A KNOWLEDGE REPRESENTATION SYSTEM BASED ON DYNAMICALLY INTERLACED HIERARCHIES: BASIC IDEAS AND EXAMPLES

by

M. R. Hieb
R. S. Michalski

Reports of the Machine Learning and Inference Laboratory, MLI 93-5, School of Information Technology and Engineering, George Mason University, May 1993.

A KNOWLEDGE REPRESENTATION SYSTEM BASED ON DYNAMICALLY INTERLACED HIERARCHIES: Basic Ideas and Examples

Michael R. Hieb & Ryszard S. Michalski

MLI93-5

P93-13

A Knowledge Representation System Based on Dynamically Interlaced Hierarchies: Basic Ideas and Examples

M. R. Hieb and R. S. Michalski

Center for Artificial Intelligence
George Mason University
Fairfax, Virginia 22030

MLI 93-5

August 1993

A Knowledge Representation System Based on Dynamically Interlaced Hierarchies: Basic Ideas and Examples

Abstract

A new approach to knowledge representation is presented that is specifically designed for supporting multistrategy learning and multitype inference. The approach, called *Dynamically Interlaced Hierarchies* (DIH), stems from research on cognitive modeling of human plausible inference and semantic memory organization. In DIH, knowledge is divided into two parts: a "static" part that represents stable knowledge consisting of concepts organized into hierarchies of different kinds, such as type, part and *precedence* hierarchies; and a "dynamic" part that represents knowledge that changes relatively frequently, consisting of *traces* that link concepts from different hierarchies. Parametric knowledge is represented as numeric quantities characterizing structural elements of knowledge, such as various measures of uncertainty. It is shown how the conceptual framework presented can represent diverse and complex forms of knowledge and can support basic knowledge transformations. These transformations are knowledge generation transmutations, as defined in the Inferential Theory of Learning (ITL). Basic knowledge transmutations, such as generalization/ specialization, abstraction/concretion and similization are illustrated in DIH. Inference is modeled as small perturbations of existing knowledge.

Key Words: multistrategy learning, inferential theory of learning, knowledge transmutation, generalization, abstraction, similization.

Acknowledgments

The authors wish to thank Tomasz Dybala, David Hille, Zenon Kulpa and Brad Utz for their comments and critical review of this paper.

This research was conducted in the Center for Artificial Intelligence at George Mason University. The Center's research is supported in part by the National Science Foundation under grant No. IRI-9020266, in part by the Advanced Research Projects Agency under the grant No. N00014-91-J-1854, administered by the Office of Naval Research and the grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research, and in part by the Office of Naval Research under grant No. N00014-91-J-1351.

1. Introduction

This paper presents underlying ideas on a knowledge representation designed for inference and learning. The motivation for research on Dynamically Interlaced Hierarchies (DIH) is to provide an adequate knowledge representation for facilitating multitype inference in multistrategy learning systems. A requirement for the development of multistrategy learning systems is a flexible, expressive and easily modifiable system of representing knowledge. In particular, multitype inference requires a conceptually simple representation that can encompass many different forms of knowledge - facts, rules, dependencies, etc. These two related ideas, multitype patterns of inference and multistrategy learning, are related in the Inferential Theory of Learning (Michalski, 1992).

The initial idea for DIH stems from the core theory of human plausible reasoning (Collins & Michalski, 1989; Boehm-Davis, Dontas & Michalski, 1990). That theory presents a formal representation of various plausible inference patterns observed in human reasoning. From the core theory and the rudiments of the knowledge representation that was developed for plausible reasoning, we design a more detailed system in order to accomplish the aims stated above.

One goal of multistrategy learning systems is reasoning in a knowledge rich environment. In order to perform this type of reasoning there must be a large, critical mass of background knowledge with a representation efficient enough to deal with this amounts of knowledge. We envision organizing this knowledge in generalization hierarchies. Research in psychology postulates that this is a plausible structure for human semantic memory (Collins & Quillian, 1972).

2. Underlying Ideas

The theory of plausible reasoning postulates that there are recurring patterns of human plausible inference. Knowledge, stored in "dynamic hierarchies", is modified in certain characteristic ways during the process of human inference. Recent studies show that a variety of operations upon these hierarchies are used in human reasoning (Boehm-Davis, Dontas and Michalski, 1990). For a complete description of knowledge transmutations see (Michalski, 1993). These patterns are represented in DIH in section III.

A basic assumption of this method is that knowledge is structurally organized in hierarchies. These hierarchies are dynamic in that they are always being updated, modified and expanded as new experience (input) is integrated into the hierarchies (background knowledge).

2.1 Major Components in DIH Representation

The DIH approach partitions knowledge into a "static" part and "dynamic" part. The static part represents knowledge that is relatively stable (such as established hierarchies of concepts), and a "dynamic" part that represents knowledge that changes relatively frequently (such as statements representing new observations or results of reasoning). The static part is organized into *type* hierarchies (TH), *part* hierarchies (PH) and *precedence* hierarchies. Precedence hierarchies include several subclasses, specifically, *measure* hierarchies (MH), *quantification* hierarchies (QH) and *schema* hierarchies (SH). The dynamic part consists of *traces* that represent knowledge involving concepts from different hierarchies. Each trace links nodes of two or more hierarchies and is assigned a degree of belief.

These hierarchies are composed of nodes representing abstract or physical entities, and links representing certain basic relationships among the entities, such as "type-of", "part-of" or "precedes". In the "pure" form, these hierarchies are single parent, that is, no node can have more than one parent. The root node is assigned the name of the class of entities that are organized into the hierarchy from a given viewpoint.

A type (or generalization) hierarchy organizes concepts in a given class according to the "type-of" relation (also called a "generalization" or "kind-of" relation). For example, different types of "animals" can be organized into a "type" hierarchy.

A part hierarchy organizes entities according to a "part-of" relationship. For example, the world, viewed as a system of continents, geographical regions, countries, etc., can be organized into a part hierarchy. While properties of a parent node in the type hierarchy are inherited by children nodes, this does not necessarily hold for a part hierarchy. There are different part relationships which have different generalization characteristics. Part relations include part-component, part-member, part-location and part-substance as well as several others (Winston, Chaffin and Herrmann, 1987). In this initial report on DIH only part-location hierarchies are utilized in inference, since this part relation has similar generalization properties to a type hierarchy. Other part relations, such as part-component, do not have satisfactory generalization properties.

To represent relationships among elements of ordered or partially ordered sets, a class of *precedence* hierarchies is introduced. Hierarchies in this class represent hierarchical structures of concepts ordered according to some precedence relation, such as "A precedes B", "A is greater than B", "A has higher rank than B", etc.

There are several subclasses of precedence hierarchies. One subclass is a *measure* hierarchy, in which leafs stand for values of some physical measurement, for example, weight, length, width, etc., and the parent nodes are symbolic labels characterizing ranges of these values, such as "low", "medium", "high", etc. Figure 1 shows a measure hierarchy of possible values of people's height. Dotted lines indicate a continuity of values between nodes. Arrows indicate the precedence order of the nodes. Another subclass hierarchy is a *belief* hierarchy, in which nodes represent degrees of an agent's beliefs in some knowledge represented by a trace.

Other subclasses of precedence hierarchies include a *rank* hierarchy and a *quantification* hierarchy. A rank hierarchy consists of values representing the "rank" of an entity in some structure, e.g., an administrative hierarchy or military hierarchy. A quantification hierarchy consists of nodes that represent different quantifiers for a set (An example is shown in Figure 3). A quantification hierarchy that is frequently used in commonsense reasoning includes such nodes as "one", "some" (corresponding to the existential quantifier), "most", and "all" (corresponding to the universal quantifier).

Each hierarchy has a heading that specifies its kind (TH, PH, MH, QH or SH) and the underlying concept (or viewpoint) used for the creation of the hierarchy. In addition, the type and part hierarchies also have a *top* node that in the type hierarchies stands for the class of all entities in the hierarchical structure, and in the part hierarchies for the complete object.

Schema hierarchies (or schema) are structures that indicate which hierarchies are connected in order to express multi-argument concepts or relationships. For example, the schema hierarchy for the concept of "physical-object" can be <shape, size>. This states that an attribute "shape" applies to any object that is a "physical-object" (a node in the "physical-object" hierarchy), and produces a shape value, which is a node in the "shape" hierarchy. The schema hierarchy for the concept of "giving" may be <giver, receiver, object, time> that states that this concept involves an agent that gives, an agent that receives, an object that is being given, and the time when the "giving" occurs. The agents, object and time are elements of their respective hierarchies.

2.2 Structural vs. Parametric Knowledge

DIH also makes a distinction between structural and parametric knowledge. Structural knowledge is assumed to be the *basis* of human reasoning. In DIH, structural knowledge is represented as single parent hierarchies and the traces between them. Hierarchies are stable groupings formed on the basis of experience and not often changed. This is not the case for the traces between the hierarchies, which are easily formed and changed and are transient in

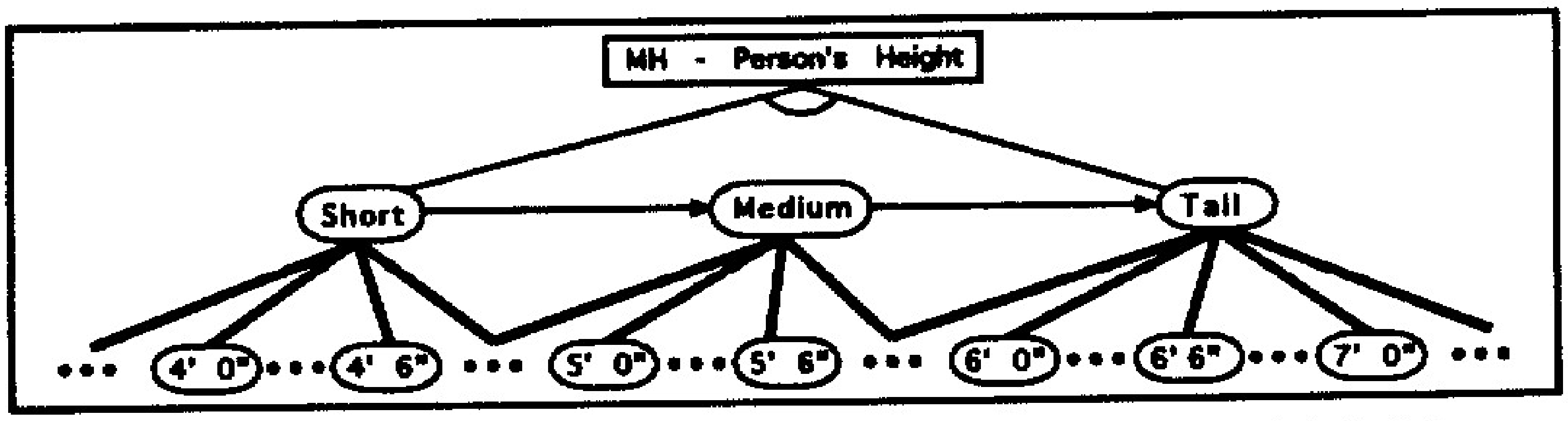


Figure 1: A measure hierarchy of values characterizing people's height.

nature. It is postulated that when people first reason about a problem or question, they first look to these structural elements which comprise their knowledge. This knowledge is easily and efficiently accessed and various inferences (primarily deductive) can be utilized with this structural knowledge to yield satisfactory results.

Parametric knowledge, on the other hand, includes probabilistic knowledge about the structural elements. In DIH, this knowledge is represented via precedence hierarchies of *merit parameters*. that are applied selectively to different types of knowledge structures. People only resort to using this type of knowledge after they have attempted to reason using the structural elements. People resist using uncertain knowledge and there is no normative model that is widely accepted for this type of probabilistic or approximate reasoning.

DIH also makes a distinction between structural and parametric knowledge. The structural knowledge is represented by hierarchies and traces that link nodes of different hierarchies. Parametric knowledge consists of numeric quantities characterizing structural elements of knowledge. The basic merit parameter is a belief measure that characterizes the "truth" relationship of a given component of knowledge representation (a trace), as estimated by the reasoning agent. Other merit parameters include the *forward* and *backward* strength of a dependency, *frequency*, *dominance*, etc. (Collins and Michalski, 1989; Michalski, 1993). In this paper, we will consider only one merit parameter, namely, the belief measure.

