all the facts or some of them.

Illustrations:

There is smoke in the East building.
Fire causes smoke.
Hypothesize that there is a fire in the East building.

Which are some other potential explanations?

Abduction

University Dr. is wet.

Raining causes the streets to be wet.
Hypothesize that it was raining on the University Dr.

Which are other potential explanations?

Provide other examples of abductive reasoning.

Abduction

Definition (Josephson, 2000):

D is a collection of data (facts, observations, givens),
H explains D (would, if true, explain D),
No other hypothesis explains D as well as H does.
Therefore, H is probably correct.

Abstract illustrations:

If $\quad B$ *is true and* $A \rightarrow B$
then $\quad$ *hypothesize* A.

If $\quad A=A_1 \,\&\, A_2 \,\&\, \ldots \,\&\, A_n$ *and* $A_2 \,\&\, \ldots \&\, A_n$ *is true*
then $\quad$ *hypothesize* A_1.

Discussion

Why is abduction a form of learning?

It discovers (learns) new facts.

Which are the basic operations in abductive learning?

- generation of explanatory hypotheses;
- selection of the "best" hypothesis;
- (testing the best hypothesis).

Overview of the Abductive Learning Approach

- **Let D be a collection of data**
- **Find all the hypotheses that (preferably causally) explain D**
- **Find the hypothesis H that explains D better than other hypotheses**
- **Assert that H is true**

How to Choose the "Best" Explanation?

Consider this: B is true and $A \to B$ *and* $C \to B$

What should we hypothesize?

- prefer to backtrace causal rules (A causes B);

- prefer to backtrace the rule that has the highest number of true left-hand side literals (where $A=A_1$ & A_2 & ... & A_n);

- prefer to backtrace the rule that has the highest number of known instances;

-prefer the simplest hypothesis, etc.

What is the justification of each approach?

The Abductive Learning Problem: Illustration

Given

• ***A surprising observation that is not explained by the background knowledge***

KILL(John, John) ; John committed suicide

Background knowledge

$\forall$x, $\forall$y, BUY(x, y) $\rightarrow$ POSSESS(x, y)

$\forall$x, $\forall$y, HATE(x, y) & POSSESS(x, z) & WEAPON(z) $\rightarrow$ KILL(x, y)

$\forall$x, GUN(x) $\rightarrow$ WEAPON(x)

$\forall$x, DEPRESSED(x) $\rightarrow$ HATE(x, x) ...

DEPRESSED(John), AGE(John, 45), BUY(John, obj1), ...

Learning goal

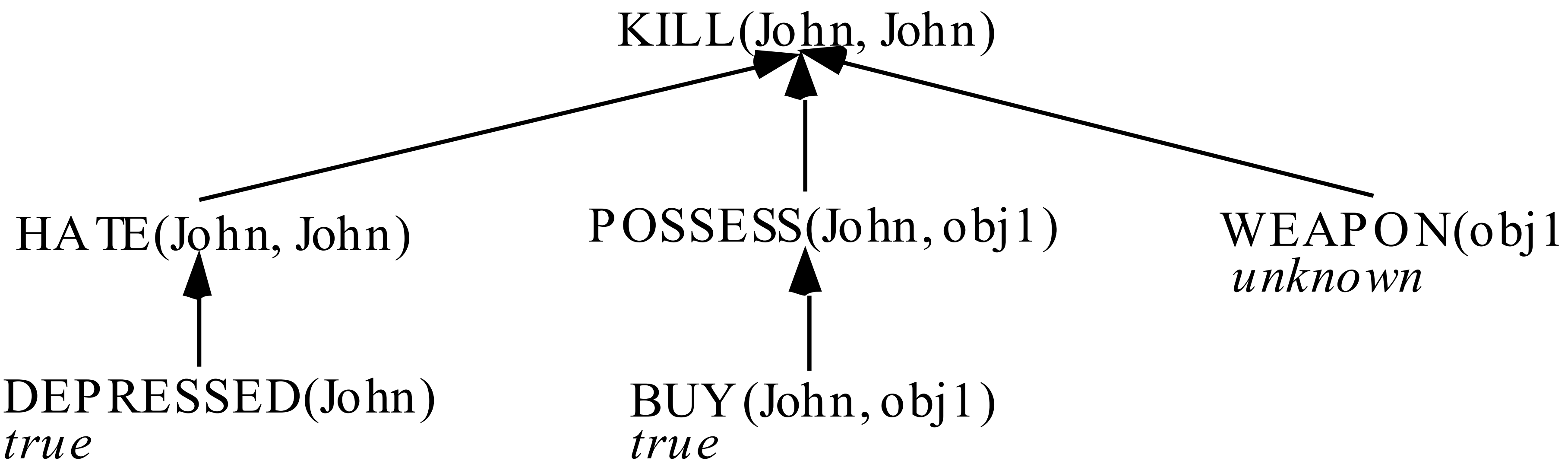
Find an assumption which is consistent with the background knowledge and represents the best explanation of the new observation.

Determine

The “best” assumption satisfying the learning goal: GUN(obj1)

The Abductive Learning Method: Illustration

Build partial explanations of the observation:

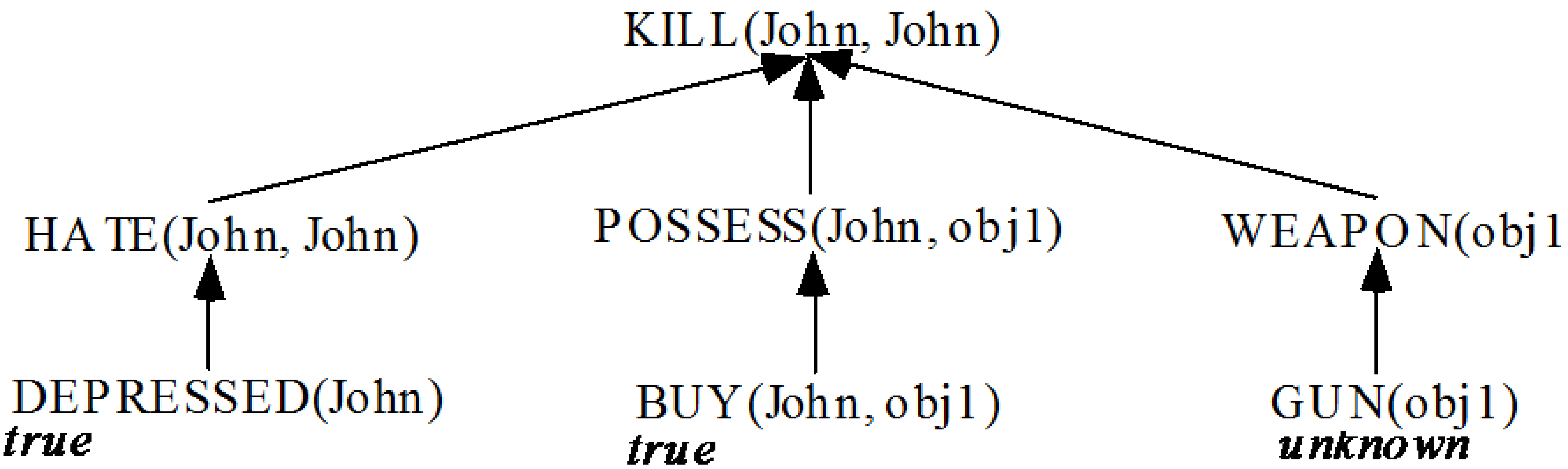


If one assumes that "WEAPON(obj1)" is true
Then "KILL(John, John)" is explained.

Therefore, a possible assumption is "WEAPON(obj1)".

The Abductive Learning Method: Illustration

Another partial proof tree:



If one assumes that "GUN(obj1)" is true
Then "KILL(John, John)" is also explained.

