Discovery of Physical Principles from Design Experiences*

Sambasiva R. Bhatta and **Ashok K. Goel**
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
bhatta@cc.gatech.edu, goel@cc.gatech.edu

In a special issue on "Machine Learning in Design" of the International Journal *AI EDAM* (AI for Engineering Design, Analysis, and Manufacturing), 8(2), Spring 1994.

Abstract

One method for making analogies is to access and instantiate abstract domain principles, and one method for acquiring knowledge of abstract principles is to discover them from experience. We view generalization over experiences in the absence of any prior knowledge of the target principle as the task of hypothesis formation, a subtask of discovery. Also, we view the use of the hypothesized principles for analogical design as the task of hypothesis testing, another subtask of discovery. In this paper, we focus on discovery of physical principles by generalization over design experiences in the domain of physical devices. Some important issues in generalization from experiences are what to generalize from an experience, how far to generalize, and what methods to use. We represent a reasoner's comprehension of specific designs in the form of structure-behavior-function (SBF) models. An SBF model provides a functional and causal explanation of the working of a device. We represent domain principles as device-independent behavior-function (BF) models. We show that (i) the function of a device determines what to generalize from its SBF model, (ii) the SBF model itself suggests how far to generalize, and (iii) the typology of functions indicates what method to use.

1 Introduction

Analogy, as it is commonly accepted, plays an important role in reasoning. Making analogies, however, is not always easy due to the difficulty of retrieving the right analog from memory

*An earlier, shorter paper appeared in the Procs. of AID'92 workshop on "Machine Learning in Design," June 1992, Pittsburgh, USA, and in the Procs. of ML'92 workshop on "Machine Discovery," July 1992, Aberdeen, Scotland.

and deciding on what to transfer from the retrieved analog to the problem at hand. One method of analogical transfer is to directly map the analog to the current problem [Gentner, 1983]. This method also forms the basis of much recent work in case-based reasoning [Kolodner and Simpson, 1989; Riesbeck and Schank, 1989; Hammond, 1989; Ashley and Rissland, 1988; Alterman, 1988]. For example, in our earlier work on case-based design, we showed how *structure-behavior-function* (SBF) models of physical devices can be used for directly mapping the designs of those devices to new problems [Goel, 1991a]. An SBF model of a device captures the reasoner's comprehension of how the device works, that is, how the structure of its design results in its output behaviors. While such methods can be very useful for making analogies within a given domain, cross-domain transfer often requires higher-level abstractions such as domain principles. Since physical principles and processes typically do not refer to any specific device, we represent them as *behavior-function* (BF) models. In this model-based method for analogical transfer, new problems are solved by accessing and instantiating the BF models of principles and processes.

An important issue in model-based analogy is how to acquire knowledge of domain principles. One solution is to acquire them from a teacher, which, in fact, is a common method for acquiring such knowledge. Another method is to incrementally *discover* the principles from experiences. For example, auto mechanics apparently learn principles of automobile engineering from their experiences with auto repair although they do not always start with a deep understanding of the domain. Similarly, electronic hobbyists often learn about electrical processes from their experiences in designing electronic circuits and designers of heating and cooling equipment might acquire an understanding (i.e., a model) of how "heat exchange" occurs and what is "heat flow." By discovery, we mean learning a "concept" description from examples without knowing the target concept *a priori*. This is unlike most explanation-based learning systems [DeJong and Mooney, 1986; Mitchell *et al.,* 1986] that assume some knowledge of the target concept that needs to be learned.

The process of discovery is generally considered to have two distinct phases [Klahr and Dunbar, 1988]: *hypothesis formation* and *hypothesis testing*. One method for hypothesis formation is to incrementally generalize over design experiences. The use of a generalization in analogy acts as a test for the hypothesis. Depending on the feedback from this evaluation, the hypothesis may get revised (generalized further or refined).

In this paper, we focus on the formation of hypotheses about physical principles such as the "zeroth law of thermodynamics" from experiences in designing physical devices and briefly touch upon how they can be used for analogical design. This law states that when a hot body is brought in thermal contact with a cold body, heat flows from the hot body to the cold body [Fermi, 1937]. We show how the BF models of physical principles can be acquired by a gradual removal of structural information from the SBF models of specific devices. This process of generalization occurs while storing a design case for potential reuse. Kerr and Duffy [1992] consider generalization of past designs as one way of rationalizing past design knowledge such that it is useful in later design.

Generalization from experiences raises three important issues:

1. **The issue of relevance:** This is the issue of deciding what to generalize from an experience. With respect to this issue, the method of pure induction over design experiences could potentially become complex. Hence there is a need for developing more efficient and effective learning methods that can bias the learning in design and reduce its complexity. We show that the specification of the function of a new device can help determine what to generalize from its SBF model, and thus alleviate the problem of complexity with subsequent induction.[1]

2. **The issue of level of generalization:** This is the issue of determining how far to generalize a chosen aspect of the device. We show that the SBF model together with the knowledge of design objects, such as components and substances, can help determine how far to generalize.

3. **The issue of method selection:** This is the issue of deciding what methods to use for generalization. We show that a typology of device functions can help to determine what strategy to use.

[1] In applying induction, most existing methods in machine learning assume that the instances/examples for induction are available in batch; in contrast, our model-based method relaxes this assumption and allows for experiences to come in incrementally.

Figure 1 presents the learning task we analyzed in this paper.

Input: • Design Experience [consisting of design problem (i.e., function), solution (i.e., structure), and explanation (i.e., SBF model)].
e.g., design of a sulfuric acid cooler.

Output: • Generic principles of the domain (represented as BF models).
e.g., the zeroth law of thermodynamics.

Method: • Model-based generalization with inductive biasing.
e.g., function of a design determines what parts of the experience to focus on.

Knowledge:
- Typology of primitive functions in the domain.
  e.g., ALLOW, PUMP.
- Typology of functions in the domain (consisting of primitive functions).
  e.g., substance-parameter transformation.
- Substances in the domain.
  e.g., nitric acid, water.
- Components in the domain.
  e.g., pipe, chamber.
- Past design experiences in memory.
  e.g., design of a nitric acid cooler.

Figure 1: **Learning task analyzed in this paper**

The proposed model-based method(s) for discovering physical principles from design experiences is implemented as a learning component of IDeAL,[2] an integrated system for design by analogy and learning of abstract models, that designs physical devices such as electrical circuits and heat exchangers.

