TYPES OF EXPLANATION AND THEIR ROLE IN MULTISTRATEGY CONSTRUCTIVE LEARNING

by

R. S. Michalski
H. Ko

Reports of the Machine Learning and Inference Laboratory, School of Information Technology and Engineering, George Mason University, Fairfax, VA, January 1989.

Reports of Machine Learning and Inference Laboratory
George Mason University
1/18/89

12/25/89

TYPES OF EXPLANATION and THEIR ROLE IN MULTISTRATEGY CONSTRUCTIVE LEARNING

Ryszard S. Michalski and Heedong Ko
Artificial Intelligence Center
George Mason University
4400 University Drive
Fairfax, VA 22030

Abstract

Constructing an explanation of some observation involves generally two components: the *explanation knowledge* that provides premises needed for explaining the observation, and the *explanation structure* that demonstrates the logical link between the premises and the observation. Two types of explanations are distinguished: *derivational* - that involve only truth-preserving *(or deductive)* knowledge transformations, and *hypothetical* - that involve creating an explanation structure assuming some hypotheses built through falsity preserving *(or inductive)* knowledge transformations. These distinctions are used to compare *empirical, analytical* and *multistrategy constructive learning* methods.

Multistrategy constructive learning requires both empirical and analytic inference mechanisms. An observation may be explained by a deductive inference when the observation is logically implied by the background knowledge, and by an inductive inference, otherwise. Through inductive inference a new piece of knowledge is hypothesized that allows one to establish the explanation structure. An example, as applied to a problem in assembly planning, is presented.

1. Introduction

An explanation of some phenomenon involves indexing knowledge in memory that logically entails the phenomenon, and creating a structure that demonstrates that this knowledge indeed entails the phenomenon. This distinction is useful in analyzing various learning paradigms to unify inductive and deductive learning as a constructive learning paradigm [Michalski and Ko, 1988; Michalski and Watanabe, 1988].

To illustrate this distinction, suppose that a person did not come to a meeting with a friend. Suppose that this person later told his friend that he did not come to the meeting because his mother got sick. This statement, called *explanation knowledge*, explains the person's absence to his friend, if the friend has the background knowledge that a crisis in a family typically overrides other commitments, and that a mother's sickness is a family crisis. The explicit demonstration that the explanation knowledge indeed explains the behavior (which in our example case is a simple two step application of modus ponens) is called *explanation structure*. The explanation structure is thus a logical derivation that the explanation knowledge together with background knowledge logically implies the behavior, or, in general, any phenomenon that is being explained.

Without knowing the illness of the person's mother, no explanation structure can be constructed. In pure analytic learning, such as explanation-based learning, the system could not learn in such a case because its learning mechanism requires a complete explanation structure. People in such situations, however, usually construct some hypothesis to explain the *observation*, in this case, the absence of a person at a meeting. The friend, for example, may hypothesize that something serious might have happened to the person. This leap of faith is a form of *constructive inductive inference* (Michalski, 1983). It could be done by employing a background knowledge rule:

"If a person, who normally comes to meetings, did not come to one, then it is likely that something serious must have happened".

For a given observation, a learning system may have sufficient background knowledge (e.g., contain the rule as above) to construct the required explanation structure, or may not. To handle both cases, the idea of *constructive closed-loop learning* (or, briefly, *constructive learning*) has been introduced [Michalski & Ko, 1988]. Here, we extend the notion to *multistrategy constructive learning* (MCL).

In multistrategy constructive learning the system uses deductive inference when what is being explained (a fact, example, and more) is logically implied by background knowledge and observations, but resorts to inductive inference when an additional (or modified) knowledge is necessary to establish the needed explanation structure. Thus, such a system can learn when its initial knowledge is inadequate.

Many practical learning situations start with such an inadequate knowledge. An important example of such a situation is learning by an autonomous robot exploring a partially known environment, or a robot assembling a device without a complete domain knowledge of constructing an assembly sequence.

In the next section we use the above ideas to distinguish between two types of explanation, *derivational (or deductive)* and *hypothetical (or inductive)*. Based on this distinction, we then characterize different learning approaches. In particular, we discuss multistrategy constructive learning in terms of these ideas, and illustrate a MCL system by an example from the area of automated robot assembly.

2. Types of Explanation

What is being explained could be an observation, a concept, an instruction, and many more, called *explanation target* (ET). The objective of an explanation to a person is to logically relate the *background knowledge* (BK) of the person to ET. In the process, the person may generate a knowledge structure explaining ET. The constructed knowledge structure, called *explanation structure* (ES), demonstrates that BK logically entails ET, which we write:

$$BK \mid\!> ET \qquad (1)$$

In other words, ET must be a logical consequence of BK. In many situations, however, the background knowledge (BK) may not be adequate to establish (1). BK may be inadequate because it may be intractable computationally (too complex), or inconsistent with the explanation target (ET). BK should be modified: the former requires reformulating BK and the latter requires correcting BK. In both cases, BK should be modified BK* to establish an ES: thus,

$$BK^* \mid\!> ET \qquad (2)$$

In some instances, BK may be inadequate because it is incomplete. It should then be enhanced by additional knowledge, called *explanation knowledge* (EK). EK may be given to us from another source, e.g., teacher or environment, or may have to be hypothesized. In this case the explanation structure (ES) is established showing:

$$\text{BK \& EK} \mid\!> \text{ET} \qquad (3)$$

Constructing an explanation knowledge may require changing (updating, correcting, etc.) BK into some modified BK* for the similar reason as the ES in (2). Thus, in general, establishing an ES involves a combination of the explanation processes for establishing (2) and (3):

$$\text{BK* \& EK} \mid\!> \text{ET} \qquad (4)$$

Based on these considerations, an explanation may construct the following two types of knowledge:

EK - additional knowledge required to establish explanation structure (ES)

ES - an knowledge structure that demonstrates that explanation knowledge (EK) together with background knowledge (maybe modified) logically entails the explanation target (ET)

Using this conceptual framework, we can analyze different methods of learning. For example, in explanation-based learning, EK is null, and one seeks the explanation structure, ES, that demonstrates (1). In general, such learning is purely analytic. For another, in pure empirical learning, BK is small and inadequate for explaining ET deductively, so one needs to hypothesize an explanation knowledge EK inductively, based on the observational data as ET.

There may be many different situations between the two extremes, the purely empirical and purely analytic learning. When BK is inadequate, and one needs either to create EK so that (3) holds, and/or modify BK so that (4) holds. Empirical learning may be extended to incorporate substantial amount of BK by a constructive induction. Constructive induction uses BK to create new descriptors missing in the observational data so that a semantically plausible EK in the task domain can be hypothesized. That is, ES demonstrates the following logical relation in constructive induction:

$$\text{BK \& EK} \mid\!> \text{ET*} \qquad (5)$$

BK is used to transform ET to ET* so that more semantically plausible EK with respect to the task domain is formed.

Constructive learning is a theoretical model of learning which can be used as a tool to relate and contrast the two extremes of learning models like constructive induction so as to improve the learning systems that can handle all such situations.

The above leads us to the distinction between two types of explanation:

• a *derivational (or deductive) explanation,* which requires a deductive inference to construct the explanation structure demonstrating that explanation target, ET, is a logical consequence of what the system already knows (BK[1]), i.e., that BK |> ET. The explanation knowledge (EK) is null .

• an *hypothetical (or inductive) explanation,* which requires an inductive inference to hypothesize an explanation knowledge (EK) in addition to a deductive inference to construct the explanation structure, which demonstrates that EK together with background knowledge, BK*, implies explanation target (ET): i.e., BK* & EK |> ET .