Therefore, another possible assumption is "GUN(obj1)".

What hypothesis to adopt?
- **the most specific one?**
- **the most general one?**

Discussion

What real world applications of abductive reasoning can you imagine?

Exercise

Given

• *Training Example*

An example of the concept "LIKES(x, y)":

HUMAN(John) & HAPPY(John) & AGE(John, 32) => LIKES(John, John)

• *Learning goal*

Find a sufficient concept definition for "LIKES", expressed only in terms of the features used in the training example (i.e. HUMAN, HAPPY, AGE)

• *Background Knowledge*

$\forall$x $\forall$y KNOWS(x, y) & PERSON-TYPE(y, nice) $\rightarrow$ LIKES(x, y)
$\forall$z ANIMATE(z) $\rightarrow$ KNOWS(z, z)
$\forall$u HUMAN(u) $\rightarrow$ ANIMATE(u)
$\forall$v FRIENDLY(v) $\rightarrow$ PERSON-TYPE(v, nice)
$\forall$w HAPPY(w) $\rightarrow$ PERSON-TYPE(w, nice)

Determine

Apply the explanation-based learning method to determine a deductive generalization of the training example that satisfies the learning goal

Exercise

Change the exercise from the previous slide to represent an abductive learning problem and then solve it.

Consider the explanation-based learning problem from the previous slide and the abductive learning problem from this exercise. Compare abductive learning with explanation-based learning, based on these problem formulations and their solutions.

Partial solution

Given

- ***A surprising observation that is not explained by the background knowledge***

  LIKES(John, John)

Background knowledge

…

Learning goal

Find an assumption which is consistent with the background knowledge and represents the best explanation of the new observation.

Determine

The "best" assumption satisfying the learning goal.

Recommended Reading

Tecuci G., These Lecture Notes (required).

P. A. Flach and A. C. Kakas (Eds.), Abduction and Induction: Essays on their Relation and Integration, Kluwer Academic Publishers, 2000.

P. A. Flach and A. C. Kakas (Eds.), Abductive and Inductive reasoning: backround and issues, in the above volume.

J. R. Josephson, Smart inductive generalizations are abductions, in the above volume.

J. R. Josephson and S. G. Josephson, Abductive inference: computation, philosophy, technology, Cambridge University Press, 1994.

O'Rorke P., Morris S., and Schulenburg D., Theory Formation by Abduction: A Case Study Based on the Chemical Revolution, in Shrager J. and Langley P. (eds.), Computational Models of Scientific Discovery and Theory Formnation, Morgan Kaufmann, San Mateo, CA, 1990.

Subramanian S and Mooney R.J., Combining Abduction and Theory Revision, in R.S.Michalski and G.Tecuci (eds), Proc. of the First International Workshop on Multistrategy Learning, MSL-91, Harpers Ferry, Nov. 7-9, 1991.

Overview

Machine Learning: Introduction

Inductive Learning

Deductive Learning

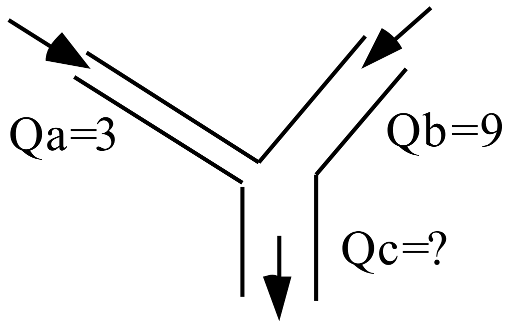
Abductive Learning

Analogical Learning

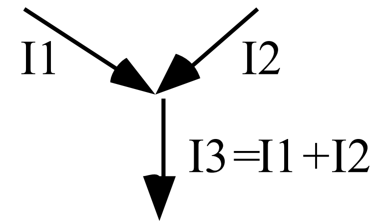
Multistrategy Learning

Learning by Analogy: Definition

Learning by analogy means acquiring new knowledge about an input entity by transferring it from a known similar entity.



Simple Hydraulics Problem



Kirchoffs First Law

One may infer, by analogy, that hydraulics laws are similar to Kirchoff's laws, and Ohm's law.

Which is the central intuition supporting the learning by analogy paradigm?

Discussion

Central intuition supporting learning by analogy:

If two entities are similar in some respects then they could be similar in other respects as well.

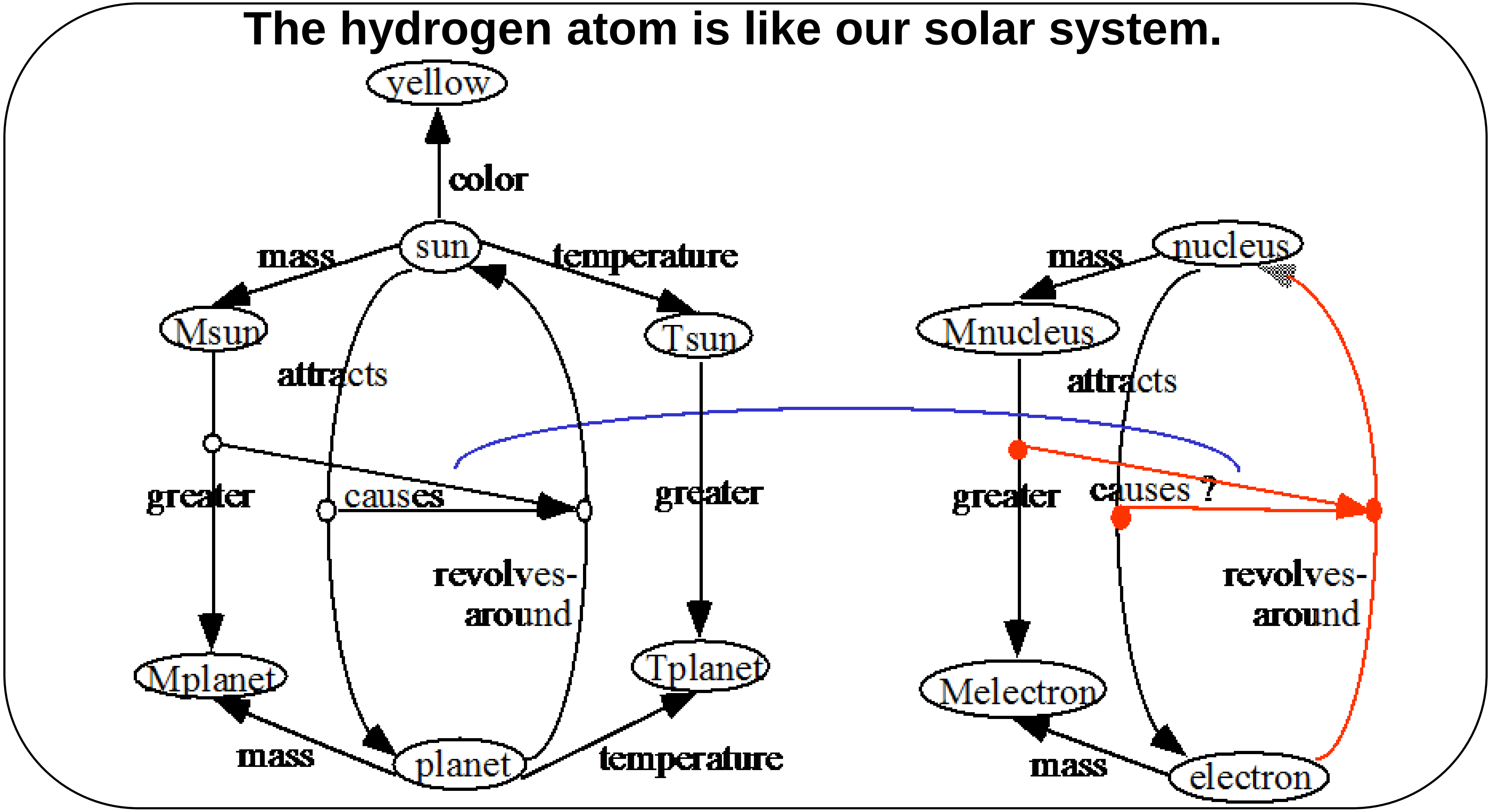
Examples of analogies:

Pressure Drop is like Voltage Drop

A variable in a programming language is like a box.