2 Design Experience

IDeAL takes as input a specification of a function of the desired design and gives as output a structure that realizes the specified function and an SBF model that explains how the structure realizes that function. A design case in IDeAL specifies (i) the functions delivered by the stored design, (ii) the structure of the design, and (iii) a pointer to the causal behaviors of the design (SBF model). Since IDeAL solves *function-to-structure* design tasks, cases are indexed by the functions that the stored designs can deliver. The design cases are organized

[2] IDeAL stands for Integrated "DEsign by Analogy and Learning."

along multiple dimensions of generalization where the dimensions pertain to the constituents of design functions.

The problem-solving component of IDeAL evolves from Kritik, an integrated case-based and model-based design system [Goel, 1991a; Goel, 1992]. Given the functional specification of a desired design, IDeAL retrieves the closest matching case from the case memory. Then it uses the SBF model of the selected design to adapt the design structure so as to meet the given functional specification. Next it revises the SBF model of the old design to incorporate the structural modifications and generates an SBF model for the new design. IDeAL uses repair plans for modifying a selected design. It verifies the new design by a qualitative simulation of the new SBF model. Finally, the new design case generated by IDeAL acts as input to its learning component. IDeAL first learns indices to the new design case [Bhatta and Goel, 1992] and stores the design for potential reuse. While storing design cases it notices similarities between the SBF models of specific designs in memory and discovers principles as described later in this article. The learned principles are intended to be evaluated and revised by their use in cross-domain analogical design.

3 Device Models

IDeAL's functional models of specific devices are represented in the form of structure-behavior-function (SBF) models. These models are based on a *component-substance ontology* [Bylander and Chandrasekaran, 1985]. This ontology gives rise to a representation language [Goel, 1992] for describing the SBF model of a design that is a generalization on Sembugamoorthy and Chandrasekaran's [1986] functional representation scheme. The constituents of the SBF model are described below.

Structure: The structure of a design is expressed in terms of its constituent components and substances and the interactions between them. Figure 2 shows the structure of a `sulfuric acid cooler` (SAC) schematically.

Function: A function is represented as a schema that specifies the behavioral state the function takes as input, the behavioral state it gives as output, and a pointer to the internal causal behavior of the design that achieves the function. Figure 3(a) shows a function of the

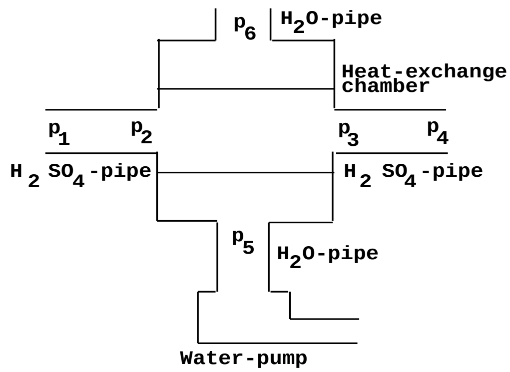


Figure 2: **Sulfuric Acid Cooler**

`SAC`, namely, heating water. The input state of the function specifies that water at `location` p5 in the topography of the device (Figure 2) has the properties `temperature` and `flow`, and corresponding parameters t_1 and r'. It also specifies that the water contains another substance `heat` whose `magnitude` is q_1. Similarly, the output state specifies the properties and the corresponding parameters of the substance at `location` p6.

This representation of functions gives rise to a typology of functions in the domain: transformation functions, control functions, maintenance functions, etc. In this article, we will be focusing on transformation functions, which again are of several types: substance transformation, substance-parameter transformation, and substance-location transformation. For example, the function of SAC is both a *substance-parameter transformation* and a *substance-location transformation* because it specifies a change in the parameter of the substance temperature as well as a change in the substance location.

Behavior: The internal causal behaviors of a device are viewed as sequences of *state transitions* between *behavioral states.* The annotations on the state transitions express the *causal, structural,* and *functional context* in which the transformation of state variables, such as substance, location, properties, and parameters, can occur. The causal context provides *causal relations* between the variables in preceding and succeeding states. The structural context specifies different kinds of structural information such as substances, components, structural relations among components and substances, and spatial locations in the device. The functional context indicates which functions of components in the device are responsible