On the basis of these concepts, we will analyze in more detail the empirical, analytical and multistrategy constructive learning.

3. Empirical Learning

Traditional empirical learning methods presuppose little *direct* (or observation-level) background knowledge to explain the observation as the explanation target so that the main concern is to hypothesize a concept or a rule explaining the conceptual regularities in the observational data from an external information source inductively [Michalski, 1983, 1987]. Since there is usually a plethora of possible hypotheses that could explain an observation, the main problem is to find the most plausible, or generally, the most preferred explanation knowledge. Note, there are two types of inductive steps taken by

---

[1] In explanation based learning literature, BK is typically called domain knowledge. In general, BK contains domain-specific knowledge, domain independent knowledge, and metaknowledge (rules of inference, constraints, etc.).

empirical learning systems: one is the formulation of the inductive hypotheses and the other is the choice of preferred hypotheses among them. The main inference scheme of these systems is inductive.

The empirical learning task is described as follows:

Given:

- Observational statements about an object, phenomenon, or a process as the explanation target (ET)
- Background knowledge (BK) which includes domain structure, the preference criterion for choosing among competing hypotheses, and inductive rules of inference.

Determine:

- Explanation knowledge (EK) that, if assumed to be true, logically entails the observation and is most plausible, or, in general, most desirable among all other such hypotheses according to a given preference criterion.

Explanation structure (ES) in most empirical learning systems is demonstrating a subsumption relationship between EK and ET. Thus, they typically use a unification procedure or a simple pattern matcher rather than a full deductive decision procedure like a resolution to establish ES.

For example, in SPARC [Michalski & Ko & Chen 1987], the observational statements are a sequence of snapshots of some unknown process. The background knowledge includes attributes used to describe each snapshot of the sequence, associated types and structures of the value sets of the attributes, a rule preference criterion, and inductive generalization rules. In addition, it includes description models. A description model determines a syntactic form of plausible explanation knowledge and the observations (ET) are transformed according to the syntactic constraints of each model into another form (ET*). Therefore, SPARC performed constructive induction according to the equation (5).

The description models were specific to the sequential nature of the observation but independent of the specific processes from which the observations were taken. Therefore, it was a strategic knowledge for inductive generalization for sequential domain. Hence, the background knowledge may be specific to processes (attributes as domain knowledge) or general (description models as strategic knowledge). Using rules from each model as EK, the system may predict the future continuation of the unknown process qualitatively (or symbolically).

4. Analytic Learning

Most well-known forms of analytical learning are explanation-based generalization [Mitchell & Keller & Kedar-Cabelli 1986] and explanation-based learning [DeJong & Mooney 1986]. In this approach the system attempts to show that the background knowledge of the observer logically implies the observation. A successful explanation enables the system to reformulate the background knowledge into a more efficient or operational form with respect to some performance task.

It is well-known that the type of inference used by the above learning schemes is deductive. However, an inductive step is taken when what form of reformulation is considered to be *operational*. [Watanabe, 1988]. This is because the system is lacking an explicit theory of operationality. In a robot planning task, this theory can not be specified completely with open-world assumption. Existing analytic learning systems are equipped with only a primitive inductive step for determining operationality: e.g., tagging a predicate as operational. Analytic learning can be described as follows:

Given:

- Observational statements about some objects, phenomena, processes, or a goal statement as the explanation target (ET)
- Background knowledge (BK) which contains domain theory, domain facts and general meta knowledge such as as well as relevant inference rules.

Determine:

- A reformulation of the background knowledge that logically entails the explanation target and is more effective and/or efficient. than the prior knowledge.

The explanation structure (ES) in these systems is a logical derivational tree generated by a theorem prover, a trace of the Horn clauses in Prolog, or some other equivalent form.

For example, in ARMS system [Segre 1987], the explanation target (ET) includes the goal statement and a sequence of actions performed by the teacher to reach the goal. The goal statement includes a set of joint relationship between components of an assembly. The background knowledge includes general plans for achieving simpler joint relationships. The explanation process indexes general plans in the background knowledge that participated in the teacher's actions and establishes ES as a sequence of instances of the general plans for establishing the joint relations in the goal statement. Then, the system learns a new general macro-plan for achieving the joint relationships of the goal statement. The macro-plan represents a reformulation of the general plans in background knowledge.

5. Multistrategy Constructive Learning

Empirical learning, described above, is one form of inductive learning [Michalski 1973]. Another form is constructive induction [Michalski 1983]. *Constructive induction* incorporates domain specific knowledge to elaborate the input observations with new descriptors in order to search for inductive hypotheses in the preferred description space of the domain concepts. This process incorporates a deductive step into an empirical learning framework when establishing the explanation structure shown in (5). Recently, the concept of constructive induction was extended to *multistrategy constructive learning*, [Michalski, 1987; Michalski & Ko, 1988; Michalski & Watanabe, 1988]. In addition to constructive induction, multistrategy constructive learning requires a *closed-loop mechanism* and a *deductive restructuring*.

A closed-loop learning extends the incremental learning In the incremental learning, the exiting or previously learned concept description was used as a "potential" explanation knowledge of a new input but when the explanation structure can not be formed with the concept description, a concept description was modified to establish the explanation structure:

$$BK \,\&\, EK^* \mid\!> ET \qquad (6)$$

Using INDUCE as an example of a single-concept increment learning system [Larson 1977], the system is given multiple exemplars of west-bound or east-bound trains (two classes) and the concept to learn is different directions of the trains (east-bound or west-bound). There may be multiple candidate descriptions of each class of the concept but there is only a single concept being learned (directionality). Therefore, indexing the memory for which concept to use for EK is trivial: e.g., this is determined by the class of the exemplar in INDUCE.

On the other hand, in a multi-concept learning situation, the system should determine which concepts in memory to be used as an explanation knowledge for ES before incrementally modifying the candidate descriptions of the chosen concept because there are more than one concepts involved in the explanation process. Hence, a closed-loop learning consists of concept determination step and incremental learning step.

Furthermore, the system should evaluate the constructed knowledge in order to decide if it is to be stored as candidate hypotheses for future use or not. Here, the form of ES may be restructured to store more "useful" EK for future explanation process. This step is called *deductive restructuring* of ES. In short,

MCL = Constructive Induction+ Closed-Loop + Deductive Restructuring (7)

Multistrategy constructive learning includes both inductive learning (empirical and constructive) and deductive (analytical) learning. Such an integration is needed for a number of applications, for example, for implementing learning by intelligent robots. A prototype MCL system called NOMAD [Ko, 1989] has been developed for the Intelligent Explorer Project (IEX), conducted by the George Mason University. The goal of the IEX project is to develop an autonomous robot capable of multi-faceted reasoning tasks including learning, reasoning and planning in a partially known environment. NOMAD is a planning system that learns empirically from planning experiences. Here we illustrate central aspects of multistrategy constructive learning, and explain how the system works by a simple learning scenario in the area of automatic assembly.

5.1. Multistrategy Constructive Learning by NOMAD

This section describes a learning element of NOMAD. The goal is to construct an assembly structure specified by its components and their mating conditions. The problem is to form a sequence of assembly steps so that there is a collision free path for each assembly step.

NOMAD constructs assembly sequences from two schemas in memory (two concepts): grouping and precedence schemas. A grouping schema consists of schemas for a base and satellite components. A base component is fixed during an assembly operation; the satellite components are moved to be mated with the base component. A grouping schema may be linked to one or more precedence schemas that specifies the ordering of mating the satellite components.