The theory of human plausible reasoning (Collins and Michalski, 1989) postulates that people rely primarily on the structural knowledge, and resort to parametric knowledge when the "structural" reasoning does not produce a unique result. They resist performing uncertain inferences based on only parametric knowledge, and they are not good at assigning a degree of certainty to a statement based only on the combination of the certainties of its constituents, without taking into consideration the meaning of the whole sentence. A reason for this may be that there does not exist a normative model for reasoning under uncertainty that is independent of the structural aspects of knowledge, i.e., its meaning. Plausible reasoning about a problem or question typically involves both structural and parametric knowledge components.

Nodes of a hierarchy are elementary units of the DIH representation. Each node represents some real or abstract entity—a concept, an object, a process, etc. A given entity can be a node in multiple hierarchies, where each hierarchy structures a set of entities from a different viewpoint. The relevant viewpoint is determined by the context of the discourse.

Nodes are the basic unit of representation in DIH. Each Node represents a *single* concept or idea, but a single node can be represented in multiple hierarchies. In this case the node is used in different contexts (is contained in different hierarchies). Because of this the hierarchies can be intricately interlaced. One can view the representation space as three-dimensional, with hierarchies having two dimensions, connected together by shared nodes, to form the third dimension. An example of a node used in multiple hierarchies is the concept of "turbine" as shown in Figure 2. This node can be simaltaneously in a part-component hierarchy of "power plant" and in a type hierarchy of "rotary engine" .

In DIH a node represents either a class or an individual, where we view an 'subnode' as a more specific set than the it's parent.

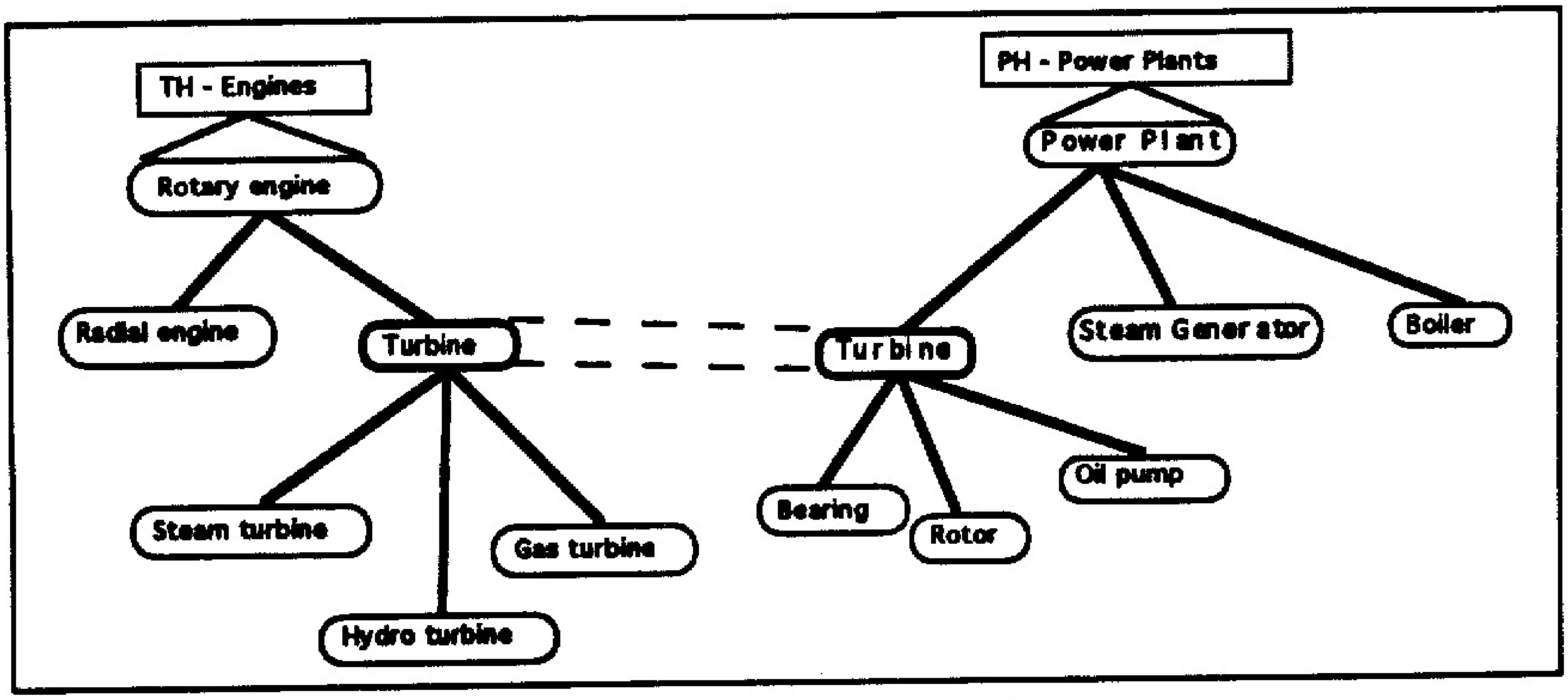


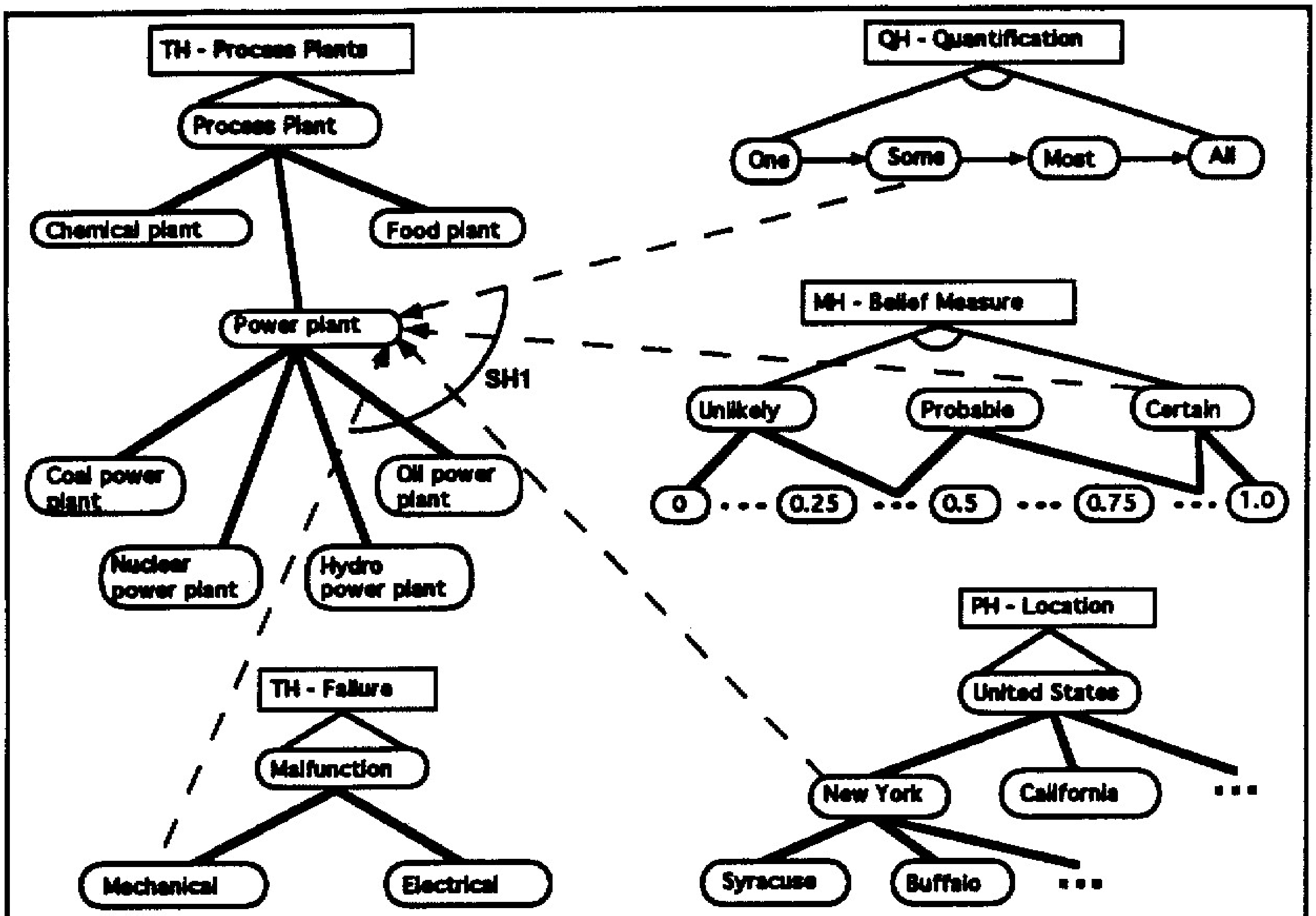
Figure 2: Interlacing Hierarchies

As mentioned earlier, the basic structures in the DIH representation are hierarchies, nodes, traces and schema. Our research on DIH demonstrates that these structures provide a very natural environment for performing basic types of inference on statements. The subsequent sections show how these inferences are performed using the DIH representation.

2.3 DIH Traces

A trace is a statement representing a path between nodes in different hierarchies. The simplest traces consist of links that point from a node in one hierarcy to another node in a second hierarchy via a descriptor.

Figure 3 presents a statement – "It is certain that some power plants in New York have mechanical failures" – as a trace connecting nodes of five hierarchies: "Process plants" and



The trace representing the sentence consists of nodes linked by dotted lines. The arrows in the trace indicate the argument (reference set) that is being described by the sentence. The interpretation of the trace is given by schema hierarchy SH1 in Figure 4.

Figure 3: A DIH trace representing the sentence "It is certain that some power plants in New York have mechanical failures."

"Failure", both type hierarchies; "Quantification", the quantification hierarchy; "Location", a part hierarchy; and "Belief measure" a measure hierarchy.

The interpretation of the trace is done on the basis of the schema hierarchy shown in Figure 4. The schema defines the universe of sentences that can be generated using concepts of these hierarchies, ordered according to the schema.

The convention for the direction of arrows in a trace is that they point from the nodes

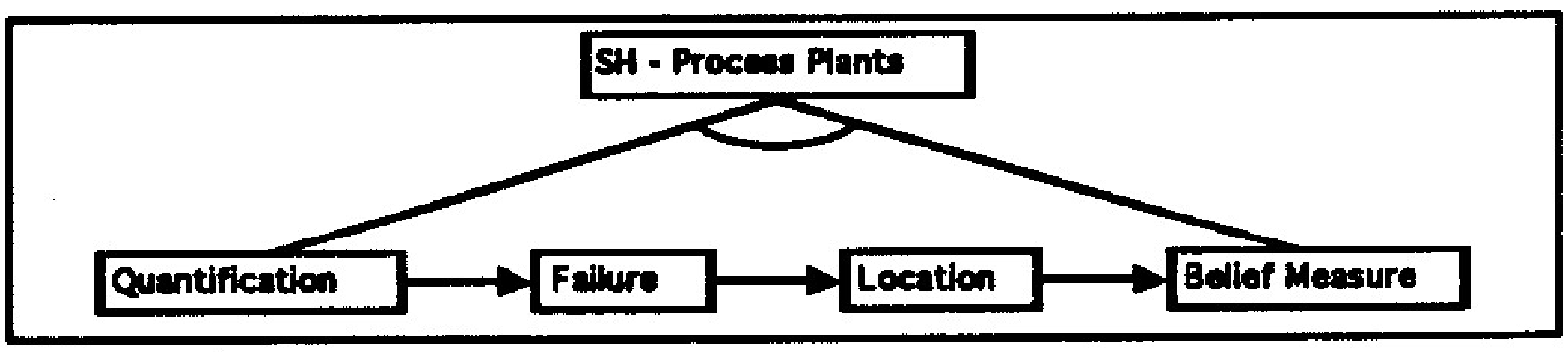


Figure 4: Schema hierarchy SH1.

denoting descriptive concepts (called *reference* nodes) to the *argument node* that stands for the set (or individual) being described or refered to. In this example, the set being described is "Power plant" in the hierarchy of Process Plants, thus the node representing it is the argument node. Other nodes (reference nodes) linked by the trace represent descriptive concepts for the argument node. The belief measure takes values from a belief hierarchy, and refers to the entire trace rather than a single node, which is indicated by the schema.