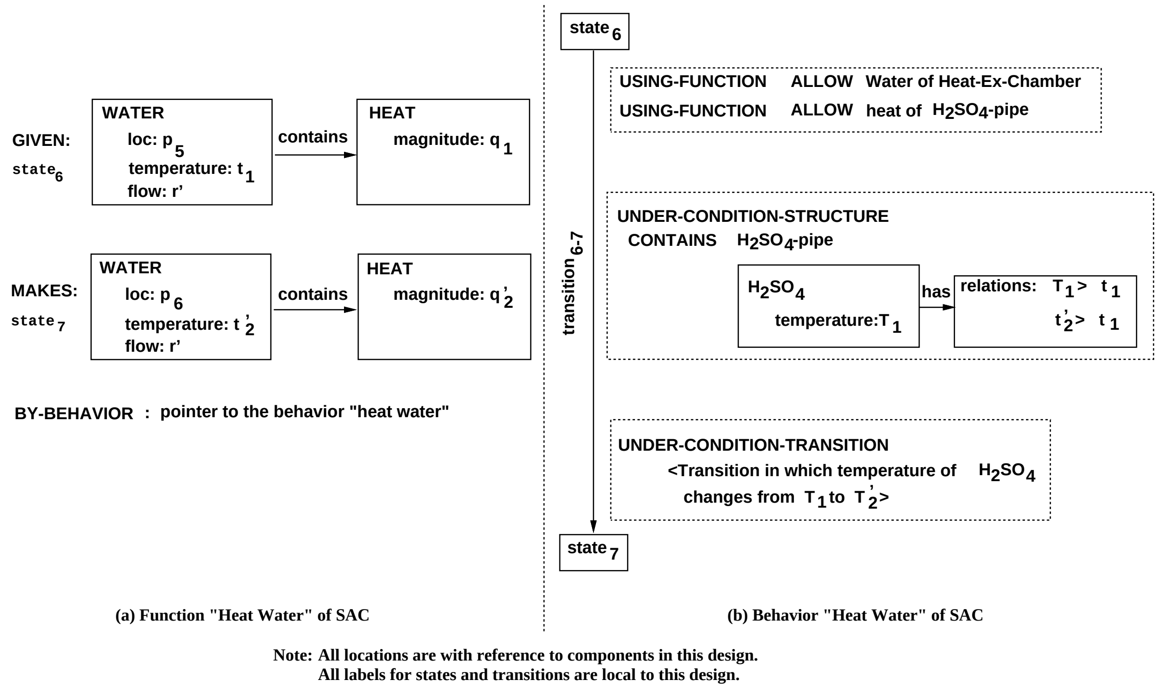


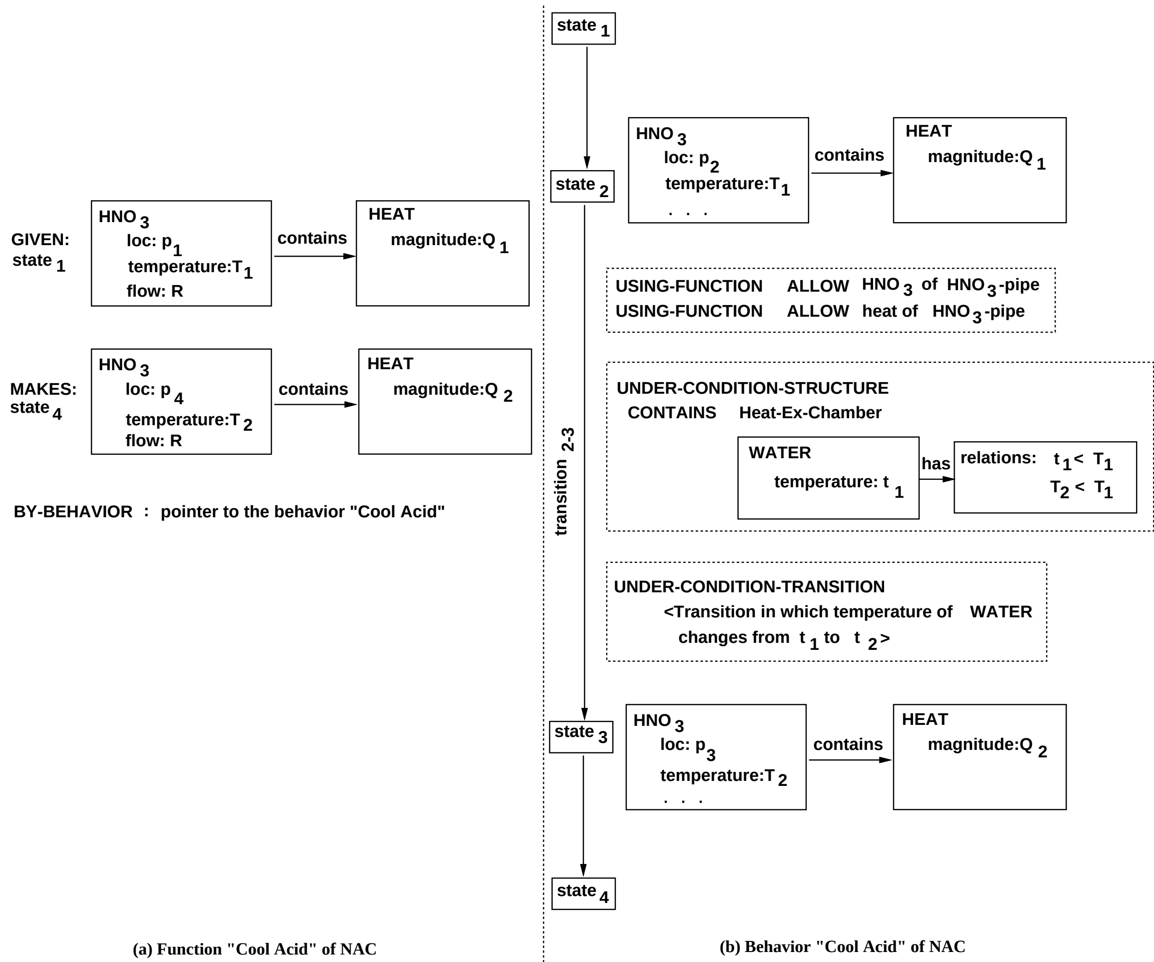
Figure 3: **Function and Behavior of Sulfuric Acid Cooler**

for the transition. Figure 3(b) shows the causal behavior that explains how water is heated from `temperature` t_1 to $t_2{}'$. $State_6$, the preceding state of $transition_{6-7}$, describes the state of water at location p5 and $state_7$, the succeeding state, at location p6.

The UNDER-CONDITION-STRUCTURE annotation in $transition_{6-7}$ specifies that the behavior **allow** of H_2SO_4-pipe can allow the flow of heat only if the H_2SO_4-pipe CONTAINS sulfuric acid with a temperature of T_1 that is greater than t_1. The qualitative parameter relations on the substance properties, such as those shown for `temperature` in Figure 3(b), are a crucial part of describing the causal process underlying a transition.

4 A Model-Based Method for Hypothesis Formation

Consider, for example, the situation in which IDeAL finds multiple (e.g., two) cases to be similar in their functions while it is storing a design case in the functionally organized case memory. We will consider the designs of sulfuric acid cooler and nitric acid cooler whose function and behavior are shown respectively in Figures 3 & 4 for the purpose of illustrating the methods. The similarity between two functions is determined by comparing the input



Note: All locations are with reference to components in this design.
All labels for states and transitions are local to this design.

Figure 4: **Function and Behavior of Nitric Acid Cooler**

---

state and output state in them. Furthermore, similarity between two states is determined by comparing different slots in the schemas, such as `substance`, `location`, and other properties. For instance, a function F_1 is more similar to another function F_2 than it is to F_3 if the `substance` in both F_1 and F_2 is same while it is different in F_3. For example, the function of a nitric acid cooler that cools nitric acid from T_1 to T_2 is more similar to another nitric acid cooler that cools nitric acid from T_1 to T_3 than it is to a sulfuric acid cooler that cools sulfuric acid from T_1 to T_2. This is based on the heuristic that changing a substance altogether in a design is harder than changing a property of a substance. These similarity measures are based on those used in Kritik for accessing cases from memory [Goel, 1992].

In addition to generalizing the functions of similar design cases, IDEAL can also generalize the associated SBF models for use in solving problems by analogy in a different domain with the experience gained in one domain. However, IDEAL does not know *a priori* what the target "concept" will be; hence, it formulates the generalized model as a hypothesis.