The multistrategy constructive learning approach by NOMAD is to:

- generate candidate grouping and precedence schemas inductively as EK,
- validate candidate schemas by experiments, and
- apply validated schema(s) in new assembly situations.

Suppose a user specifies the Bell Head assembly structure in terms of three components (Bell Head, Pin, and Ring) and their mating conditions. The completed assembly is shown in Fig. 1. The Global Frame is a reference coordinate system of all the components in the environment: z axis is pointing vertically upward. Each component has its own local reference frame called base frame.. Fig. 1 shows the base frame of Pin.

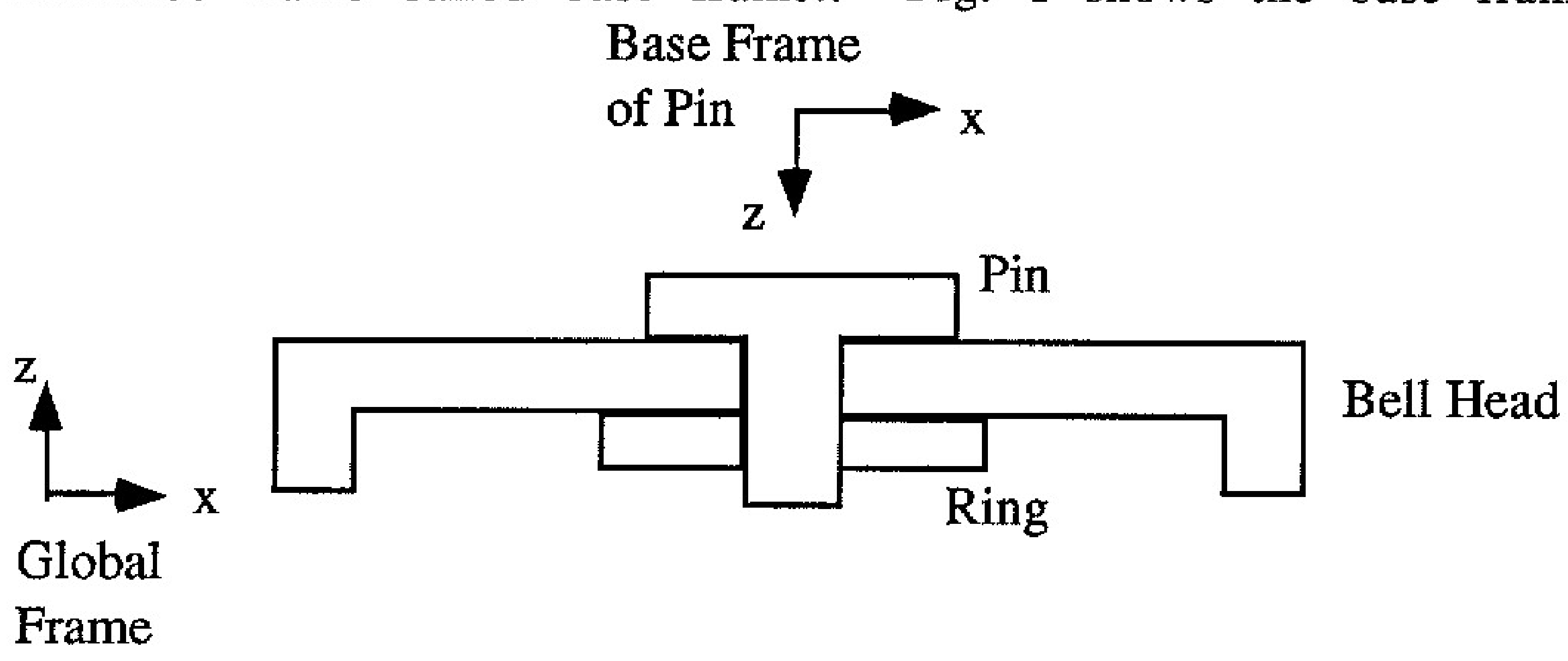
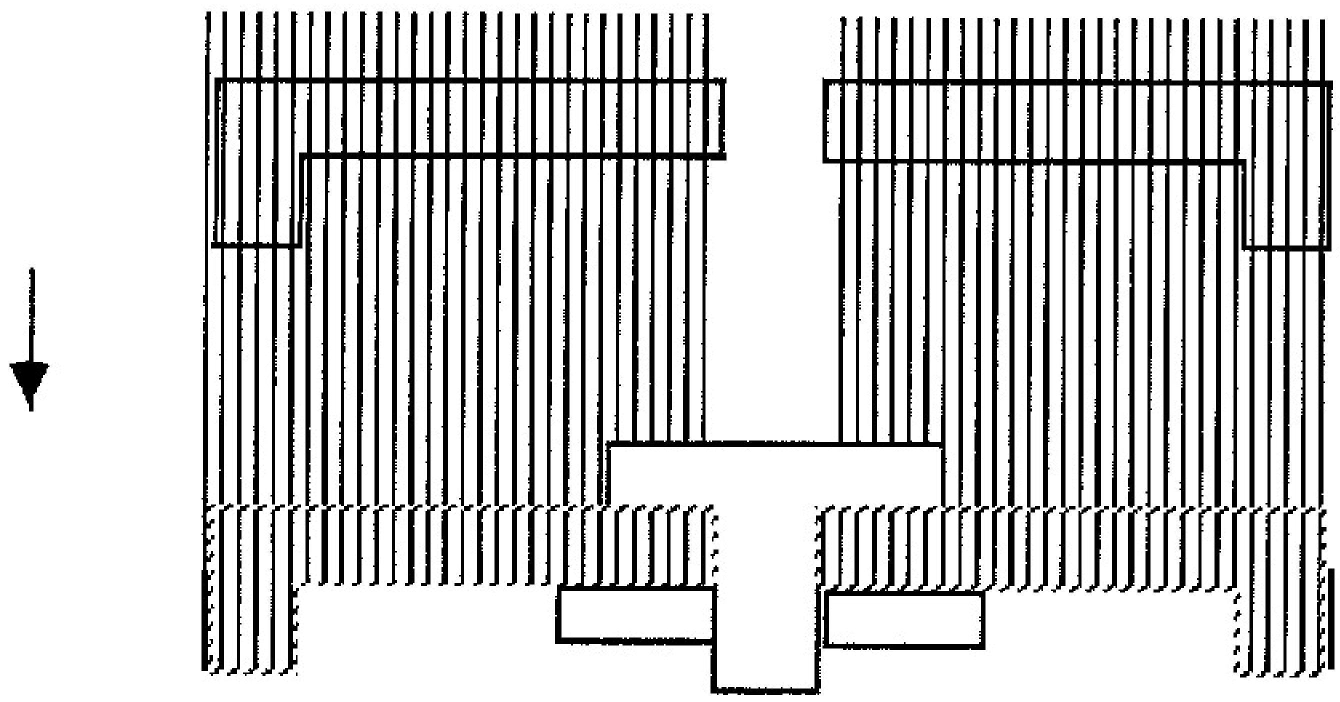