Provide other examples of analogies.

Rutherford's Analogy



The Sun has a greater mass than the Earth and attracts it, causing the Earth to revolve around the Sun.

The nucleus also has a greater mass then the electron and attracts it. Therefore it is plausible that the electron also revolves around the nucleus.

Learning by Analogy: The Learning Problem

Given:

- **A partially known target entity T and a goal concerning it.**

Partially understood structure of the hydrogen atom under study.

- **Background knowledge containing known entities.**

Knowledge from different domains, including astronomy, geography, etc.

Find:

- **New knowledge about T obtained from a source entity S belonging to the background knowledge.**

In a hydrogen atom the electron revolves around the nucleus, in a similar way in which a planet revolves around the sun.

Learning by Analogy: The Learning Method

- **ACCESS: find a known entity that is analogous with the input entity.**

In the Rutherford's analogy the access is no longer necessary because the source entity is already given (the solar system).

- **MATCHING: match the two entities and hypothesize knowledge.**

One may map the nucleus to the sun and the electron to the planet, allowing one to infer that the electron revolves around the nucleus because the nucleus attracts the electron and the mass of the nucleus is greater than the mass of the electron.

- **EVALUATION: test the hypotheses.**

A specially designed experiment shows that indeed the electron revolves around the nucleus.

- **LEARNING: store or generalize the new knowledge.**

Store that, in a hydrogen atom, the electron revolves around the nucleus. By generalization from the solar system and the hydrogen atom, learn the abstract concept that a central force can cause revolution.

Discussion

How does analogy help?

Why not just study the structure of the hydrogen atom to discover that new knowledge?
We anyway need to perform an experiment to test that the electron revolves around the hydrogen atom.

Analogy allows replacing a more complex problem (e.g. discovering new knowledge about the hydrogen atom) with a simpler problem (e.g. verifying plausible knowledge about the hydrogen atom).

Learning by Analogy: Design Issues

- **ACCESS: find a known entity that is analogous with the input entity.**

Given a target, how to identify a few potential sources in a very large storage?

- **MATCHING: match the two entities and hypothesize knowledge.**

Given a potential source, how to identify the knowledge to hypothesize?

- **EVALUATION: test the hypotheses.**

Why and how to test the hypothesized knowledge?

- **LEARNING: store or generalize the new knowledge.**

How to learn?

Case Study Discussion: Rutherford's Analogy

"The hydrogen atom is like our solar system".

In this case, the fact that S and T are analogous is already known. Therefore, the access part is solved and the only purpose of the matching function remains that of identifying the correct correspondence between the elements of the solar system and those of the hydrogen atom.

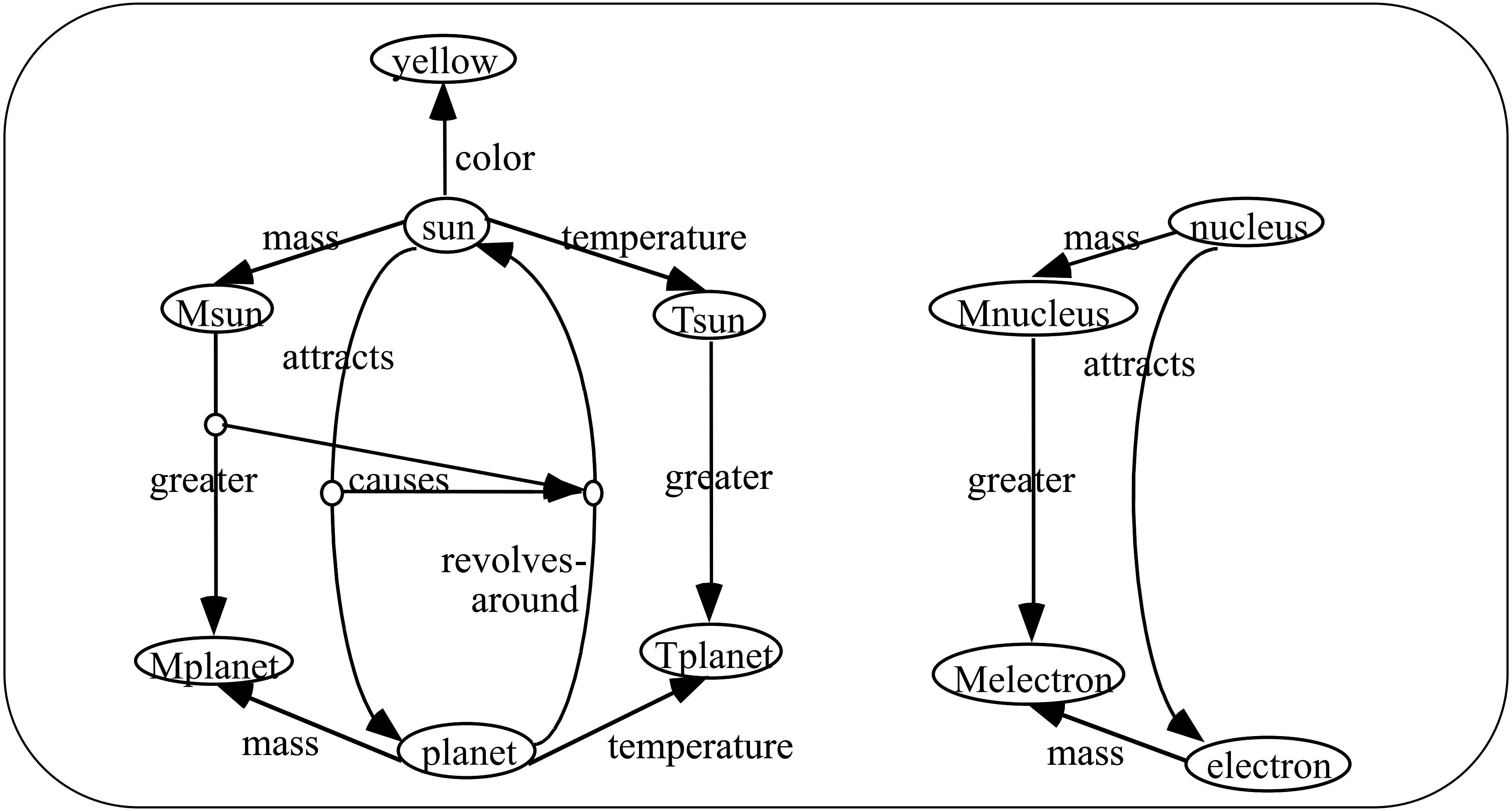
This is an example of a special (simpler) form of analogy:

"A T is like an S."

This is useful mostly in teaching based on analogy.

Case Study Discussion: Potential Matchings

Which are the possible matchings between the elements of S and the elements of T?