As mentioned earlier, the function of a device determines what parts of its model to generalize. If the function is a *transformation function* (e.g., substance transformation, substance-parameter transformation, substance-location transformation) then any relations in the different types of context annotating the transitions in the behavior that describe the corresponding change and the transitions themselves can be generalized to form meaningful abstractions of behaviors. For example, since the function "heat water" of sulfuric acid cooler is to transform the `temperature` of the substance `water` from one value to another, the transition $transition_{6-7}$ in Figure 3(b) is useful to focus on. The relations on the parameters of `temperature` describing the change can be generalized along with the similar behavior of another cooler or heater. In addition to the parametric relations, other aspects of the context, such as conditions on substance and conditions on structural relations that involve the parameter being transformed, also form an important part of the content to be generalized.

After identifying what parts of the specific models to focus on, the issue is to determine what kinds of changes along a dimension are meaningful for generalization. In other words, the issue is what similarities between the two models (in the focused segments of the behaviors) are retained, as they are, in the generalization and what differences are generalized. The same kind of similarity metrics as those for comparing functions are used for this purpose as well, because a focused segment of behavior includes a sequence of states and state-transitions. However, in addition to comparing states, the annotations on the transitions are also compared as guided by the functions (explained above). Since generalizations tend to deal with more qualitative parameters than specializations, we consider *positive* changes (i.e., increase) and *negative* changes (i.e., decrease) in the parameter of the chosen property for generalization. The changes across different models under consideration suggest the level of generalization. Since SBF models specify different kinds of structural information (e.g., locations, structural relations, components etc.), successive removal of each kind leads to the formation of models at different levels of abstraction. By removal, we mean two things: (i) substitution of specific values (e.g., low and medium) by a value from a more general class of values (e.g., qualitative-value) in a value hierarchy; and (ii) a complete deletion of specific structural information (e.g., deleting the information that some substance *moves* from one location to another). These will become clearer from the example illustrated below.

Since some functions such as that of a `sulfuric acid cooler` can be classified in multiple ways, multiple subtasks of generalization can be performed—generalization over parameter changes and generalization over changes in location. Depending on which generalization is performed on given experiences, different types of abstract models will be formed. However, in some cases, both might be applicable; in such a case of multiple subtasks, generalization occurs to multiple levels. IDeAL applies both methods, when applicable, in a specific order, that is, it generalizes over parameter changes prior to changes in location. Models at intermediate levels of abstraction are models of prototypical devices (similar to design prototypes [Gero, 1990]) such as the model of a heat exchanger that is applicable to both coolers and heaters. Models at still higher levels of abstraction are such as the model of a physical principle "the zeroth law of thermodynamics" or the physical process "heat flow."

Consider the design of a sulfuric acid cooler (Figure 2) and its function of heating water for the purpose of illustrating the methods. The type of this function (i.e., substance-parameter transformation as well as substance-location transformation) suggests two methods for generalization: (i) generalization over substance-parameter transformation (Figure 5) and (ii) generalization over substance-location transformation (Figure 6). The transitions $transition_{2-3}$ in the behavior "cool acid" of NAC (Figure 4(b)) and $transition_{6-7}$ in the behavior "heat water" of SAC (Figure 3(b)) are selected for generalization because they transform parameters of the substance temperature and the substance location.

The application of the method shown in Figure 5 to these two behaviors results in the description of a generalized model as shown in Figure 7, which is the SBF model of a heat exchanger (a prototypical device). Note that the structural information in the behaviors of SAC and NAC is generalized and so are the parametric relations in the corresponding

Input: • E_1, the new design experience.
• E_2, a design experience found to be similar to E_1 under the same node in memory.
Output: • Generalized model from E_1 and E_2.
Procedure:
if (function of E_1 is substance-parameter-transformation)
  then
    begin
    (1) Get transitions, TR_1 and TR_2, corresponding to the transformed parameter
        in E_1 and E_2 respectively.
    (2) Compare the change in parameters in TR_1 and TR_2 qualitatively.
        if (direction of change is same in TR_1 and TR_2)
          then generalize over "range" of the parameters;
          else generalize over the direction of change;
    (3) Modify other context in TR_1 and TR_2 that specifies this parameter. That is,
        if (any "inequalities" exist on the parameter-relations)
          then generalize the inequalities to conditional inequalities;
    (4) Propagate this generalization to other dependent parameters and transitions,
        and then repeat step (3) until all the context is generalized.
    (5) Store the generalized model from E_1 and E_2.
    end.

Figure 5: **A model-based method for generalizing over parameter transformation**

transitions (Figure 7). For instance, the specific components H_2SO_4-pipe and HNO_3-pipe that achieve the function "**allow** heat" are generalized to the abstract component `pipe` achieving the same function, which is prototypical of a heat exchanger.

IDeAL's knowledge of components that H_2SO_4-pipe and HNO_3-pipe belong to the class of *pipes* helps in doing this generalization. Also, the parametric relations in Figure 7 cover both possibilities, that is, *increase* and *decrease* in the substance temperature, unlike those in the behavior of either SAC or NAC alone. This is essential to describing the behavior of a heat exchanger. Further, the generalizations are propagated to the behaviors of those substances on which the transitions depend, which is indicated by UNDER-CONDITION-TRANSITION in the Figures 3(b) & 4(b). That is, in step 4 of the method (Figure 5), for instance, the generalizations performed on the behavior segment (say, "heat water" of sulfuric acid cooler) are propagated to the dependent transition (i.e., "cool acid" of sulfuric acid cooler) which results in the generalized segment "cool substance" shown in Figure 7.

Input: • E_1, the new design experience or newly generalized experience.
• E_2, a design experience (perhaps generalized before) found to be similar to E_1, if any, under the same node in memory.

Output: • Generalized model from E_1 (and E_2).

Procedure:

if (function of E_1 is substance-location-transformation)
  then
    begin
    (1) Get transitions, TR_1 and TR_2, corresponding to the location in $E_1$and E_2 respectively.
    (2) Compare the causal context that involves location in TR_1 and TR_2.
      if (causal context is similar in TR_1 and TR_2)
        then generalize/variablize locations;
        else generalize over the associated structural elements;
    (3) Modify other context that involves locations and associated structural information. That is,
      if (any structural conditions exist in TR_1 and TR_2 and they are similar)
        then remove the structural conditions;
        else check for similarity at a more abstract level of components involved;
    (4) Propagate this generalization to other dependent parameters and transitions, and then repeat step (3) until all the context is generalized.
    (5) Store the generalized model from E_1 and E_2.
    end.