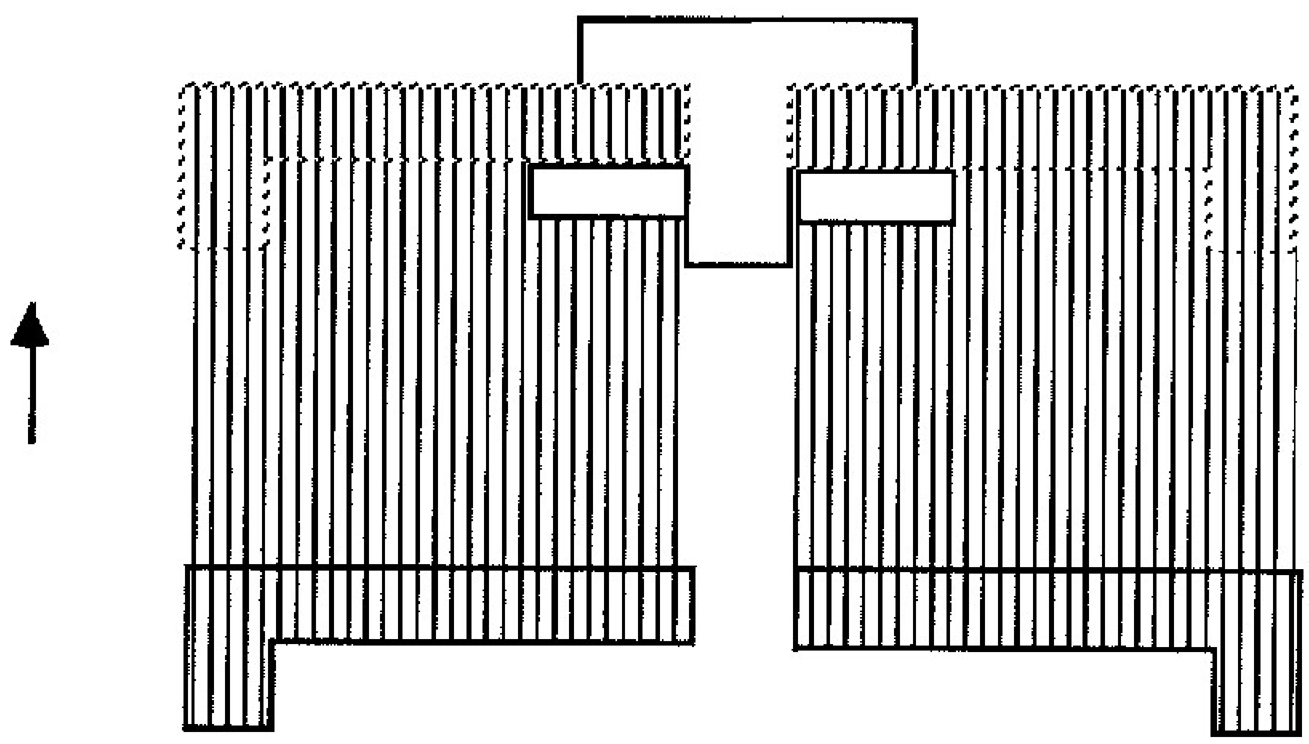
Figure 1. Assembly of Bell Head



Swept Volume
of HEAD

(a)

Interference between Bell Head and Pin



Swept Volume
of HEAD

(b)

Interference between Bell Head and Ring

Figure 2. Spatial Interferences

Initially, NOMAD has no schemas for the Bell Head assembly task at hand. So, the user may specify a grouping: Pin as a base and Ring and Bell Head as satellites. Then, the system may experiment with alternative plans randomly because it does not have any experience to guide the planning process.

When Ring is assembled to the Pin before Bell Head, there is no collision free path for Bell Head as shown in Fig. 2. However, when Bell Head is assembled first, the Ring can be assembled to the Pin. This successful planning episode is summarized by Fig. 3..

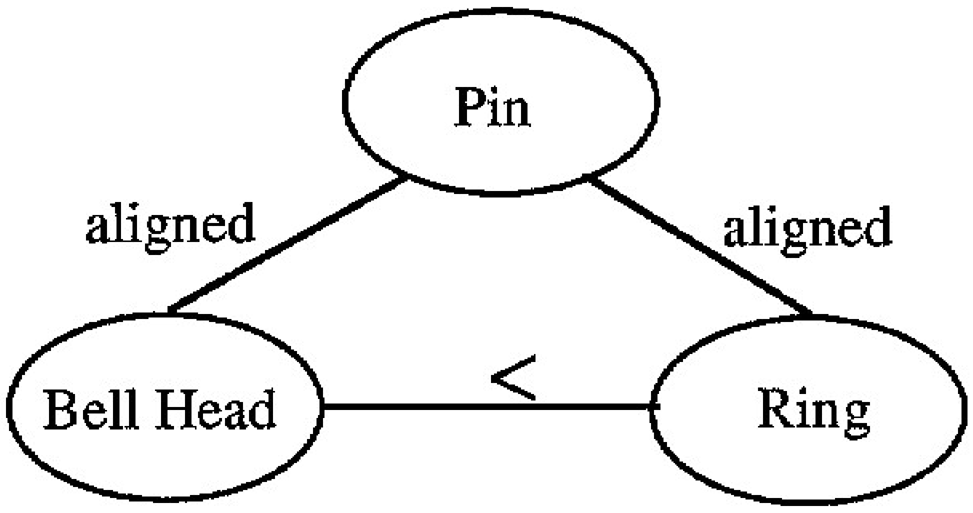


Figure 3. Bell Head Assembly Episode

where "<" in Fig. 3 represents that the assembly of Bell Head temporally precedes that of Ring.

To explain the observation (a successful planning episode), using a constructive induction rule, the following descriptors are generated from the locations of individual satellite components with respect to the two reference frames, the global frame and the base frame of the base component, Pin:

• the location of the Bell Head is above Ring with respect to Global Frame or higher than Ring: [dloc-g (Ring, Bell Head) < 0]

• the location of the Ring is above Bell Head with respect to the base frame of Pin or Bell Head is closer to Pin than Ring: [dloc-b(Ring,Bell Head) > 0)]

In addition, Pin is T-shaped, Bell Head and Ring are ring-shaped.

Then, NOMAD generates alternative schemas from the planning episode, using the "dropping a conjunct" inductive generalization rule [Michalski, 1983], as explanation knowledge.. Two candidate precedence schemas are generated, P1 and P2,.with respect to the grouping schema. P1 with its grouping schema is shown in Fig. 4. $p1, $p2, and $p3 are schemas for components.

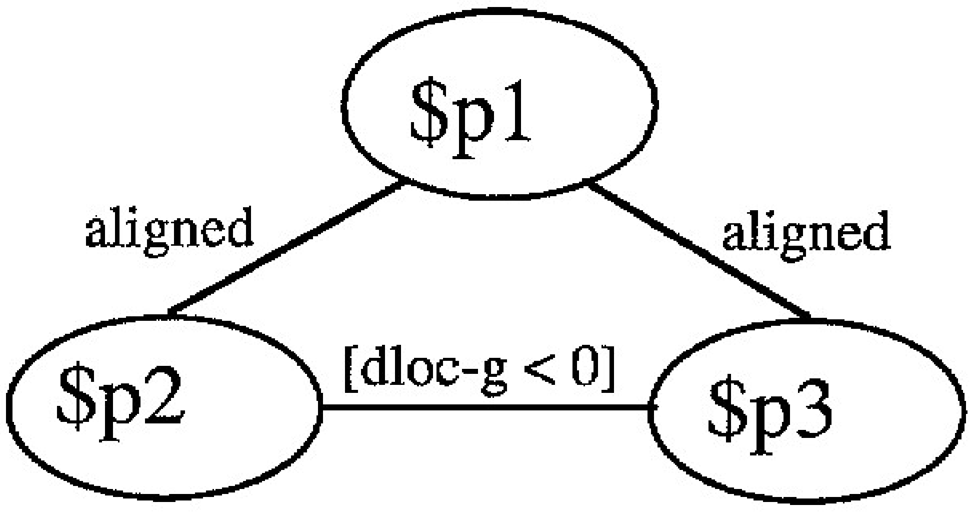


Figure 4. Candidate Precedence Schema P1

Fig. 4 shows that the base component schema, $p1, is aligned with satellite schemas, $p2 and $p3, and $p2 is assembled before $p3 because $p2 is higher than $p3 with respect to the global reference frame. P2 with its grouping schema is shown in Fig. 5. Fig. 5 shows that the base component schema, $p1, is aligned with satellite schemas, $p2 and $p3, and $p2 is assembled before $p3 because $p2 is closer to $p1 The component schemas of Fig. 4 and 5 stores that fact that $p1 is pin-shaped and $p2 and $p3 are ring-shaped.