Case Study Discussion: Potential Matchings

There are several possible matchings between the elements of S and the elements of T and one has to select the best one:

Matching1:

sun ↔ nucleus, planet ↔ electron, Msun ↔ Mnucleus, Mplanet ↔ Melectron, which is supported by the following correspondences

mass(sun, Msun) ↔ mass(nucleus, Mnucleus)
mass(planet , Mplanet) ↔ mass(electron, Melectron)
greater(Msun, Mplanet) ↔ greater(Mnucleus, Melectron),
attracts(sun, planet) ↔ attracts(nucleus, electron)

Matching2:

sun ↔ nucleus, planet ↔ electron, Tsun ↔ Mnucleus, Tplanet ↔ Melectron, that is supported by the following correspondences

greater(Tsun, Tplanet) ↔ greater(Mnucleus, Melectron),
attracts(sun, planet) ↔ attracts(nucleus, electron)

Matching3:

sun ↔ electron, planet ↔ nucleus, Msun ↔ Melectron, Mplanet ↔ Mnucleus

Similarity Estimation

1. How to search the space of all possible matchings ?

Exhaustive search.
Other solutions?

2. How to measure the similarity of two elements ?

Two elements are similar if they represent the same concept or are subconcepts of the same concepts. In such a case their similarity may be considered 1 (on a 0-1 scale).
Other solutions?

3. How to combine the estimated similarities of the parts in order to obtain the similarity between S and T ?

The similarity of two entities is the sum of the similarity of their elements.
Other solutions?

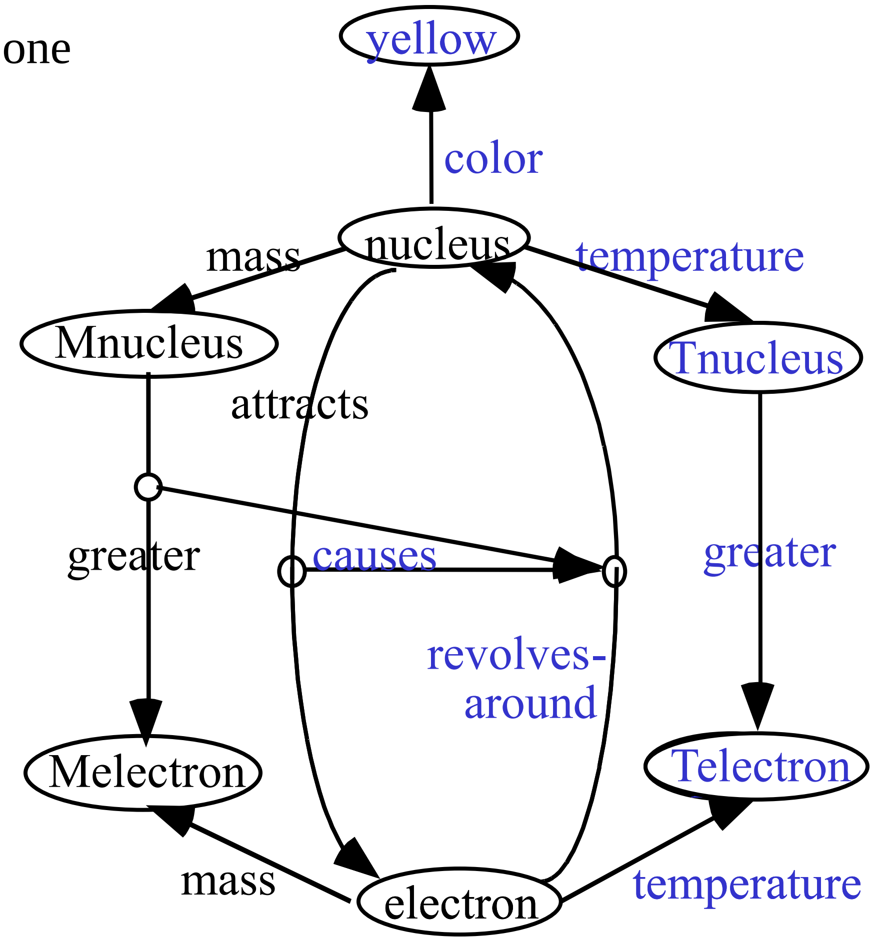
4. How to define the similarity threshold ?

Similarity threshold defined by the designer (a hard critical issue).
Other solutions?

Case Study Discussion: Matching Result

The best matching is Matching1 (because it leads to the highest number of common features of the solar system and the hydrogen atom) that gives the following substitution:
σ = (sun ← nucleus, planet ← electron, Msun ← Mnucleus, Mplanet ← Melectron)

By applying the substitution to the solar system, one obtains the following structure:



The features in blue color are those that could be transferred to the hydrogen atom as a result of the analogy with the solar system:

- color(nucleus, yellow)
- temperature(nucleus, Tn)
- temperature(electron, Te)
- greater(Tn, Te)
- revolves-around(nucleus, electron)
- causes((attracts(nucleus,electron),
  greater(Mnucleus, Melectron)),
  revolves-around(nucleus, electron))

Case Study Discussion: Evaluation

The evaluating phase shows that

The hydrogen atom has the features:
- revolves-around(nucleus, electron)
- causes((attracts(nucleus,electron), greater(Mnucleus, Melectron)),
  revolves-around(nucleus, electron))

The hydrogen atom does not have the features:
- color(nucleus, yellow)
- temperature(nucleus, Tn)
- temperature(electron, En)
- greater(Tn, En)

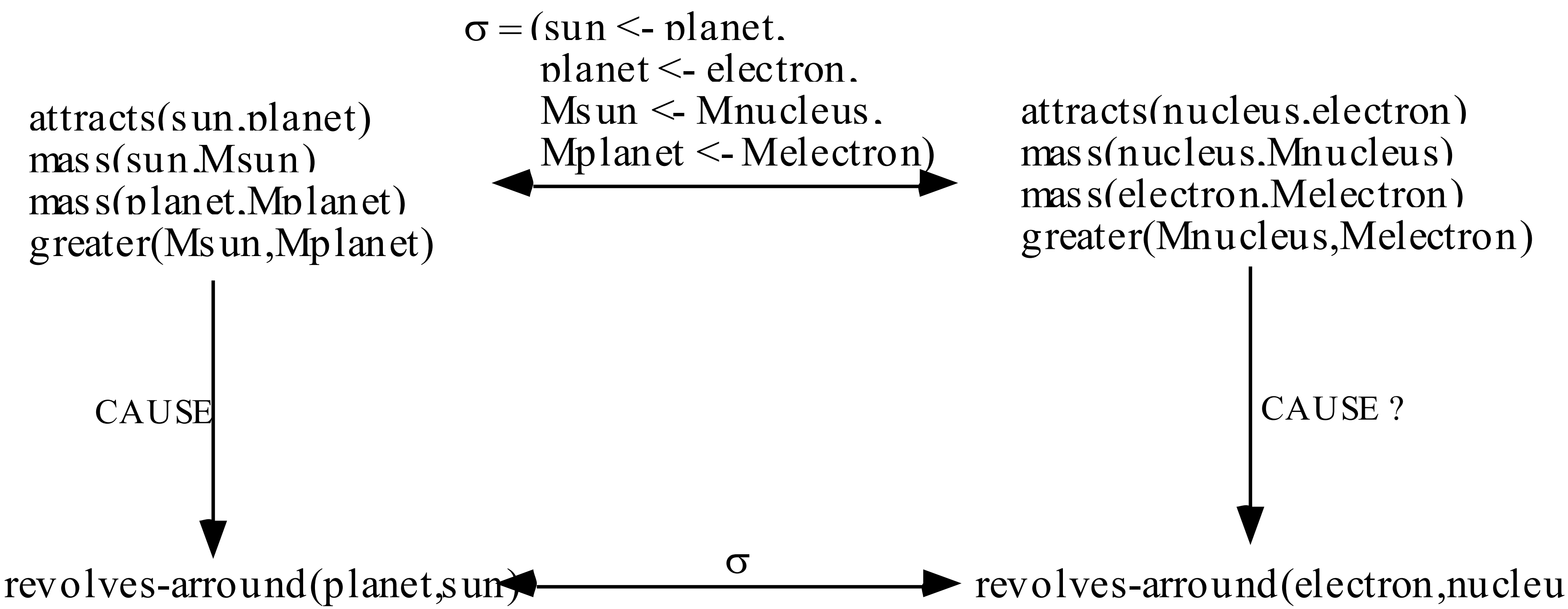
Which is, in your opinion, the most critical issue in analogical learning?