Using the formalism of the annotated predicate logic (Michalski, 1983), this trace can be interpreted as: "(Some)x, [type(x) = Power plant) & [location(x) = New York] & [failure(x) = mechanical]: Belief = 1.0." This statement is a quantified conjunction of several *elementary* statements. An elementary statement expresses one property of the reference node (set), for example, "Location(Power plant) = New York."

In a formal expression of an elementary statement, the reference set ("Power plant") is called an *argument*, the predicate ("Location") is called a *descriptor*, and the value of the descriptor ("New York") is called the *referent*. Thus, an elementary statement is formally expressed in the form "descriptor(argument) = referent". Predefined applicability conditions determine which descriptors can apply to argument nodes.

In Figure 3, the square boxes contain the heading of the hierarchy. The concept specified in the heading is the general descriptor for the hierarchy. The nodes in the hierarchy are possible values of this descriptor.

The schema hierarchy, SH1, in Figure 4 is used for the interpretation of the trace represented in Figure 3. The heading indicates the type of hierarchy (SH: Schema Hierarchy) and the reference set of the trace. Since the schema hierarchy is a precedence hierarchy, a valid interpretation of the schema requires each of the descriptors in order. Thus the first element of the trace must be from the quantification hierarchy, the second from the failure hierarchy, the third from the location hierarchy and the last from the hierarchy of belief measures. This schema hierarchy is also utilized for examples in Section 4.

Adding knowledge to the DIH representation is done by creating hierarchies and specifying traces that express statements involving nodes of different hierarchies. To allow proper interpretation of a trace, the schema is also specified by indicating relevant descriptors and their order.

2.4 Other Constructs in DIH

Besides the basic constructs of DIH, we have additional elements of complex traces and merit parameters. Schema are ordered sets of traces. The merit parameters are fundamentally different in form and function.

A complex trace is able to represent more complicated forms of knowledge, involving predicates and various types of logical operators. Multiple links are used to form a statement in DIH for a n-ary predicate. Complex traces may use traces as referents (forming nested traces). Such a complex trace might be a generalized trace, where the trace is perhaps an inference step that is verified by experience, or a higher order concept, such as a rule. It may represent an implication, dependency, or any other type of casual association between traces.

An example of such a complex trace could be "A component failure of a power plant causes a maintenance outage". Here, cause is a 2 place predicate as in causes (input, effect). We use the arguments of the predicate as the descriptor ("input(causes) = [failure(power plant) = mechanical failure]" and "effect(causes) = maintenance outage").

Merit Parameters represent numerical or qualitative properties of the association designated by a trace or schema. These include:

- Certainty of Belief (applies to all traces and schema)
- Conditional likelihood (applies to schema)
- Typicality (applies only to nodes)
- Frequency (applies only to traces)
- Dominance(applies only to nodes)
- Multiplicity (applies only to traces)
- Similarity (applies only to traces)

Merit parameters work to guide inductive inference in DIH. Most will be calculated when needed, based upon the background knowledge. It is beyond the scope of this report to further describe the merit parameters. The interested reader is referred to the "Logic of Plausible Reasoning" (Collins & Michalski, 1989).

2.5 Learning in DIH

Learning in DIH can be represented as either structural or parametric changes in the representation. If viewed as structural, learning can be either creating entirely new knowledge or modifying existing knowledge. The creation of new knowledge in DIH is performed by creating hierarchies, adding nodes to hierarchies or creating traces between nodes. The modification of knowledge in DIH is changing the placement of nodes in hierarchies, or modifying traces or schema. If viewed as parametric, or dealing with the certainty of knowledge, learning includes strengthening or weakening merit parameters.

3. Implementing Multiple Inference Types in DIH

The core theory of plausible reasoning introduced in (Collins & Michalski, 1989) gives four knowledge transmutation operators (also called transforms) – generalization, specialization, similization and dissimilization. The Inferential Theory of Learning (Michalski, 1993) specifies several additional operators, of which abstraction and concretion are incorporated into DIH. (In (Collins and Michalski, 1989), the abstraction and concretion transmutations were called referent generalization and referent specialization, respectively.)

Generalization (specialization) transmutations extend (contract) the reference set. They are done either by argument generalization (specialization) or by quantification generalization (specialization). Argument generalization is accomplished by moving above the node representing the reference set in a type hierarchy. Quantification generalization is accomplished by moving up the quantification hierarchy.

Abstraction (concretion) transmutations decrease (increase) the amount of information about the reference set. A way to accomplish such a transmutation is by moving above the node in the type or part hierarchy that corresponds to a value of some descriptor in the sentence represented by the trace.

Similization (dissimilization) transmutation is done by replacing a node corresponding to the reference set (argument) or a descriptor value (referent) by a node at the same level of hierarchy, which corresponds to a similar (dissimilar) concept within the context of the given hierarchy. In the case of dissimilization, the resulting trace is linked with a negation node, because the generated inference is a negation of the original sentence (Michalski, 1993).

These transmutations can be given a simple conceptual interpretation, if one assumes that nodes at each level of hierarchy are ordered by the relation of similarity, that is, nodes that correspond to similar concepts (in the context of the given hierarchy) are located near each other, and nodes that correspond to dissimilar concepts are placed far away from each other. Such an arrangement is natural for precedence hierarchies. In sum, similization and dissimilization transmutations are performed by sideways node movements, while generalization (specialization) and abstraction (concretion) are performed by upward (downward) node movements.

Table 1 gives the basic knowledge transmutations possible in DIH and the kind of hierarchy that they can operate on. Each of the transmutations is characterized according to inference type. While Dissimilization operators are possible, they can be thought of as an "inverse" Similization operator, and are not shown. While all of the other transmutations considered here move up and down in the hierarchies, the Similization operator moves sideways among the same level in the hierarchy and thus there is no clear way to distinguish Similization from Dissimilization. The various kinds of part hierarchies are not shown, but are distinguished in

DIH. Additional constraints are necessary in some kinds of part hierarchies to maintain the validity of the transmutation.

Transmutation	Symbol	Relevant Hierarchies	Inference Type
Argument Generalization	AGen	Type, Part	Deductive
Argument Specialization	ASpec	Type, Part	Inductive
Quantification Generalization	QGen	Quantification	Inductive
Quantification Specialization	QSpec	Quantification	Deductive
Abstraction	Abs	Type, Part, Precedence	Deductive
Concretion	Con	Type, Part, Precedence	Inductive
Argument Similization	ASim	Type, Part	Analogical
Argument Dissimilization	ADis	Type, Part	Analogical
Referent Similization	RSim	Type, Part, Precedence	Analogical
Referent Dissimilization	RDis	Type, Part, Precedence	Analogical

Table 1: Basic DIH knowledge transmutations

Figure 5 shows diagrametrically how the knowledge transmutations 'modify' a trace. The arrows mean that the trace is moving to a new argument or referent node, except for the Quantification transmutations. The Quantification transmutations operate over the entire trace, rather than on a single node. The thickness of the lines indicates the certainty of the inference. It is assumed that the Argument Generalization/Specialization and Abstraction/Concretion operations only move the trace one level up or one level down. In contrast, the Quantification operations typically range from the extremes of the hierarchy (from "every" to "one"). The Similization operator operates on the same level, and is unrestricted except that it must operate on nodes at the same level of the hierarchy.

A dotted line represents a link in a trace. An arrow means that the trace is moving to a new node in the indicated direction by performing the indicated transmutation. The merit parameters transmutations operate over the entire trace, rather than on a single node, as can the transformations involving the quantification hierarchy.

One form of generalization transmutation moves a node in the quantification hierarchy upward, another form moves a node (argument) in the type hierarchy upward. The "+" indicates a strengthening of a merit parameter, or the movement of the link to a node that is "higher" in the particular merit parameter measure hierarchy. The "–" indicates a weakening of the merit parameter, or the movement of the link down in the hierarchy.

Moving a node in a trace in a manner that corresponds to a deductive inference (Table 1) produces a new trace (statement) with the same truth status as the original trace. In the case of node movement that corresponds to inductive or analogical inference, the smaller the node movement ("perturbation"), the more plausible the resulting inference.

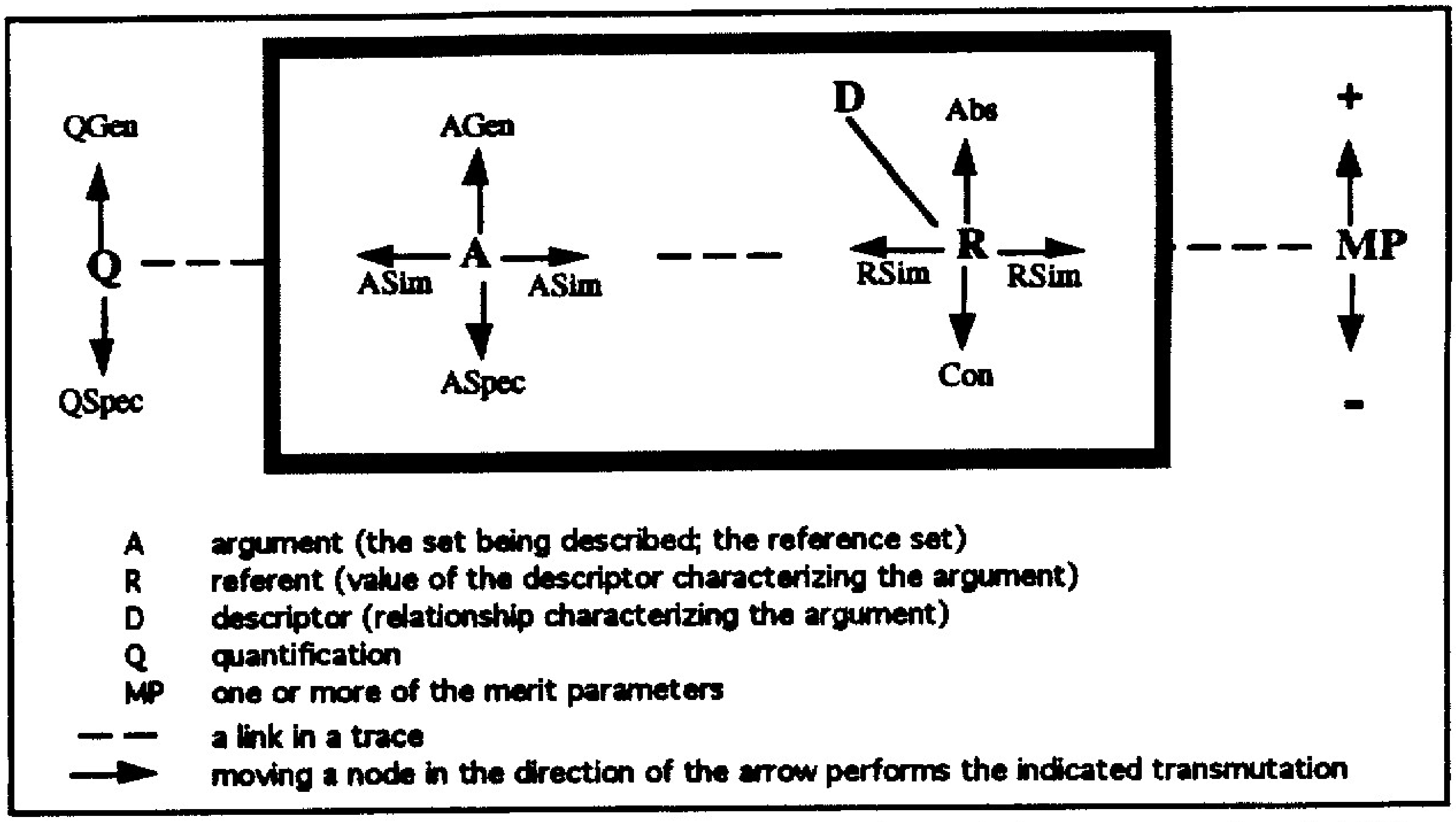


Figure 5: A schematic illustration of different types of knowledge transmutations in DIH

The Argument Generalization transmutation represents a deductive inference. The abstraction operation is also deductive. In contrast, Argument Specialization, Quantification generalization and Concretion are inductive, because they produce traces (statements) that logically entail the original traces (statements).

The above transmutations can be usually done in a number of different ways, by moving to different alternative nodes. The plausibility of the generated statements depends on additional merit parameters.