Figure 6: **A model-based method for generalizing over location transformation**

The application of the method shown in Figure 6 to the result of applying the first method, that is, to the model of the heat exchanger, leads to the formation of an even further generalized model as shown in Figure 8. This is the generic principle that we call the zeroth law of thermodynamics. This model is also a description, although partial, of the process "heat flow" because the process of heat flow is the behavior that the zeroth law of thermodynamics epitomizes.[3] However, the system, conforming to the classical "term problem" in learning, does not realize that this is *the* zeroth law of thermodynamics nor does it realize that this is a partial description of the process "heat flow," but rather considers it simply as an abstract model possibly applicable to a wider class of devices. Again, note that the structural information in the behavior of heat exchanger is further generalized in the zeroth law of thermodynamics. For instance, the component `pipe` that achieves the function "**allow** heat" is generalized to an abstract component `connector` achieving the

[3]A complete description should also indicate that heat continues to flow from a hot body to a cold body only until an equilibrium temperature is reached.

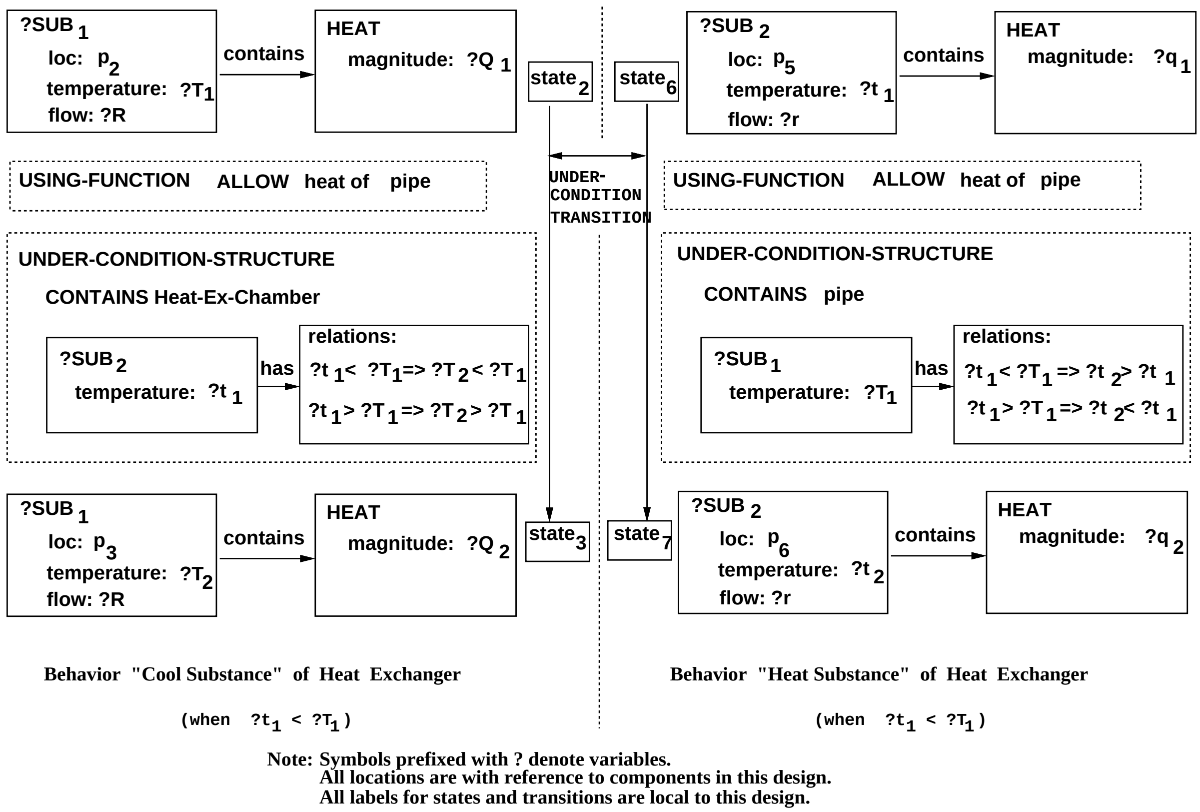
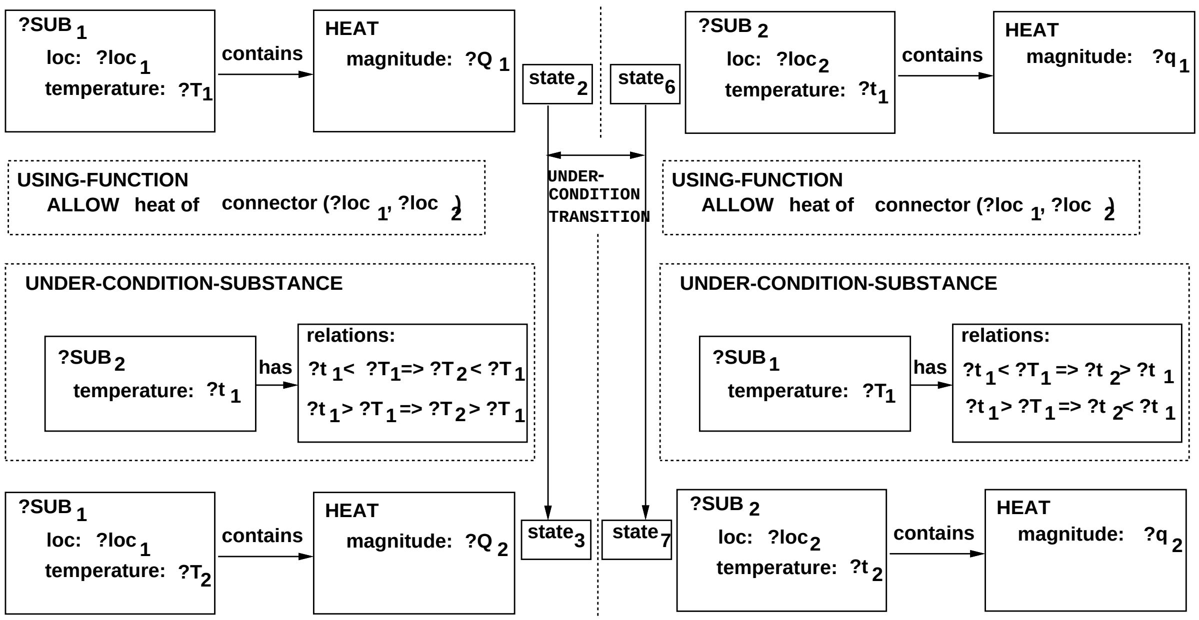


Figure 7: **Behavior of a Heat Exchanger generalized from SAC and NAC**

---

same function.