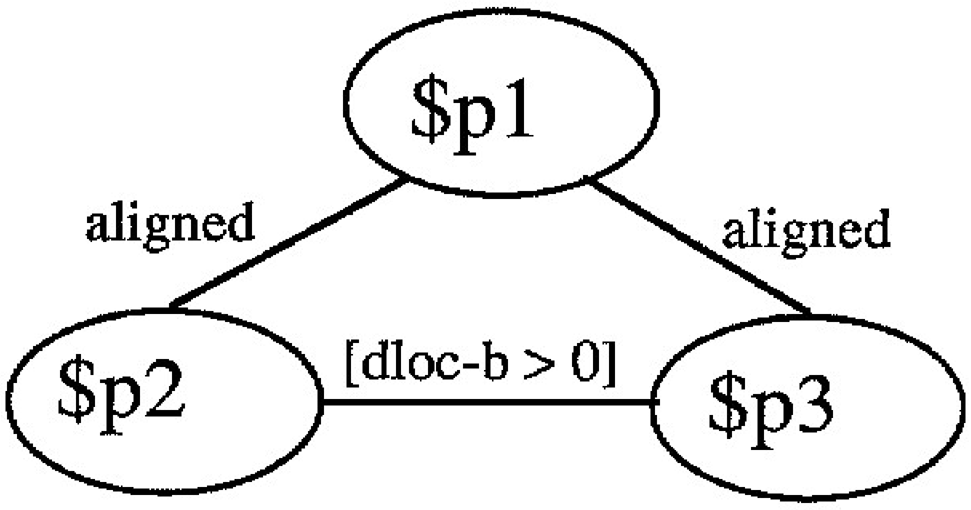


Figure 5. Candidate Precedence Schema P2

Having learned two candidate precedence schemas, P1 and P2 and their grouping schema, the system is given another assembly problem shown in Fig. 6. Now, NOMAD applies these candidate schemas to test their validity empirically in solving the new problem. This problem is very similar to the Bell Head assembly of Fig. 3: Bolt is pin-shaped and Nut and Bracket are ring-shaped. Using the grouping schema of P1 and P2, NOMAD forms a grouping where Bolt as the base component and Bracket and Nut are satellite components .

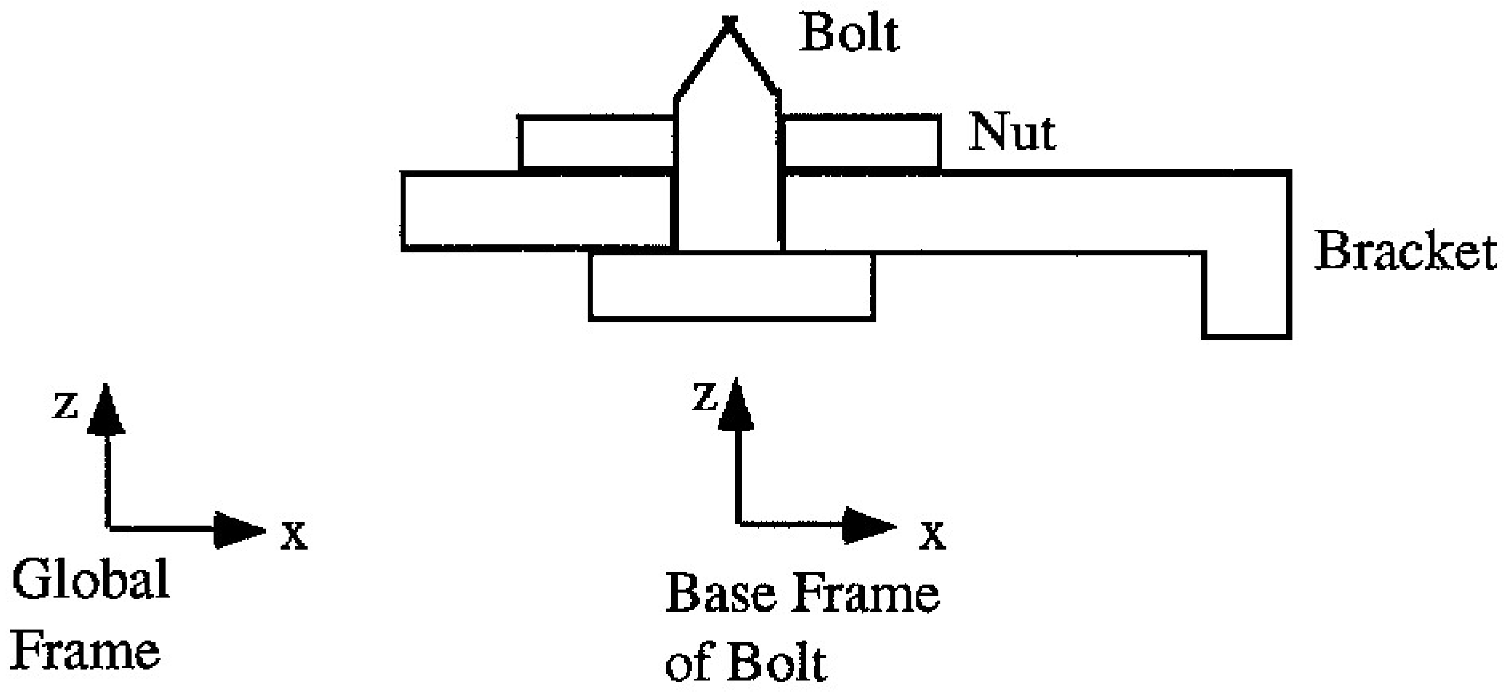


Figure 6. Bracket Mounting Assembly

Using P1, the system plans that the Nut is attached to the Bolt first because Nut is higher than Bracket as shown in Fig. 7. However, the plan (or a hypothesis) fails for the similar reason as the case shown Fig. 2: Bracket can not be assembled if Nut is attached first. because of the spatial interferences between the bracket with both the Bolt and the Nut. This failed planning episode weakens the validity of P1.