These transmutations are illustrated through a series of examples concerning power plant operation. Figure 6 illustrates the transmutations diagrametrically for the simple trace (input) "location(power plant) = New York". The resulting statements (output) show the results of

the given transmutation assuming that there are no merit parameters that assist in the specialization and that the similization operator finds a single "most similar" node using the stated context.

Figure 6 assumes the following background knowledge. Both Power plants and Chemical plants are types of Process plants, and Nuclear, Oil and Hydro plants are types Power plants. New York is the designation for the location bounded by the state of the same name, as is

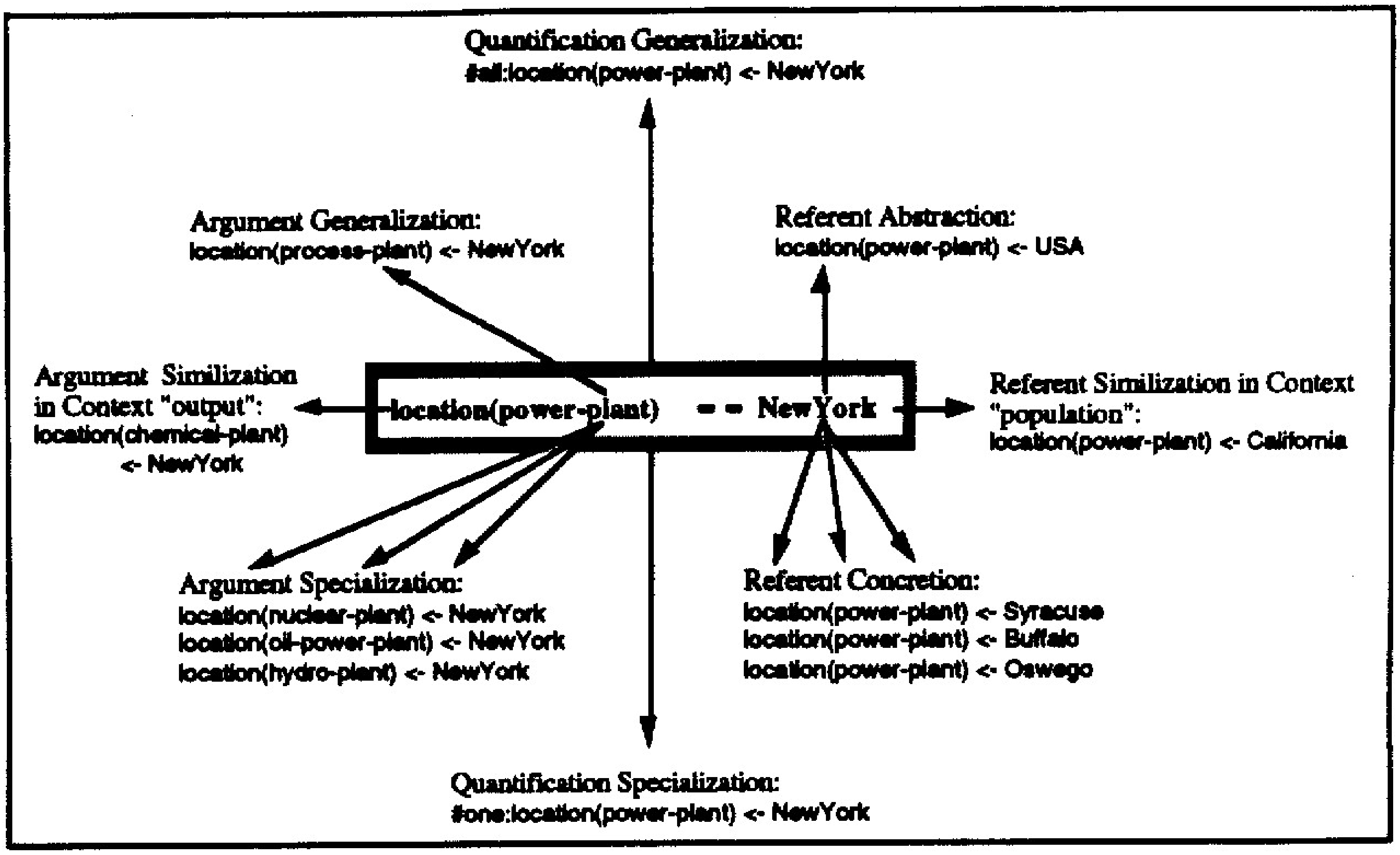


Figure 6: Diagram of example statement transmutations

California. New York and California are both parts of the USA, just as the regions Syracuse, Buffalo and Oswego are parts of New York. The designation "#x:" indicates that the quantity x is to be quantified over the entire trace. The absence of a designation indicates a default quantification corresponding to "there exists".

The deductive nature of the Argument Generalization transmutation is shown in the following examples. There is only one node (the argument node's parent) to generalize up to. Thus, given that there exists a plant that is of type Power plant, and has the location of New York, we can deductively assert that there is a plant of type Process plant that has the location New York. The Abstraction process has the same deductive nature.

Argument Specialization, like Concretion, must have some additional criteria, learning goal or bias to perform the selection of the output node. Unless there is only one child of the input node, there is a selection process, which must be assisted by the parametric aspects of DIH, the merit parameters. Such an output node may be more typical or dominant in respect to the input node.

Table 2 gives a summary of 11 transmutations performed on a single input statement with the given background knowledge. The actual scenario is that two nuclear power plants in New York are shut down due to mechanical problems with their turbine generators. Specifically, their Westinghouse turbines have cracked blades due to a materials problem with the rotor

Input

Some power plants in New York have mechanical failures.

Background Knowledge

Process Plant Type Hierarchy	Quantity Hierarchy
Failure Type Hierarchy	Location Part of Hierarchy

"Turbines are part of Airplanes"
"Bad design causes power plant mechanical failures."
"Power plant mechanical failures cause expensive maintenance outages."

Output

1.	*QGen:*	*All* power plants in New York have mechanical failures.
2.	*QSpec:*	*A* power plant in New York has a mechanical failure.
3.	*AGen:*	Some *process plants* in New York have mechanical failures.
4.	*ASpec:*	Some *nuclear* power plants in New York have mechanical failures.
5.	*Abs:*	Some power plants in *the United States* have mechanical failures.
6.	*Abs:*	Some power plants in New York have a *failures*.
7.	*Con:*	Some power plants in *Syracuse* have a mechanical failure.
8.	*Con:*	Some power plants in New York have *a component defect.*
9.	*ASim:*	Some *chemical plants* in New York have mechanical failures.
10.	*RSim::*	Some power plants in *California* have mechanical failures.
11.	*RSim:*	Some power plants in New York have *system failures*.
12.	*Anl:*	Some airplanes have mechanical failures.
13.	*Pred:*	Some New York power plants have expensive maintenance outages.
14.	*Abd:*	Some power plants have a bad design.

Table 2 - Major transmutations of the statement:
"Some power plants in New York have mechanical failures."

fabrication. Additional schema based inferences are represented by analogy (Anl), prediction (Pred) and abduction (Abd).

Each of the eight basic transmutations are illustrated in DIH in Figures 8 through 14, in conjunction with a formalization of the particular transmutation. Figure 7 shows a legend used for interpretation of the following figures. The input statement can be thought of as a conjunction of three traces, "quantity(power-plant) = some & location(power-plant) = New York & failure(power-plant) = mechanical failure". Thus, there are two referents, excluding the special case of quantity, in the trace.

To characterize the transmutations, a convention is used where $N_{x,y}$ in H indicates node N at the x level of a hierarchy H and at the y rank. The convention is only relative and does not correspond to any ordering of the hierarchy.

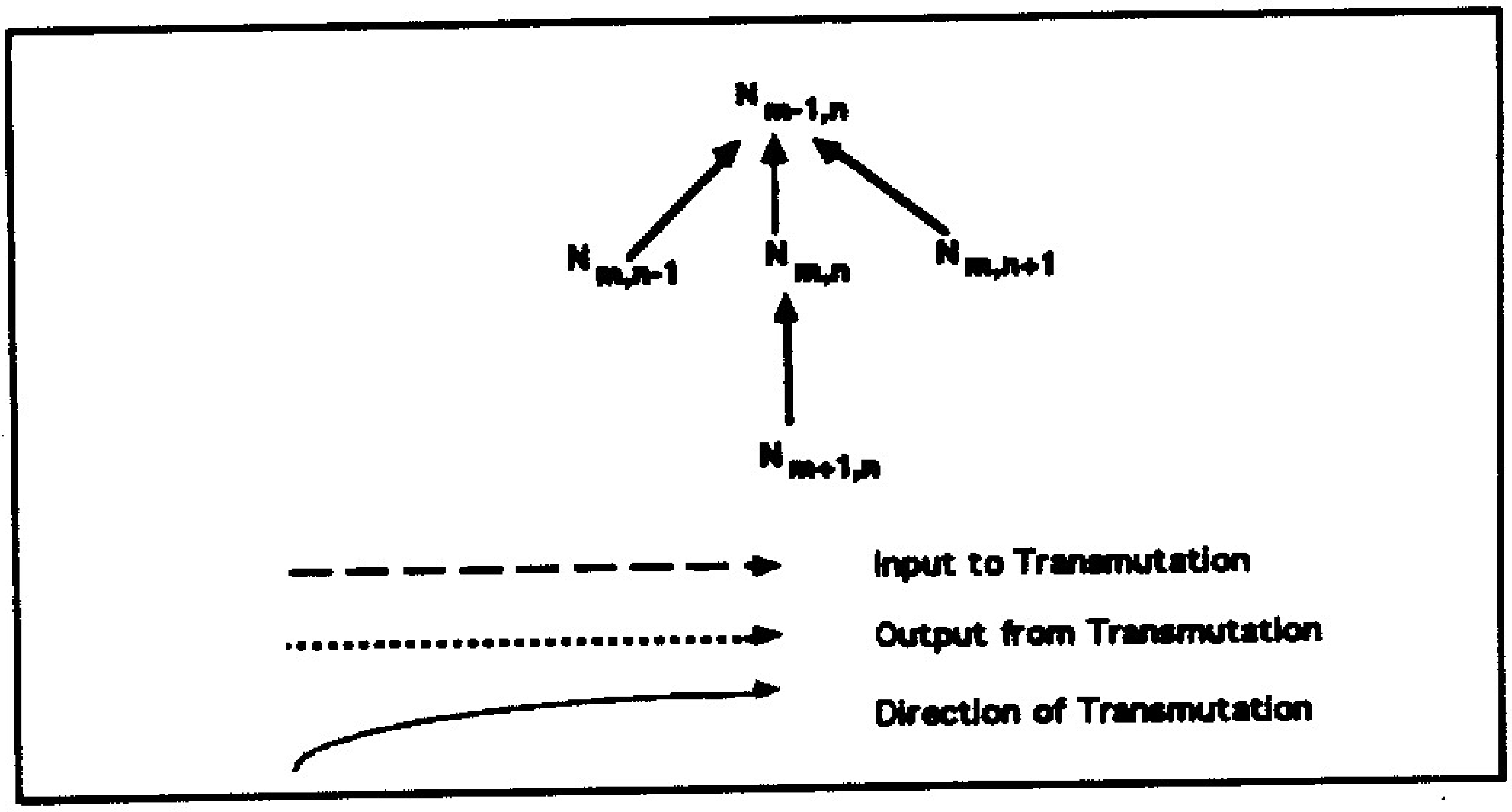


Figure 7 - Legend for Transmutations

The input statement is the same as that of Figure 3, without the belief measure hierarchy. All of the examples are interpreted according to the schema SH1 shown in Figure 4.

There are two referents in the input statement. The resulting statements (output) show the results of the given transmutation assuming that there are no merit parameters that assist in the specialization or concretion and that the similization operator finds a single "most similar" node using the descriptors given. The Background Knowledge (BK) is the learner's prior knowledge that is relevant to the learning process.

The transmutations are characterized above in terms of Input, Output and Background Knowledge (BK). Note that a range must be given to determine how far to generalize for the Quantification operators. No context (or selection of the correct hierarchy, H, to utilize) is necessary as the quantity hierarchy is specified implicitly. In all other cases, a context must be specified, however, as a given node can be in multiple hierarchies.