Discussion

Which is the most critical issue in analogical learning?

What kind of features may be transferred from the source to the target so as to make sound analogical inferences?

Case Study: Transfer of Causal Relations

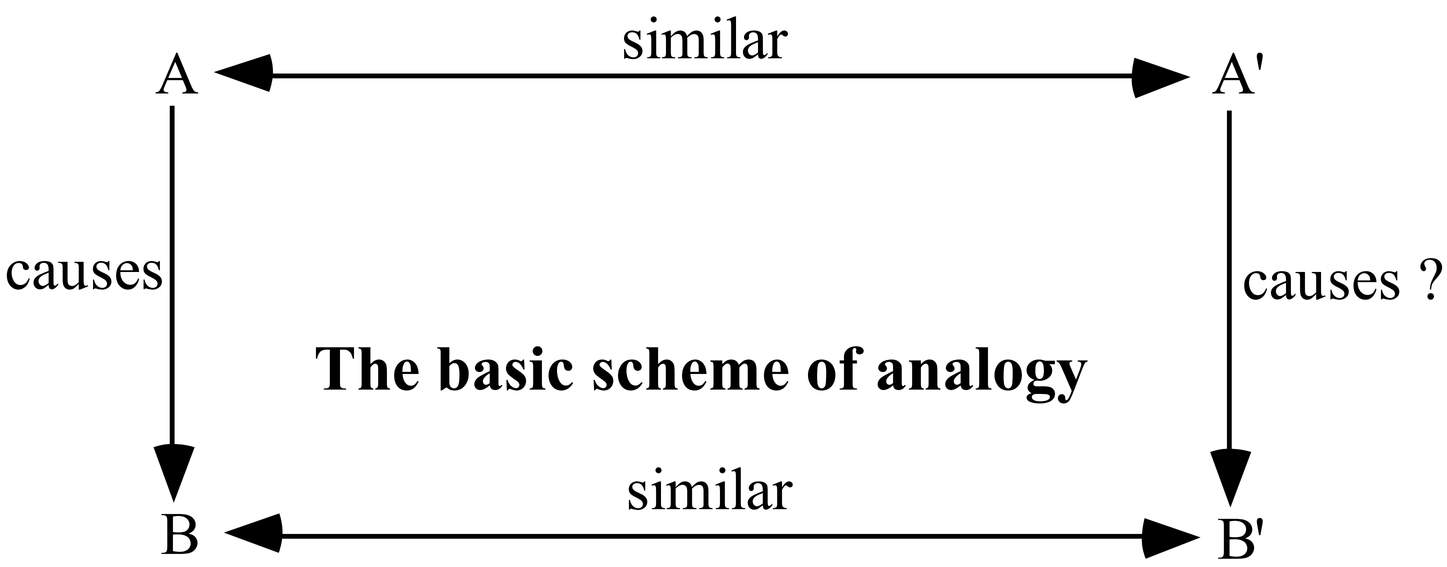


Causal Networks of Relations

An important result of the learning by analogy research is that analogy involves mapping some underlying causal network of relations between analogous situations.

By causal network of relations it is generally meant a set of relations related by special higher order relations such as 'physically-causes(ri, rj)', 'logically-implies(ri, rj)', 'enables(ri, rj)', 'justifies(ri, rj)', determines(ri, rj), etc.

The idea is that similar causes are expected to have similar effects:



The basic scheme of analogy

Case Study Discussion: Learning

Store the new acquired knowledge about the hydrogen atom:

- revolves-around(nucleus, electron)
- causes(attracts(nucleus,electron), greater(Mnucleus, Melectron)),
  revolves-around(nucleus, electron))

By generalization from the solar system and the hydrogen atom one may learn the abstract concept that a central force can cause revolution:

- causes(attracts(x, y), greater(Mx, My)), revolves-around(x, y))

Question:

When to store the acquired knowledge and when to generalize it?

Problem Solving by Analogy

Analogy means deriving new knowledge about an input entity by transferring it from a known similar entity.

How could we define problem solving by analogy?

Problem Solving by Analogy: Definition

Problem solving by analogy is the process of transferring knowledge from past problem-solving episodes to new problems that share significant aspects with corresponding past experience and using the transferred knowledge to construct solutions to the new problems.

What could be the overall structure of a problem solving by analogy method?

The Problem Solving by Analogy Method

Let P be a problem to solve.

First, look into the knowledge base for a previous problem solving episode which shares significant aspects with the problem to solve.

Next transform the past episode to obtain a solution to the current problem.

The Problem Solving by Analogy Method

Let P be a problem to solve.

First, look into the knowledge base for a previous problem solving episode which shares significant aspects with the problem to solve.

Next transform the past episode to obtain a solution to the current problem.

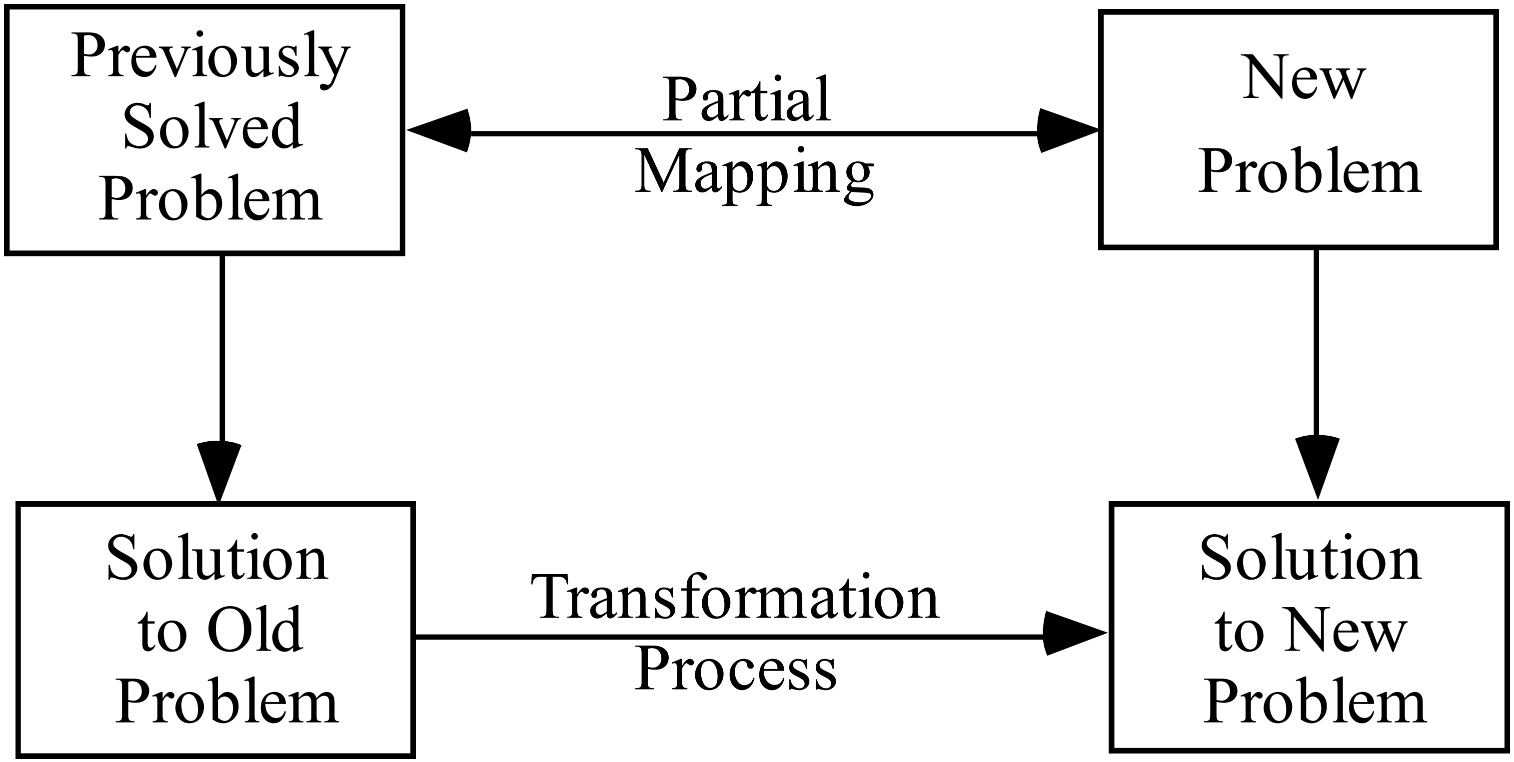
What questions need to be answered to develop such a method?