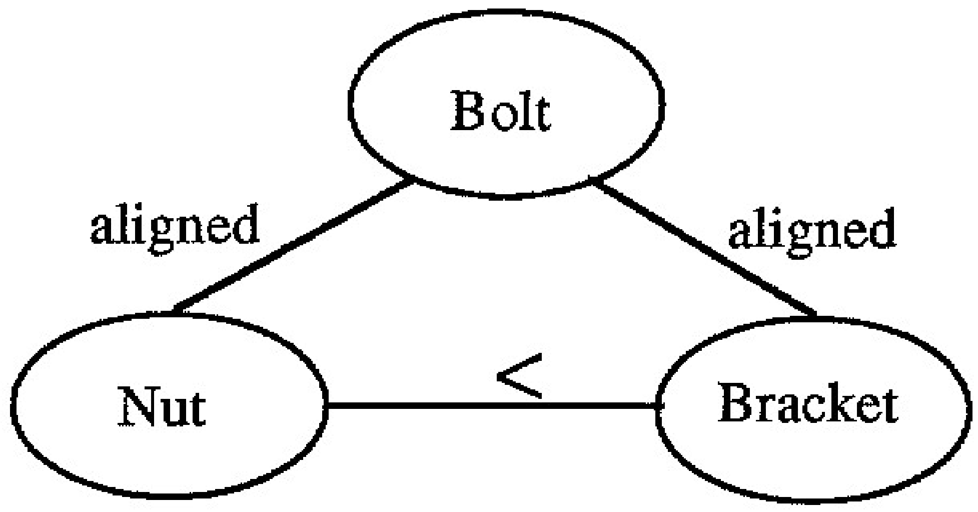


Figure 7. Failed Application of the Candidate Schema P1

Using P2, an alternative plan says that Bracket is attached to the Bolt first as shown in Fig. 8 because Bracket is closer to Bolt than Nut. This planning episode is successful and enforces the belief of P2: so far, P2 has been valid empirically. As an end result, P2 is going to be applied in the future planning situations over invoking P1.

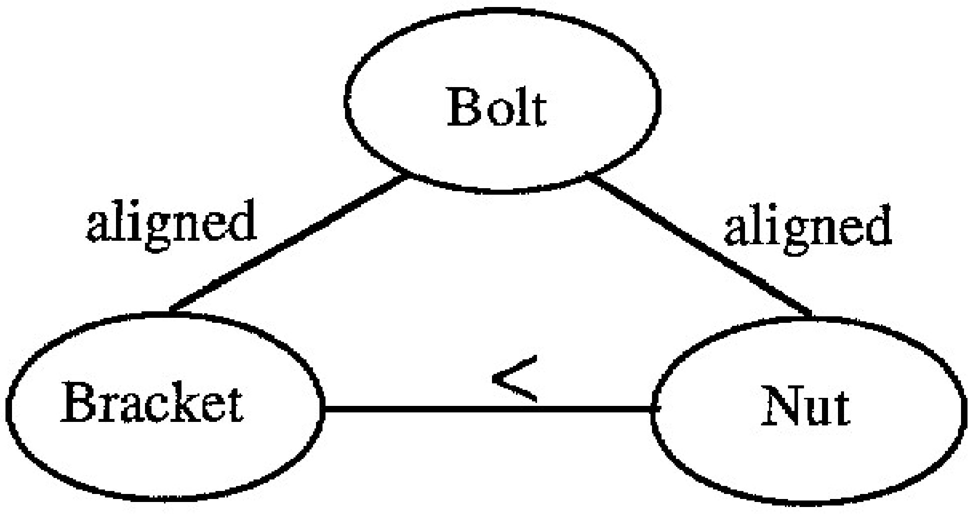


Figure 8. Successful Application of the Candidate Schema P2

Consider a screw assembly shown in Fig. 9. Using the grouping schema, the system recognizes the screw as a base component because it is pin-shaped. Washers 1 and 2 are candidate components of the grouping because they are ring-shaped. Then, the precedence schemas of the grouping schema NOMAD invokes P2 over P1. Using P2, the system generates a precedence relation that Washer 1 is assembled before Washer 2 because Washer 1 is closer to Screw than Washer 2 as shown in Fig. 10.

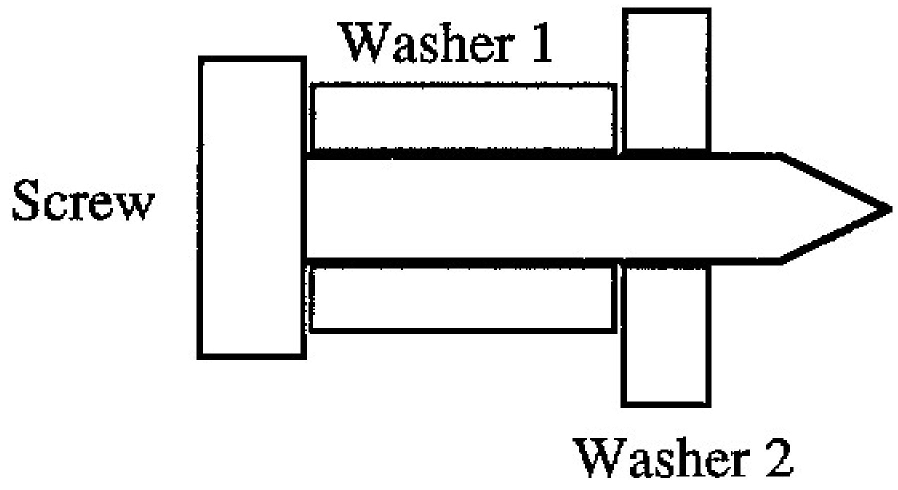


Figure 9. Screw Assembly

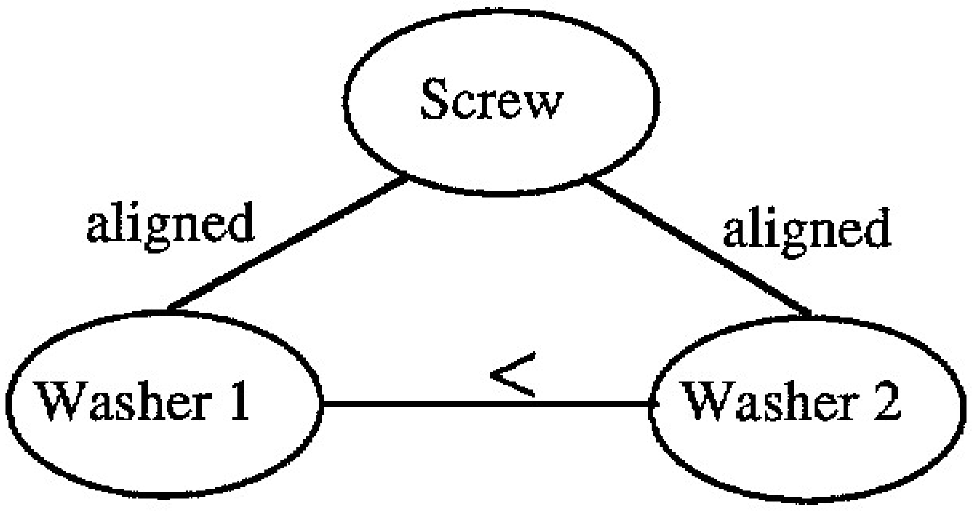


Figure 10. Screw Assembly Episode with P2

To summarize, this shows how the base and candidate components of an assembly are recognized and the candidate descriptions of precedence constraints are maintained or pruned in memory by their experimentations in a new planning situation. The learning scenario has shown the multistrategy constructive learning approach of creating explanation knowledge and validating that knowledge with increasing experience. Now, the next example will show the deductive restructuring mechanism of NOMAD. With the empirically validated schema p2, the system is presented with a new and more complicated screw assembly with three washers as shown in Fig. 11.