In addition to the result of generalization over structure, the generalized parametric relations in Figure 8 that cover both *increase* and *decrease* in the substance temperature are also crucial to representing the zeroth law of thermodynamics. These relations are represented as conditions on substance properties indicated by the annotation UNDER-CONDITION-SUBSTANCE in Figure 8 because the structural conditions (in Figure 7) are removed by the application of step 3 in Figure 6. Again, step 4 in the method shown in Figure 6 leads to the propagation of generalizations performed in one behavioral segment to the dependent ones.



Note: Symbols prefixed with ? denote variables.
All locations are with reference to components in this design.
All labels for states and transitions are local to this design.

Figure 8: **The Zeroth Law of Thermodynamics**

4.1 Learning Indices to Hypothesized Models

When a piece of knowledge is learned, its usefulness relies on the ability to also learn the appropriate conditions under which it might be used. In other words, learning a piece of knowledge inevitably involves learning its *indices.* So, when the models of the heat exchanger and the zeroth law of thermodynamics are hypothesized, one subtask is to learn their indices. Since these models are learned in the context of analogical design and are intended for the design task, like design cases, they could also be indexed by different types of indices—functional and structural.

Storing the hypothesized models in a hierarchically organized memory implies two distinct issues in index learning: *learning the indexing vocabulary* and *learning the right level of generalization* [Bhatta and Goel, 1992]. Deciding on the indexing vocabulary generally requires some notion of what is important about the new knowledge and the task for which it is likely to be reused. The level of generalization depends in part on the knowledge already stored in memory and the inductive biases that can be generated at storage time.

We have earlier shown how the SBF model of a design, together with a specification of the task for which the design case might be reused, provides the vocabulary for indexing the design case in memory [Bhatta and Goel, 1992] . Further, we have also shown how the model-based method, together with similarity-based learning (using earlier design cases in memory) helps to determine the level of index generalization. Insofar as the same types of indices are used for storing the models of heat exchangers and the models of the zeroth law of thermodynamics, the same methods as presented in [Bhatta and Goel, 1992] apply to the task of learning indices to the hypothesized models. The only difference is that the indices for these models will be more general than those for design cases. Hence these models will be stored at a more general level in a hierarchical organization of memory. However, space constraints do not permit us to describe these methods here.

5 Evaluation

The proposed model-based method can be evaluated for different things: (i) Computational Efficiency; (ii) Domain Generality; and (iii) Performance Task.

Computational Efficiency: The issue here is whether IDeAL requires a number of examples in order to learn a target concept such as the zeroth law of thermodynamics. Due to the constraints that SBF models provide on the generalization process, IDeAL does not require more than a few (e.g., 3-4) examples for learning the zeroth law of thermodynamics. From the examples illustrated in this paper, it is clear that IDeAL required only two examples for learning a reasonably complete description of the zeroth law of thermodynamics.

Domain Generality: The question here is whether the proposed methods are applicable in different domains? Currently, they have been tested only in the domain of heat exchangers; that is, for learning the model of a heat exchanger and the zeroth law of thermodynamics. However, from our experience with other tasks for which SBF models have been used, such as case-based design [Goel, 1989; Goel, 1991a; Goel, 1991b] and index learning by model-based generalization [Bhatta and Goel, 1992; Bhatta and Goel, 1993], in different domains, it appears that the proposed method is also applicable in other domains of physical devices such as electric circuits (i.e., for instance, in learning Ohm's law). This is because the main power of the method comes from the representational framework that the SBF models provide.

Performance Task: The question here is whether model-based learning of principles or processes affect some performance task. The motivation is that a target concept is best learned if done in the context of a performance task in which it gets used. We consider two related but different performance tasks, namely, device redesign and design of physical devices by analogy in which learned principles are useful.

(i) Device Redesign: The device redesign task takes as input a failed design and the feedback from the environment in which the device operates, and gives as output a new modified design. The physical principles learned by IDeAL such as the zeroth law of thermodynamics are useful in device redesign. For instance, Prabhakar and Goel [1992] have described how the zeroth law of thermodynamics (similar to the representation learned by IDeAL) is useful in redesigning a failed coffee maker. Device redesign task in their work involves four subtasks: formation of causal explanations of failures, discovery of new design constraints, formulation of internal behaviors that accommodate the modified constraints, and redesign of the device structure for realizing the modified internal behaviors. In particular, they describe how the zeroth law of thermodynamics is useful in the formation of causal explanations of why the device failed.

Consider, for instance, the design of a simple coffee maker whose structure is shown in Figure 9(a) in two states ((i)before and (ii) after coffee decoction is formed in `Container-2`). Its function of making coffee is to produce coffee decoction in `Container-2`, given hot water and coffee powder in `Container-1`. This design satisfies the function desired of the coffee maker, but its behavior is suboptimal. That is, there are two problems: (i) coffee decoction formed in `Container-2` is only lukewarm, and (ii) it does not stay warm in `Container-2`.

The first subtask in redesigning this coffee maker is to form a causal explanation for the failure. One way to accomplish this task is by instantiating an abstract principle, such as the zeroth law of thermodynamics, in the context of the current design and its environment.