What it means for problems to share significant aspects?

How is the past problem solving episode transformed so as to obtain the solution to the current problem?

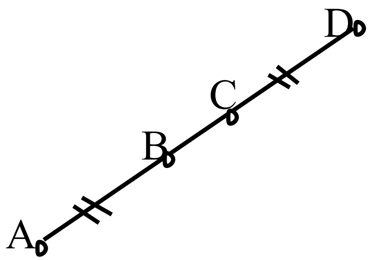
Transformational Analogy Method (Carbonell)

- Two problems share significant aspects if they match within a certain threshold, according to a given similarity metric.
- The solution to the retrieved problem is perturbed incrementally until it satisfies the requirements of the new problem.



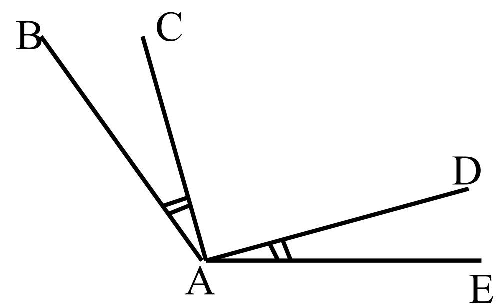
Transformational Analogy Method: Illustration

GIVEN: AB = CD
PROVE: AC = BD



σ = (AB <- <BAC
CD <- <DAE
AC <- <BAD
BD <- <CAE)

GIVEN: <BAC = <DAE
PROVE: <BAD = <CAE



AB = CD
BC = BC
AB + BC = BC + CD
AC = BD

<BAC = <DAE
<CAD = <CAD
<BAC + <CAD = <CAD + <DAE
<BAD = <CAE

Discussion

How does analogy facilitate the problem solving process?

How does the transformational analogy method relates to the generally accepted idea that the relations which are usually imported by analogy from a source concept S to the target concept T are those belonging to causal networks?

Discussion

How does this method relates to the generally accepted idea that the relations which are usually imported by analogy from a source concept S to the target concept T are those belonging to causal networks?

Intuition: The relation between a problem and its solution is a kind of cause-effect relationship.

Consider the following problem solving situation:

Problem: Find integer solutions of the problem $x^3 + y^3 = z^3$

Previously solved problem: Find integer solutions of the problem $x^2 + y^2 = z^2$

Fermat's last theorem: There is no integer solutions of $x^n + y^n = z^n$ for $n>2$

What does this example suggests?

Discussion

Except for the trivial problems, a solution does not emerge immediately from the problem formulation, as would be the case in a cause-effect relation.

What other relation from the problem solving process might be closer to a cause-effect relation?

Discussion

What other relation from the problem solving process might be closer to a cause-effect relation?

The relation between a problem and its derivation trace (i.e. solution process).

What is transferred from a past problem solving episode is not a problem solution but the problem solving process itself, what questions have been asked, what factors have been considered, etc. One would try to repeat the same process in the context of the new problem.

With this interpretation we retrieve the derivational analogy method.

The Derivational Analogy Method (Carbonell)

Two problems are considered to share significant aspects if their initial analysis yields the same reasoning steps, that is, if the initial segments of their respective derivations start by considering the same issues and making the same decisions;

The derivation of the solved problem may therefore be transferred to the new problem by reconsidering the old decisions in the light of the new problem situation, preserving those that apply, and replacing or modifying those whose supports are no longer valid in the new situation.

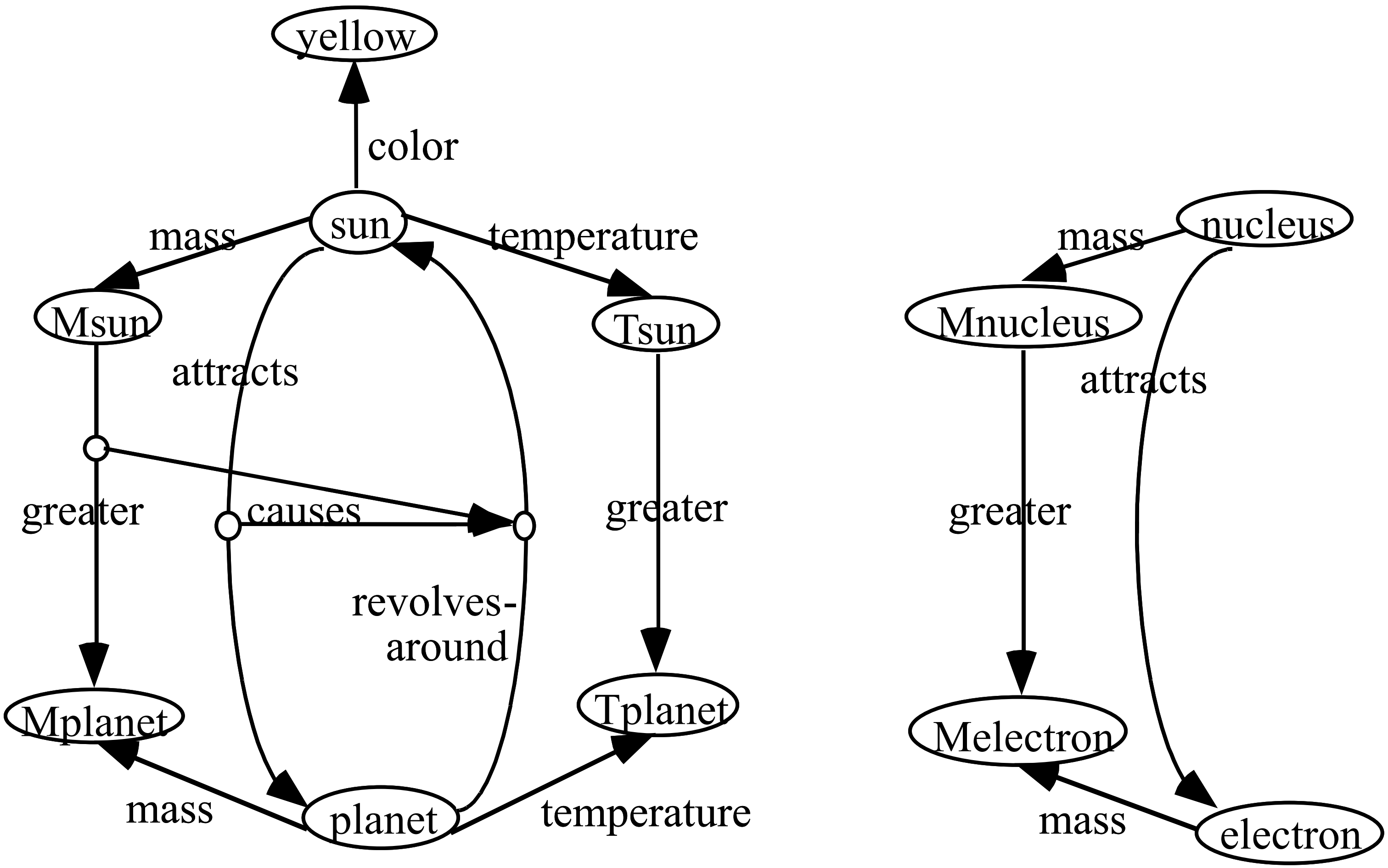
Derivational analogy gives better results than transformational analogy. However, it has the disadvantage to manipulate complex structures representing derivational traces.