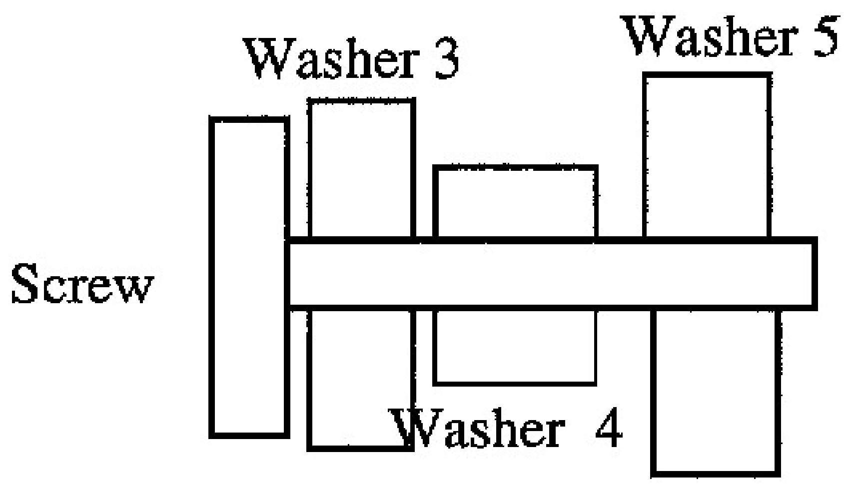


Figure 11. Screw Assembly with Three Washers

Using the grouping schema of p2, Screw is the base component because it is pin-shaped. All washers, 3, 4, and 5 are ring-shaped. Then, there are six possible bindings between the washers and the satellite schemas, \$p2 and \$p3. However, because of the precedence of p2, only three of them are possible and the rest are pruned. They are shown in Fig. 12. The first episode predicts that assemble washer 3 before washer 4; because washer 3 is closer to the screw; the second episode predict that washer 3 should be assembled before washer 5 because washer 3 is closer to screw than washer 5; likewise for the third episode, washer 4 before washer 5. These precedence constraints are all equally correct and non-conflicting with each other but none of them are *complete* in planning for the assembly structure of Fig. 11.

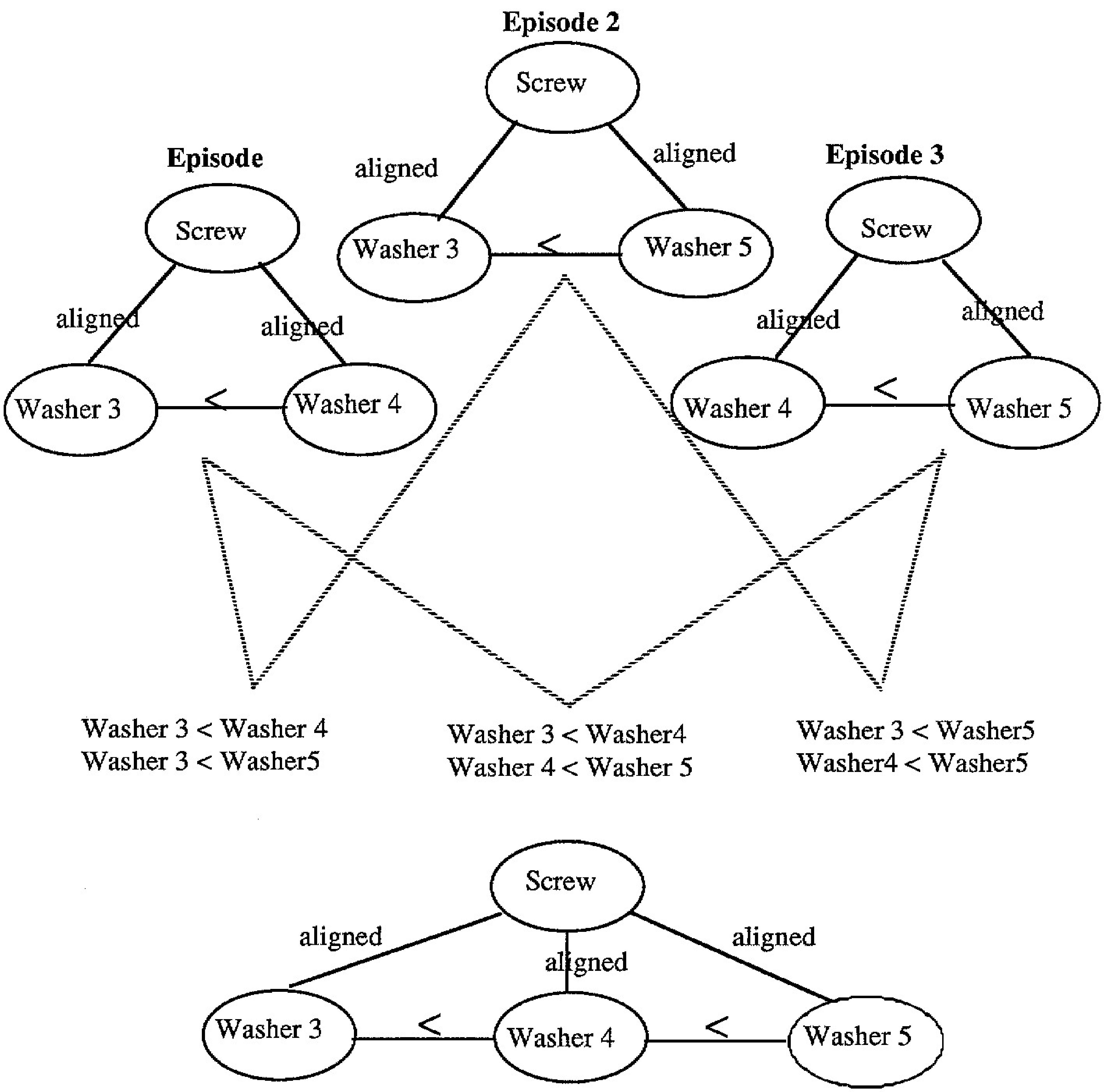


Figure 12. Three Assembly Episodes with the Candidate Schema P2

For creating a complete assembly plan, it is not necessary to combine all the episodes of Fig. 12. For example, episodes 1 and 3 are sufficient to predict the precedence constraint of Episode 2 and therefore, it is redundant. This redundancies is detected by a deductive restructuring process that merges candidate episodes. The process is carried out by an operator, called a *collapsing operator*. Here, the collapsing operator is given to episodes at a time.. From the collapse of Episodes 1 and 2, the system can predict that washer 3 is the very first wahser to assemble but no precedence between washers 4 and 5. From the collapse of Episodes 2 and 3, the system can predict that washer 5 is the last washer to assemble but no precedences between washers 3 and 4. From the collapse of Episodes 1 and 3, the system predicts washers 3, 4, and 5 are assembled in sequence. Therefore, the last collapse subsumes the precedence constraints from other collapses.