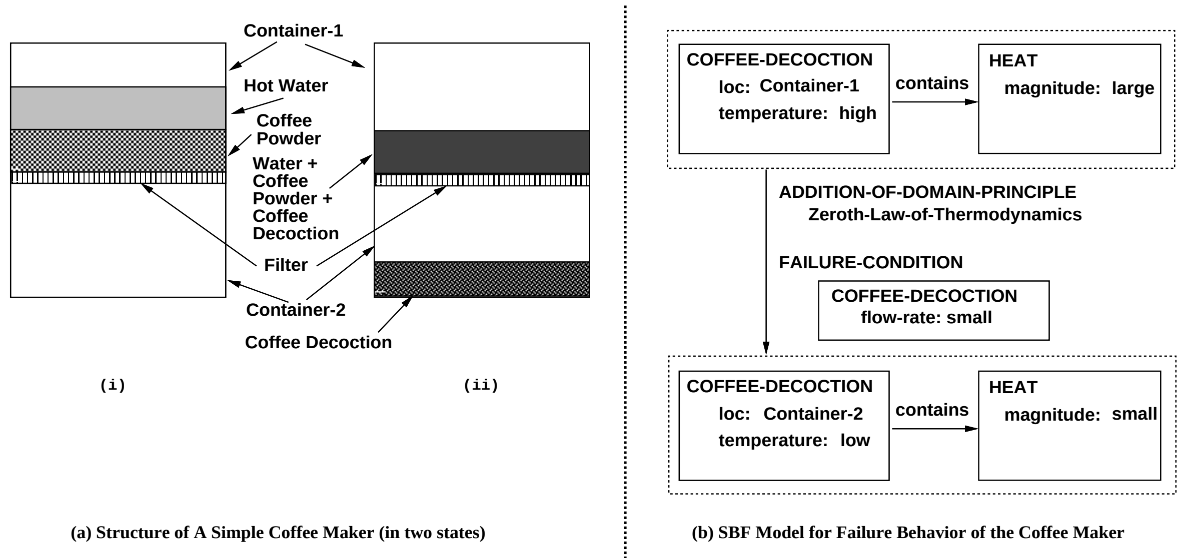


Figure 9: **Redesigning A Simple Coffee Maker**

The zeroth law of thermodynamics can be accessed by using an abstraction of the failure in the coffee maker (which is "loss of heat to the environment") as a probe into memory. Instantiating the law in the context of the design of coffee maker results in the SBF model for the failure behavior of coffee maker as shown in Figure 9(b). This helps in formulating new design constraints and in solving the subsequent subtasks in the redesign. The redesigned coffee maker may have a plunger in `Container-1` to pump the decoction into `Container-2` and a hot plate beneath `Container-2` to keep the decoction in `Container-2` warm (see [Prabhakar and Goel, 1992] for details).

(ii) Analogical Design: The hypothesized abstract models, that is, the models of the heat exchanger and the zeroth law of thermodynamics, can also be tested by using them for design by analogy. In cross-domain analogical transfer, the applicability of a source analog to a target problem is often due to sharing of some high-level (abstract) principles governing both the source and the target domains. In IDeAL, we are currently exploring the role of prototypical devices and physical principles in cross-domain transfer. Transfer in this scheme involves accessing the abstractions associated with a source analog and then instantiating them in the target domain rather than directly mapping source-specific substructures onto the target problem. We call this method *model-based analogy.* (See [Bhatta, 1992] for details). The models of physical principles and processes are more abstract than the models of prototypical devices. Hence, physical principles and processes are applicable to a wider class of design problems and thus they facilitate analogical transfer between distant domains.

6 Related Work

Learning Task: Our work is similar to JULIANA [Shinn, 1989] and ASIS [Roverso *et al.*, 1992] in learning abstractions from specific cases. But our learning task is different in the type of abstractions learned: JULIANA forms abstract cases and ASIS forms abstractions of structural models of specific situations while IDeAL discovers models of prototypical devices, physical principles, and processes.

Explanations in Learning: The proposal that learning from experience is facilitated by explanations of specific experiences dates at least as far back as Winston [1980]. Winston's model assumed knowledge of "what" is the concept being learned and relied on information concerning whether an example is a positive instance or negative instance of the concept. Our approach is similar to Winston's later models [1982; 1986] that show that learning can be done by analogically transferring causal links in the explanation of an example to the target concept.

Our approach is also similar to explanation-based learning (EBL) [DeJong and Mooney, 1986; Mitchell *et al.,* 1986] in using explanations (SBF models) to constrain the learning of "concepts." However, most EBL systems assume some knowledge of the target concept *a priori;* our model-based approach attempts to discover them.

Also, our model-based approach differs from EBL in the kind of explanations it uses. First, while the explanations in EBL are purely causal, the explanations in SBF models are functional in nature, i.e., they not only provide a causal account, they also show how causal processes result in the achievement of specific functions. Second, the explanations in EBL specify how an example is an instance of a target concept while SBF models are explanations of the functioning of devices. Besides, models also provide functional and structural decomposition knowledge for the devices that is useful in constraining the generalization process. Third, the explanations in EBL are constructed at run-time from domain specific rules whereas SBF models are formed by revising old models as part of the problem solving. Fourth, SBF models are grounded in a well-defined component-substance ontology.

Integration of Learning Methods: In addition, our work integrates the model-based approach with similarity-based methods for learning abstractions. In this respect, our work is similar to Pazzani's OCCAM [1991] which integrates similarity-based learning, EBL, and theory-driven learning (TDL) for learning of concepts.

Learning by Discovery: Our approach can be compared to work in scientific discovery such as BACON [Langley *et al.,* 1987], FAHRENHEIT [Zytkow, 1987], and ABACUS [Falkenhainer and Michalski, 1986]. These systems require a large amount of data because they use inductive approaches to discover regularities and form laws. In contrast, IDeAL is designed to incrementally discover physical principles using models to guide the discovery process. Hence, we expect IDeAL to require fewer examples for discovering useful principles. Most of the above systems use predesigned experiments to test their hypotheses. On the other hand, our approach takes a different stance on experimentation—it views *problem solving* using hypothesized "concepts" as testing the hypotheses. Thus hypothesis testing is not planned but rather is a consequence of solving design problems in the real world.

7 Conclusions

The models of specific devices (SBF models) provide both the content and the constraints for learning the models of physical principles (BF models) by incremental generalization over design experiences. In particular, we showed that the function of a device determines what to generalize from its SBF model, the SBF model suggests how far to generalize, and the typology of functions indicates what method to use for generalization.

Without the constraints from models (or similar knowledge) the method of induction for generalization can be potentially very complex. So the moral is that the existing machine learning techniques can be adapted for learning design knowledge, but they may need to be constrained by deep knowledge such as models in order to circumvent the complexity problem. Furthermore, most existing machine learning techniques have been developed in isolation of a performance task, but we believe that the acquisition of knowledge cannot be separated from the problem-solving tasks in which the learned knowledge might be used.

Finally, we believe that the issue of learning abstract models such as the models of physical principles and processes that facilitate cross-domain analogical design provides a great potential for machine learning in design because cross-domain analogies often play a crucial role in non-routine design.

Acknowledgements. This research has benefited from discussions with Eleni Stroulia. This work has been supported by research grants from NSF (grant IRI-92-10925), ONR (contract N00014-92-J-1234), Northern Telecom, ARPA, and equipment grants and donations from IBM, NCR, and Symbolics. We thank Kavi Mahesh and anonymous reviewers for their comments on an earlier draft of this paper.