Quantification Generalization

Input :	Some power plants in New York have mechanical failures.
BK:	Quantity hierarchy
Output :	All power plants in New York have mechanical failures.

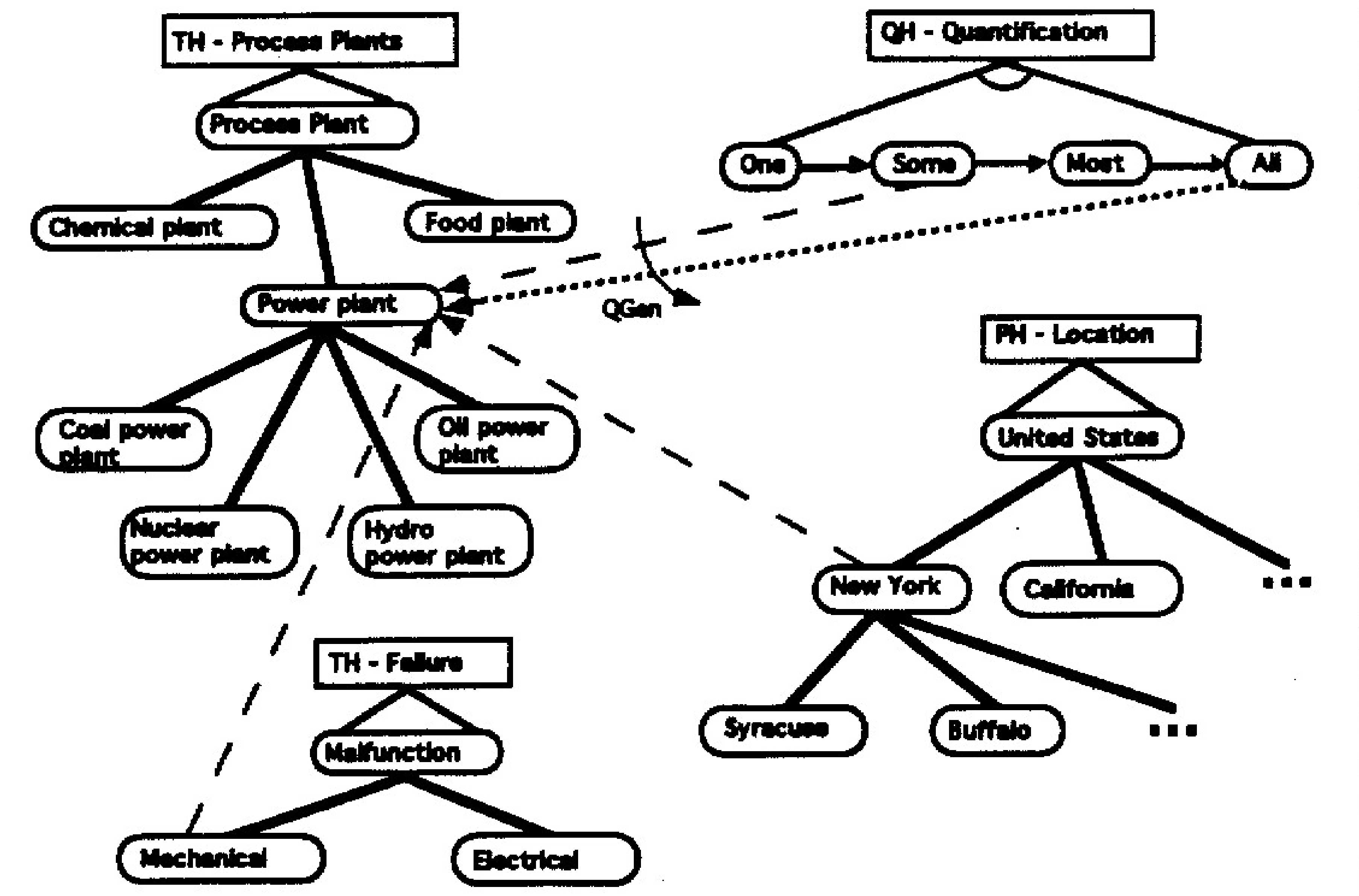


Figure 8: Inductive generalization based on quantification.

<u>Quantification Generalization Transmutation</u>

Input:	Trace "D(Node $A_{m,n}$)=$QR_{i,k}$", range
BK:	Quantity Hierarchy H containing $QR_{i,k}$
Output:	Trace "D($A_{m,n}$)=$QR_{i-1,k}$"

Quantification Specialization

Input :	Some power plants in New York have mechanical failures.
BK:	Quantity Hierarchy
Output :	One power plant in New York has a mechanical failure.

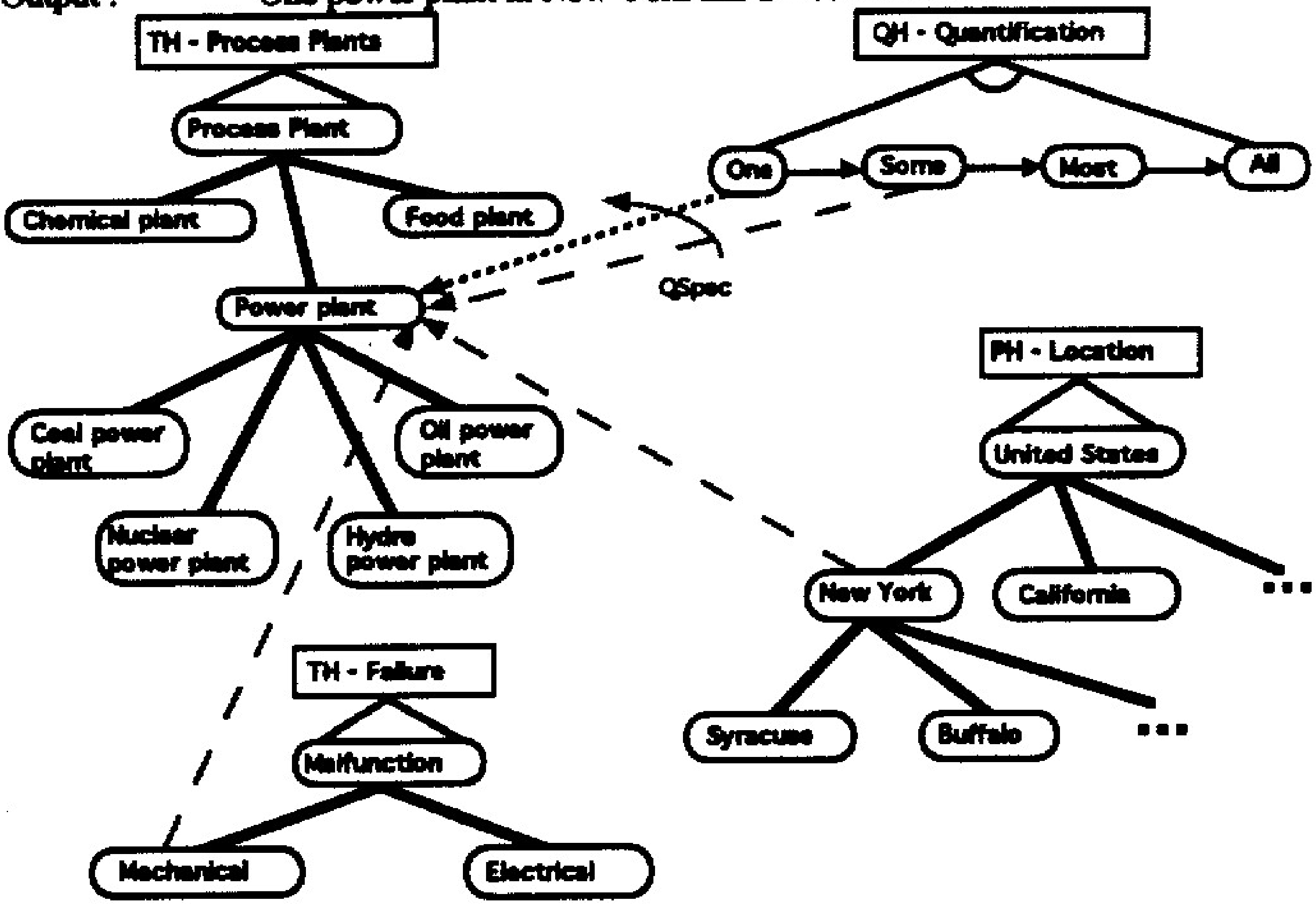


Figure 9: Deductive specialization based on quantification.

<u>Quantification Specialization Transmutation</u>

Input:	Trace "D(Node $A_{m,n}$)=$QR_{i,k}$", range
BK:	Quantity Hierarchy H containing $QR_{i,k}$
Output:	Trace "D($A_{m,n}$)=$QR_{i+1,k}$"

Argument Generalization

Input : Some power plants in New York have mechanical failures.
BK: Power plants are a type of Process plants
Output : Some process plants in New York have mechanical failures.

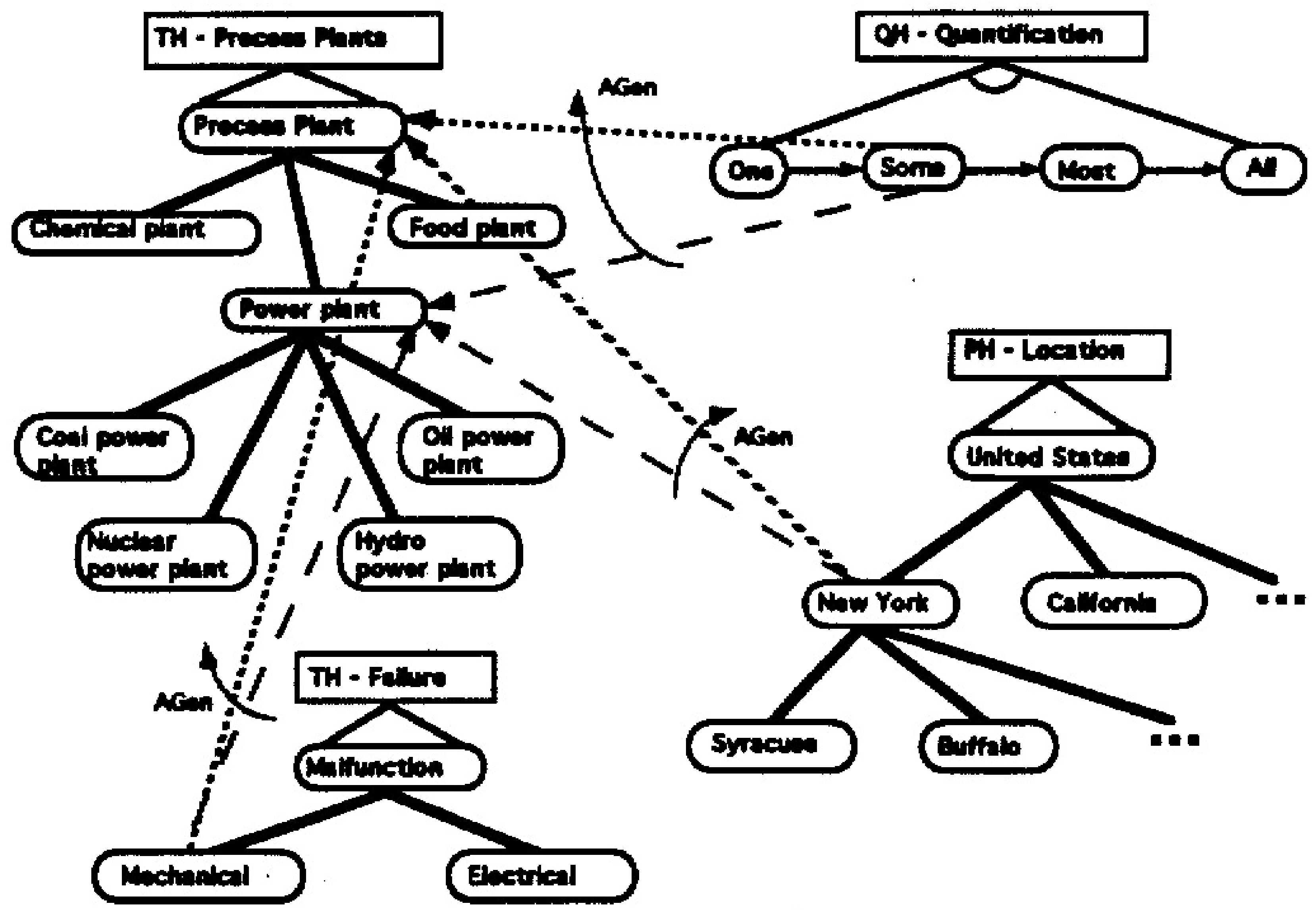


Figure 10: Deductive generalization based on the argument.