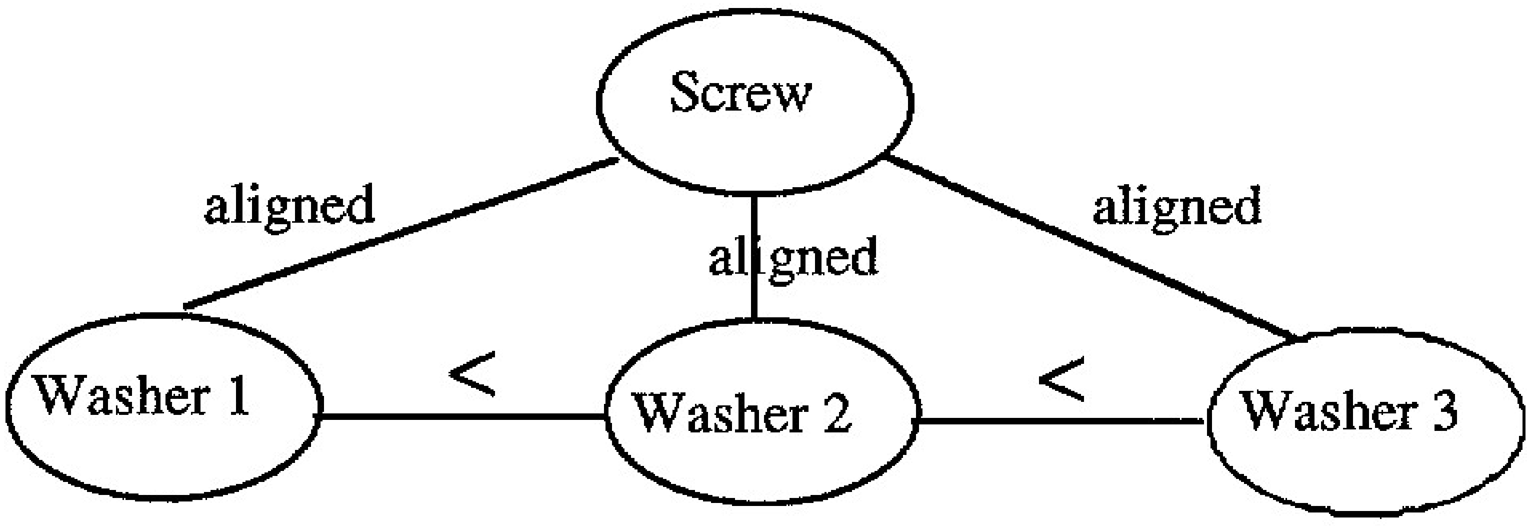


Figure 13 The Collapsed Assembly Episode

The collapsed structure represents a specialization of the original episode by a deductive restructuring. The collapses structure is similar to the macro-operator of ARMS [Segre, 1987] but the general plans (schemas) are inductively created and therefore they may be invalidated in the future. As an inductive generalization, the deductive restructuring is a method for creating "interesting" specialization of the candidate hypothesis description where in NOMAD, the specialization is guided by the planning consideration that prefers the specific prediction to less specific ones.

5.1.1 NOMAD as an MCL System

The descriptors such as dloc-b and dloc-g are created *constructively* from positional attributes of the components. Also, the collapsing operator modifies the explanation structure (ES) that is used in creating a new EK. Furthermore, the collapsing operator collapses the candidate episodes by a *deductive restructuring* using the transitivity property of temporal precedences.

NOMAD is a *multi-concept* learning system. In particular, it operates with two distinct concepts, the grouping and precedence schemas. The grouping and precedence schemas are qualitatively different type of a concept rather than different classes of a single concept: the former is about the assembly structure and the latter is about the assembly plans.

These schemas participate in creating an assembly plan whose result of execution is used to incrementally update the validity measure of the participating schemas. The validity measure is used to index the memory for future schema invocations. NOMAD is an incremental learning system under multi-concept learning context. Therefore, NOAMD is a *closed-loop* system. In short, NOMAD is an MCL system.

5.1.2. Comparison with UNIMEM

In UNIMEM [Lebowitz 1986], generalization-based memory (GBM) consists of the gen-nodes related by the generalization relationship as the candidate hypotheses similar to the version space [Mitchell 1982]. A gen-node consists of instances that are previously explained by the properties of the gen-node and sub gen-nodes that are specializations of the gen-node. A new instance is categorized into the most specific gen-node explaining the instance. If there are instances under the found gen-node that share a sufficient number of common properties with the new instance, then, a new gen-node is created as a sub gen-node of the found gen-node with the new instance and those existing instances as its instances.

In the framework of this paper, the gen-nodes represent EK or schemas according to NOMAD. The instances represent ET or episodes by NOMAD. EK is modified by generating a new sub gen-node: EK*. Therefore, the explanation process of UNIMEM establishes (6): increment learning. UNIMEM is a single concept learning system: the memory of gen-nodes is about a single type of concept: e.g., various groupings of states according to their common properties.

In UNIMEM, when the gen-nodes are successful in categorizing the new instance, the confidence of the properties of the gen-node increases; otherwise, their confidences are reduced, a similar mechanism to validity measure of NOMAD.. This is an important mechanism for a multistrategy constructive learning system whose EK is created inductively so that their validity should be examined empirically.

6. Conclusion

We have introduced a distinction between a derivational and hypothetical explanation. This distinction was then used to characterize empirical and analytic learning systems. We have then discussed multistrategy constructive learning, which integrates empirical and analytic learning. A multistrategy constructive learning system was described by a detailed analysis of example problems and their solutions in the area of automated assembly by NOMAD.

In the conceptual framework of multistrategy constructive learning, NOMAD is a partial and incomplete learning system and many avenues of machine learning research remain unexplored. MCL opens many demanding issues to be resolved by future learning systems to be built in the present and the future and challenges future machine learning researches.

Acknowledgements

The authors thank the members of the Intelligent Systems Group, and the Intelligent Explorer Group. for discussion and criticism of ideas presented here

This research was supported in part by the Office of Naval Research under grant No. N00014-82-K-0186, and in part by the Defence Advanced Project Agency under the grant administered by the Office of Naval Research No. N00014-K-85-0878.

References

DeJong, G. and Mooney, R., Explanation-Based Learning: An Alternative View, *Machine Learning Journal, vol 2*, 1986.

de Kleer, J., An Assumption-Based Truth Maintenance System, *Artificial Intelligence*, vol. 28, no. 1, 1986.

Ko, H., Reasoning about Kinematics from Mating Conditions, submitted for publication to *IEEE Journal of Robotics and Automation,* October, 1987.

Lebowitz, M., Concept Learning in a Rich Input Domain: Generalization-Based Memory, a chapter in *Machine Learning: An Artificial Intelligence Approach, Volume 2*, R. S. Michalski, J.G. Carbonell, T.M. Mitchell (eds), Morgan Kaufmann, 1986.

Michalski, R.S., Discovering Classification Rules Using Variable-Valued Logic System VL1, *Proceedings of the Third IJCAI*, Stanford, Calif., pp. 162-72, 1973.

Michalski, R. S., Theory and Methodology of Inductive Learning, *Machine Learning: An Artificial Intelligence Approach,* R. S. Michalski, J. G. Carbonell, T. M. Mitchell (eds.), Tioga Publishing Co., 1983.

Michalski, R. S., Concept Learning, *Encyclopedia of Artificial Intelligence*, E. Shapiro (ed.), John Wiley & Sons, January, 1987.

Michalski, R. S., Ko, H. and Chen, K, Qualitative Prediction: The SPARC/G Methodology for Describing and Predicting Discrete Processes, in the *Current Issues in Expert Systems*, P. Dufour and A. Van Lamsweerde (eds), Academic Press Incorporated, pp 125-158, 1987.

Michalski, R.S.and Ko, H.,. On the nature of explanation or Why did the wine bottle shatter, *AAAI Workshop on Explanation-based Learning,* Stanford University, March 1988.

Michalski, R. S., Watanabe, L., Constructive Closed-loop Learning: Fundamental Ideas and Examples, Working paper, George Mason University, 1988.

Mitchell, T.M., Generalization as Search, *Artificial Intelligence*, Vol. 18, pp203-26, 1982.

Mitchell, T. M., Keller, T., and Kedar-Cabelli, S., Explanation-Based Generalization: A Unifying View, *Machine Learning Journal*, vol 1, January 1986.

Segre, A. M., Explanation-Based Learning of Generalized Robot Assembly Plans, PhD thesis, UILU-ENG-87-2208, Coordinated Science Laboratory, University of Illinois, January 1987.

Watanabe, L., personal communication, 1988.