References

[Alterman, 1988] R. Alterman. Adaptive Planning. *Cognitive Science*, 12:393–422, 1988.

[Ashley and Rissland, 1988] K. Ashley and E. Rissland. A Case-Based Approach to Modeling Legal Expertise. *IEEE Expert*, 3(3):70–77, 1988.

[Bhatta and Goel, 1992] S. Bhatta and A. Goel. Use of Mental Models for Constraining Index Learning in Experience-Based Design. In *Proceedings of the AAAI workshop on Constraining Learning with Prior Knowledge*, pages 1–10, San Jose, CA, July 1992.

[Bhatta and Goel, 1993] S. Bhatta and A. Goel. Model-Based Learning of Structural Indices to Design Cases. In *Proceedings of the IJCAI workshop on "Reuse of Designs: An Interdisciplinary Cognitive Approach"*, pages A1–A13, Chambery, Savoie, France, August 1993.

[Bhatta, 1992] S. Bhatta. A Model-Based Approach to Analogical Reasoning and Learning in Design. Technical Report GIT-CC-92/60, Georgia Institute of Technology, College of Computing, Atlanta, GA, November 1992. Ph.D. Thesis Proposal.

[Bylander and Chandrasekaran, 1985] T. Bylander and B. Chandrasekaran. Understanding Behavior Using Consolidation. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 450–454, 1985.

[DeJong and Mooney, 1986] G. DeJong and R. Mooney. Explanation-Based Learning: An Alternative View. *Machine Learning*, 1(2):145–176, 1986.

[Falkenhainer and Michalski, 1986] B. Falkenhainer and R. Michalski. Integrating Quantitative and Qualitative Discovery: The ABACUS System. *Machine Learning*, 1:367–401, 1986.

[Fermi, 1937] E. Fermi. *Thermodynamics*. Prentice-Hall, New York, NY, 1937.

[Gentner, 1983] D. Gentner. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7:155–170, 1983.

[Gero, 1990] J. Gero. Design Prototypes: A Knowledge Representation Schema for Design. *AI Magazine*, 11(4):26–36, 1990.

[Goel, 1989] A. Goel. *Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving.* Ph.D. thesis, The Ohio State University, Department of Computer and Information Science, Columbus, Ohio, 1989.

[Goel, 1991a] A. Goel. A Model-Based Approach to Case Adaptation. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 143–148, Chicago, August 1991.

[Goel, 1991b] A. Goel. Model Revision: A Theory of Incremental Model Learning. In *Proceedings of the Eighth International Conference on Machine Learning*, pages 605–609, Chicago, June 1991.

[Goel, 1992] A. Goel. Representation of Design Functions in Experience-Based Design. In D. Brown, M. Waldron, and H. Yoshikawa, editors, *Intelligent Computer Aided Design*, pages 283–308. North-Holland, Amsterdam, Netherlands, 1992.

[Hammond, 1989] K. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task.* Academic Press, Boston, MA, 1989.

[Kerr and Duffy, 1992] S. Kerr and A. Duffy. Dynamic Memory by Automating Rationalization of Past Designs. In *Proceedings of the AID'92 Workshop on Machine Learning in Design*, pages 28–30, Pittsburgh, PA, USA, June 1992.

[Klahr and Dunbar, 1988] D. Klahr and K. Dunbar. Dual Space Search During Scientific Reasoning. *Cognitive Science*, 12:1–48, 1988.

[Kolodner and Simpson, 1989] J. L. Kolodner and R. L. Simpson. The MEDIATOR: Analysis of An Early Case-Based Problem Solver. *Cognitive Science*, 13(4):507–549, 1989.

[Langley *et al.*, 1987] P. Langley, H. Simon, G. Bradshaw, and J. Zytkow. *Scientific Discovery: An Account of the Creative Processes.* MIT Press, Boston, MA, 1987.

[Mitchell *et al.*, 1986] T. M. Mitchell, R. Keller, and S. Kedar-Cabelli. Explanation-Based Generalization: A Unifying View. *Machine Learning*, 1(1):47–80, 1986.

[Pazzani, 1991] M. Pazzani. Learning to Predict and Explain: An Integration of Similarity-Based, Theory-Driven, and Explanation-Based Learning. *The Journal of the Learning Sciences*, 1(2):153–199, 1991.

[Prabhakar and Goel, 1992] S. Prabhakar and A. Goel. Performance-Driven Creativity in Design: Constraint Discovery, Model Revision, and Case Composition. In J. Gero and F. Sudweeks, editors, *Proceedings of the Second International Round-Table Conference on Computational Models of Creative Design*, pages 101–128, Heron Island, Australia, December 1992.

[Riesbeck and Schank, 1989] C. Riesbeck and R. Schank. *Inside Case-Based Reasoning.* Erlbaum, Hillsdale, NJ, 1989.

[Roverso *et al.*, 1992] D. Roverso, P. Edwards, and D. Sleeman. Machine Discovery by Model Driven Analogy. In J. M. Zytkow, editor, *Proceedings of the ML-92 workshop on Machine Discovery*, pages 87–97, Aberdeen, Scotland, July 1992.

[Sembugamoorthy and Chandrasekaran, 1986] V. Sembugamoorthy and B. Chandrasekaran. Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems. In J.Kolodner and C.Riesbeck, editors, *Experience, Memory and Reasoning*, pages 47–73. Lawrence Erlbaum, Hillsdale, NJ, 1986.

[Shinn, 1989] H. S. Shinn. *A Unified Approach to Analogical Reasoning*. Ph.D. thesis, Georgia Institute of Technology, School of Information and Computer Science, Atlanta, GA, 1989.

[Winston, 1980] P. Winston. Learning and Reasoning by Analogy. *Communications of the ACM*, 23(12):689–703, 1980.

[Winston, 1982] P. Winston. Learning New Principles from Precedents and Exercises. *Artificial Intelligence*, 19(3):321–350, 1982.

[Winston, 1986] P. Winston. Learning by Augmenting Rules and Accumulating Censors. In R.Michalski, J.Carbonell, and T.Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach, Vol. II*, pages 45–61. Morgan Kaufmann, Los Altos, CA, 1986.

[Zytkow, 1987] J. Zytkow. Combining Many Searches in the FAHRENHEIT Discovery System. In *Proceedings of the Fourth International Workshop on Machine Learning*, pages 281–287, Los Altos, CA, 1987. Morgan Kaufmann.