Exercises

1. Define learning by analogy and give an example of analogy.

2. Describe the four stages of learning by analogy.

3. Illustrate learning by analogy with the help of the following example:



Recommended reading

Tecuci G., These Lecture Notes. (required)

Gentner D., Holyoak K.J., Kokinov B.N. (eds.), The Analogical Mind: Perspectives from Cognitive Science, The MIT Press, 2001.

Carbonell J.G., Learning by analogy: formulating and generalizing plans from past experience, Machine learning I, 1983.

Carbonell J.G., Derivational analogy: a theory of reconstructive problem solving and expertise acquisition, in Shavlik J. and Dietterich T. (eds), *Readings in Machine Learning*, Morgan Kaufmann, 1990. Also in Readings in Machine Learning and Knowledge Acquisition.

Davies T.R., Russell S.J., A logical approach to reasoning by analogy, in Shavlik J. and Dietterich T. (eds), *Readings in Machine Learning*, Morgan Kaufmann, 1990.

Gentner D., The mechanisms of analogical reasoning, in J.W.Shavlik, T.G.Dietterich (eds), Readings in Machine Learning, Morgan Kaufmann, 1990.

Winston P.H., Learning and reasoning by analogy, Communications of the ACM, 23, pp.689-703, 1980.

Forbus K.D., Exploring Analogy in the Large, in Gentner D., Holyoak K.J., Kokinov B.N. (eds.), The Analogical Mind, 2001

Tecuci, Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies, Academic Press, 1998, pp: 101-108.

Overview

Machine Learning: Introduction

Inductive Learning

Deductive Learning

Abductive Learning

Analogical Learning

Multistrategy Learning

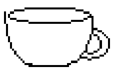
Multistrategy Learning

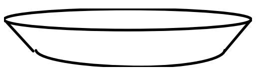
Multistrategy learning is concerned with developing learning agents that synergistically integrate two or more learning strategies in order to solve learning problems that are beyond the capabilities of the individual learning strategies that are integrated.

Learning from Examples

Compares the positive and the negative examples of a concept, in terms of their similarities and differences, and learns a concept as a generalized description of the similarities of the positive examples. This allows the agent to recognize other entities as being instances of the learned concept.

Illustration:

Positive examples of cups:  **P1**  **P2** **...**

Negative examples of cups:  **N1** **…**

Description of the cup concept: **has-handle(x), ...**

Requires many examples

Does not need much domain knowledge

Improves the competence of the agent

Explanation

The goal of this learning strategy is to learn a general description of a concept (for instance the concept of “cup”) by analyzing positive examples of cups (i.e. objects that are cups) and negative examples of cups (i.e. objects that are not cups). The learning agent will attempt to find out what is common to the cups and what distinguishes them from non-cups. For instance, in this illustration, the agent may learn that a cup should have a handle because all the positive examples of cups have handles, and the negative examples of cups do not have handles. However, the color does not seem to be important for a cup because the same color is encountered for both cups and non-cups.

To learn a good concept description through this learning strategy requires a very large set of positive and negative examples. On the other hand, this is the only information that the agent needs. That is, the agent does not require prior knowledge to perform this type of learning.

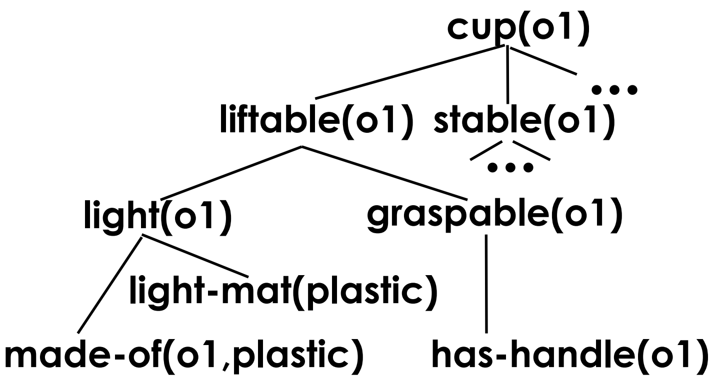
The result of this learning strategy is the increase of the problem solving competence of the agent. Indeed, the agent will learn to do things it was not able to do before. In this illustration it will learn to recognize cups, something that it was not able to do before.

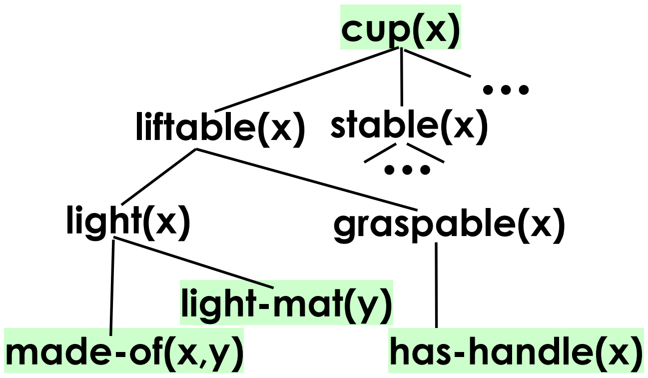
Explanation-based Learning

Learns to recognize more efficiently the examples of a concept by proving that a specific instance is an example of it, and thus identifying the characteristic features of the concept.

A example of a cup

cup(o1): color(o1, white), made-of(o1, plastic), light-mat(plastic), has-handle(o1), has-flat-bottom(o1), up-concave(o1),...





The proof identifies the characteristic features:

- **made-of(o1, plastic) is needed to prove cup(o1)**
- **has-handle(o1) is needed to prove cup(o1)**
- **color(o1,white) is not needed to prove cup(o1)**

Proof generalization generalizes them:

- **made-of(o1, plastic) is generalized to made-of(x, y);**
- **the material needs not be plastic.**