<u>Argument Generalization Transmutation</u>

Input: Trace "D(Node $A_{m,n}$)=$R_{i,k}$"; Context
BK: Hierarchy H containing $A_{m,n}$
Output: Trace "D($A_{m-1,n}$)=$R_{i,k}$"

Argument Specialization

Input :	Some power plants in New York have mechanical failures.
BK:	A Nuclear power plant is a type of Power Plant.
Output :	Some Nuclear power plants in New York have mechanical failures.

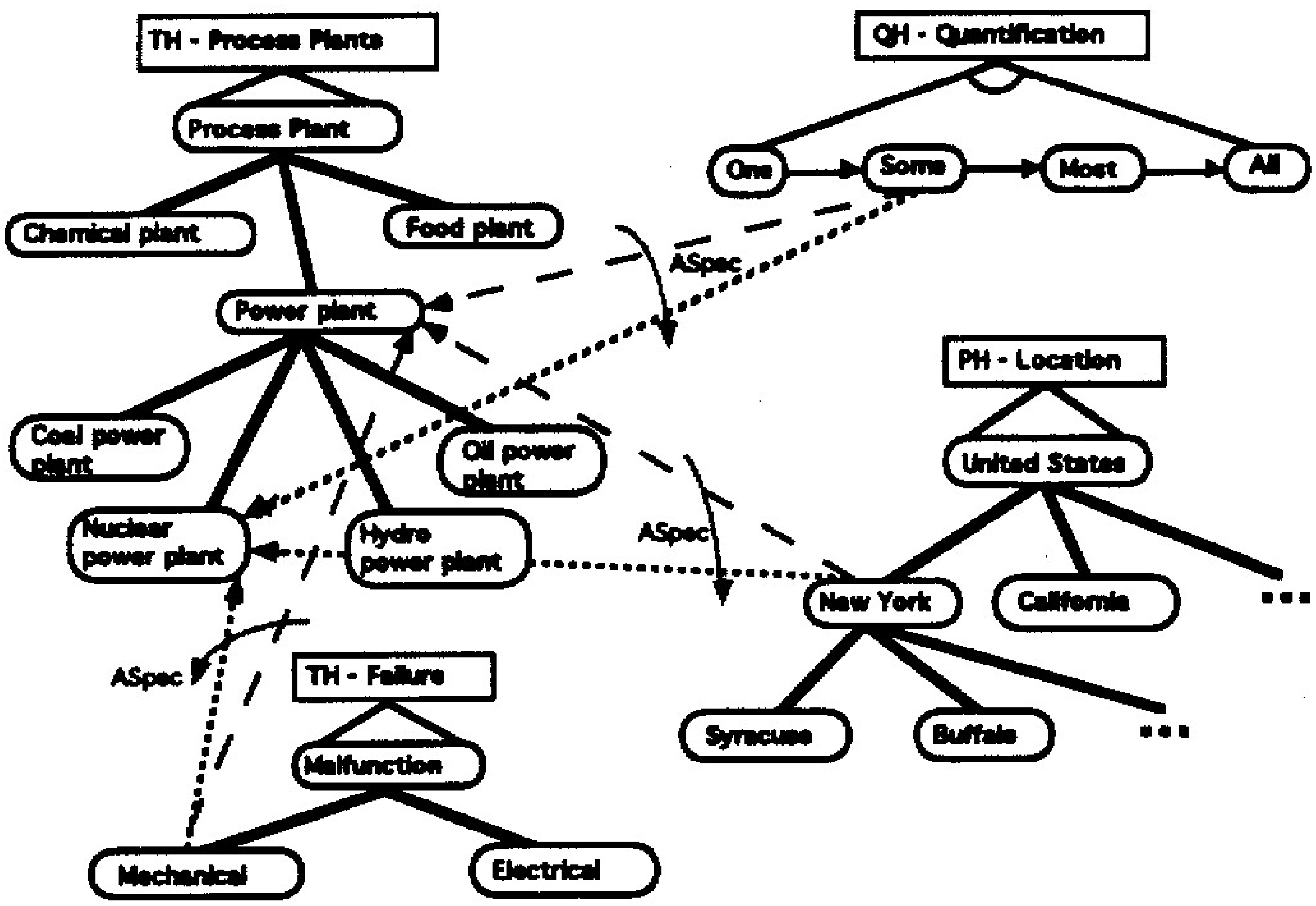


Figure 11: Inductive specialization based on the argument.

<u>Argument Specialization Transmutation</u>

Input:	Trace "D(Node $A_{m,n}$)=$R_{i,k}$"; Context
BK:	Hierarchy H containing $A_{m,n}$; merit parameters
Output:	Trace "D($A_{m-1,n}$)=$R_{i,k}$"

Abstraction

Input : Some power plants in New York have mechanical failures.
BK: A mechanical failure is a type of failure.
Output : Some power plants in New York have failures.

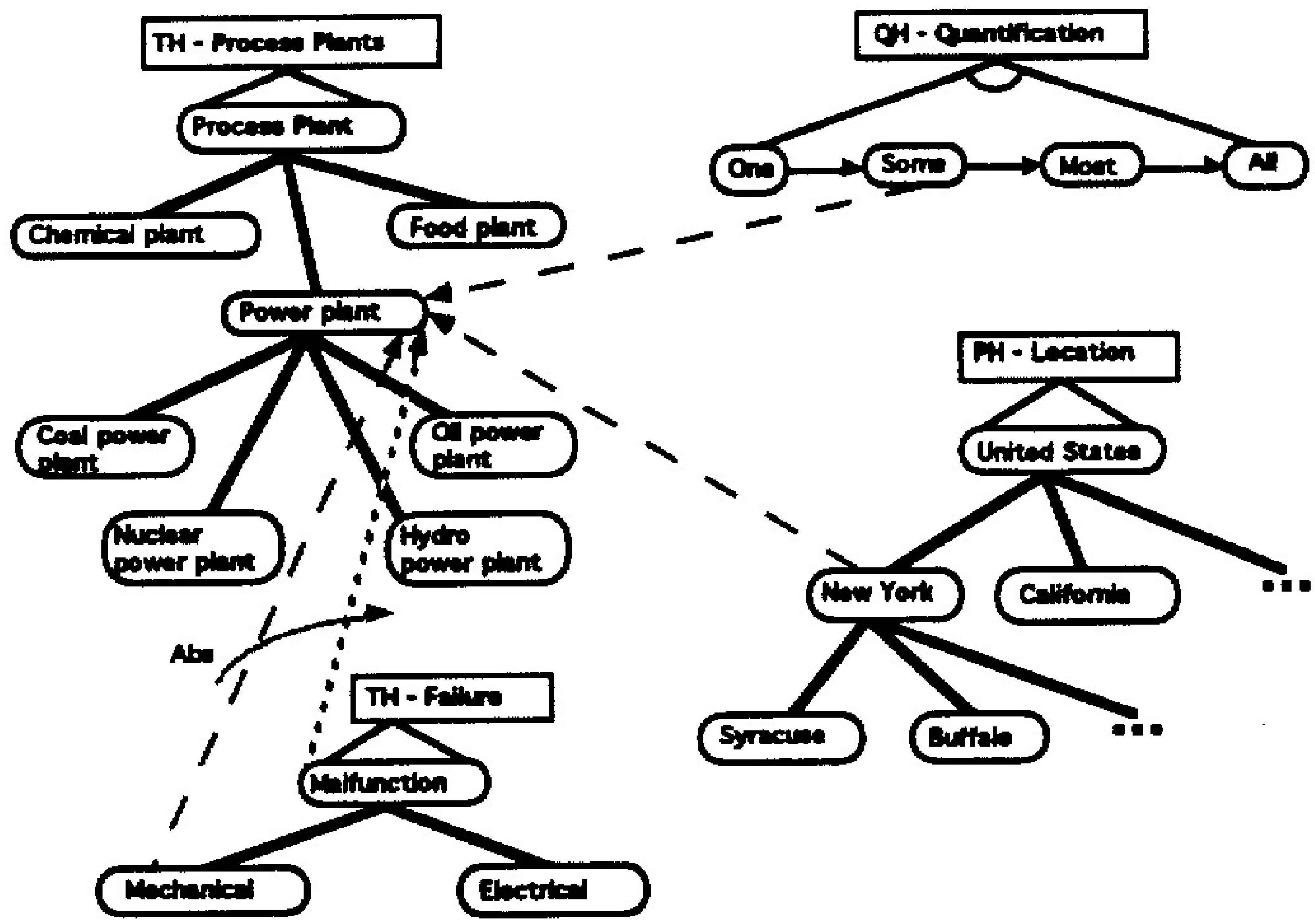


Figure 12: Abstraction transmutation

Abstraction Transmutation

Input: Trace "D(Node $A_{m,n}$)=$R_{i,k}$"; Context
BK: Generalization Hierarchy H containing $R_{i,k}$
Output: Trace "D($A_{m,n}$)=$R_{i-1,k}$"

Concretion

Input :	Some power plants in New York have mechanical failures.
BK:	A component defect is a type of mechanical failure.
Output :	Some power plants in New York have component defects.

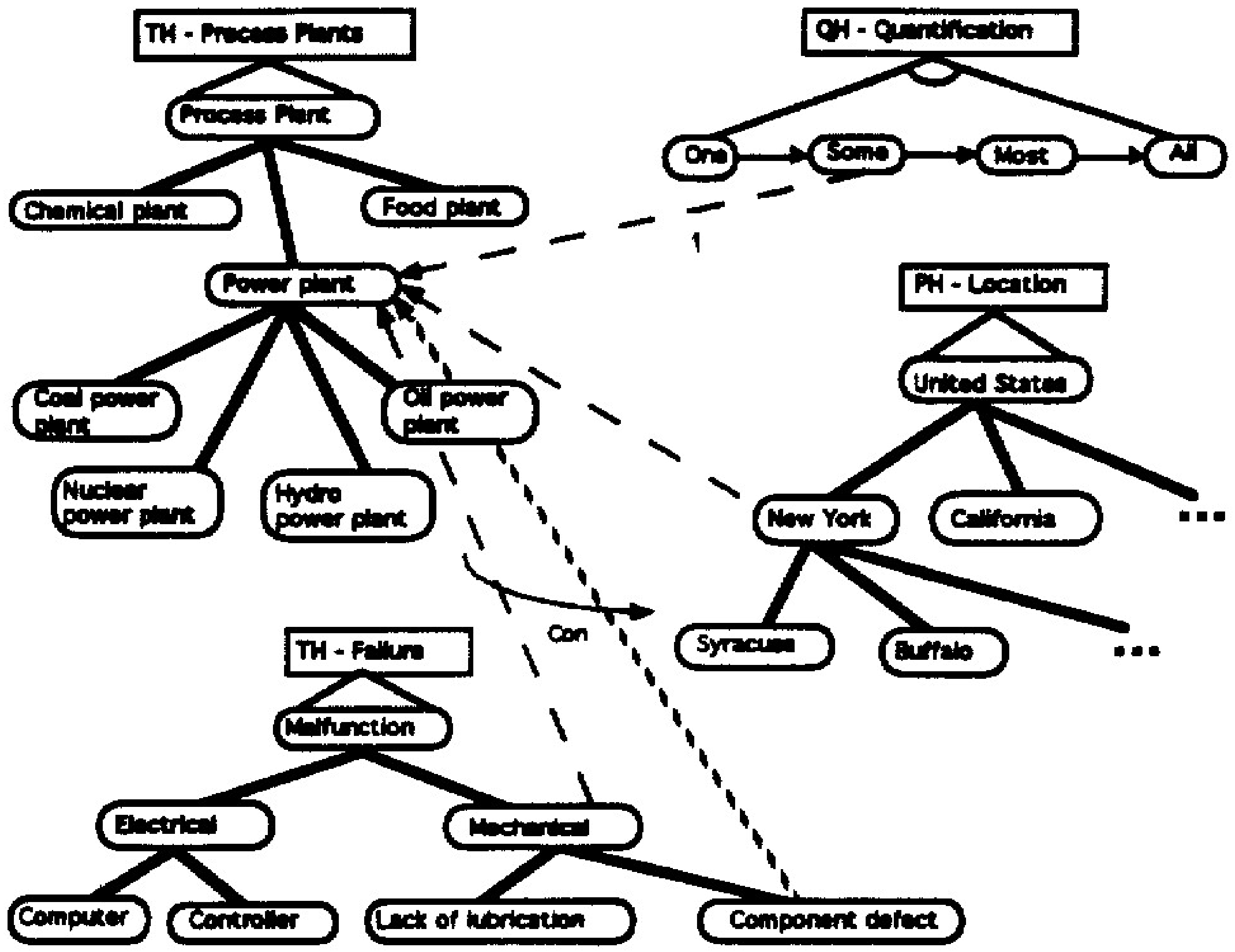


Figure 13: Concretion transmutation

<u>Concretion Transmutation</u>

Input:	Trace "D(Node $A_{m,n}$)=$R_{i,k}$"; Context
BK:	Generalization Hierarchy H containing $R_{i,k}$; merit parameters
Output:	Trace "D($A_{m,n}$)=$R_{i+1,k}$"

Argument Similization

Input :	Some power plants in New York have mechanical failures.
BK:	Both power plants and chemical plants are process plants.
Output :	Some chemical plants in New York have mechanical failures.

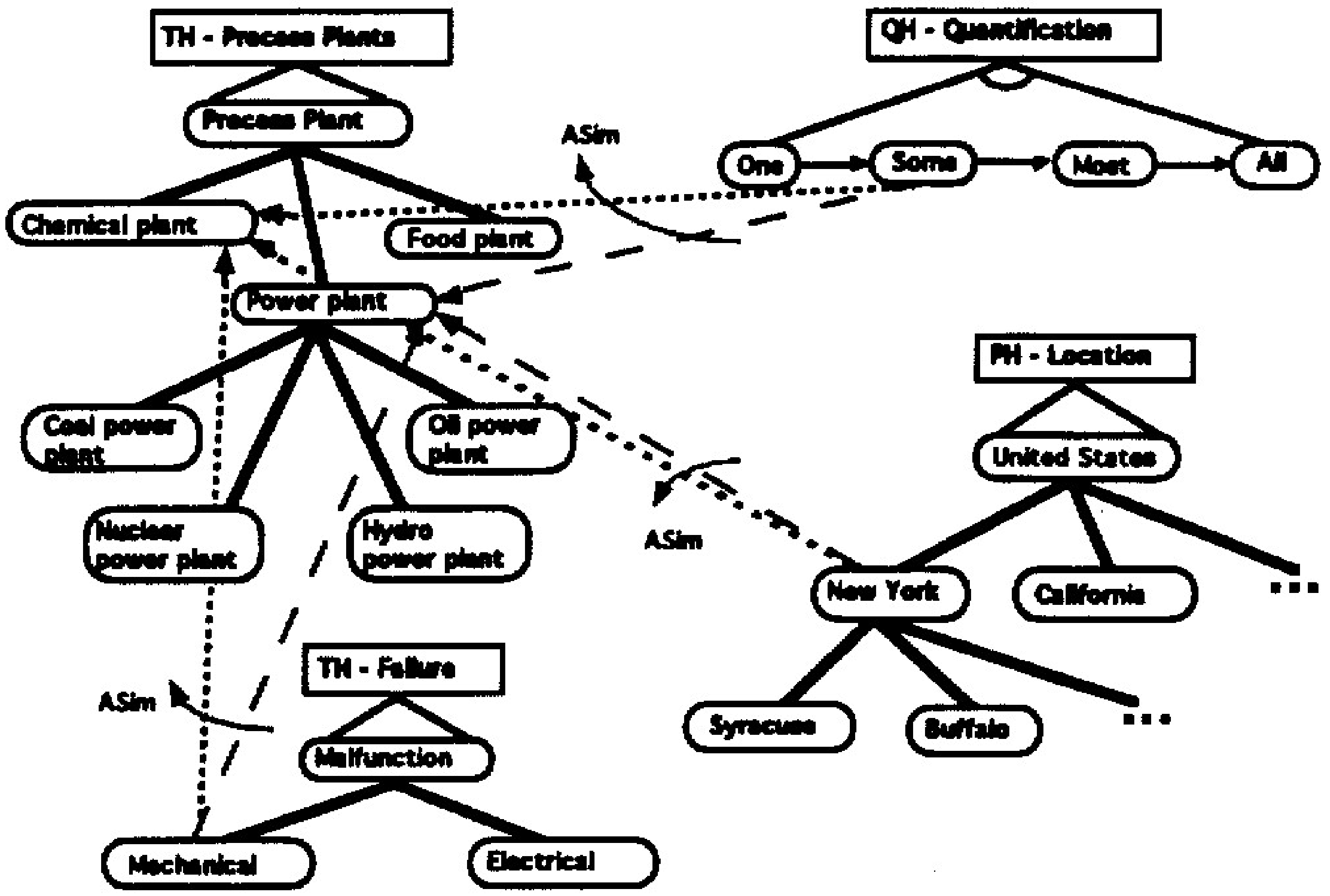


Figure 14: Similization transmutation based on the argument

Argument Similization Transmutation

Input:	Trace "D(Node $A_{m,n}$)=$R_{i,k}$"; Context Set of relevant Descriptors {D_p...D_q};
BK:	Hierarchy H containing $A_{m,n}$; y Children of $A_{m-1,n}$
Output:	Trace "D($A_{m,n+x}$)=$R_{i,k}$" where $x \le y$

Referent Similization

Input :	Some power plants in New York have mechanical failures.
BK:	Both California and New York are part of the United States.
Output :	Some power plants in California have mechanical failures.

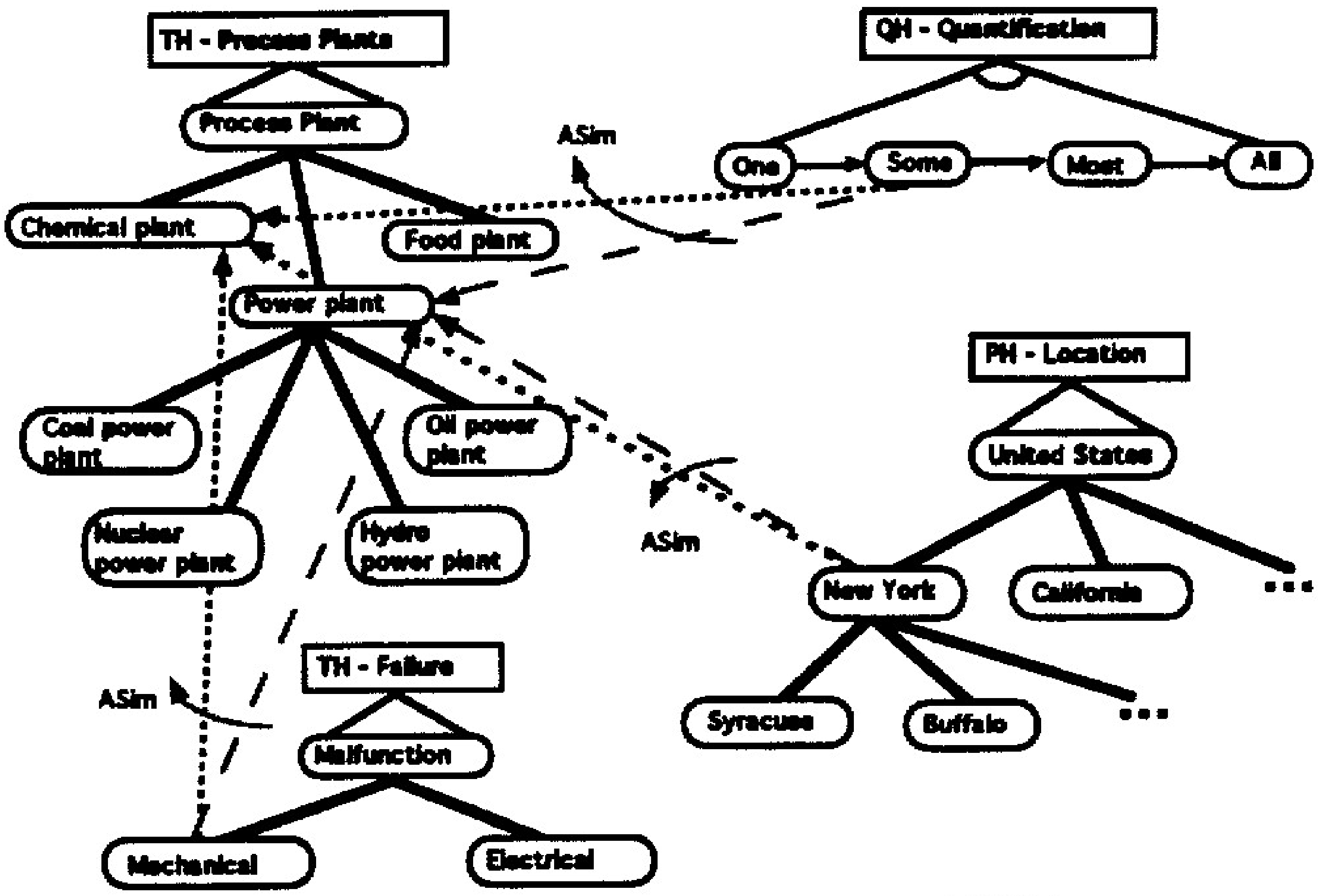


Figure 15 - Referent Similization Transmutation

Referent Similization Transmutation

Input:	Trace "D(Node $A_{m,n}$)=$R_{i,k}$"; Context; Set of relevant Descriptors $\{D_p...D_q\}$;
BK:	Hierarchy H containing $R_{i,k}$; y Children of $R_{i-1,k}$
Output:	Trace "D($A_{m,n}$)=$R_{i,k+x}$" where $x \leq y$

4. Relevant Research

Historically, the idea of plausible inference has been dwelt onby several philosophers, in particular, Ajdukiewics (Ajdukiewics, 1965), Polya (Polya, 1968) and more recently, Rescher (Rescher, 1976). They attacked the question of how people draw conclusions from uncertain and possibly conflicting evidence using both structural and parametric knowledge, not only parametric knowledge (eg. frequency-based probability theories). Polya in particular posits that there are certain 'patterns of inference' that people utilize.

Other relevant past research includes the development of systems that can reason with commonsense knowledge, as well as representations that are explicitly concerned with multistrategy learning.

One such system is Cyc, developed by Doug Lenat and his collaborators at MCC (Lenat & Guha, 1990a; Lenat & Guha, 1990b). This system is designed to represent commonsense knowledge with the stated aim of developing a system that can understand newspaper articles. This project is characterized by the design and construction of a very large knowledge base that uses frames and predicate calculus for its basic representation. Cyc has a wide variety of different forms of inference implemented. While controversial, the research on Cyc has raised many ontological issues which any commonsense reasoning system must address.

While Cyc is the most ambitious of the projects that are mentioned here, it does not have any unifying theoretical basis for its various learning methods. In addition, Cyc uses a nonmonotonic logic scheme to deal with uncertainty, which is quite contrary to the theory of plausible reasoning. Thus Cyc's representation takes a quite different path to the goal of knowledge rich reasoning then is taken by DIH.

Another relevant representation system is the Common Knowledge Representation Language (CKRL), an ESPRIT project (Morik, Causse & Boswell, 1991). CKRL offers a canonical form in which knowledge can be exchanged between machine learning tools. To this end the project has explored the issue of expressibility for each of the individual languages for the various machine learning tools.

In this design, the representation is primarily used for communication between these tools. In other words, CKRL sits between the human user and the machine learning tools. In the most basic scenario, any input is given in CKRL, and then translated into the proper form for the appropriate machine learning tool. When the learning algorithm has results to report, these must be translated into CKRL and then given to the user or another learning algorithm.

While this project is explicitly dealing with the problem of accommodating many different types of learning within a single multistrategy system, it adopts the practical aim of working with existing systems, rather than designing an optimal representation. To be fair, the designers point out that is unlikely that any one representation is optimal for all learning tasks. This is a possible objection to DIH, in that we present a canonical system. However, we are looking ahead to new systems which are waiting to be built and not trying to mend many different tools together. Our aims, stated above, are also concerned with inference. Thus CKRL is valuable in that it shows one approach to designing a representation for multistrategy learning. Also CKRL has analyzed what types of background knowledge are necessary for different machine learning tasks.