Explanation

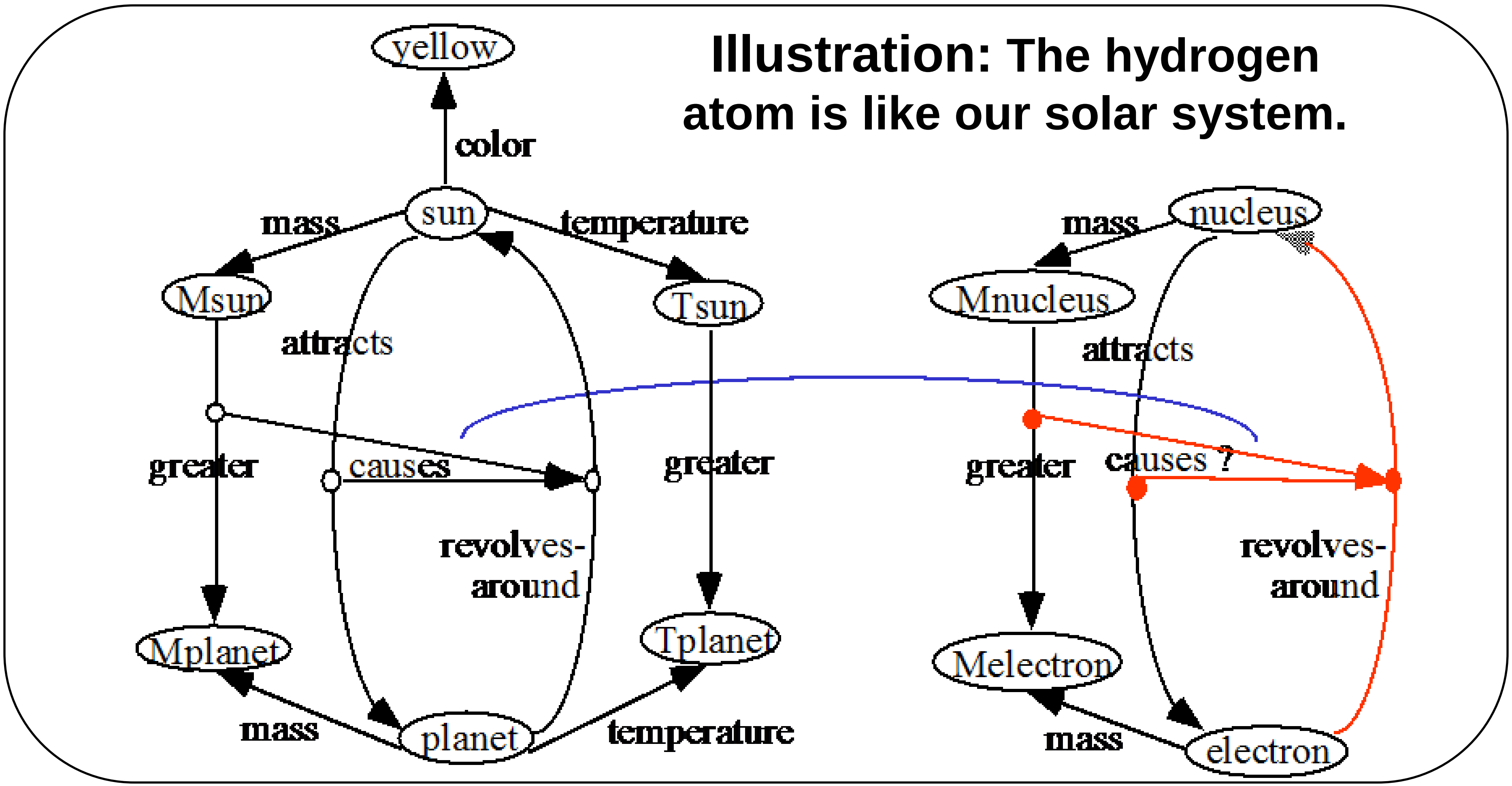
The goal of this learning strategy is to improve the efficiency in problem solving. The agent is able to perform some task but in an inefficient way. We would like to teach the agent to perform the task faster. Consider, for instance, an agent that is able to recognize cups. The agent receives a description of a cup that includes many features. The agent will recognize that this object is a cup by performing a complex reasoning process, based on its prior knowledge. This process is illustrated by the proof tree from the left hand side of this slide. The object o1 is made of plastic which is a light material. Therefore o1 is light. o1 has a handle and therefore it is graspable. Being light and graspable, it is liftable. And so on … being liftable, stable and an open vessel, it is a cup. The agent will learn from this process to recognize a cup faster. Notice that the agent used the fact that o1 has a handle in order to prove that o1 is a cup. This means that having a handle is an important feature. On the other hand the agent did not use the color of o1 to prove that o1 is a cup. This means that color is not important. Notice how the agent reaches the same conclusions as in learning from examples, but through a different line of reasoning, and based on a different type of information.

The next step in the learning process is to generalize the tree from the left hand side into the tree from the right hand side. While the tree from the left hand side proves that the specific object o1 is a cup, the tree from the right hand side shows that any object x that is made of some light material y, has a handle and some other features is a cup. Therefore, to recognize that an object o2 is a cup, the agent only needs to look for the presence of these features discovered as important. It no longer needs to build a complex proof tree. Therefore cup recognition is done much faster.

Finally, notice that the agent needs only one example to learn from. However, it needs a lot of prior knowledge to prove that this example is a cup. Providing such prior knowledge to the agent is a very complex task.

Learning by Analogy

Learns new knowledge about an input entity by transferring it from a known similar entity.



General idea of analogical transfer: similar causes have similar effects.

Explanation

Learning by analogy is the process of learning new knowledge about some entity by transferring it from a known entity.

For instance, I can teach students about the structure of the hydrogen atom by using the analogy with the solar system. I am telling the students that the hydrogen atom has a similar structure with that of the solar system, where the electrons revolve around the nucleus as the planets revolve around the sun.

The students may then infer that other features of the solar system are also features of the hydrogen atom. For instance, in the solar system, the greater mass of the sun and its attraction of the planets cause the planets to revolve around it. Therefore, we may conclude that this is also true in the case of the hydrogen atom: the greater mass of the nucleus and its attraction of the electrons cause the electrons to revolve around the sun. This is indeed true and represents a very interesting discovery.

The main problem with analogical reasoning is that not all the facts related to the solar system are true for the hydrogen atom. For instance, the sun is yellow, but the nucleus is not. Therefore, facts derived by analogy have to be verified.

A general heuristic is that similar causes have similar effects. That is, if A is similar to A’ and A causes B, then we would expect A’ to cause B’ which would be similar to B.

Complementariness of Learning Strategies

	Learning from examples	Explanation-based learning	Multistrategy learning
Examples needed	many	one	several
Knowledge needed	very little	complete knowledge	incomplete knowledge
Type of inference	induction	deduction	induction and/ or deduction
Effect on agent's behavior	improves competence	improves efficiency	improves competence or/ and efficiency

Explanation

The individual learning strategies have complementary strengths and weaknesses. For instance learning from example requires a lot of example while explanation-based learning requires only one example. On the other hand, learning from examples does not require any prior knowledge while explanation-based learning requires a lot of prior knowledge.

Multistrategy learning attempts to synergistically integrate such complementary learning strategies, in order to take advantage of their relative strengths to compensate for their relative weaknesses.

The Disciple agent uses a form of multistrategy learning that synergistically integrates learning from examples, learning from explanations, and learning by analogy and experimentation.

Exercise

Compare the following learning strategies:
- Inductive learning from examples
- Deductive (Explanation-based) learning
- Abductive learning
- Analogical learning

From the point of view of their input, background knowledge, type of inferences performed, and effect on system’s performance.

Recommended Readings

Tecuci G., These Lecture Notes (required)

Michalski R.S. and Tecuci G. (eds), *Machine Learning: A Multistrategy Approach,* vol. IV, 782 pages, Morgan Kaufmann, San Mateo, 1994.

Tecuci G., "An Inference-Based Framework for Multistrategy Learning," in Michalski R.S. and Tecuci G. (eds), *Machine Learning: A Multistrategy Approach,* vol. IV, pp. 107-138, Morgan Kaufmann, 1994.

Tecuci G., "Multistrategy Approaches to Learning: Why and How," *Informatica,* Special issue on multistrategy learning, vol.17, no.4, pp. 327-330, December 1993.

Tecuci G. (guest editor), *Informatica*, Special issue on multistrategy learning, vol.17, no.4, December 1993.

Tecuci G., Building Intelligent Agents, Academic Press, 1998, pp. 1-12 (1. Intelligent Agents), pp.50-65 (3.2 Generalization in the representation language of the agent) (recommended).