A project very similar to CKRL is the Darpa Knowledge Representation Standards Effort (Neches et al., 1991). This wide ranging research initiative has several components. The Interlingua group has developed KIF, a Knowledge Interchange Format which is designed to serve as a language for communicating knowledge between computer programs. It is similar to CKRL in that it translates into and from a common, canonical language. It is, however, not optimized for a small area of application (as CKRL was with machine learning), but is intended to be a standard for many different systems. As with CKRL, a translator must be written for each different language that is included

A different group (Knowledge Representations System Specification Working Group) is attempting to develop core Knowledge Representation Systems for each of the major paradigms (e.g.. they are initially working on a language in the KL-One family), but the group likens its effort to that of standardizing the different variants of a language. The problem they face is specifying what inferences their core system will perform.

A knowledge structure named the PAR (Parameterized Association Rule) was proposed in the Inferential Learning Theory by (Michalski, 1990). This was an initial attempt to develop a uniform, comprehensive knowledge representation for Multistrategy Task-adaptive Learning as defined in the Theory. The PAR is frame-based, with merit parameters characterizing the certainty of a relationship. The PAR has the ability to represent not only rules but many other types of associations, such as dependencies and casual relationships. It is expressly created for the purpose of representing any form of knowledge which might be produced or needed by a multistrategy learner.

While the PAR has many similarities with DIH, it does not include the basic structural organization of hierarchies and is more suited for representing complex relationships than general knowledge.

A confirmation of the ideas in DIH came from the WordNet project at Princeton (Beckwith et al., 1991; Miller et al., 1990). WordNet is an enormous lexical database with approximately 50,000 different word forms. WordNet divides the lexicon into four categories: Nouns; Verbs; Modifiers (adjectives and adverbs); and Function Words. We are concerned with the first three categories (function words are considered special cases e.g.. "the" and "when")

Each of the categories are organized differently. Significantly, the nouns are stored in topical hierarchies (both type and part), lending support to the DIH representation. Verbs are organized according to different entailment relationships, however there is enough information on the relationships between verbs to form generalization hierarchies. Adjectives are stored in what is called "N-dimensional hyperspaces". In DIH, any node can be a descriptor of another node, given that it has a predefined applicability relationship. But certain nodes may be only used as descriptors, and it may be useful to structure these descriptor nodes to perform transmutations on them. The research on organization of adjectives may provide useful insight into how to organize descriptors.(Gross, Fischer and Miller, 1991)

The core theory of Plausible Reasoning presents a system that formalizes various plausible inference patterns and "merit parameters" that affect the certainty of these inferences. This system combines structural aspects of reasoning (determined by knowledge structures) with parametric aspects that represent quantitative belief and other measures affecting the reasoning process.

Various components of the "Logic of Plausible Reasoning" have been implemented in several systems (Baker, Burstein & Collins, 1987; Dontas & Zemakova, 1988; Kelly, 1988). These implementations used various subsets of the inferences ("statement transforms") described in the core theory to investigate the parametric aspects of the theory. The implementations demonstrated how the core theory of plausible reasoning can be applied to various domains. DIH specifies a broader set of knowledge transmutations in a general and well-defined knowledge representation. These transmutations are part of a framework for both reasoning and learning.

5. Summary and Open Problems

In this early stage of DIH, the investigation here may seem to raise more questions than it answers. Many important problems remain to be addressed for DIH to emerge as a viable form of representation and inference. These include the graphical depiction of DIH, inferencing techniques, developing a way to treat context and relevancy, and determining probability models for the merit parameters.

Investigating the interactive display and modification of hierarchies is one area of current research. Currently the visual display of inferences is only useful on a very simplified level, as shown in the examples in this report. At any realistic level of complexity, the links become too complex and lose the ability to indicate the knowledge transmutations performed. One approach is to abstract the information contained in the inferences and only show this. Experiments need to be carried out, however, since this is an domain where human cooperation and user evaluation is essential.

One of the most difficult problems is treating the context of a node properly. Certainly the similarity-based transfers depend on identifying relevant attributes and determining the context. As outlined above, the context is the viewpoint of the hierarchy or the organizing principles of the hierarchy.

There has been no discussion of how to apply the inferencing capabilities described together in a coordinated fashion, or indeed, what their control structure would be, although a framework is outlined in a multistrategy task-adaptive learning (MTL) methodology (Hieb and Michalski, 1993).

Recently there has been much interest in building systems that reason plausibly. Kochen and Resnick describe the design of PM, a plausible reasoner in the area of Mathematics (Kochen & Resnick, 1987). They cite the need for a more efficient knowledge representation when performing plausible inference. Several implementations have been made of the "Logic of Plausible Reasoning" theory (Baker, Burstein, & Collins, 1987; Dontas & Zemakova, 1988; Kelly, 1988). The most recent implementation was in the domain of music (Widmer, 1991). All of these implementations have used different treatments of the merit parameters.

Currently there is no accepted probability model for the merit parameters, particularly the parameter of certainty. (This can be seen from the various implementations referenced above. Each of them uses a quite different probability model.) This parameter can be interpreted in a Pascalian or Bayesian model (varying from 0-1 from disbelief to belief), a Dempster-Shafer model using belief functions (capturing the states of doubt and ignorance) or a "fuzzy probability" model (where we have gradations of truth values). It is probable that no one model will capture the necessary behavior for all of the parameters, but by a careful analysis, we may determine which of the many various probability models are best for any given parameter.

5.1 Advantages of DIH

In DIH, the system learns in terms of what you already know. The more you have learned, the easier it is to learn. The more structures you have as background knowledge, the easier it is to assimilate new knowledge or to plausibly explain input statements. DIH is an efficient, representation, because most knowledge modifications consist of forming or changing traces, without affecting the established hierarchies. Inferences are easily and efficiently performed by changing links between structures. Experience from the world is integrated and used to strengthen associations and relationships.

Plausible reasoning is based on psychological evidence as outlined in the "Logic of Plausible Reasoning" (Collins & Michalski, 1989). Knowledge transmutations in DIH are easily understood and manipulated by humans.

A more flexible representation is needed to support multistrategy learning and inference. In this report we have outlined the DIH system, and shown how various basic transmutations can be represented. The results here support the Inferential Theory of Learning and provide a basis for the implementation of MTL systems.

References

Baker, M., Burstein, M.H., and Collins, A.M., "Implementing a Model of Human Plausible Reasoning," *Proceedings of the Tenth International Joint Conference of Artificial Intelligence*, pp. 185-188, Los Altos, CA: Morgan Kaufman, 1987.

Beckwith, R., Fellbaum, C., Gross, D., and Miller, G.A., "WordNet: A Lexical Database Organized on Psycholingistic Principles," *Using On-line Resources to Build a Lexicon*, U. Zernick (Ed.), Hillsdale, NJ: Erlbaum, 1991.

Boehm-Davis, D., Dontas, K., and Michalski, R.S., "A Validation and Exploration of the Collins-Michalski Theory of Plausible Reasoning," *Reports of the Machine Learning and Inference Laboratory*, Center for Artificial Intelligence, George Mason University, Fairfax, VA, MLI-90-5, 1990.

Brachman, R.J., McGuinness, D.L., Patel-Schneider, P.F., Resnick, L.A. and Borgida, A., "Living with CLASSIC: When and How to Use a KL-ONE-Like Language," *Principles of Semantic Networks - Explorations in the Representation of Knowledge*, J.F. Sowa (ed.), Morgan Kaufmann Publishers, 1991.

Collins, A., and Michalski, R.S., "The Logic of Plausible Reasoning: A Core Theory," *Cognitive Science,* Vol. 13, pp. 1-49, 1989.

Collins, A.M., and Quillian, M.R., "How to make a language user," *Organization of Memory*, E. Tulving and W. Donaldson (Eds.), New York: Academic, 1972.

Dontas, K., and Zemakova, M., "APPLAUSE: An Implementation of the Collins-Michalski Theory of Plausible Reasoning," *Proceedings of the Third International Symposium on Methodologies for Intelligent Systems*, Torino, Italy, 1988.

Gross, D., Fischer, U., and Miller, G.A., "The Organization of Adjectival Meanings," *Journal of Memory and Language,* Vol. 28, pp. 92-106, 1989.

Hieb, M.R. and Michalski, R.S., "Multitype Inference in Multistrategy Task-adaptive Learning: Dynamic Interlaced Hierarchies," *Proceedings of the Second International Workshop on Multistrategy Learning*, Harpers Ferry, West Virginia, 1993.

Kelly, J., *PRS: A System for Plausible Reasoning*, M.S. Thesis, Department of Computer Science, University of Illinois, Urbana, 1988.

Kochen, M., and Resnick, P., "PM: A Plausible Mathemachine," *Human Systems Management,* Vol. 7, pp. 163-169, 1987.

Lenat, D.B., and Guha, R.V., *Building Large Knowledge-Based Systems*, Reading, Massachusetts: Addison-Wesley, 1990a.

Lenat, D.B., and Guha, R.V., "Cyc: A Midterm Report," *AI Magazine,* Vol. 11, No. 3, pp. 32-59, 1990b.

Michalski, R. S., "A Theory and Methodology of Inductive Learning," Chapter in the book, *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. Carbonell and T. Mitchell (Eds.), TIOGA Publishing Co., Palo Alto, 1983, pp. 83-134.

Michalski, R.S., "Inferential Learning Theory as a Basis for Multistrategy Task-Adaptive Learning," *First International Workshop on Multistrategy Learning*, pp. 3-18, Harpers Ferry, West Virginia, 1991.

Michalski, R.S., "Inferential Theory of Learning: Developing Foundations for Multistrategy Learning," *Reports of the Machine Learning and Inference Laboratory*, Center for Artificial Intelligence, George Mason University, Fairfax, VA, MLI-92-3, 1992.

Michalski, R.S., "Inferential Theory of Learning: Developing Foundations for Multistrategy Learning," in *Machine Learning: A Multistrategy Approach, Volume 4*, R.S. Michalski & G. Tecuci (Eds.), Morgan Kaufmann Publishers, 1993.

Miller, G.A., Fellbaum, C., Kegl, J., and Miller, K., "WordNet: An Electronic Lexical Reference System Based on Theories of Lexical Memory," *Revue Québécoise Linguistique*, Vol. 17, pp. 181-213, 1990.

Morik, K., Causse, K., and Boswell, R., "A Common Knowledge Representation Integrating Learning Tools," *First International Workshop on Multistrategy Learning*, pp. 81-96, Harpers Ferry, West Virginia, 1991.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W.R., "Enabling Technology for Knowledge Sharing," *AI Magazine*, Vol. 12, No. 3, pp. 36-56, 1991.

Polya, G., *Patterns of Plausible Inference*, Princeton, New Jersey: Princeton University Press, 1968.

Rescher, N., *Plausible Reasoning: An Introduction to the Theory and Practice of Plausibilistic Inference*, Amsterdam, The Netherlands: Van Gorcum, Assen, 1976.

Tecuci, G. & Michalski, R.S., "A Method for Multistrategy Task-adaptive Learning Based on Plausible Justifications," in Birnbaum, L., & Collins, G. (Eds.) *Machine Learning: Proceedings of the Eighth International Workshop*, San Mateo, CA, Morgan Kaufmann, 1991.

Tecuci, G., "An Inference-Based Framework for Multistrategy Learning" in *Machine Learning: A Multistrategy Approach, Volume 4*, R.S. Michalski & G. Tecuci (Eds.), Morgan Kaufmann Publishers, 1993.

Widmer, G., "Learning By Plausible Reasoning and its Application to a Complex Musical Problem," *First International Workshop on Multistrategy Learning*, pp. 411-418, Harpers Ferry, West Virginia, 1991.

Winston, M.E., Chaffin, R., and Herrmann, D., "A Taxonomy of Part-Whole Relations," *Cognitive Science*, Vol. 11, pp. 417-444, 1987.

Wnek, J. and Michalski, R.S., "An Experimental Comparison of Symbolic and Subsymbolic Learning Paradigms: Phase I - Learning Logic-style Concepts," *First International Workshop on Multistrategy Learning*, pp. 324-339, Harpers Ferry, West Virginia, 1991.